UAM
UNIVERSIDAD AUTONOMA
DE MADRID

Escuela Politécnica Superior

Departamento de Ingeniería Informática

ARTIFICIAL COGNITIVE ARCHITECTURE WITH SELF-LEARNING AND SELF-OPTIMIZATION CAPABILITIES. CASE STUDIES IN MICROMACHINING PROCESSES

PhD DISSERTATION

Author:

Gerardo Beruvides López

Supervisors:

PhD. D. Rodolfo E. Haber Guerra

PhD. D. Ramón Quiza Sardiñas

Madrid, April 2017

Título de la Tesis:	Artificial Cognitive Architecture with self-learning and self-optimization capabilities. Case studies in Micromachining processes.
Autor:	D. Gerardo Beruvides López
Directores:	PhD. D. Rodolfo E. Haber Guerra Prof. D. Ramón Quiza Sardiñas
Tutor:	Prof. D. Manuel Sánchez-Montañés Isla

Miembros del Tribunal de Defensa de la Tesis Doctoral:

Presidente:	Prof. D. Fernando Matía Espada
Vocales:	Prof. D. Marcelino Rivas Santana Prof. Dña. Matilde Santos Peñas Dra. Dña. María Carmen García-Alegre Sánchez
Secretario:	Prof. D. Pablo Varona Martínez
Suplentes:	Prof. D. Luca Fumagalli Prof. D. Manuel Ferre Pérez

Aprendemos de los grandes sabios del pasado,

… que no hay fuerza motriz mayor que la voluntad,

… que no hay éxitos sin perseverancia y trabajo duro,

… que solo en tus sueños encontraras el camino de tu realidad,

Gracias Mamucha por tener la voluntad de apostar por mis sueños,

Gracias Peluchina por no permitir que mis sueños se apaguen jamás.

AGRADECIMIENTOS

No podría empezar estas palabras sin citar a mi madre (Delia Margarita), has sido mi luz, mi farro, mi guía y mi ejemplo desde que me trajiste a este mundo. Mamucha has sabido transformar mis inmadureces con tacto y la elegancia en el camino a seguir, sin imponerme nada, sino ayudarme a razonar por mí mismo que los años no son por gusto y que una madre no puede desear otra cosa que no sea lo mejor para sus hijos, aunque esto repercuta en no tenerlos cerca, gracias por tus enseñanzas, por tus consejos y por tus sacrificios para que pudiera tener la oportunidad que te fue negada, gracias por nunca mostrar ante mí una señal de tristeza o dolor por verme partir, gracias por alimentar mis sueños y confiar en que tenía todo lo necesario para poderlos cumplir. Por todo esto te dedico esta tesis, te dedico mi vida.

A mi peluchina linda (Claudia), amor no sé si por suerte o desgracia para ti has estado más presente que nadie en el todo el proceso de confección de dicha tesis, has estado a mi lado tanto en los momentos de éxito como en los que he flaqueado, apoyándome y teniendo en mí una fe infinita que muchas veces ni to mismo soy capaz de encontrar; muchas gracias por enfrentar los problemas con una gran sonrisa y por llenar nuestras vidas de felicidad, sin duda este triunfo también es tuyo mi amor.

A mi familia, a mi padre por ser un ejemplo de superación y lucha. Al jefe (Puig) por darme tantos consejos durante mi formación en Cuba, por acompañar y cuidar a mi vieja, saber que estás con ella siempre ha sido una satisfacción y tranquilidad para mí. A mi abuelo (Alberto López Miña), del que nunca olvidaré sus cuentos de siendo solo un estudiante de economía basándose en la confianza en sí mismo auditaba empresas, gracias por las enseñanzas, por decirme que no basta con estudiar, sino que debía ser humilde pero valientes ante la vida y de que cada persona nos puede trasmitir una enseñanza sin importar tu estatus o escolaridad. Por último, darle las gracias a Fernando (*brother*) que a pasar de no ser hermanos de sangre se ha portado en cada momento como si lo fuera, muchas gracias por tu ayuda incondicional y al cariño recibido por tu familia en cada momento de mi estancia en este país.

A mis tutores, Ramón Quiza (Gran Maestro Hechicero), donde empezó todo hace ya algunos años cuando este estudiante llamaba a la puerta de su despacho para pedirle si podía formar parte de su grupo de investigación, muchas gracias por dejarme entrar al equipo y mostrarme el camino y las técnicas necesarias para emprender mi carrera investigadora. Durante estos años hemos logrado muchas cosas juntos: artículos, conferencias, softwares, etc., y sé que si hoy día estoy acá es por ti de lo cual siempre estaré agradecido. Además, gracias por ser más que mi director de tesis, ser mi amigo, mi hermano mayor. Por otra parte, a mi director Rodolfo, primero por confiar en mí y darme la oportunidad de formar parte del grupo GAMHE haciéndome crecer como profesional y como persona. También, por todas las lecciones y consejos de como un investigador o un aspirante en este caso, debe como saber estar, de que es más allá de un título, es una actitud y un compromiso moral de cambiar, de crear, de innovar y sobre todo por servir de ejemplo con tu trabajo para los que queremos formar parte de este mundo donde la pasión por la ciencia no debe escatimar en tiempo ni esfuerzo para lograr aportar tu granito de arena, dentro de ese gran todo que ha hecho más grande a la humanidad.

A mis compañeros de trabajo, en especial a Raúl por el apoyo brindado desde el día que llegue a España, por aconsejarme y mostrarme muchas veces el camino que en el pasado él tuvo que recorrer, muchas gracias por tus consejos y reflexiones, espero que encuentres reflejadas en esta tesis, algunas de tus enseñanzas. Una mención especial también a Artu (Antonio), por invitarme a su tierra sin apenas conocerme, pero sobre todo por ser mi amigo incondicional desde que nos conocimos, muchas gracias *Bro*, de corazón. A mi compatriota (Jorge), por ser mi asesor en trámites y otras gestiones necesarias para ir cumpliendo cada una de las etapas a las que me he enfrentado en este país. También al resto de compañeros (Nacho, Lola, Jorge Villagras, Fernando Seco y Antonio Jiménez), de los cuales he aprendido de sus reflexiones y debates cada día.

Por otro parte, no quisiera dejar agradecer a mis colegas de carrera: Yuyo, Benito, Danny y Yeisel, junto a ellos comencé a crecer como profesional, encontrando el apoyo y la confianza para lanzarme hacia nuevos retos en mi carrera. También agradecer a mis gente del barrio de Versalles (Matanzas), al Miche, Mascara, Albe (pequeño aprendiz) por brindarme su amistad.

Por último, a mis nuevas amistades de Madrid, Tomás y Tatiana por acogerme en su casa en momento donde esta tratado de adaptarme a un nuevo mundo, por aconsejándome y brindándome cariño a lo largo de estos años, muchas gracias de corazón. También a Alain ("Mini") y Mike por las incontables muestras de apoyo e incorporarme a su lista de amigos, apostando siempre de que sería capaz de culminar dicha etapa.

Por todas estas razones, le agradezco a todos los que he nombrado y lo que he olvidado nombrar pero que han formado parte de este proceso o travesía que hoy cierra otra página con la presente tesis doctoral, muchas gracias de corazón por creer en mí y continuará…

Lo logramos mamá ya sumamos una más.

RESUMEN

La fabricación del futuro afronta cuestiones esenciales relacionadas con la sostenibilidad y eficiencia que continúan siendo abordadas parcialmente, donde las Ciencias de la Computación e Inteligencia Artificial y la Automática son fundamentales para alcanzar altos niveles de modularidad, conectividad, autonomía y digitalización. La industria de fabricación requiere aumentar la eficiencia con menores tiempos de entrega al mercado y donde la optimización de la producción a través de los sistemas ciberfísicos de producción, el auto-aprendizaje y la auto-organización son esenciales.

Esta Tesis Doctoral está enfocada hacia el diseño e implementación de una arquitectura cognitiva artificial, de inspiración biológica, dotada de estrategias de autoaprendizaje y auto-optimización para realizar tareas de monitorización y control. En primer lugar, el fundamento nace en el nexo entre el paradigma del control por modelo interno y la conectividad cerebro-cerebelo como base de la inteligencia humana. La principal hipótesis radica precisamente en que el control por modelo interno a través de la conectividad cerebro-cerebelo es un componente único de la inteligencia humana. El segundo principio está basado en el modelo de los circuitos compartidos y la emulación de las capacidades y experiencias socio-cognitivas de los seres humanos.

Tres cuestiones esenciales han sido el desarrollo y perfeccionamiento de un método libre de gradiente para permitir la auto-optimización multiobjetivo, el desarrollo de una estrategia de aprendizaje por refuerzo para el autoaprendizaje, y finalmente la evaluación experimental y validación en dos procesos esenciales en la micro-escala (microfresado y microtaladrado).

La concepción de una metodología basada en la combinación e integración de métodos de investigación teóricos y experimentales se nutre de la selección, modificación, adaptación e integración de los más convenientes paradigmas dentro las Ciencias de la Computación, la

Automática y las técnicas de Inteligencia Artificial. Además, la elección del micromecanizado mecánico como caso real, se debió a razones científico-técnicas y económicas. El micromecanizado mecánico tiene un gran impacto en la economía mundial, en sectores como: la electrónica, el aeroespacial y la biomedicina, entre otros.

Desde el punto de vista científico y de tratamiento de la información estos procesos se caracterizan por un comportamiento no lineal de las variables, gran sensibilidad e influencia del entorno en los procesos tales como temperatura, humedad, contaminación, ruido mecánico y eléctrico, incertidumbre en la información sensorial y dependencia de la composición y del tipo de material. De este modo, se produce un incremento exponencial en la especificidad de las tareas en la micro-escala dependiendo de las fuerzas predominantes, de las propiedades físicas, geométricas y químicas de la superficie y de las condiciones de entorno y las perturbaciones. Además, aumenta la complejidad funcional de la microfabricación debido a las no linealidades. Desde el punto vista de la monitorización y el control, aumentan exponencialmente los requisitos funcionales y de precisión de sensores, de los medios de cómputo y de las estrategias de procesamiento y de toma decisión.

En esta memoria científica se presentan los algoritmos y métodos que componen las diferentes etapas o modos de actuación de la arquitectura propuesta. Se describen las capacidades para el procesamiento, modelado, optimización, monitorización y control a partir de señales captadas en tiempo real en sistemas complejos. En general, los sistemas clásicos de monitorización de los procesos en la microescala fallan debido a que carecen de información sensorial relevante, o porque las estrategias de toma de decisión no están suficientemente preparadas para hacer frente a determinados comportamientos emergentes y responder a determinados eventos.

El diseño e implementación de la arquitectura computacional modular, en red y reconfigurable para la monitorización y el control en tiempo real, tiene en cuenta los análisis de diferentes tipos de sensores, estrategias de procesamiento y metodologías de extracción de patrones de comportamiento de las señales representativas en estos procesos complejos. La capacidad de reconfiguración y portabilidad de esta arquitectura está sustentada por dos grandes niveles: el nivel cognitivo (núcleo de la arquitectura) y el nivel ejecutivo (intercambio directo con el proceso) que a su vez están compuestos por los diferentes módulos que interactúan con el proceso que se va a monitorizar y/o controlar. Estos procedimientos, que son brevemente descritos a continuación, tienen una precisión dependiente de los diferentes modelos y algoritmos integrados en la arquitectura. El nivel cognitivo está compuesto por tres módulos fundamentales para el modelado, la optimización y el aprendizaje, necesarios para la toma de decisiones (acciones de control) desde el punto de vista computacional, así como la caracterización experimental en tiempo real de procesos complejos. En el caso específico de los procesos de microfabricación se obtuvieron un grupo de modelos basados en técnicas de regresiones lineales y no lineales, además de técnicas de Inteligencia Artificial. Las variables principales que se han considerado han sido las componentes principales de fuerzas y vibraciones y el par del eje del husillo. Para la estimación de indicadores de calidad en las piezas elaboradas, se han utilizado cifras de mérito tales como la calidad de los agujeros, el descentrado (*run-out*) de la herramienta y la rugosidad superficial. Por otra parte, el nivel ejecutivo tiene una constante interacción con el proceso a monitorizar y/o controlar. Dicho nivel recibe la configuración y parametrización del nivel cognitivo para realizar las tareas de monitorización y control deseadas. El diseño e implementación de la arquitectura es la contribución y el elemento cohesionador de este trabajo.

Otra de las contribuciones más importantes de la tesis es el desarrollo y perfeccionamiento de un método de optimización basado en entropía cruzada con una serie de modificaciones en cuatro parámetros (número máximo de iteraciones, tamaño de población, número de intervalo de histograma y fracción de élite). Se trata de mejorar la convergencia de un algoritmo de optimización de múltiples objetivos mediante entropía cruzada, demostrando la notable influencia del número de época y el tamaño de la población en el tiempo de ejecución y en la calidad del frente de Pareto. Se presenta un estudio comparativo utilizando cifras de mérito reportadas en la literatura para validar los cambios propuestos en el método de entropía cruzada, con resultados prometedores (mejor distancia generacional, hipervolumen, etc.) en relación con la calidad de los frentes de Pareto con respecto a otras técnicas reportadas.

La tercera contribución es el diseño e implementación de un método de aprendizaje por refuerzos (*Q-learning*) para dotar de capacidad de auto-aprendizaje a la arquitectura propuesta. Se introdujeron algunas modificaciones y consideraciones para facilitar el despliegue en la definición de los conceptos de estado y acción, así como la función de recompensa en un sistema de autoaprendizaje. El enfoque se centra en el ajuste de los parámetros de los controladores.

Como parte de la metodología científica de la Tesis Doctoral, todas las estrategias desarrolladas han sido validadas rigurosamente en una plataforma experimental, utilizada como soporte tecnológico. Desde el punto de vista de la microfabricación en esta Tesis Doctoral se presentan resultados muy positivos. En primer lugar, la caracterización experimental se ha corroborado mediante la comparación entre los resultados teóricos y experimentales obtenidos y la utilización de diferentes cifras de mérito o índices de comportamiento (p.ej., histogramas, estadígrafos, errores relativos, medios, cuadráticos, entre otros) durante operaciones de microtaladrado y microfresado. Durante la investigación se desarrollaron dos metodologías para la detección del descentrado y la predicción de la calidad de los agujeros en los procesos de microtaladrado, así como una metodología para la estimación de la rugosidad superficial en procesos de microfresado. Finalmente, además de las aportaciones expuestas anteriormente, se realizaron más de 400 pruebas, combinando todos los modos de control de los que está dotada la arquitectura (p.e., monitorización, lazo simple, anticipación, lazo simple + aprendizaje y anticipación + aprendizaje) descritos en la Tesis Doctoral.

De forma resumida, la contribución técnica fundamental de esta Tesis Doctoral es que a partir de la mínima información sensorial posible (señales de aceleración y señales de fuerzas) y de la mínima cantidad de información sobre las condiciones de corte (velocidad de corte, avance por diente y penetración axial), se puede monitorizar en tiempo real el estado del proceso de corte en la micro-escala y realizar acciones de control para garantizar buenos acabados superficiales y alargar la vida útil de la herramienta. Este resultado técnico supone un salto cualitativo importante sin precedentes en la investigación industrial en el campo de la microfabricación.

El trabajo desarrollado y presentado en esta Tesis Doctoral ha sido posible gracias a una beca de formación de personal investigador (FPI), concedida en el proyecto del plan nacional de I+D DPI2012-35504 CONTROL COGNITIVO ARTIFICIAL EN PROCESOS DE MICROMECANIZADO MECÁNICO. MÉTODO Y APLICACIÓN (CONMICRO). Finalmente, un gran número de estos resultados parciales han sido sometidos a consideración de la comunidad científica internacional a través de la presentación en congresos internacionales y de las publicaciones en revistas de reconocido prestigio de diferentes áreas de conocimiento que han valorado positivamente el carácter heterogéneo, multidisciplinar e interdisciplinar de las aportaciones realizadas. La contemporaneidad de estos resultados armoniza con las líneas de investigación futuras en el campo de los sistemas ciberfísicos, la Industria 4.0 y la Fábrica del Futuro, en las que los sistemas cognitivos y las capacidades auto-reconfiguración, auto-optimización y autoaprendizaje son claves para los sistemas productivos sostenibles del siglo XXI.

ABSTRACT

The sustainability and efficiency are essential issues on the future manufacturing industry. Nowadays, the main challenges around the manufacturing industry are only partially addressed in fields such as Computer Science, Artificial Intelligence, Mechanical Engineering and System Engineering, all relevant to achieve high levels of modularity, connectivity, autonomy and digitization. The manufacturing industry needs to increase efficiency with shorter delivery times, where the production optimization through cyber-physical systems, self-learning and self-adaptation are essential.

This Doctoral Thesis is focused on the design and implementation of an artificial cognitive architecture, biologically inspired with strategies of self-learning and self-optimization to carry out monitoring and control tasks. Firstly, the foundations rely on the nexus between the paradigm of internal model control and the cerebellum-brain connectivity as the pillar of human intelligence. The main hypothesis is precisely that internal model control through the brain-cerebellum connectivity is a unique component of human intelligence. The second principle is based on the shared circuits model and the capacities to emulate socio-cognitive skills of human beings.

Three key issues have been addressed in this Thesis, as follows: the development and refinement of a gradient-free method to enable multi-objective self-optimization, the development of a reinforcement learning strategy to carry out self-learning and finally, the experimental evaluation and validation in two manufacturing processes at the micro-scale (i.e., micro-milling and micro-drilling).

The conception of a methodology is based on the combination and integration of theoretical and experimental methods by selecting, modifying, adapting and integrating the most convenient paradigms within the Computer Sciences, Control Engineering and Artificial

Intelligence. In addition, the choice of micromachining processes as case study is based on scientific-technical and economic reasons. The micromachining has a great impact on the world economy, in sectors such as: electronics, aerospace and biomedicine, among others.

From the scientific point of view, these processes are characterized by a non-linear behaviour of the variables, great sensitivity and influence of the environment such as temperature, humidity, pollution, mechanical and electrical noise, uncertainty in the sensory information and dependence of the composition and the material used for manufacturing. In this way, there is an exponential increase in the specificity of the tasks in the micro-scale depending on the predominant forces, the physical, geometric and chemical properties of the surface, the environmental conditions and disturbances. In addition, the functional complexity increases in the microfabrication due to nonlinearities. From the monitoring and control point of view, the functional and precision requirements of sensors, computing processing requirements and decision-making strategies exponentially increase.

In this PhD thesis, the algorithms and methods for the different operating modes of the proposed architecture are presented. Furthermore, the capabilities for processing, modeling, optimization, learning, monitoring and control from signals captured in real time in complex systems are also described. In general, the classical monitoring systems for micro-scale process fail because they lack relevant sensory information or the decision-making strategies are not sufficiently prepared to deal with certain emerging behaviours and respond to particular events.

The design and implementation of a computational architecture (modular, network and reconfigurable for real-time monitoring and control) take into account the analysis of different types of sensors, processing strategies and methodologies for extracting behaviour patterns from representative signals in complex processes. The reconfiguration capability and the portability of this architecture are supported by two major levels: the cognitive level (core of architecture) and the executive level (direct exchange with the process). At the same time, this is composed by different operating modes that interact with the process to be monitored and/or controlled. These procedures (briefly described below) depend on the models and algorithms integrated in the architecture. The cognitive level is composed by three fundamental modes: modeling, optimization and learning, necessary for decision-making (control signals) from the computational point of view, as well as the real-time experimental characterization of complex processes. In the specific case of the micromachining processes, a series of models based on linear regression, nonlinear regression and artificial intelligence techniques were obtained. The main considered variables were the principal components of forces, vibrations and the spindle torque. For the workpiece-tool quality indicators, estimation merit figures such as the quality of the holes, the run-out and the surface roughness were considered. On the other hand, the executive level has a constant interaction with the process to be monitored and/or controlled. This level receives the configuration and parameterization from the cognitive level to perform the desired monitoring and control tasks. The design and implementation of the architecture is one of the main contributions and the cohesive element of this work.

Another important contribution of this Doctoral Thesis is the development and improvement of a cross entropy- based optimization method with a series of modifications in four parameters (maximum epoch number, population size, histogram interval number and elite fraction). The objective is to improve the convergence of a multi-objective optimization cross-entropy based algorithm, demonstrating the notable influence of the epoch number and the population size in the execution time and the quality of the Pareto front. Furthermore, a comparative study using multiples merit figures reported in the literature to validate the proposed changes in the cross-entropy method is presented, obtaining promising results (better generational distance, hypervolume, etc.) in relation to the quality of the Pareto fronts respect to other reported optimization techniques.

The third contribution is the design and implementation of a reinforcement learning method (Q-learning) to provide self-learning capabilities for the proposed architecture. Some modifications and considerations were introduced to facilitate the deployment in the definition of the state and action concepts, as well as the reward function in a system of self-learning. This approach is focused on tuning controller parameters.

As part of the scientific methodology of the Doctoral Thesis, all strategies developed have been rigorously validated in an experimental platform, used as technological support. From the micromachining point of view, this Doctoral Thesis presents very positive results. Firstly, the experimental characterization has been corroborated by comparing the theoretical and experimental results obtained and the use of different merit figures or behaviour indexes (e.g., histograms, statisticians, relative errors, means, quadratics, among others) during micro-drilling and micro-drilling operations. During the investigation, two methodologies were developed for the run-out detection and the hole quality prediction in micro-drilling processes. In addition, a methodology for the surface roughness estimation in micro-milling processes was introduced. Finally, more than 400 tests were performed, combining all the control modes incorporated to the architecture (e.g., monitoring, simple loop, anticipation, simple loop + learning and anticipation + learning).

In summary, the fundamental technical contribution from this Doctoral Thesis is the use of the minimum possible sensory information (force and vibration signals) and the minimum cutting conditions information (cutting speed, feed rate per tooth and axial cutting deep) in order to monitor and to perform control actions to guarantee high surface finish quality and to extend the useful life of the tool in micromachining process. This technical result represents an unprecedented qualitative leap in micromachining industrial research.

Finally, a large number of these partial results has been submitted to consideration by the international scientific community by the presentation at international congresses and publications in prestigious journals. Referees from different knowledge areas have positively valued the heterogeneous, multidisciplinary character and interdisciplinary of the contributions made. The contemporaneity of these results harmonizes with the future research in the field of cyber-physical systems, Industry 4.0 and the Factory of the Future, in which cognitive systems, self-reconfiguration, self-optimization and self-learning capabilities are the key to the sustainable production systems in the 21st century.

To conclude, this Doctoral Thesis has been developed within the framework of a research staff training grant (FPI) approved in the research project DPI2012-35504 ARTIFICIAL COGNITIVE CONTROL FOR MICROMECHANICAL MACHINING. METHOD AND APPLICATION (CONMICRO).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

For a long time, since the born of cybernetics in 1948, artificial cognitive systems have been associated with robotics because of the scientific assumption of robot design with human-like intelligence in terms of perception, performance and high level of cognition. Artificial cognitive systems have automatic adjustment capabilities based on continuous interaction with the permanent change of the environment. In order to understand the system behavior, for each particular problem a series of goals are defined or a set of objectives to be achieved by the cognition or learning process. This automatic tuning capability is also known as self-capacity, encompassing many techniques and procedures for self-learning, self-optimization and self-organization [1, 2].

Many optimization methods are reported in the literature, ranging from genetic algorithms to particle swarm optimization [3-6]. Evolutionary algorithms (EAs) have demonstrated their suitability as a method for multi-objective optimization. EAs store a set of simultaneous trade-off solutions with the potential to exploit the synergies of a parallel search across all possible solutions. However, EAs are usually experimentally assessed through various test problems because an analytical assessment of their behavior is very complex. Thus, the appropriate performance on some problems cannot be guaranteed *a priori*. Estimation-of-Distribution Algorithms (EDAs) have emerged in the middle ground between Monte-Carlo simulation and EAs [7, 8]. One of the main advantages of EDAs is that the fusion of prior information into the optimization procedure is straightforward, thereby reducing convergence time when such information is available. From the standpoint of computational costs, it involves fewer heuristics than the other gradient-free optimization methods.

The optimal setting of models and controllers is a real challenge. Some reports on stochastic and gradient-free based optimization in fuzzy control systems demonstrate the potential of these techniques [9, 10]. However, many of these optimization techniques have not been applied to real industrial processes, due to the high complexity of optimization algorithms, the need to define appropriate cost functions and performance indexes, the insufficient performance and the absence of commonly used empirical formulas in industrial contexts. The concept of cross-entropy (CE) has been thoroughly addressed in the literature from different perspectives. For instance, very promising results have recently been described in the literature in relation to cross entropy for uncertain variables, the minimum cross-entropy principle for uncertain optimization and machine-learning problems [11]. Moreover, the theoretical background of CE enables theoretical studies of the method, which can provide sound guidelines on the potential applications of this algorithm in artificial cognitive architectures.

By the other hand, the need of embedding new learning capability in robots and machines in complex and dynamic systems grows up very fast every day. One definition of the machine learning was given by Arthur Lee Samuel "[…] *gives computers the ability to learn without being explicitly programmed*" [12]. In general terms, it is about *to learn* in the presence of uncertainty, noise and the absence of a traditional or conventional model of the process. In order to carry out this task, in the field of Artificial Intelligence several methods have been developed for supervised and unsupervised learning, depending on the information received.

Reinforcement learning (RL) is one of the most popular unsupervised learning techniques. Numerous applications of reinforcement learning techniques applied to the model adjustment in the presence of dynamic environments can be found in the literature [13]. Initially, reinforcement learning techniques were conceived for a finite and discrete states and actions. Nowadays, for instance learning algorithms are capable to establish the relationship between states and action to obtain a determined current flow by observing the electric current behavior. RL has also been applied to dynamic systems whose states and actions to be taken are continuous. This feature can be seen, for example, in the application of reinforcement learning to the complex piloting of a helicopter [14]. Another application to realize the optimal energy allocation between the engine-generator and battery of a hybrid vehicle is reported by [15], based on a reinforcement learning-based real-time energy-management strategy.

There is as yet no such complete scientific theory of intelligence [16]. Recent results in different disciplines, such as neuroscience, psychology, artificial intelligence, and robotics, and results related with new machines and intelligent processes, have laid the foundations for a computational theory of intelligence [17]. There are many definitions of intelligence, one of them is the ability of human beings to perform new, highly complex, unknown or arbitrary cognitive tasks efficiently and then explain those tasks with brief instructions. It has spurred many researchers in areas of knowledge such as control theory, computer science, and artificial intelligence (AI) to explore new paradigms to achieve a qualitative change and then to move from intelligent control systems to artificial cognitive control strategies [17, 18].

A natural cognitive system displays effective behavior through perception, action, deliberation, communication, and both individual interaction and interaction with the environment. What makes a natural cognitive system different is that it can function efficiently under circumstances that were not explicitly specified when the system was designed. In other words, cognitive systems have certain flexibility for dealing with the unexpected [19]. A cognitive system can also *reason* in different ways, using large quantities of knowledge adequately represented in advance. In addition, a cognitive system can learn from experience to improve how it operates. Furthermore, it can explain itself and accept new directions, it can be aware of its own behavior and reflect upon its own capabilities, and it can respond robustly to unexpected changes. Thus, artificial cognitive agents must share with natural cognitive systems key traits and some *cognitive and neurobiological* principles.

General systems analysis about the heterogeneous aspects of cognitive phenomena demonstrates that, bearing in mind the known mechanisms of human mind, cognition can be defined as model-based behavior [20-22]. During cognitive or executive control, the human brain and some animal brains process a wide variety of stimuli in parallel and choose an appropriate action (task context), even in the presence of a conflict of objectives and goals. Thus, there is a shift from attention control (a selective aspect of information processing that enables one to focus on a relevant objective and ignore an unimportant one) to cognitive change in itself.

Nowadays, there is a wide variety of strategies and initiatives related with the partial or full emulation of cognitive capacities in computational architectures. Each one is based on different viewpoints regarding the nature of cognitive capacity, how behaves a cognitive system, and how to analyze and synthesize such a system. However, there are two widespread trends, the *cognitivist* approach (reflected, for example, in architectures such as *Soar*, *EPIC*, and *ICARUS*), based on representational systems as a tool for processing information symbolically [23], and the approach that describes *emerging systems* (*AAR*, *Global Workspace*, and *SASE*), which include connectionist systems, dynamic systems, and enactive systems [24]. They are all based to a greater or lesser extent on the principles of self-organization [24, 25]. The cognitivist approach rests on cognition's being developed on the basis of symbolic representations, while the connectionist approach treats cognition as an inactive system, that is, a system defined "… as a simple network of processes that it produces itself and that constitutes its identity". This *sense-making* [26] has its roots in autonomy, an autonomous system being a distinguishable individual [27]. There are also hybrid models that combine the two visions; i.e., they use representations which are only constructed by the system itself when it interacts with and explores its environment.

On the other hand, there are thousands of complex systems and processes which are waiting for artificial cognitive control strategies in order to behave adequately before disturbances and uncertainties [28]. In this century, the manufacturing is a clear example of a dynamic social and technical system operating in a turbulent environment characterized by continuous changes at all levels, from the overall manufacturing system network right down to individual factories, production systems, machines, components, and technical processes. Nowadays, the highest priority goes to the development of technologies that enable faster, more efficient manufacturing by means of cooperative, self-organized, self-optimized behavior through process control systems. In addition, manufacturing processes are influenced by nonlinear and time-variant dynamics that are determined by forces, torques and other variables—even, in the case of nano-scale processes, with strong interactions at intermolecular level. These characteristics increase the functional complexity of manufacturing due to nonlinearities, and they exponentially increase the functional and precision requirements of sensors, actuators, and computing resources.

The need to apply computational intelligent techniques to the conventional or non-conventional machining processes increase over the years. In this multidisciplinary area, methods based on Artificial Intelligence techniques are determinant to model, to control, and to optimize high-performance machining processes focus on the zero defect and high productivity concepts. In Spain universities and research centers such as University of the Basque Country (EHU), IK4-Tekniker, University of Girona, among others are reported results in the applications of soft-computing techniques in machining processes. At international level, universities such as: Georgia Tech (Prof. Liang), Berkeley (Prof. P. Wright), Maryland (Prof. G. Walsh), M.I.T. (Prof. S. Mitter), among others are leaders in the research topics relative to monitoring, modeling, prediction and optimization system for manufacturing processes. Some projects at European level from FP7 to H2020 have been carried out such as DEXMART, ROBOCAST, HANDLE, HUMANOBS, AMARSi, NOPTILUS and NIFTi. For a better understanding of the impact and the novelty of the use of computational intelligent techniques for industrial applications, the general description, techniques, contributions and applications of multiples European projects are included in annex II.

The fruitful interaction among Fuzzy Logic, Evolutionary Computation, Machine Learning and Probabilistic Reasoning techniques for dealing with industrial challenges is very rich, multidisciplinary and an important research area, covering from pure simulation and development software (Vanderbilt University), artificial cognitive architectures (Birmingham University), complex environment (CALTECH), real time systems (Lund University) to embedded and automatic solutions (IBM).

Despite the large number of investigations reported in the current state of the art, there are still few implementations of reinforcement learning and optimization techniques in artificial cognitive architectures for control systems. In most cases, the principal challenge is to represent the full process complexity by good fitting model. In the particular case of micro-manufacturing processes, the design and implementation of a methodology to predict the overall behavior of process, it is still unsolved problem.

Objectives

The main objective of this Doctoral Thesis is the design and implementation of a biological-inspired artificial cognitive architecture with self-learning and self-optimization strategies, able to monitor and to produce control signals to deal with micromachining processes.

An important aspect to take into account is the use of computationally efficient strategies to guarantee the correct application in industrial scenarios with multiple complex behavior variables in process time. In this sense, the selection and implementation of reliable models to represent the process behavior, as well as, the captured and processing signals cover an important magnitude to obtain the defined goals during execution.

Three fundamental questions are the focus of this Doctoral Thesis from the design and implementation point of view:

(i) a stochastic method to enable self-optimization capability;

(ii) a reinforcement learning strategy to provide self-learning capabilities to the cognitive architecture and,

(iii) the validation in two industrial scenarios: a micro-milling and a micro-drilling process.

In order to achieve the above-mentioned goals, the following scientific and technical challenges are addressed:

1. The study of different strategies for signal processing, feature extraction, modeling and self-capabilities techniques for complex processes reported in the literature, in particular, the influence of these techniques on the manufacturing processes.
2. The development of an empirical model library to correlate the influence of the cutting parameters, forces and vibration signals for predicting key performance indicators such as the run-out, the hole quality and the surface roughness for multiple materials in micromachining process.
3. The design and implementation of a gradient-free multi-objective optimization algorithm; adjust using a well-known complex test suites reported in the literature and validate based on signals captured during micro-milling and micro-drilling operations.
4. The design and implementation of a reinforcement learning algorithm focus on a concept of self-capability to adjust direct and inverse models on the basis of the process behavior during the execution.
5. Integration of modeling, self-optimization and self-learning techniques in different modules to compose an artificial cognitive framework.

6. Validation of the procedures for emulating socio-cognitive skills implemented in the suggested artificial cognitive architecture to control relevant process variables during micromachining operations.

Methodology

The present doctoral thesis is inspired on the combination and integration of theoretical and experimental research methods. The design of leaning and optimization strategies are based on the selection, modification, adaptation and integration of the most convenient paradigms within the Computer Sciences, System Engineering and Artificial Intelligence techniques.

At the same time, the proposed methods are re-elaborated and re-designed according to the specific characteristics of the signals generated in the manufacturing processes. Data collection, signal processing and modeling tasks are performed and assessed using the multiples signals acquired in runtime.

The adaptivity and reliability of the proposed architecture are studied and analyzed in industrial conditions with high variability and strong influence of noise is an essential aspect in this research. It is important the use of a representative and extensive experimental database, combined with the actual approaches reported in the literature, integrating the soft-computing techniques with the phenomenological models of these industrial processes.

The selection of the case study (i.e., micromechanical machining) is supported by scientific, technical and economic reasons. In the last decades, mechanical machining processes have been used for manufacturing components made from a wide variety of different materials. In particular, the processing of metals and alloys such as titanium, aluminum, copper, brass and steels is used for precision fabrication, handling and joining of miniature parts for complex miniature electronic and mechanical products. The biomedical, electronic and aerospace industries are the major end users of high precision engineering components with micrometer-precision in the last decades. Furthermore, the nonlinear and time-varying nature of the micro-cutting processes influenced on this selection too. From the physical and information processing point of view, these processes are characterized by: (i) nonlinear behavior of the variables; (ii) high sensitivity and influence of the environment in the processes, such as: temperature, humidity, pollution, mechanical and electrical noise, etc.; (iii) high uncertainty in sensory information and (iv) dependence on composition and type of material.

These characteristics and properties produce: (i) an exponential increase in the specificity of the microscale operations, depending on the predominant forces, the physical, geometric and chemical properties of the surface and the environmental conditions and disturbances; (ii) in the functional complexity of micromachining due to nonlinearities and (iii) in the precision requirements of sensors, computing capability and decision-making strategies.

The design and the implementation of artificial cognitive techniques and specifically learning and optimization to control manufacturing processes, are key issues to improve productivity in micromachining operations by controlling forces, to increase the surface quality of the workpieces and to extend the tool life of micro-drills and micro-mills based on the monitoring and supervision of the vibration signals.

In order to achieve above-mentioned objective, it is necessary to design and implement an artificial cognitive control system for cutting forces, measured directly or indirectly. Facilities are essential to carry out experimental studies. The availability of a unique laboratory in the Community of Madrid equipped with the necessary devices (sensors, real time platform, etc.), is undoubtedly an important issue in the scientific methodology based on the experimental evaluation. An ultra-precision Kern Evo machining center (E 0.5i.tm and Ra <0.1 pm) equipped with a Heidenhain iTNC-530 numerical control and Blum Laser Measurement system is will be the base to run the experimental validation of the proposed methods.

Different measuring devices (sensors and DAQ cards) are also available to acquire force, acceleration and, acoustic emission signals. A tri-axial force and torque dynamometric sensor is from Kistler MiniDyn 9256C1 and the amplifier used is a Kistler 5070A 02100 (8 channels). The acceleration sensors are from Piezotronics pcb model WJT 352B and a Brüel & Kjaer model DeltaTron 4519-003 and an acoustic emission sensor model 8152B2 from Kistler with an impedance converter for the measurement above 50 kHz in machines is also available. A National Instruments PCI-6251 acquisition card is used for the signal processing captured from the force and acceleration sensors. For acoustic emission signals, a National Instruments PCI 5922 (better sampling rate is demanded) acquisition card is available with a sampling rate or resolution from 24 bits to 500 kS/s up to 16 bits to 15 MS/s. All the acquisition cards are installed in an industrial PC.

By the other hand, five different materials were selected for the experimental step. Two titanium-based alloys (Ti grade 2 and Ti_6Al_4V), a cupper-based alloy ($W_{78}Cu_{22}$), an iron-based alloy ($Fe_{64}Ni_{36}$) and an aluminum-based alloy (Al7075). The criterion for selecting these materials was the demand and applications in in biomedical and electronic industries. Finally, micro-drills and micro-mills tools with diameters ranging from 0.1 in to 1mm were used in the experimental phase. Based on the strong experimental equipment described before, one of the expected contribution of the present doctoral thesis is the validation of the proposed artificial cognitive system in a real and complex industrial process.

Thesis structure

The present dotoral thesis consists of 4 chapters, in addition to the conclusions, summarizing the most relevant contributions and future research lines.

Chapter 1 presents a critical review of the state of the art in the field of soft computing techniques applied to artificial cognitive architectures, theoretical and practical investigations, is addressed. The current trend in the conjunction of different scientific areas such as: the General Theory of Systems, Control Systems, Artificial Intelligence techniques and Computer Sciences to emulate human socio-cognitive skills in artificial cognitive control architectures is highlighted with emphazis on manufacturing processes.

In Chapter 2 an overview of the most commonly used techniques for modeling micromachining processes is presented, summarizing a group of the analytical, numerical, statistical, and intelligence-based tools reported in the literature in the last decades. Furthermore, the experimental setup (Kern Evo Ultra Precision Center and sensory equipment) and the experimental designs used during the research are described. Finally, a summary of the empirical micromachining (micro-drilling and micro-milling) models yielded in this research is presented. The influence of the force and vibration signals are modeled to estimate important behaviors and final product features such as: the run-out; the hole quality and the surface roughness in multiple materials.

In Chapter 3 a Multi-Objective Cross Entropy method (MOCE) is introduced based on a new procedure for addressing constraints, i.e., the use of variable cutoff values for selecting the elitist population and filtering of the elitist population after each epoch. Furthermore, a comparison with some other well-known optimization methods is established. Finally, the proposed method is validated in the multi-objective optimization of a micro-drilling process. Two conflicting targets are considered, i.e., total drilling time and vibrations on the plane that is perpendicular to the drilling axis. The Pareto front, obtained through the optimization process, is analyzed through quality metrics and the available options in the decision-making process.

In Chapter 4, an artificial cognitive architecture based on the shared circuits model for emulating socio-cognitive skills is proposed. The design and implementation is concentrated on self-optimization and self-leaning capabilities by an estimation of distribution method and a reinforcement-learning mechanism. Finally, the results of simulation and real-time application for controlling force in micromachining processes are presented as a proof of concept.

Chapter 1
STATE OF THE ART

In this chapter, a critical review of the scientific literature related with soft computing algorithm concepts based on cognitive architectures is presented. This chapter highlights the current trend to bring human socio-cognitive skills (self-capabilities) theoretical developments to computers enabling new communication and decision-making tasks and the application to manufacturing processes. Furthermore, the optimization techniques for complex processes are outlined.

This chapter consists of five sections. In the first section (section 1.1) a review of the cognitive control architectures and application fields is presented. After that, some of the most widely applied techniques for multi-objective optimization are presented, from the genetic algorithms to particles warm optimization. Subsequently, a review of different reinforcement learning techniques is outlined in section 1.3. Furthermore, a classification of the multiple micro-manufacturing techniques is presented in section 1.4. Finally, the chapter conclusions are presented in section 1.5.

1.1 Cognitive control architecture: An introduction

1.1.1 Introduction to the cognitive control architectures

In traditional artificial intelligence, the intelligence is often programmed on the basis of the programmer who is the creator and makes something and imbues it with its intelligence. Indeed, many traditional AI systems were also designed to learn (e.g. improving their game-playing or problem-solving competence) [29, 30]. On the other hand, biologically inspired computing gets sometimes a more bottom-up and decentralized approach. Bio-inspired techniques often involve methods of specifying a set of simple generic rules or a set of simple nodes, from the interaction of which emerges the overall behavior [31, 32]. It is desired to build up complexity until the final result is something markedly complex [33]. However, it is also arguable that systems designed top-down on the observation basis what humans and other animals can do rather than on observations of brain mechanisms, are also biologically inspired, though in a different way [34, 35].

In particular, the artificial cognitive architectures belong to the bio-inspired techniques family. Cognitive architectures can be symbolic, connectionist, or hybrid [36]. Some cognitive architectures or models are based on a set of generic rules, as the Information Processing Language, e.g., Soar based on the unified theory of cognition [37]. Many of these architectures are based on the-mind-is-like-a-computer analogy. On the contrary, sub-symbolic processing specifies no such rules a priori and relies on emergent properties of processing units (e.g. nodes).

Herbert A. Simon, one of the artificial intelligence field founders, stated that the 1960 thesis by his student Edward Feigenbaum, the Elementary Perceiver and Memorizer (EPAM) providing a possible **architecture for cognition** [38] . In this work were included some commitments for how more than one fundamental aspect of the human mind work.

John R. Anderson started to research on human memory in the early 1970s and his 1973 thesis with Gordon H. Bower provided a theory of human associative memory [39]. He included more aspects of his research on long-term memory and thinking processes into this research and eventually designed a **cognitive architecture** he eventually called Adaptive Control of Thought (ACT). He and his student used the term cognitive architecture in his lab to refer to the ACT theory as embodied in the collection of papers and designs since they didn't yet have any sort of complete implementation at the time.

In 1983 John R. Anderson published the seminal work in this area, entitled The Architecture of Cognition [40]. The theory of cognition outlined the structure of the various parts of the mind and made commitments to the use of rules, associative networks, and other aspects. The cognitive architecture implements the theory on computers. Thus, a cognitive architecture can also refer to a blueprint for intelligent agents. It proposes artificial computational processes that act like certain cognitive systems, most often, like a person, or acts intelligent under some definition. Cognitive architectures form a subset of general agent architectures. The term **architecture** implies an approach that attempts to model, not only behavior, but also structural properties of the modelled system.

There is an abundant literature on artificial cognitive architectures in the fields of sensory motor control and robotics. Although the actual application of artificial cognitive architectures in industry is still at embryonic [41, 42]. Hybrid cognitive architecture that relies on the integration of emergent and cognitivist approaches using evolutionary strategies is proposed in

[43], with a cognitive level controlled by artificial immune systems based on genetic algorithms. Bannat, et al. [44] presented a seminal paper on how artificial cognition can be applied in production systems. The authors noted that self-optimizing and self-learning control systems are a crucial factor for cognitive systems and identified important gaps such as the individual worker internal model. Sanchez-Boza *et al.* [45] proposed an artificial cognitive control architecture based on the shared circuit model (SCM). Its main drawback is a lack of systematic procedures for learning and optimization in the proposed five-layer architecture.

The way in which neuro-physiological mechanisms such as: reinforcement learning and cognitive control are integrated in the brain to produce efficient behavior has yet to be understood with sufficient clarity for effective systems to be modeled [46]. Nevertheless, reinforcement learning has been explored in artificial cognitive control by means of computational models to control robotic systems [47]. Recent investigations corroborate what is well known for a long time: automatic and flexible decision-making procedures are the cornerstone to reduce human intervention in the presence of complexity, uncertainty, background noise and large data volumes typical of production systems [48, 49].

New initiatives in artificial cognitive systems are now emerging in response to specific challenges in industry and services [50]. Recent results in disciplines such as the neurosciences, psychology, artificial intelligence, robotics and other researches related to new machines and intelligent processes have begun to approach the foundation of a computational theory of intelligence [18]. Thus, the main purpose of this study is to emulate human socio-cognitive skills, to approach control engineering problems in an effective way at an industrial level. An integrated cognitive architecture from a control perspective can be defined as a system that is able to reproduce all aspects behavior, while remaining constant across different domains and knowledge bases [51-53]. Integrated cognitive architectures that seek to imitate the major capabilities of human intelligence have been used to explain a wide spectrum of human behavior [54]. Moreover, numerous publications reflect the current pace of its progress cognitive science, all of which cannot be summarized in the context of the present study [55].

Nevertheless, all above-mentioned researches are based on the role of internal (direct and inverse) models in cognitive tasks. From a physiological point of view, the connection between the paradigm of internal control and brain-cerebellum connectivity has been advanced as a basis for explaining human intelligence [56]. Researchers have corroborated this link as a key component of human intelligence from a functional point of view [21]. Moreover, the use of internal models to explain some socio-cognitive skills based on human experience is evident from a psychological point of view [57]. The architecture, based on the model of socio-cognitive skills, overcomes the limitations of the neuroscientific approach [58-60] and takes into account the principles of simplicity and scalability.

1.1.2 Artificial cognitive architectures

It would be difficult to summarize and review all the well-known cognitive architectures; some of the most relevant architecture reports in the literature are following described.

Bratman, et al. [61] propose a belief-desire-intention (BDI) architecture with a high-level specification of the practical-reasoning component for a resource-bounded rational agent. He corroborates that an architecture for a rational agent should serve for means-end reasoning, for the weighing of competing alternatives and for interactions between these two forms of reasoning. Besides, a major role of the agent's plans to constrain the amount of further practical reasoning is performed.

Lehman, et al. [62] consider the architecture such as a theory about what is common to cognition, the content in any particular model is a theory about the knowledge the agent has that contributes to the behavior. They propose the SOAR-architecture based in the theory posits that cognitive behavior has at least the following characteristics [53]:

- **Goal-oriented:** Despite how it sometimes feels, we do not stumble through life, acting in ways that are unrelated to our desires and intentions. If we want to cook dinner, we go to an appropriate location, gather ingredients and implements. Then chop, stir and season until we have produced the desired result. We may have to learn new actions (braising rather than frying) or the correct order for our actions (add liquids to solids, not the other way around), but we do learn rather than simply act randomly.
- **Rich, complex, detailed environment:** Although the ways in which we perceive and act on the world are limited, the world we perceive and act on is not a simple one. There are a huge number of objects, qualities of objects, actions and so on, any of them may be the key to understand how to achieve our goals. Think about what features of the environment you respond to when driving some place new, following directions you have been given. Somehow, you recognize the real places in all their details from the simple descriptions you were given and respond with gross and fine motor actions that take you to just the right spot, although you have never been there before.
- **Large amount of knowledge:** Try to describe all the things you know about how to solve equations. Some of them are obvious: get the variable on one side of the equal sign, move constant terms by addition or subtraction and coefficients by multiplication or division. But you also need to know how to do the multiplication and addition, basic number facts, how to read and write numbers and letters, how to hold a pencil and use an eraser, what to do if your pencil breaks or the room gets dark, etc.

- **Use of symbols and abstractions:** Let’s go back to cooking dinner. In front of you sit a ten-pound turkey, something you have eaten but never cooked before. How do you know it is a turkey? You have seen a turkey before but never this one and perhaps not even an uncooked. Somehow some of the knowledge you have can be elicited by something other than your perceptions in all their detail. We will call that thing a symbol (or set of symbols), because we represent the world internally using symbols and we can create abstractions. You cannot stop seeing this turkey, but you can think about it as just a turkey. You can even continue to think about it if you decide to leave it in the kitchen and go out for dinner.
- **Flexible and a function of the environment:** Driving to school along your usual route you see traffic jam ahead, so you turn the corner in order to go around it. Driving down a quiet street, a ball bounces in front of the car. While stepping on the brakes, you glance quickly to the sidewalk in the direction the ball came from, looking for a child who might run after the ball. As these examples show, human cognition is not just a matter of thinking ahead: it is also a matter of thinking in step with the world.
- **Learning from the environment and experience:** We are not born knowing how to tell a joke, solve equations, play baseball or cook dinner. Yet, most of us become proficient (and some of us expert) at one or more of these activities and thousands of others. Indeed, perhaps the most remarkable thing about people is how many things they learn to do given how little they seem to be born knowing how to do it.

Another cognitive architecture in Newell’s sense of that phrase also called ICARUS is presented by [63]. Like its predecessors, it makes strong commitments to memories, representations and cognitive processes. Another common theme is the incorporation of key ideas from theories of human problem solving, reasoning and skill acquisition. However, ICARUS is distinctive in its concern with physical agents that operate in an external environment and the framework also differs from many previous theories by focusing on the organization, use, and acquisition of hierarchical structures. These concerns have led to different assumptions than those found in early architectures such as ACT and Soar.

ICARUS architecture is fed by five high-level principles about the rationale of intelligent systems: (1) cognition is based on perception and action; (2) concepts and skills have different cognition structures; (3) long-term memory is organized in a hierarchical fashion; (4) skill and concept hierarchies are acquired in a cumulative manner, and (5) long-term and short-term structures have a strong correspondence.

The interplay between rule-based reasoning, (implicit) similarity-based reasoning and (implicit) associative memory (intuition) is explored by [64]. Doing this, both explicit and implicit forms of human reasoning are incorporated in a unified framework, which is embodied in a cognitive architecture, CLARION [65]. By the other hand, Mathews, et al. [66] propose an integrated model of prediction, anticipation, sensation, attention and response (PASAR) for artificial autonomous systems. Franklin, et al. [67] describe a cognitive architecture learning intelligent distribution agent (LIDA). Furthermore, a number of neuro-computational control mechanisms is presented by [68].

Different criteria such as properties and features, agent capacities, factors in the environment, generality, psychological validity and effectiveness have formed the basis for their comparisons in various cases. Vernon et al. [19] conducted a review of various cognitive architectures such as SOAR, ICARUS, Adaptive Control of Thought-Rational (ACT-R) [69] and others, which was limited to an analysis of relevant design aspects.

Sanchez-Boza and Guerra [70] reported an initial attempt to design an artificial cognitive control system, although with two main limitations: a lack of specific procedures for enabling self-capacities such as self-optimization, learning and non-generalizable computational systems that could be deployed on low-cost computing platforms.

An interesting approach is proposed by [57], based on the shared circuits model of socio-cognitive skills, seeks to overcome limitations from the perspectives of computer science, neuroscience and systems engineering. The SCM is supported on a layered structure that reflects socio-cognitive skills (i.e., imitation, deliberation, and mindreading) by means of control mechanisms such as mirroring and simulation. Basically, SCM is based on the observation of the human brain. Some brain regions are in charge of coding actions for reaching objectives and how other regions code means for reaching objectives. So, the brain may be envisaged as making use of not only inverse models that estimate the necessary motor plan for accomplishing an objective in a given context, but also a forward model that enables the brain to anticipate the perceivable effects of its motor plan, with the object of improving response efficiency. The first kind of behavior is covered by the action of SCM layer 1, while the behavior described in the forward model is covered by SCM layer 2. Layer 4 of the scheme is in charge of controlling when one type of behavior or another should be performed.

Imitations, in addition to playing an important role in both human sociability and development, are means of learning. Imitative learning takes place when mirroring the actions of others in response to the circumstances. The observer first copies previously observed input/output associations, in order to perform this task, which inhibits the mirroring mechanism. SCM represents this mirroring capacity in its layer 3. The interaction between layer 3 and the inhibition control performed by layer 4 serves to emulate the agent’s capability to distinguish self from others.

SCM also describes, from a functional point of view, the way in which the agent can carry out the cognitive skill of mindreading. This capacity is emulated by the operation of layer 5, which is in charge of simulating other possible related inputs that are external (exogenous) to the agent. A layer-based scheme of SCM is depicted in Figure 1.1.

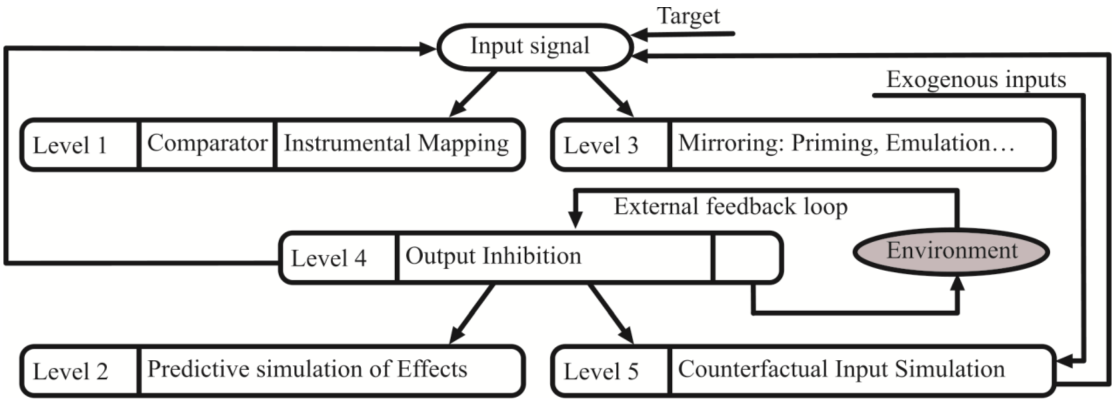


Figure 1.1 Conceptual scheme of the Shared Circuits Model approach based on [57]

A modified shared circuits model (MSCM) based on Hurley's work is proposed in [45]. Five modules, made up of one or more processes performed by the SCM layers, were implemented. The MSCM proposal defines each module in terms of an emulative cognitive ability. MSCM embodied a computational infrastructure that is plausible from a neuroscientific and psychological perspective, but which lacks a generalizable approach with optimization and learning mechanisms. More details about the five modules and the overall performance can be found in Figure 1.2.

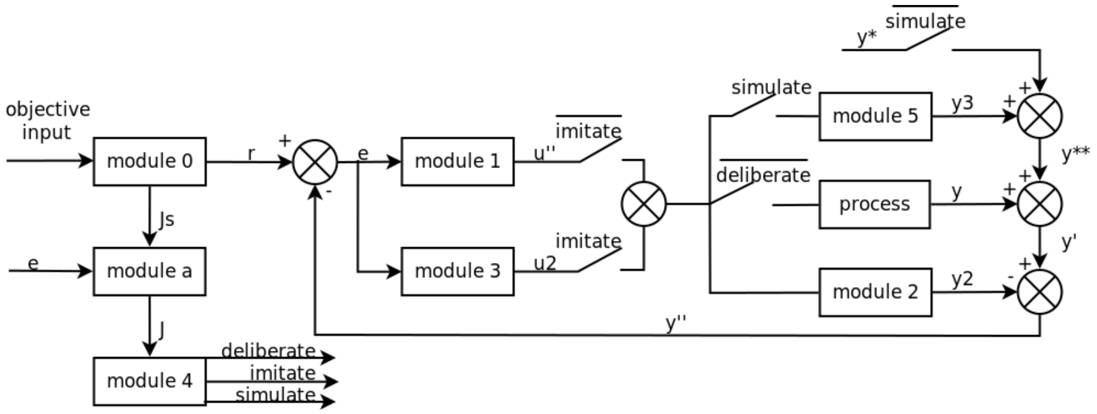


Figure 1.2 Expanded block diagram of MSCM from [45]

The main drawbacks are:

- A tailored design of the architecture without a systematic methodology means that it is not extendable to other types of processes or even to other execution configurations.
- A lack of computational strategies to enable self-optimization and learning. These strategies improve the performance of the artificial cognitive control system facing different situations.

- Module-driven architecture is mapped from Hurley’s layer concept, but is solely based on a single type of model. For instance, only fuzzy models can be used in the single loop configuration.

1.1.3 Applications

Despite the importance of cognitive architectures as a research area, strategies for the application of artificial cognitive control at industrial level have many constraints and there are very few formal reviews on control engineering [71-73]. Moreover, relevant aspects of cognitive control architectures have to be addressed in detail: firstly, self-learning and self-optimization based on interaction; secondly, procedures for assessing cognitive architectures are limited and their availability is often restricted.

Several cognitive architectures are used in many applications [74-76]. Although their implementations are not publicly available with only few exceptions. The assessment of their evaluation criteria and performance indices is therefore not easy for control engineering and computers in industry. Such a task would require associating and defining figures of merit related to transient behavior, dynamic and steady state systems and control effort, among others, all of which hinders any comparison of the present-day capabilities and the performance of these architectures. Finally, many cognitive architectures lack biological inspiration. It is essential that computational implementation of architectures have both biological and psychological roots in real applications [77]. Computational architectures are at present somewhat limited to cognitive **psychological** validity.

Manufacturing processes are characterized by the presence of nonlinear and time-variant dynamics that emerge from the behavior of temperature, forces, torques and other representative variables, characteristics that increase the functional complexity of micro-manufacturing and the functional requirements and precision of sensors, actuators and computing resources [9, 78].

Hong-Seok, et al. [79] present a new technology called cognitive agent to control the machining system. Cognitive agents with intelligent behaviors such as perception, reasoning and cooperation allow the manufacturing to overcome the disturbances. An innovative architectural solution to improve the capabilities and performance of modern production plants, so-called Cognitive Middleware for manufacturing is presented by [80]. They propose the use of ICT technologies (e.g. semantic, middleware, optimization algorithms) into a holistic framework to be transparently adopted into existing factories as well as embedded in the future designs.

In particular, micro-manufacturing is a clear example of a dynamic system operating in an environment characterized by continuous change, being a perfect stage to proof new cognitive control and decision making strategies [81]. In this scenario, one of the main objectives is the development of technologies and algorithms that enable faster, self-organized, self-optimized behavior process control systems [82, 83].

Nowadays, it is impossible to talk about cognitive systems regardless concepts such as: cyber-physical systems (CPS), cloud computing and Internet of things (IoT). Lee, et al. [84] present the different steps of a CPS implementation, being the self-capabilities the top of this pyramid (see Figure 1.3).

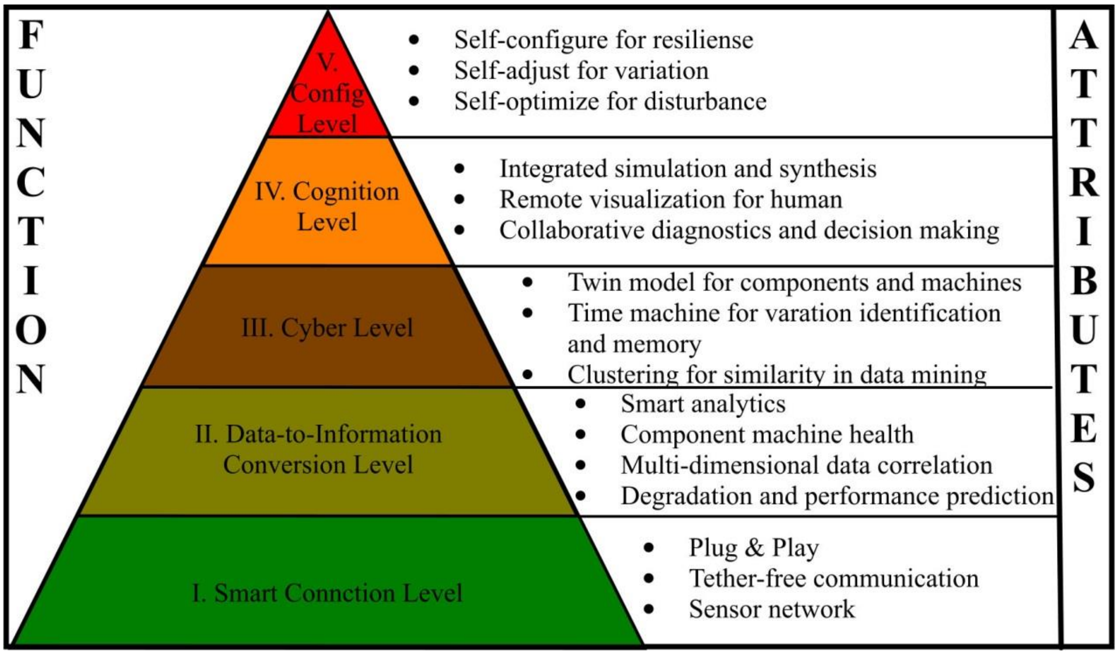

FUNCTION
V. Config Level
IV. Cognition Level
III. Cyber Level
II. Data-to-Information Conversion Level
I. Smart Connction Level
Self-configure for resiliense
Self-adjust for variation
Self-optimize for disturbance
Integrated simulation and synthesis
Remote visualization for human
Collaborative diagnostics and decision making
Twin model for components and machines
Time machine for varation identification and memory
Clustering for similarity in data mining
Smart analytics
Component machine health
Multi-dimensional data correlation
Degradation and performance prediction
Plug & Play
Tether-free communication
Sensor network
ATTRIBUTES

Figure 1.3 Five steps pyramid for CPS implementation [84]

1.2 Optimization techniques

1.2.1 General aspects

Optimization is the problem of, given some scalar function, $f(\bullet)$ (called objective function), of a vector variable, $x \in \Omega \subset \mathbb{R}^n$ (called decision variable), finding the value x* such that $f\left(x^*\right) \leq f\left(x\right)$ for all $x \in \Omega$, and also fulfill the conditions $g_i(x) \leq 0$, $i = 1, \ldots, m$ (called inequality constraints) and $h_i(x) \leq 0$, $i = 1, \ldots, p$ (called equality constraints).

Certainly, this definition refers only to a minimization, but it is not a limitation as any maximization problem can be transformed into a minimization one just by multiplying the objective function by minus one.

Some practical problems, however, require the simultaneous minimization (or maximization) of several objective functions. This is called multi-objective or multi-criteria optimization.

There are two main approaches for solving a multi-objective optimization: the first one, called *a priori* approach, is carried out by supplying information about the preferences between the objectives before executing the optimization. In this group are included the linear and nonlinear aggregation of objectives, the lexicographic method and the goal programming.

In the second approach, called *a posteriori*, no information is supplied about the preferences between the objectives. The optimization process is carried out and then the decision maker chooses the most convenient solution from a set of non-dominated solutions. The optimal is in the wide sense and there is no other solution in the considered search space that improves at least one objective function without detriment of another function [85].

These methods are based on the so-called Pareto front, which contains these non-dominated solutions. The set of Pareto solutions in the objective space is referred to as Pareto frontier or efficient frontier. Usually we select one final solution from Pareto solutions taking into account the total balance among objectives. As it is shown in Figure 1.4, while fixing an objective at the level of the weak Pareto solution, we can improve another objective. Therefore, weak Pareto solutions are not welcome as solutions in actual decision making, but we often encounter cases in which only weak Pareto optimality is guaranteed theoretically [86].

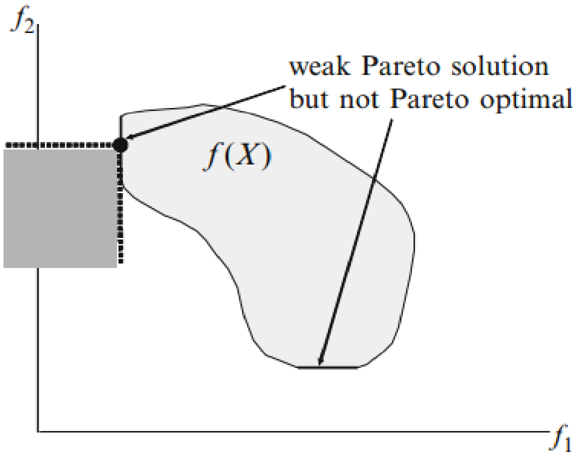


Figure 1.4 Weak Pareto solutions in the objective space from [86]

In general, many Pareto solutions are available. The final decision should be made as a trade-off of criteria. This is a judgment problem for expert's decision-making. The totally balancing over criteria is usually called trade-off. It should be noted that there are over one hundred criteria in some practical problems such as erection management of cable stayed bridge and camera lens design. Therefore, it is important to develop effective methods for helping decision making to trade-off easily even in problems with very many criteria.

1.2.2 Types

1.2.2.1 Analytic Optimization Techniques

Analytic are the oldest and the most exact optimization methods. They are based on determine the stationary points, i.e., the points where the first derivative of the objective function is zero (see Figure 1.5 (a)):

$$\frac{df}{dx} = 0 \tag{1.1}$$

When the decision variable is not scalar but vector, stationary points exists where all the components of the gradient of the function are zero (see Figure 1.5 (b)):

$$\nabla f = \frac{\partial f}{\partial x_1} e_1 + \ldots + \frac{\partial f}{\partial x_n} e_n = 0 \tag{1.2}$$

As this condition is necessary but not sufficient, the second derivative (or the Hessian matrix, for vector decision variable) must be checked in order to known if the given stationary point corresponds to a maximum, a minimum or a saddle point.

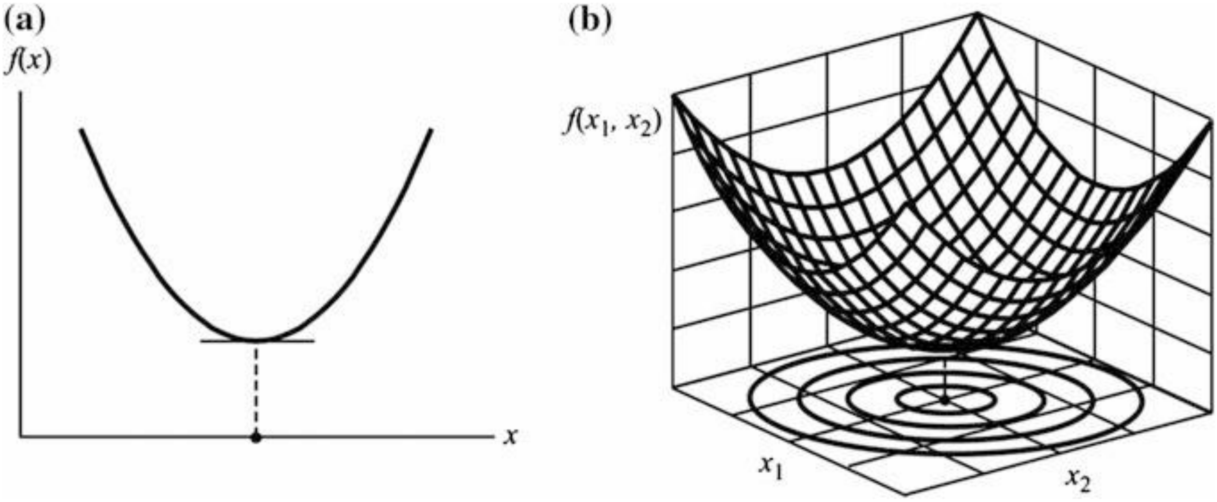


Figure 1.5 Stationary points (a) Scalar decision variable (b) Vector decision variable

Analytic techniques have strong mathematic foundations, but they only work properly in relatively simple problems. Another limitation is related to constrained optimization, although some methods such as the Lagrange multipliers have been developed to overcome this limitation.

1.2.2.2 Numeric Optimization Techniques

Numeric optimization techniques are also based on the gradient of the objective function, but unlike analytic methods do not require the computation of the root derivatives, which are often transcendental equations.

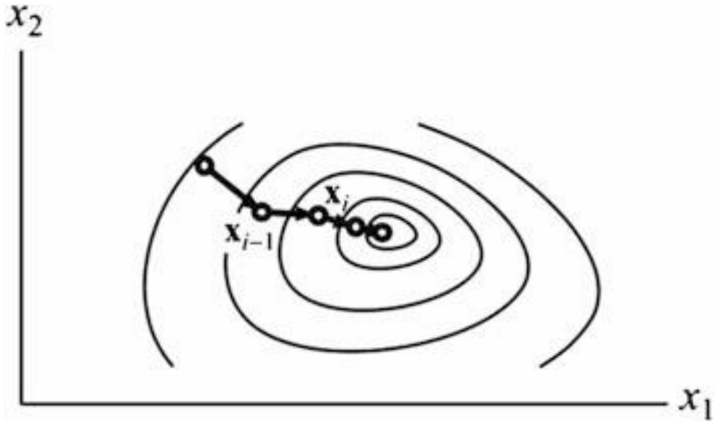


Figure 1.6 Iterative optimization method

Numeric techniques start from some point, x_0, and computes iteratively new points, x_i, from previous point, $x_{\mathrm{i}-1}$, by following the gradient until some stop condition is reached (see Figure 1.6).

Within the numeric optimization techniques two major categories are the Hessian-dependent approaches (such as the Newton's method) and the conjugate gradient/gradient descent) techniques. Often the derivatives are not directly evaluated but approximated using finite differences.

These iterative methods have two main drawbacks. Firstly, the selection of the start point heavily influences on the convergence of the method. Secondly, the found solution may be a local optimum instead a global one. Furthermore, for a successful application of most of the iterative methods, the objective function must be continuous and differentiable.

1.2.2.3 Heuristic Optimization Techniques

Many optimization problems in engineering, especially with regard to mechanical and manufacturing systems do not fulfill the conditions of continuity, differentiability and unimodality, required for applying the analytical or numeric methods. For solving this kind of problems many gradient-free techniques, so called heuristics methods, have been developed.

Heuristic optimization, unlike analytical or numeric methods, do not rely on a solid mathematic foundation; they are inspired on natural systems, either physical or biological, and try to find near-optimal solutions, which although being different from the actual optimum, are good enough to be applied in practical situations. Currently there are a lot of heuristics for optimization. A thorough revision goes beyond the scope of this Dissertation. The reader may find a deep review in [87]. Some of the most popular ones from the viewpoint of monitoring and control systems are briefly described as follows.

- Simulated annealing is based on the annealing processes in metals and other lattice structures, where the systems are conduced to a minimum energy state. The method starts from some point creating and evaluating some neighbor points in each iteration. If the objective function is lower in the new point than in the actual one, this is replaced. However, this replacement is not deterministic by random, depending on some prescribed parameter called temperature by analogy with the physical annealing process [88].

- Evolutionary optimization is inspired by the natural evolution of biological organisms. These methods carry out parallel searches starting from an initial solution set (called population) and create, in each iteration, a new *child* population that *inherits* the characteristics of the best ***parents***. Both, the selection of the *parents* and the creation of *children* include random processes. Evolutionary algorithms include two main branches, evolutionary strategies (European school) and genetic algorithms (American school) [89].
- Swarm intelligence is inspired by the behavior of natural decentralized systems, composed by a group of individuals which work together to achieve some common goal. This paradigm comprises a lot of algorithms, including the most popular, but not limited to: ant colony optimization, cuckoo search and particle swarm optimization [90].

All these methods are strongly influence by randomly procedures that lead to local minimum instead of the global one. For this reason, these approaches are often referred to as stochastic optimization.

1.2.3 Estimation-of-distribution algorithms

In this chapter, the attention is concentrated on Estimation-of-distribution algorithms. The current literature on Estimation-of-distribution algorithms contains an abundant range of deterministic and stochastic methods for solving multi-objective optimization problems [91, 92]. Figure 1.7 shows the basic steps to formulate a multi-objective optimization for a general process. The optimization of physical processes involves the use of models, represented by functions that will never fulfill the conditions of continuity, differentiability and unimodality, which are usually required for conventional analytical and numerical techniques [93]. The alternative is the use of heuristics for optimization, based on soft computing techniques [94]. Soft-computing is especially useful in multi-objective optimization when several different and often interconnected objectives are considered [95].

In fact, many optimization methods can be applied ranging from genetic algorithms to particles warm optimization [3, 5, 96-100]. Evolutionary algorithms have demonstrated their suitability as a method for multi-objective optimization [101]. EAs store a set of simultaneous trade-off solutions with the potential to exploit the synergies of a parallel search across all possible solutions. However, EAs are usually experimentally assessed through various test problems, because an analytical assessment of their behavior is very complex. Thus, their performance on random problems cannot be guaranteed prior to application [102, 103]. Estimation-of-Distribution Algorithms have emerged in the middle ground between Monte-Carlo simulation and EAs [7, 8]. A probabilistic model based on elite individuals is built and subsequently sampled to produce a new population of better individuals. One of the main advantages of EDAs is that the fusion of prior information into the optimization procedure is straight forward, thereby reducing convergence time when such information is available. From the standpoint of computational costs, it involves fewer heuristics than the other gradient-free optimization methods [104].

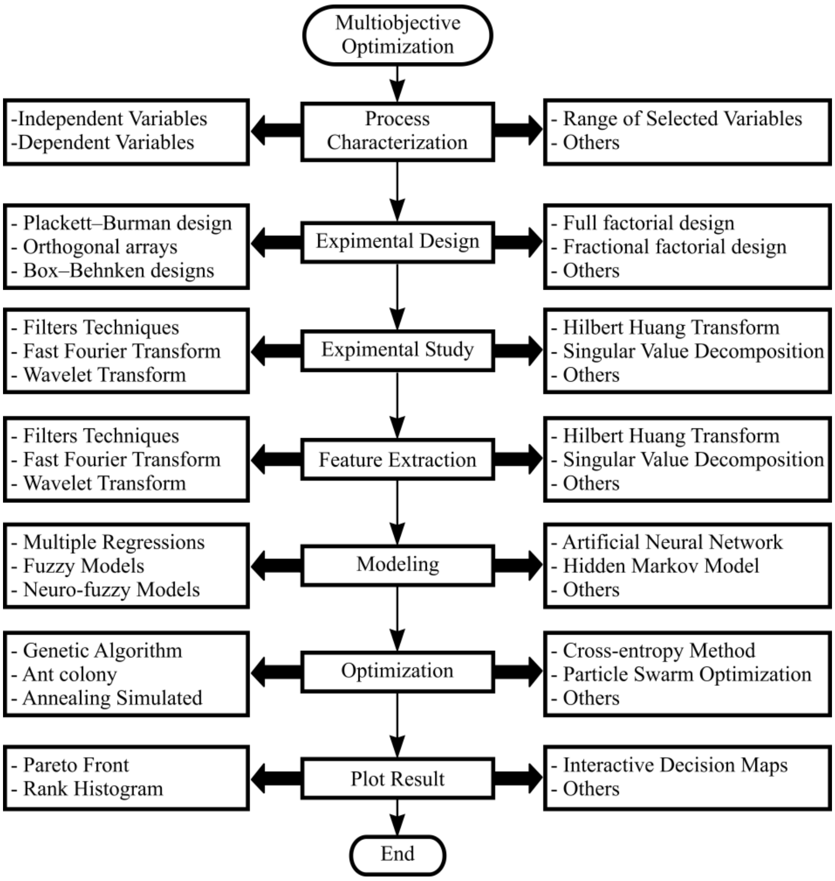


Figure 1.7 Classic procedure for optimization of physical processes

Among the broad range of optimization possibilities for computational architectures, optimal tuning of the parameters was adopted, rather than the optimization of the structure or topology. It is computationally simpler and sometimes brings better results than non-linear systems [105]. One of the main applications of these techniques is the optimal setting of the parameters (scaling factors or gains) for non-trivial and sometimes intractable tasks [106].

The optimal setting of fuzzy controller strategies based on stochastic gradient-based optimization has been reported in different works [10, 107, 108]. However, many of these optimization techniques have not been applied to real industrial processes yet, due to the high complexity of optimization algorithms, inappropriate cost functions and performance indexes, insufficient performance, and limited empirical formulas in industrial context. Cross-Entropy method [109, 110] is indeed a good candidate for multi-objective optimization. The most attractive feature of cross entropy method is that, for a certain family of instrumental densities, the updating rules can be analytically calculated, making them extremely efficient and fast. Moreover, the theoretical background of CE enables theoretical studies of the method, which can provide sound guidelines on the potential applications of this algorithm in artificial cognitive control architectures.

The concept of cross-entropy has been thoroughly addressed in the literature from different perspectives. For instance, very promising results have recently been described in the literature in relation to cross entropy forum of certain variables and the minimum cross-entropy principle for uncertain optimization and machine-learning problems [11]. The concept has also served as a foundation for developing a cross-entropy clustering method [111]. The optimal parameter settings are still an open research issue, mainly due to their varied influence on improving process behaviors. Evolutionary-based optimization algorithms have already shown improvements in performance with very promising results for iterative feedback tuning methods indiscrete-time single-input single-output systems currently reported in the literature [112]. On-going research in this field includes new methods derived from those that are already well-established such as the five-stage adaptive PSO algorithm proposed by [107]. Moreover, Fu, et al. [113] applied the cross-entropy method to optimize the scaling factors and the membership functions of fuzzy controllers and guarantee fail-safe navigation of unmanned autonomous vehicles. Finally, Giagkiozis, et al. [7] and Bekker and Aldrich [114] recently published two seminal papers on cross-entropy optimization.

1.3 Machine learning techniques. General concepts

1.3.1 General aspects

Machine learning (ML) is the subfield of computer science that gives computers the ability to learn without being explicitly programmed [12]. Evolved from the study of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data [115]. Machine learning can be used for several different software data analytics tasks, providing useful insights into software processes and products. For example, it can reveal what software modules are most likely to contain bugs, what amount of effort is likely to be required to develop new software projects, what commits are most likely to induce crashes, how the productivity of a company changes over time, how to improve productivity, etc. The right machine learning algorithm depends on the data and the environment being modeled. Therefore, in order to create good data models, it is important to investigate the data analytics problem in hand before choosing the type of machine learning algorithm to be used [116].

Machine learning is closely related to computational statistics, which also focuses in prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Furthermore, ML sometimes converges to data mining techniques, where the latter subfield focuses more on exploratory data analysis and it is more concentrated on supervised learning [117, 118]. Besides, it can also be unsupervised and be used to learn and establish baseline behavioral profiles for various entities and then used to find meaningful anomalies [119]. Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics [120, 121]. These analytical models allow researchers, data scientists, engineers, and analysts to produce reliable, repeatable decisions and results and uncover hidden insights through learning from historical relationships and trends in the data. Machine learning tasks are typically classified into three broad categories, depending on the nature of the learning signal or feedback available to a learning system [122]:

- Supervised learning consists in inputs-outputs data used as examples, given by a teacher, and the goal is to learn a general rule that maps inputs to outputs.
- Unsupervised learning, on the contrary, no labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).
- Reinforcement learning, as particular case of unsupervised learning, a computer program interacts with a dynamic environment in which it must perform a certain goal, such as: driving a vehicle or playing a game against an opponent. The program produces a feedback in terms of rewards and/or penalties as it navigates its problem space.

Between supervised and unsupervised learning is semi-supervised learning, where the teacher gives an incomplete training signal: a training set with some (often many) of the target outputs missing. Transduction is a special case of this principle where the entire set of problem instances is known at learning time, except the specific targets which are missing.

Developmental learning, elaborated for robot learning, generates its own sequences (also called curriculum) of learning situations to cumulatively acquire repertoires of novel skills through autonomous self-exploration and social interaction with human teachers and using guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

Finally, bioinformatics, brain interfaces, computer vision, pattern recognition, game theory, medical diagnosis economics, natural language processing, optimization and metaheuristic, robot locomotion, search engines, sequence mining, stock market analysis, user behavior analytics are the principal application fields of the machine learning techniques [123-126].

1.3.2 Reinforcement learning algorithms

In reinforcement learning, an agent tries to maximize the accumulated reward over its life-time. In an episodic setting, where the task is restarted after each end of an episode, the objective is to maximize the total reward per episode. If the task is on-going without a clear beginning and end, either the average reward over the whole life-time or a discounted return (i.e., a weighted average where distant rewards have less influence) can be optimized. In such reinforcement learning problems, the agent and its environment may be modeled being in a state $s \in S$ and can perform actions $a \in A$, each of which may be members of either discrete or continuous sets and can be multidimensional. A state s contains all relevant information about the current situation to predict future states (or observables); an example would be the current position of a robot in a navigation task1. An action a is used to control (or change) the state of the system. For example, in the navigation task we could have the actions corresponding to torques applied to the wheels. For every step, the agent also gets a reward r, which is a scalar value and assumed to be a function of the state and observation [127].

The reinforcement learning agent needs to discover the relations between states, actions, and rewards. Hence exploration is required which can either be directly embedded in the policy or performed separately and only as part of the learning process. Classical reinforcement learning approaches are based on the assumption that we have a Markov Decision Process (MDP) consisting of the set of states S, set of actions A, the rewards R and transition probabilities T that capture the dynamics of a system. Transition probabilities (or densities in the continuous state case) $T(s', a, s) = P(s'| s, a)$ describe the effects of the actions on the state. Transition probabilities generalize the notion of deterministic dynamics to allow for modeling outcomes are uncertain even given full state. The Markov property requires that the next state s' and the reward only depend on the previous state s and action a [13], and not on additional information about the past states or actions. In a sense, the Markov property recapitulates the idea of state, a state is a sufficient statistic for predicting the future, rendering previous observations irrelevant.

Different types of reward functions are commonly used, including rewards depending only on the current state $R = R(s)$, rewards depending on the current state and action $R = R(\mathrm{s, a})$, and rewards including the transitions $R = R(s', a, s)$. Most of the theoretical guarantees only hold if the problem adheres to a Markov structure, however in practice, many approaches work very well for many problems that do not fulfill this requirement.

The goal of reinforcement learning is to discover an optimal policy π_p that maps states (or observations) to actions to maximize the expected return J_{max}, which corresponds to the cumulative expected reward. There are different models of optimal behavior [128] which result in different definitions of the expected return. A finite-horizon model only attempts to maximize the expected reward for the horizon *H*, i.e., the next *H* (in time) steps *i*:

$$J_{max} = \sum_{i=0}^{H} R_i \tag{1.3}$$

This setting can also be applied to model problems where it is known how many steps are remaining.

Alternatively, future rewards can be discounted by a discount factor γ ($0 \leq \gamma < 1$)

$$J_{max} = \sum_{i=0}^{\infty} \gamma^i \bullet R_i \tag{1.4}$$

This is the setting most frequently discussed in classical reinforcement learning texts. The parameter γ affects how much the future is taken into account and needs to be tuned manually. As illustrated in [128], this parameter often qualitatively changes the form of the optimal solution. Policies designed by optimizing with small γ are *myopic* and *greedy*. It is straightforward to show that the optimal control law can be unstable if the discount factor is too low (e.g., it is not difficult to show this destabilization even for discounted linear quadratic regulation problems).

Off-policy methods learn independent of the employed policy, i.e., an explorative strategy that is different from the desired final policy can be employed during the learning process. *On-policy* methods collect sample information about the environment using the current policy. As a result, exploration must be built into the policy and determines the speed of the policy improvements. Such exploration and the performance of the policy can result in an exploration-exploitation trade-off between long- and short-term improvements of the policy. Modeling exploration models with probability distributions has surprising implications, e.g., stochastic policies have been shown to be the optimal stationary policies for selected problems [129, 130] and can even break the curse of dimensionality [126]. Furthermore, stochastic policies often allow the derivation of new policy update steps with surprising ease.

A wide variety of methods of value function based reinforcement learning algorithms that attempt to estimate the optimal value function (*V**(*s*')) or the state-action value function Q*(*s*, *a*) have been developed. They can be split mainly into three classes [131]: (i) dynamic programming based optimal control approaches such as policy iteration or value iteration, (ii) rollout-based Monte Carlo methods and (iii) temporal difference methods such as: Temporal Difference learning (TD), Q-learning, and State-Action-Reward-State-Action (SARSA).

In particular, Q-learning is a model-free reinforcement learning technique. Specifically, Q-learning can be used to find an optimal action-selection policy for any given (finite) Markov decision process [132]. It works by learning an action-value function that ultimately gives the expected utility of taking a given action in a given state and following the optimal policy (*off-policy learner*) thereafter. When such an action-value function is learned, the optimal policy can be constructed by simply selecting the action with the highest value in each state [133]. One of the strengths of Q-learning is that it is able to compare the expected utility of the available actions without requiring a model of the environment. Additionally, Q-learning can handle problems with stochastic transitions and rewards, without requiring any adaptations. It has been proven that for any finite MDP, Q-learning eventually finds an optimal policy, in the sense that the expected value of the total reward return over all successive steps, starting from the current state, is the maximum achievable.

The algorithm therefore has a function that calculates the Quantity of a state-action combination (Q: $S \times A \rightarrow \mathbb{R}$). Before learning has started, Q returns an (arbitrary) fixed value, chosen by the designer. Then, each time the agent selects an action, and observes a reward and a new state that may depend on both the previous state and the selected action, Q is updated as it is described in Eq. (1.5). The core of the algorithm is a simple value iteration update. It assumes the old value and makes a correction based on the new information [134].

$$Q(s_{t+1}, a_{t+1}) \leftarrow Q(s_t, a_t) + \alpha \bullet (R_{t+1} + \gamma \bullet \max_{a \in A} Q(s_{t+1}, a_t) - Q(s_t, a_t)) \tag{1.5}$$

where, r_{t+1} is the reward observed after performing a_t in s_t, and α is the learning rate ($0 < \alpha \leq 1$).

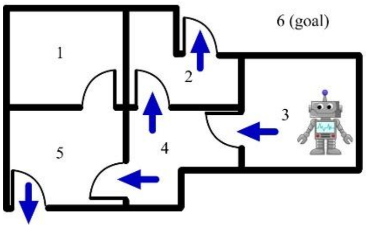
The learning rate or step size determines to what extent the newly acquired information will override the old information. A factor equal to 0 makes the agent not learn anything, while a factor of 1 would make the agent consider only the most recent information. In fully deterministic environments, a learning rate of $\alpha = 1$ is optimal. When the problem is stochastic, the algorithm still converges under some technical conditions on the learning rate that require it to decrease to zero. In practice, often a constant learning rate is used between $\alpha = 0.1 \dots 0.3$ [126].

In a real implementation of the Q-learning algorithm, two possible scenarios can be found in function of the rewards: R is known (e.g. game theory, localization problems, etc.) [135, 136] or R is estimated for each iteration (e.g. industrial applications) [82, 137]. The first scenario where the goal find the best route to way out, starting from a random s is shown in the Figure 1.8. As it can be observed, after 100 iterations stating from an s = 3 the best route estimated for the algorithms are s(3,4,2,6) or s(3,4,5,6). In the second scenario, the main modification from the general implementation (r is known) is assume that *a* state is a set of parameters of the model/models and the estimation of the reward in function of the possible actions in each state.

Each parameters K_i has its own limits ($K_i^{\min}; K_i^{\max}$), considering m_K possible values of each parameter, the range of values of this parameter would be:

$$K_{i1} = K_i^{\min}, K_{i2} = K_{i1} + \frac{K_i^{\max} - K_i^{\min}}{m_K - 1}, \ldots, K_{im_K} = K_i^{\max} \text{c} \tag{1.6}$$

As a consequence of the exposed above, the space of states is finite whose dimension is m_K^{n}. For a given state $s_t \rightarrow \left\langle K_1^t, K_2^t, ..., K_n^t \right\rangle$ its available actions will be those that change s_t to $s_{t+1} \rightarrow \left\langle K_1^{t+1}, K_2^{t+1}, ..., K_n^{t+1} \right\rangle$, namely in each action only one step in each parameter can be done.



Rewards matrix

$$R = \begin{array}{c|cccccc} S\backslash A & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & -1 & -1 & -1 & -1 & 0 & -1 \\ 2 & -1 & -1 & -1 & 0 & -1 & 1 \\ 3 & -1 & -1 & -1 & 0 & -1 & -1 \\ 4 & -1 & 0 & 0 & -1 & 0 & -1 \\ 5 & 0 & -1 & -1 & 0 & -1 & 1 \\ 6 & -1 & 0 & -1 & -1 & 0 & 1 \end{array}$$

After $t = 100$ iterations

$$Q = \begin{array}{c|cccccc} S\backslash A & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 7.6 & 0 \\ 2 & 0 & 0 & 0 & 0.6 & 0 & 98.5 \\ 3 & 0 & 0 & 0 & 0.6 & 0 & 0 \\ 4 & 0 & 7.6 & 0.5 & 0 & 7.6 & 0 \\ 5 & 0.6 & 0 & 0 & 0.6 & 0 & 98.5 \\ 6 & 0 & 7.6 & 0 & 0 & 7.6 & 98.5 \end{array} \times 10^{-3}$$

Figure 1.8 Q-learning algorithm applied to a localization problem (R is known)

Finally, RL resembles methods for classical optimal control, as well as dynamic programming, stochastic programming, simulation-optimization, stochastic search, and optimal stopping. Both RL and optimal control address the problem of finding an optimal policy (often also called the controller or control policy) that optimizes an objective function (i.e., the accumulated cost or reward), and both rely on the notion of a system being described by an underlying set of states, controls and a plant or model that describes transitions between states. However, optimal control assumes perfect knowledge of the system's description in the form of a model. For such models, optimal control ensures strong guarantees, which, nevertheless, often break down due to model and computational approximations. In contrast, reinforcement learning operates directly on measured data and rewards from interaction with the environment. Reinforcement learning research has placed great focus on addressing cases that are analytically intractable using approximations and data-driven techniques [13].

1.4 Micro-manufacturing processes: A brief introduction

Nowadays, conventional industrial manufacturing has an enormous impact on the global economy. Large companies around the world have allocated funds and infrastructure to develop new machines, tools, sensors, and control systems, in order to increase their levels of production and competitiveness. Over the past two decades, the progressive growth of micro-manufacturing processes has been reported in the industrial sector. Some processes unrelated to manufacturing, such as lithography and etching may be traced back over the decades. Actually, micro-manufacturing can be defined as a collection of technologies that are used to make micro-devices or micro-scale components for a wide range of aerospace, medical, and electronic industrial applications. Micro-manufacturing largely uses non-traditional manufacturing methods, scaling down or modifying the traditional methods, as appropriate, to fully address micro-scale manufacturing issues. Overall, designers and manufacturers are obliged to consider new aspects, due to the miniaturization of tools and workpieces. Grain size, tool stiffness and workpiece weaknesses are often not taken into account in traditional macro-manufacturing; most are established and standardized over decades of industrial applications [138].

1.4.1 Micro-manufacturing methods and processes

Micro-manufacturing techniques are often categorized as micro-electro-mechanical systems (MEMS) manufacturing and non-MEMS manufacturing. In the last decades, the MEMS, optical-MEMS, radio frequency (RF-MEMS), Power-MEMS, Bio-MEMS have been extended into the micro-manufacturing methods. MEMS manufacturing largely involves techniques such as: photolithography, chemical etching, plating, lithography electroplating and moulding (LIGA), laser ablation, etc., while non-MEMS manufacturing often involves techniques such as: Electrical discharge machining (EDM), micro-mechanical cutting, laser cutting/patterning, micro-embossing, micro-injection moulding, micro-extrusion, microstamping among others.

Micro-manufacturing processes can also be classified in five categories as follows: subtractive, additive, mass containing, joining and hybrid process as shown in Figure 1.9, based on the report of [139]. For the sake of clarity, only some examples of micro-manufacturing techniques are depicted in Figure 1.9. In the following subsections, the additive, mass containing, forming, hybrid process and subtractive subgroups are briefly addressed. All the acronyms used in the Figure 1.9 are explained in the following sections.

Many additive processes in the micro-scale are associated with micro-electro-mechanical systems. Under this category, chemical vapor deposition (CVD); physical vapor deposition (PVD); stereolithography; and LIGA (German acronym: Lithographie Galvanoformung Abformung - Lithography, Electroplating and Molding) are some of the most widely reported in the industry and the scientific literature. The coating for wear resistance, corrosion resistance, high temperature protection and erosion protection applied to semiconductors industry, optical fibers and composites are the main application of the CVD process. In contrast, PVD is a coating technique in which material is transferred at the atomic level. A PVD coating is applied to improve hardness and wear resistance, to reduce friction, and to improve oxidation resistance increasing performance and lengthening component life cycles. In the context of MEMS, electron beam, ion beam, ion track and x-ray lithography are listed as possible techniques. In particular, the stereolithography is based on a rapid prototyping additive process, also called layered manufacturing uses ultraviolet or laser light to cure resins on a selective basis. LIGA is also an additive process used in the manufacture of complex microstructures with very high aspect ratios. By the other hand, 3D printers for extrusion, light polymerized, powder bed, laminated and wire are the most common types using materials from polymers, ceramics to complex metal alloys, composites and thermoplastics. Most of them are based in deposition, lithography and laser technologies having an accelerated application increase in the prototype industries.

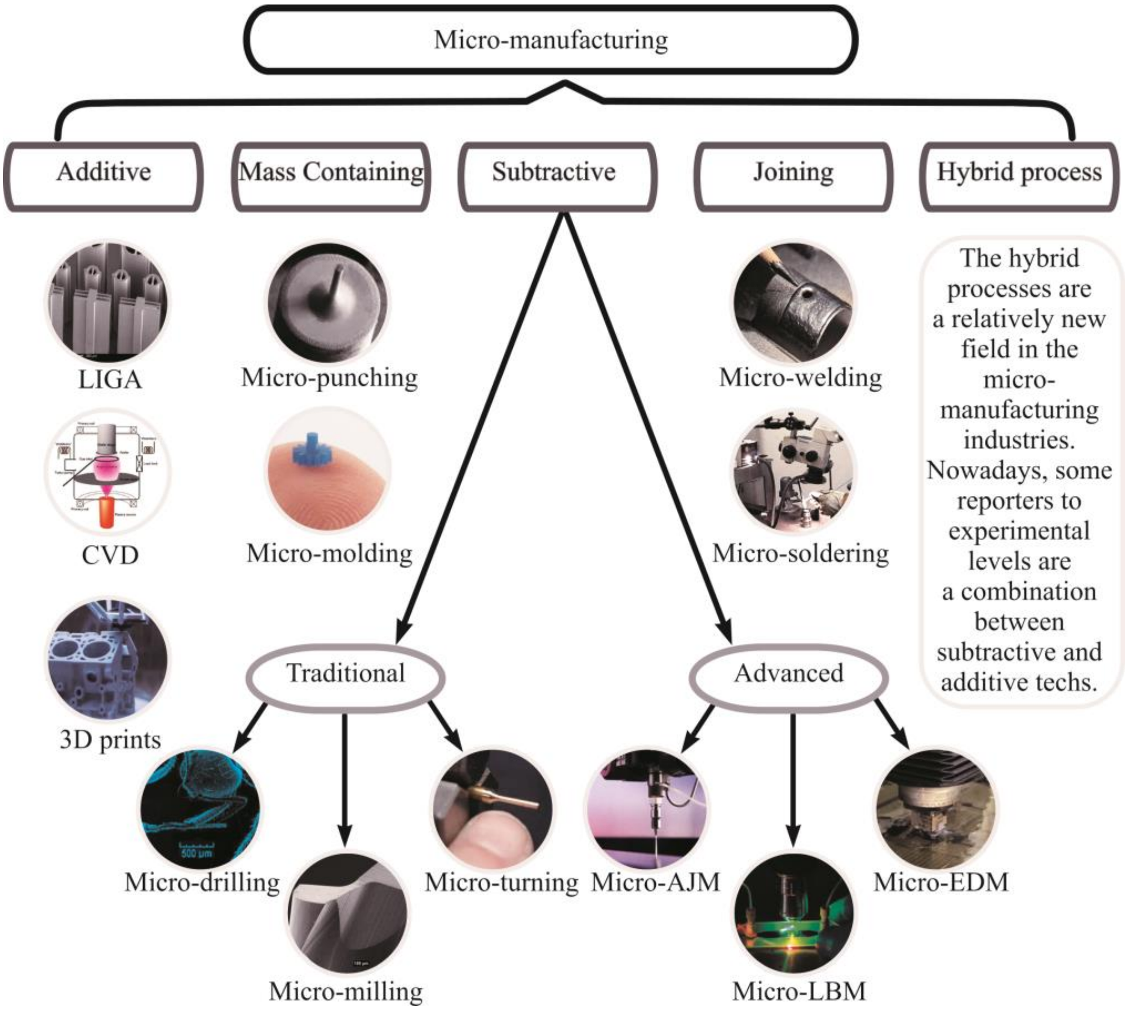


Figure 1.9 A short review of micro-manufacturing techniques

More challenges arise when the sizes/features are reduced to tens or hundreds of microns. Major issues to be addressed include understanding of material deformation mechanisms and material/tool conditions, materials property characterization, process modeling and analysis, process design optimization, etc., with emphasis on the related size effects [140, 141]. Metal forming offers some attractive characteristics that are superior to those of other processes, e.g. machining and chemical etching, considering such features as higher production rates, better material integrity, less waste, lower manufacturing costs, etc. Furthermore, it is a key process in the fabrication of micro-tools and the preparation of micro-materials. Micro-casting processes can be classified into four different categories: capillary action, investment casting, vacuum pressure casting, and centrifugal casting. All of them are used in the production of metallic micro-workpieces with high mechanical strengths and a high aspect ratio [142].

Finally, micro-moulding can be classified as injection moulding, reaction injection moulding, hot embossing and injection compression moulding [143].

The micro-joining process has been an integral part of the microelectronics, pharmaceutical, aerospace and others micro-manufacturing industries for many decades. Micro-welding, the micro-soldering, the micro-brazing and the adhesive bonding belong to the category of manufacturing and assembly processes. These micro-joining operations can be mechanical connections, electrical connections, and optical coupling [144]. Electron beam is a popular process for macro-scale applications and it is characterized by a low thermal load, precise energy input, and fine beam manipulation capabilities. However, this method is also gaining importance in micro applications, because of its capability to focus the beams exactly within a diameter of a few microns. Laser welding is a non-contact process that requires only single-sided access [145]. Micro-adhesive bonding is also considered a micro-joining process. In the micro-scale workpiece, the assembly of micro-parts by the use of mechanical means such as screws, rivets, pins are an uncommon solution, becoming micro-adhesive bonding a feasible solution in these cases. Electronics in automotive components, high density packing of microelectronics, fiber-optic couplers, electro-optic transducers telecommunications, biotechnology and high definition sealing in microfluidics are the main applications of this process [146].

Nowadays, hybrid machines are one of the leading programs for researchers and manufacturers around the world. The latest concept for hybrid machine development is focused on the use of a single workstation for manufacturing micro-components. The idea is the reduction or the elimination of as many post-assembly operations as possible, which may involve changes in handling and high precision positioning [147, 148]. Ultra-precision manufacturing of self-assembled microsystems is another example of this emerging development, which combines ultra-precision micro-machining such as milling, turning, drilling, and grinding with sacrificial/structural multilayer manufacturing processes to produce self-assembled, 3D micro-systems and associated meso-scale interfaces from a variety of materials for MEMS applications [149] .

1.4.2 Subtractive micro-manufacturing processes

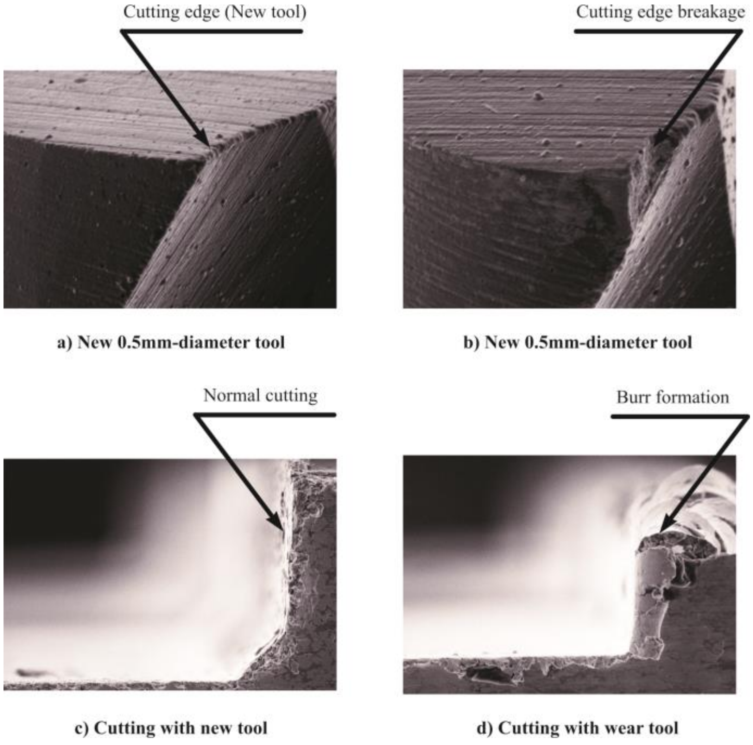
Special interest is put on the subtractive operations based on the case study presented in the doctoral thesis. The subtractive micro-manufacturing processes can be classified in two classes: traditional and advanced. Micromachining implies the removal of material in the form of chips or debris with tool sizes in the range of 1 mm to 999 microns. It may be also seen as an ultra-precision material removal process that is able to achieve micro-form accuracy and roughness of several nanometers [150, 151]. These operations are still powerful technologies for the development of micro-components with systems that use electronic, mechanical, fluidic, optical and radiative signals. The micro-instrumentation, inertial sensing; biomedical devices, wireless communication, high density data storage, as well as producing dies, micro-forming and injection moulding are among the main development lines [152] .

1.4.2.1 Traditional micromachining processes

Micro-turning, micro-milling, micro-drilling, micro-grinding, etc. are considered traditional micromachining processes. These operations are the result of the micro-scaled and continuous improvements in the design of the new machine-tool families. Several types of cutting processes are suitable for micromachining. Typical examples of traditional micro-cutting are drilling of micro-holes, milling of microgrooves and micro 3D shapes, and turning of micro-pins [153].

The most attractive advantage of traditional micro-cutting technologies is the possibility of machining 3D microstructures characterized by a high aspect ratio and comparatively high geometric complexity. Traditional micro-cutting processes involve critical issues: cutting force must be as low as possible, rigidity of the machine tool should be high enough to minimize machining errors and tool edge radius must be smaller than the dimension of the feature that will be created.

In particular, the capability of micro mechanical machining especially micro milling to manufacture a wide range of workpiece materials and complex three-dimensional geometries makes it one of the best candidates to produce the micro parts. The material removal in micro-milling differs from macro-milling processes, due to the presence of minimum chip thickness, size effect, elastic recovery and the ploughing mechanism. These effects must be taken into consideration in micro-milling research.



a) New 0.5mm-diameter tool

b) New 0.5mm-diameter tool

c) Cutting with new tool

d) Cutting with wear tool

Figure 1.10 Micro-milling tools and operations measured by NeoScope JCM-5000

A new 0.5mm-diameter micro-end-milling tool was shown in Figure 1.10a). Chattering and run-out are normally presented during the cutting process as factors that cause external/internal perturbation of tool-workpiece vibrations [154, 155]. Furthermore, micro-cutting tools are tiny, fragile and easily broken by excessive deflections, forces, and vibrations (see Figure 1.10b)). Therefore, appropriate cutting tool geometry and cutting conditions must be selected in micro milling (see Figure 1.10c)) [156]. Finally, most micro products such as: medical equipment, micro-mould, micro tubular components need a high quality finished surface. Avoidance of cutting radius breakage and increments in vibrations are crucial to good surface roughness and will also lessen tool wear. Besides, researchers have found that burr size and surface roughness increase at a low ratio of feed per tooth to cutting edge radius [157]. Specifically, the formation of a burr during micro-end-milling is associated with such factors as: size effect, comparatively large edge radius and minimum chip thickness, etc. [158]. In particular, burr formation shown in the Figure 1.10d) of tool-radius breakage during the cutting process.

1.4.2.2 Non-traditional micromachining processes

Non-traditional micromachining processes aim to deal with stringent parts requirements used in high-tech industries in relation to new materials that are difficult to work with. Another interesting change arising from progress in micro-manufacturing is the use of non-conventional material removal procedures. Instead, these methods make direct use of some form of energy for micromachining. These processes may be classified into different groups on the basis of the working principle [139].

The first group consists of mechanical micromachining processes: ultrasonic micromachining (USM), abrasive jet micromachining, abrasive water-jet micromachining and water-jet micromachining employing the kinetic energy of either abrasive particles or a water jet or both to remove waste material from a workpiece. The cutting process of brittle and ductile materials has some differences, being determinant factors the material hardness, strength and other mechanical properties of the workpiece material in the process performance [159, 160].

Another group is composed of beam-based micromachining processes, utilizing different forms of thermal energy such as: electron-beam [161], laser beam [162], the heat energy of sparks in electric-discharge, plasma-arc and the kinetic energy of ions in ion-beam micromachining [163]. The thermal energy (except in ion beam) is concentrated on a small area of workpiece, resulting in melting or vaporization or both. These processes are widely used for machining hard and tough materials. Electrical discharge machining is also considered a beam-based micromachining technique. The manufacturing of micro-components by thermal material removal mechanism allows almost force-free process machining independently of the mechanical properties of the processed material. For this reason, EDM may be applied to functional materials such as: hardened steel, cemented carbide and electrically conductive ceramics with a high precision [164]. Its applications have extended so far beyond dies and moulds fabrication such as micro-gears, microfluidic devices, medical implants, etc. [165].

The last group is composed of chemical or electrochemical micromachining processes, including mainly electrochemical (micro-ECM) and photochemical (micro-PCM). Micro-ECM has numerous advantages: there is no tool wear, no residual stresses developed in the machined workpiece, its performance does not depend on the physical and mechanical properties of the work material and using an anodic dissolution process yields a high material removal rate. Other attractive characteristics include burr-free surfaces, no thermal damage, and no distortion of the part. Based on its earlier descriptions, this process has a wide field of applications [166-168]. However, the work material should be electrically conductive. Moreover, micro-PCM is a type of etching process in which the workpiece is selectively etched by masking the area where no machining is required. The selection of an appropriate etchant depends on the properties of the workpiece material to be machined [169].

1.5 Conclusions

In this chapter a review of the state of the art in of the field of cognitive architectures is presented with strong emphasis on those architectures and frameworks for monitoring and control. Moreover, gradient free optimization techniques and reinforcement learning strategies are roughly outlined. The study is also centered on the necessary techniques to enable self-capabilities in artificial cognitive architectures for industrial applications. During the investigation, the embryonic state of this type of solutions for real industrial setup was demonstrated according to the scientific and technical literature considered in this Thesis.

Furthermore, the review of some optimization techniques, relevant from system engineering viewpoint, was done. Among them, estimation of distribution algorithms is considered with attention on cross entropy technique to enable the self-optimization capability. Subsequently, the importance to include a reinforcement learning algorithm to enable self-learning capabilities in the architecture is also pointed out.

Finally, a review of the different micro-manufacturing processes was done, explaining the principal applications fields and the impact in of the micromachining processes in the modern industry.

In the next chapter, different techniques for analytical, numerical and empirical modeling are described. Modeling is one of the cornerstones of the theoretical and experimental research for extracting patterns and correlating the process behavior with representative signals and variables of micromachining processes.

Chapter 2
MODELING TECNIQUES FOR MICROMACHINING PROCESSES

Modeling is essential from system engineering viewpoint to drive quality improvement in physical processes. Therefore, modeling of micro-scale machining processes is a key issue for efficient manufacturing with dozens of worldwide reports. Good models not only reduce the need for expert operators, thereby lowering costs, but it also decreases the probability of unexpected tool breakage, which may involve damage to the workpiece or, even, to the machine-tool. Process monitoring is also of immense importance in view of the tiny tool diameters used in micro-mechanical machining and the high surface roughness quality desired in this operations. Some numerical, analytical and empirical modeling techniques will be analyzed in this chapter, concluding with a set of empirical micromachining models (i.e., micro-drilling and micro-milling) obtained in different stages of the research. Furthermore, the experimental setup used in the multiples case studies is also described in the present chapter.

The chapter is divided in six sections. Firstly, an overview of some of the most commonly used analytical, numerical, statistical, and intelligence-based techniques for modeling micromachining processes is presented in section 2.1. Following, the experimental setup (Kern Evo Ultra Precision Center and sensory equipment) used during the research is described in section 2.2. Subsequently, the forces and vibrations influence in function of the nominal cutting parameters are modeled in section 2.3. Furthermore, two models to estimate the run-out and the hole quality in micro-drilling processes were included in this section. After that, the surface roughness behavior is modeled in micro-milling processes for tungsten–copper alloys (section 2.4). In section 2.5, a summary of the all models obtained during the research period is presented. Finally, conclusions of this chapter are presented in section 2.6.

2.1 Modeling techniques of micromachining processes

2.1.1 General aspects

A conceptual model is a model made of the composition of concepts, which are used to help people know, understand, or simulate a subject the model represents. The term conceptual model may be used to refer to models which are formed after a conceptualization or generalization process [170]. A conceptual model's primary objective is to convey the fundamental principles and basic functionality of the system in which it represents. Moreover, a conceptual model must be developed in such a way as to provide an easily understood system interpretation for the models users [171]. Data flow, Metaphysical, Logical, Business process, Statistical models, Mathematical models are the most common reported in the literature.

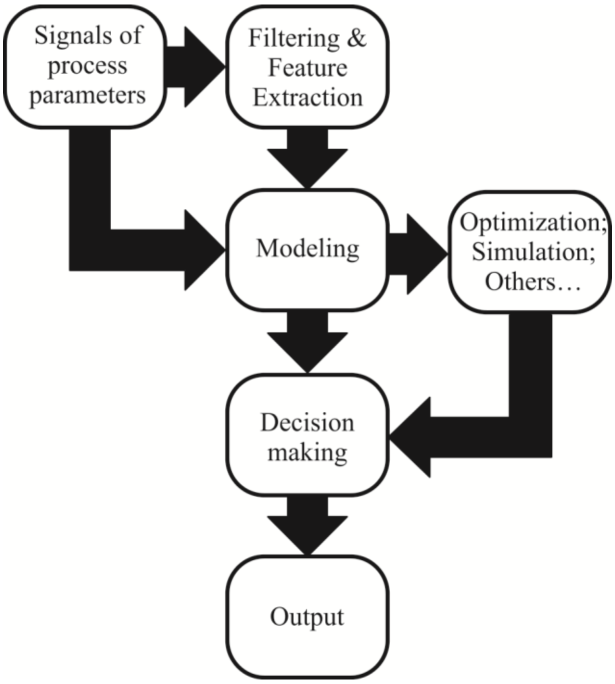

Signals of process parameters
Filtering & Feature Extraction
Modeling
Optimization; Simulation; Others…
Decision making
Output

Figure 2.1 General scheme of modeling process and utility

In last decades, Computer Sciences and soft-computing techniques open up a new arsenal of methods for modeling processes. The computational model is a mathematical model that requires computer resources to estimate the behavior of a system by means of computer simulation. The system under study is often a complex nonlinear system for which simple, intuitive analytical solutions are not readily available. Examples of common computational models are weather forecasting models, earth simulator models, flight simulator models, molecular protein folding models, and neural network models. Artificial cognitive-based models are usually focused on a single cognitive phenomenon or process (e.g., learning), how two or more processes interact (e.g., visual search and decision making), or how to make behavioral predictions for a specific task or tool.

Undoubtedly, one of the most important steps for monitoring and control is to obtain at least a rough model to identify the relationship between the dependent and independent variables Figure 2.1 showed the general flow scheme with the modeling step as key issue. First of all, it is important to choose the correct input signals if you want to produce an appropriate model. Sometime, the input signals captured during a real process have a high level of noise or contain insufficient information. For this reason, a filtering or feature extraction is needed before the modeling step. In engineering, modeling can be divided into two major groups (see Figure 2.2): phenomenological modeling including analytical and numeric solutions and empirical modeling where statistical and Artificial Intelligence-based methods are very representative and useful [172]. Once the modeling phase is finished, other tasks are indeed facilitated depending on the model characteristics. For example, different simulation studies can be conducted, optimization tasks can be run, and even decision making can be performed on the basis of the model behavior (see Figure 2.1). Finally, the correlation coefficient and generalization capability influence on the model precision, being determinant metrics to select the correct model to represent a simulated or real process.

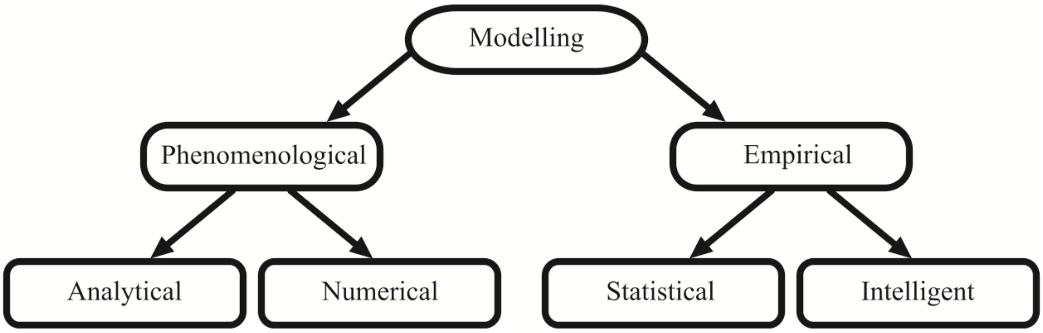


Figure 2.2 Classification of modeling techniques based in [172]

2.1.2 Filtering and feature extraction techniques for micromachining processes

In signal processing, a filter is a device or process that removes from a signal some unwanted component or feature. Filtering is a class of signal processing, the defining feature of filters being the complete or partial suppression of some aspect of the signal [173]. Most often, this means removing some frequencies and not others in order to suppress interfering signals and reduce background noise. However, filters do not exclusively act in the frequency domain; especially in the field of image processing many other targets for filtering exist. There are different criteria for classifying filters with evident overlapping, therefore there is no simple hierarchical classification. Filters may be classified as [174]: linear or non-linear; time-invariant or variant; causal or not-causal; analog or digital; discrete-time or continuous-time; passive or active and infinite impulse response (IIR) or finite impulse response (FIR). Some terms used to describe and classify linear filters is the frequency response [175]: low-pass filter; high-pass filter; band-pass filter and band-stop filter are the most common. Besides, notch filter, to rejects just one specific frequency; comb filter; cutoff frequency; transition band and ripple is also reported in the literature. Other filters such as the infinite impulse response and the finite impulse response are intensively applied [176].

Feature extraction is essential for reducing the amount of resources required to describe a large set of data. When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power or a classification algorithm which overfitting the training sample and generalized poorly to new samples [177, 178]. The features can be extracted in the time, frequency and amplitude domains or with a combination between them [179]. Data types is an important characteristic to select a feature extraction method [180-182]. In the particular case of the manufacturing processes techniques such as temporary statisticians, time series, Fast Fourier transform (FFT), Wavelet transform (WT) and Hilbert-Huang transform (HHT) among others are the most widely applied.

Statistics in the time domain are based on the application of statistical functions such as mean, minimum (MIN), maximum (MAX), standard deviation (STD), root mean square (RMS), skewness (SKEW) or the kurtosis (KURT). It is important to say that in most of the articles reviewed time domain methods are often combined with two or more techniques to guarantee completeness of the information extracted. In micromachining processes, feature extraction is quite frequently supported on the force signals [183], but some papers also report the use of the RMS component in vibration [184] and acoustic emission signals [185].

Features in frequency domain are determined through methods to estimate the energy distribution on the frequency spectrum. The fast Fourier transform is used to generate a spectral density function of energy. However, the FFT are not suitable for analyzing nonstationary signals. Nevertheless, the FFT is one of the most reported techniques to process signal captured by force, vibration and acoustic emissions sensors during micromachining [186, 187].

In mathematics, a wavelet series is a representation of a square integral function with real or complex values by a certain orthonormal series generated [188]. Nowadays, wavelet transform is one of the most popular of the time-frequency analysis. The WT shows an advantage over the Fourier transform, the temporal resolution. That is the ability to capture the frequency and location information (on time). Furthermore, the WT of a signal is calculated through a series of filters (filter bank), decomposing simultaneously using a low pass and a high pass filter [189]. It is important that the two filters are related to each other and this is known as the mirror filter numerical integration. Unlike the FFT, the WT has the ability to build a time-frequency diagram of a signal behavior [190].

According to Huang and Shen [191] past applications of the Hilbert transform are limited only for narrow-band signals. Today, the real advantage of this transformation became evident when was introduced the empirical mode decomposition (EMD) dealing with nonlinear and nonstationary processes. Hilbert-Huang transform is divided into two parts: an empirical mode decomposition and Hilbert spectral analysis. This method is strong in the nonstationary and nonlinear data analysis, especially in the amplitude-time-frequency domains for micromachining processes [192, 193].

Others techniques reported with a lesser extent are the time series. They have the advantage of do not requiring tedious calculations for editing; compiling; debugging programs and acquisition of information quickly and easily. However, these features are less informative about the process and they are affected by noise and disturbances of the system. Although some authors have reported the used of these algorithms applied to micromachining processes [194]. Finally, the Table 2.1 described a comparison between three of the most used feature extraction techniques.

Table 2.1 Comparison between FTT, WT and HTT presented by [193]

Parameters	*Fast-Fourier*	*Wavelet*	*Hilbert-Huang*
Basis	A priori	A priori	Adaptive
Frequency	Convolution: global uncertainty	Convolution: regional uncertainty	Differentiation: local certainty
Presentation	Amplitude-frequency	Amplitude-time-frequency	Amplitude-time-frequency
Nonlinear	No	No	Yes
Nonstationary	No	Yes	Yes
Feature extraction	No	Continuous: Yes	Yes
Theoretical base	Theory complete	Theory complete	Empirical

2.1.3 Analytical modeling in micromachining processes

Analytical models are widely reported in macro-scale cutting process. Besides, a micromachining process is quite similar to a conventional cutting operation, i.e. all the geometrical features and the kinematic characteristics of the tool and the workpiece can be identified. However, downscaling all the phenomena, in order to apply the same theories in both the micro and the macro regimes has proven itself ineffective. There are features of machining and phenomena that are considerably different in micromachining that make no allowance for any simplifications; differences arise when considering the chip-formation process, the resulting cutting forces, surface integrity, and tool life [195]. Analytical models are considered the predecessors of numerical models, although they will never substitute them. Even today, lower and upper bound solutions, shear plane models, slip-line field models and shear zone models are still using analytical modeling [196].

Another important aspect is the chip flow in all wedged-tool machining processes, in theory, in a common way by two different cutting schemes termed orthogonal cutting (Figure 2.3a)) and oblique cutting (Figure 2.3b)). Several models for force prediction [197] , chip formation [198, 199], temperature [200] and flow stress [201] have been developed using the analytical modeling in micromachining processes, principally based in orthogonal cutting models [202].

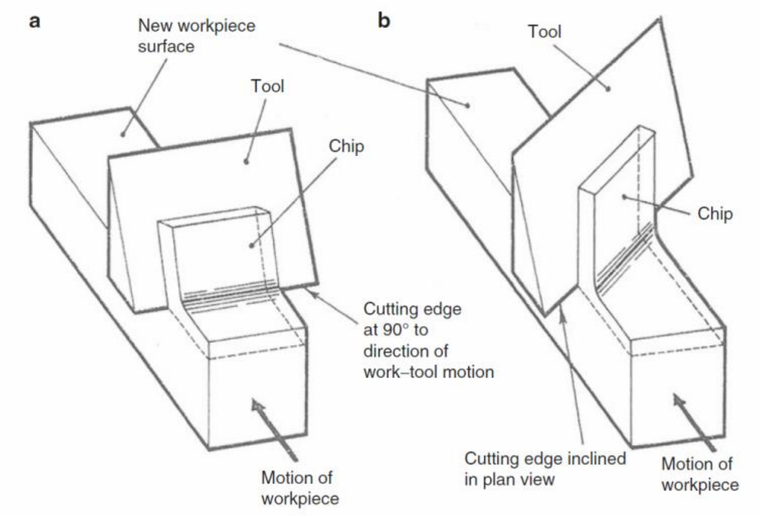

a
New workpiece surface
Tool
Chip
Cutting edge at 90° to direction of work–tool motion
Motion of workpiece
b
Tool
Chip
Cutting edge inclined in plan view
Motion of workpiece

Figure 2.3 Orthogonal cutting (a) and oblique cutting (b) conditions [153]

Besides, analytical models are reported when the influence on machining quality, efficiency, material removal rates, surface roughness and dimensional accuracy of the workpiece, even the tool and machine life are uncontrolled [203-205]. In almost all cases, efficient chatter vibration models will be critical to an understanding of chatter phenomena and to its prediction or avoidance during cutting processes [206, 207]. Likewise, Shi, et al. [208] demonstrated the effects of regenerative chatter on micro-end mill dynamics and stability behavior.

2.1.4 Numerical modeling of micromachining processes

Most engineering cases cannot be solved analytically and require a numerical solution. The numerical model usually needs to be carefully calibrated and validated against pre-existing data. Numerical modeling to solve forward and inverse problems has found extensive uses in industry. Forward problems include simulation of space shuttle flight, ground water flow, material strength, earthquakes, and molecular and medication formulae studies. Inverse problems consist of non-destructive evaluation, tomography, source location, image processing, and structural deformation during loading tests [209]. With the increase in computational technology, many numerical models and software have been developed for various engineering fields.

Specifically, in micromachining numerical techniques have grown in importance, in spite of the difficulties of modeling chip formation. Except for the physical phenomena explained above, two more challenges need to be addressed. The first one is to input accurate data into the model: this appears to be common sense, however, it can be problematic. The second is to choose a finite elements method [210], from among the different approaches or strategies proposed for metal machining modeling with FEM pertaining to formulation, friction treatment, material behavior, iteration schemes etc. used for approximating a solution [211-213].

Over the past decade, micromachining has established itself as a very important micro-manufacturing process. Compared with other micro-manufacturing processes, such as advanced machining, its prominence is partly because, it can provide complex shapes in a wide variety of materials [214]. Many modeling and simulation techniques have been applied in micromachining and FEM is one of them. However, in the micro-scale some differences are considered with respect to macro scale processes. For instance, the assumption of a perfectly sharp cutting tool is non-realistic when micromachining is studied. In metal cutting, size effect, the non-linear increase in the specific energy and the depth of cut, influences process parameters, e.g. the minimum cutting edge radius, and therefore the analysis of the size effect is very important [215-217].

Various 3D FEM models that model micromachining processes have been reported in the literature. One example is the three-dimensional finite element for micro-cutting simulation based on the concept of a representative volume element (RVE) and constitutive material modeling proposed by [218]. The idea was to validate chip formation, feed force, size effects and torque with a realistic prediction model in micro drilling tests. Another interesting 3D model, in this case to estimate chip flow and tool wear in a micro-milling process is presented by [219]. The Figure 2.4 shows a comparison of predicted and measured 3-D chip formation and chip flow for half-immersion down micro-end milling. In addition, continuous chip formation and steady-state workpiece and tool cutting temperatures were analyzed by numerical modeling methods in micromachining. The empirical Johnson-Cook (JC) model to describe the thermomechanical flow behavior over the entire strain rate and temperature range is reported in several works [220, 221] to solve thermal modeling predictions.

Finally, an emerging technique over the past decade to obtain numerical models is the molecular dynamic (MD). In principle, MD is used for simulating nano-metric cutting [222, 223], i.e., the uncut chip thickness within the range of nanometers is possible to find some applications in the micro-scale [224] . MD models are used for the investigation of chip removal mechanisms, tool geometry optimization, cutting force estimations, subsurface damage identification, burr formation, surface roughness and surface integrity prediction. The results indicated that MD is a possible modeling tool for micro-cutting processes; atomistic modeling can provide a better representation of micro and nano-level characteristics than other modeling techniques [225, 226]. Some disadvantages are the significant computational power required and the cutting speed that is considered to be unrealistically high [195].

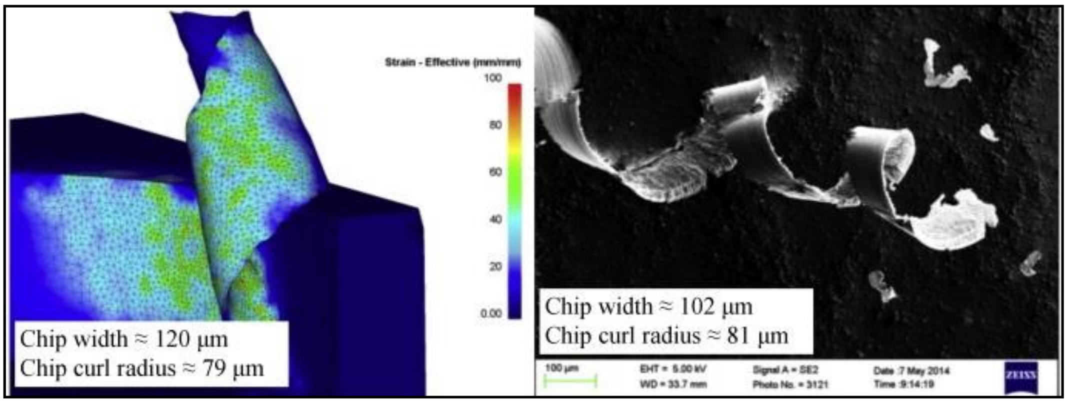


Figure 2.4 Comparison of predicted and measured 3-D chip formation and chip flow for half-immersion down micro-end milling [219]

2.1.5 Empirical modeling of micromachining processes

An empirical model can be defined as the representation obtained from the relationships between inputs and outputs based on experimental data. A mathematical empirical model can relate a non-random n-input variable, $x \in \mathbb{R}^n$ with a random variable scalar output variable, $y_{(x)}$. In multiple-input multiple-output models, the outputs can be separately handled and may therefore be considered multiple-input single-output models: the previous definition is sufficiently general [227]. Empirical modeling can also be conducted by any kind of computer modeling technique based on empirical observations rather than on descriptive mathematical relationships in the system that is modeled. This technique is a novel approach to computer-based modeling that was developed at Warwick University, England [228]. In summary, the empirical modeling can be classified into statistical and Artificial Intelligence-based groups, both of which are described in the following subsections.

2.1.5.1 Statistical models

A statistical model embodies a set of assumptions concerning the generation of the observed data and similar data from a larger population. Often, a model represents the data-generating process in considerably idealized form. Model assumptions describe a set of probability distributions, some of which are assumed to provide a suitable approximation of the distribution to a particular data set [229]. In statistical modeling, regression analysis is a statistical process for estimating the relationships among variables. There are many regression modeling techniques reported in the literature, starting with linear and non-linear regression, through to Bayesian linear regression, and non-parametric regression, among others. The simplest multiple regression models are represented by the linear equation [193]:

$$y^{(P)} = b^T x = [b_0, b_1, \ldots, b_n] \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \tag{2.1}$$

relating some input variables, *x*, with an output, *y*. Coefficients, *b*, are obtained by minimizing the sum of the square of the difference between the predicted and observed values, $y_{(P)}$ and y, for a set of m input–output pairs:

$$X = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1m} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \cdots & x_{nm} \end{bmatrix}; \qquad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \tag{2.2}$$

the regression coefficients can be estimated through:

$$b = \left(X^T X \right)^{-1} X^T y \tag{2.3}$$

Although most of the models used in science and engineering are linear or can be linearized, some of them are nonetheless strictly nonlinear. In these kinds of models, the number of parameters, q_{reg}, can differ from the number of input variables, *n*, but always less so than the number of input–output pairs in the dataset, N_{reg}, so as to prevent mathematical non-determination [230] . An important issue in the use of non-linear models is overfitting, which takes place when the model not only fits the relationship between the input and output variables, but also the noise present in the data. This unwanted phenomenon negatively affects the generalization capability of the model and mainly takes place when the number of parameters becomes too large. The regression models are commonly applied in the micromachining processes to establish a parametric correlation between force, vibration, and acoustic emission, among other factors, and productivity, surface quality and material removal rate [231-233].

In the same way, Bhandari, et al. [234] develop a multi-lineal regression modeling to create a burr-control chart based on experimental results for micro-drilling process. In this investigation, three types of burrs depending on location are defined: entrance burrs, interlayer burrs and exit burrs, using an orthogonal array experimental design proposed by Genichi Taguchi to determine the relationship between the drilling parameters and the burr height. Something similar, but in this case applied to a micro-milling process is proposed by [235]. A statistical ANOVA is employed to determine the significance of mathematical model taken from regression analysis to characterize the surface roughness and depth of machined area. Finally, a predictive model used for micro-turning of ceramics was proposed by [236]. A mathematical model was developed to establish the various micro-machining parameters such as: laser-beam power, workpiece rotational speed, assisted air pressure and y-axis feed rate with response criteria such as: surface roughness and deviation in turned depth, so as to achieve the desired surface quality and dimensional accuracy during micro-turning operations using laser systems. The multi-lineal regression correlation coefficients of the models obtained by the analysis of surface roughness and micro-turning depth deviations were 0.961 and 0.958, respectively. These correlation coefficients are near to 1 and clearly indicate that there is a very good fit between the predicted and the experimental values.

2.1.5.2 Modeling on the basis of Artificial Intelligence

Artificial intelligence methods refer to a set of tools and paradigms that are designed to understand and to emulate human intelligence and other complex natural systems. Although far from these final objectives, IA-based techniques have found a wide spectrum of applications in several knowledge branches. The most widely used techniques for modeling physical processes inspired in Artificial intelligence are Artificial Neural Networks, fuzzy & neuro-fuzzy systems, and probabilistic methods. These topics will be briefly reviewed in the following paragraphs [237, 238].

The development of AI technologies offers new opportunities to address not only conventional applications (expert systems, intelligent data bases, technical diagnostics, etc.), but also for full automation, control and monitoring of manufacturing processes [239]. Artificial neural networks are the most popular and well-established AI paradigm. This technique, inspired in the network structure of biological brains, has been extensively used for classification tasks and function approximation. The whole idea of the artificial neural network is based on the concept of the artificial neuron, a rough mathematical simulation of the biological neuron. Haber, et al. [240] was one the seminar works to analyze how Fuzzy Logic and Neural networks are essential for knowledge-based systems in industrial applications.

The McCulloch-Pitts neuron [241] is a binary device, with two stable states of a neuron as shows the Figure 2.5. The McCulloch-Pitts model can be considered as a computing unit with an input n-vectors, $x \in \mathbb{R}^n$, which are weighted (ω), linearly combined and them transformed by some transfer function (θ). Usually it can be: a sigmoid, a hyperbolic tangent, or simply a linear function to give a scalar output, *y*. A scalar term, *bias*, is often added to the linear combination which is called bias or threshold, representing the predisposition of the neuron to be activated. Each neuron has a fixed threshold. Furthermore, every neuron has excitatory and inhibitory synapses, which are inputs of the neuron. But if the inhibitory synapse is active, the neuron cannot turn on. If no inhibitory synapses are active, the neuron adds its synaptic inputs. If the sum exceeds or equals the threshold, the neuron becomes active. So the McCulloch-Pitts neuron performs simple threshold logic [242].

If some neuron layers only receive inputs from neurons located in previous layers, then the network is called a feed-forward network (Figure 2.6 a)) [243]. Examples of this structure are the multilayer perceptron (MLP) and the radial basis function networks (RBFN). In contrast, if connections exist between any one layer to a previous one, then they form a recurrent network

(Figure 2.6 b)) [244]. Examples of this architecture are the de Hopfield and the de Elman networks. The free parameters that select values (weights and biases) in the network are trained in what are known as learning or training processes. There are two main approaches to training a neural network: if the whole set of input–output pairs are known, supervised learning can take place.

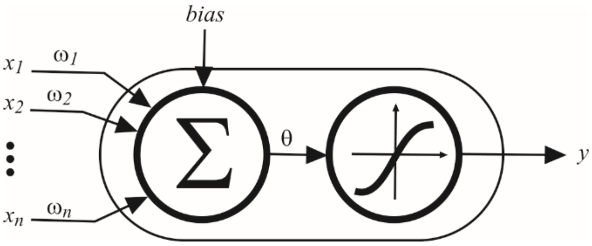


Figure 2.5 McCulloch-Pitts neuron model

The most popular supervised learning algorithm is error back propagation, which is used for training MLPs and is, actually, a generalization of the Delta rule, a gradient descent algorithm. On the contrary, when the output information of the training dataset is not completely known, unsupervised learning algorithms must be used. This kind of learning is used principally for classification tasks and is widely used in self-organized maps (SOM) and adaptive resonance theory (ART) networks.

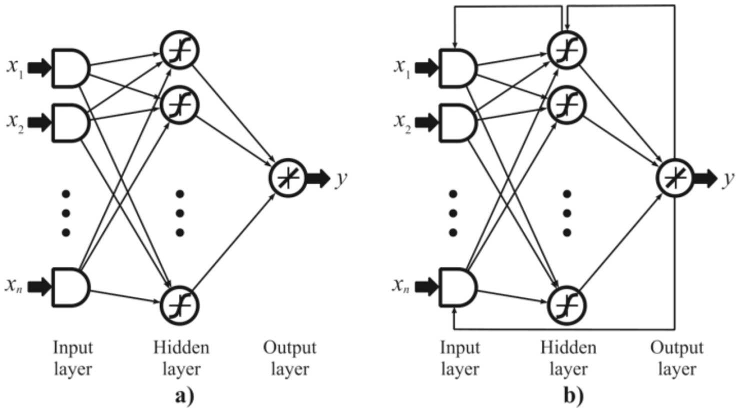


Figure 2.6 a) Feed forward architecture and b) Recurrent architecture

Fuzzy logic, unlike traditional or Boolean logics deals with uncertain relationships. In this approach, a statement can have a degree of truth ranging from zero to one, while in Boolean logic this degree of truth can be just zero (false) or one (true). The so-called membership functions determine the degree of membership of some element to some fuzzy subset [245] . A fuzzy inference system is a set of IF–THEN fuzzy rules, assembled together to offer some response to a given input. A fuzzy inference system is composed by a fuzzifier that converts the crisp inputs into fuzzy values. Then the inference engine applies the IF–THEN rules to produce a fuzzy output representing the response of the system and finally, this fuzzy set is transformed back into a crisp value by the defuzzifier. The design bottleneck is the selection and tuning of the membership functions, although, fuzzy systems have been successfully applied for solving many control and modeling problems [246-248].

An approach for solving this problem is the adaptive neuro-fuzzy inference system (ANFIS), an inference system organized by layers, some of which using fuzzy rules and others are tuned by an adaptive learning process (see Figure 2.7), like a neural network [249]. Rules provide the necessary information on the global behavior of the system. Once defined, the rules were normalized according to their importance. The next step was to calculate the consequent parameters, i.e., the Takagi–Sugeno function for each fuzzy rule. Finally, defuzzification is performed in the last layer to generate the crisp values.

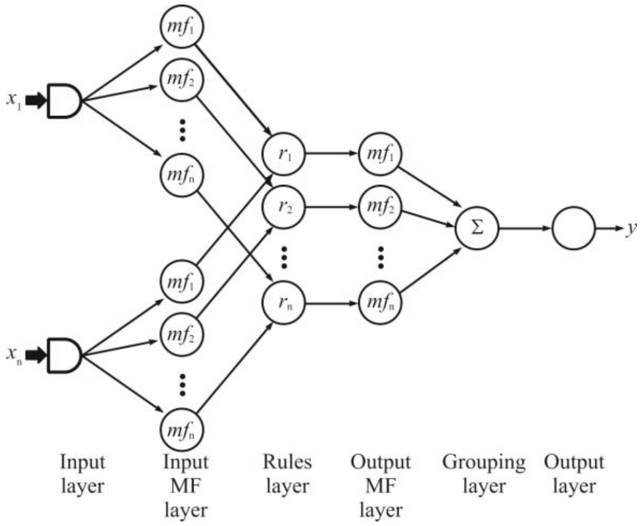


Figure 2.7 ANFIS model

The probabilistic methods for uncertain reasoning represent another group of techniques. Although apparently similar to fuzzy logic, both approaches are fundamentally different. Probability theory predicts events from a state of partial knowledge. On the other hand, fuzzy logic deals with situations where vagueness is intrinsic. In probability theory, the truth is unknown but absolute, while it is by nature a relative concept in fuzzy logic.

Bayesian and Hidden Markov models are the most widely applied probabilistic techniques in the various fields of modeling mechanical systems. A Bayesian network is a directed acyclic graph consisting of a set of nodes, representing random variables and a set of directed edges, representing their conditional dependencies. The dependencies in a Bayesian network can be adaptively determined from a dataset through a learning process. The objective of this training is to induce the network with the best description of the probability distribution over the dataset and can be considered as an unsupervised learning method, because the attribute values are not supplied in the dataset [250].

Hidden Markov models consist of a set of hidden states that form a chain described by a transition probability distribution over these states and an associated set of emission probability distribution for the observed symbols. Unlike Bayesian networks, hidden Markov models can be represented as cyclic graphs and they have the ability to model the temporal evolution of signals [251]. The learning process in a hidden Markov model aims to find the best chain corresponding to a given set of output sequences. Several approaches have been proposed for obtaining local solutions in a computationally efficient way.

Intelligence techniques are widely used due to the complexity and non-linearity of signals and events present in the micromachining processes. The principal objective is to correlate the information captured (force, vibrations, acoustic emissions and internals variables such as: position, current, power, etc.) during the cutting process with quality parameters (surface roughness, burr formation, geometry errors). Furthermore, models to predict, monitoring and controls to reduce costs and to increase the productivity (material removal rate) have been described in the literature. For a better understanding, some recent works were summarized in Table 2.2.

Table 2.2 Intelligence models: a short review

*Operation**	*Objective*	*Variables*	*Algorithms*	*Authors*
Milling	Wear monitoring	Vibrations	Backpropagation neural network	[252]
Turning	Machine stiffness and material characteristics	Cutting force	Fuzzy rule based	[253]
Drilling	Online run out detection	Force signals and cutting parameters	Feed-forward neural network	[254]
Milling	Surface roughness modeling	Vibration signals	Adaptive network-based fuzzy inference system	[255]
Turning	Tool-wear prediction	Pattern-recognition	Neural network	[256]
Milling	Tool flank wear state identification	Cutting force features	Continuous Hidden Markov model	[257]
Drilling	Tool breakage and wear detection	Input current driving motor	Neural network	[258]
Milling	Tool life	Cutting force component	Takagi-Sugeno-Kang fuzzy logic modeling	[259]

* All the operations are referred to micro-scale processes.

Quite often, the modeling techniques, especially those based on Artificial Intelligence techniques are the middle step for achieving the optimal setting of some parameters in the production chain [260, 261]. Undoubtedly, at least a rough model is a must for performing the optimization, and this representation is needed to establish the correlation between the dependents and the independents variables. One example is adaptive control optimization in the micro-milling of hardened steels presented by [262]. These authors proposed an adaptive control optimization system to optimize cutting parameters using an artificial neural network to estimate cutting-tool wear.

2.2 Experimental setup

2.2.1 Kern Evo Ultra-precision machine

As it was previously described, a solid experimental setup is required to obtain empirical models. In order to carry out this task during the experimental period, all experiments were conducted in a three axes Ultra Precision Kern Evo Machine. Furthermore, the machine is equipped with a laser control Nano NT to measure the tool geometry. The main components of the machine and the laser control system are shown in the Figure 2.8.

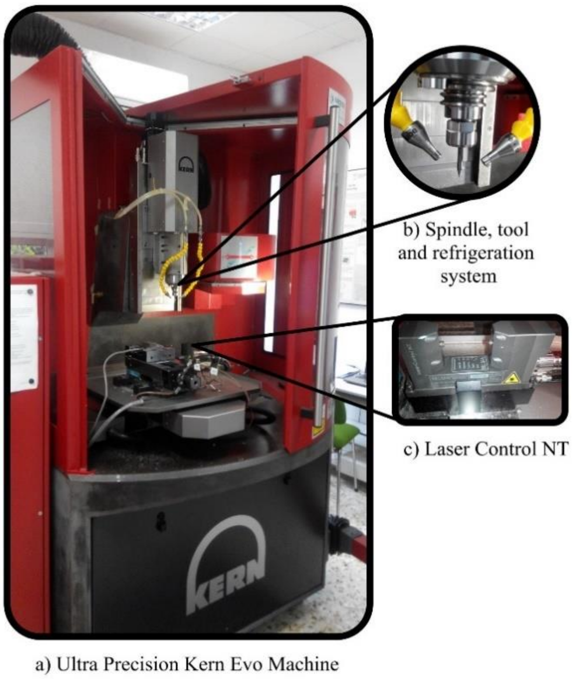

b) Spindle, tool and refrigeration system
c) Laser Control NT
a) Ultra Precision Kern Evo Machine

Figure 2.8 Ultra Precision Kern Evo Machine

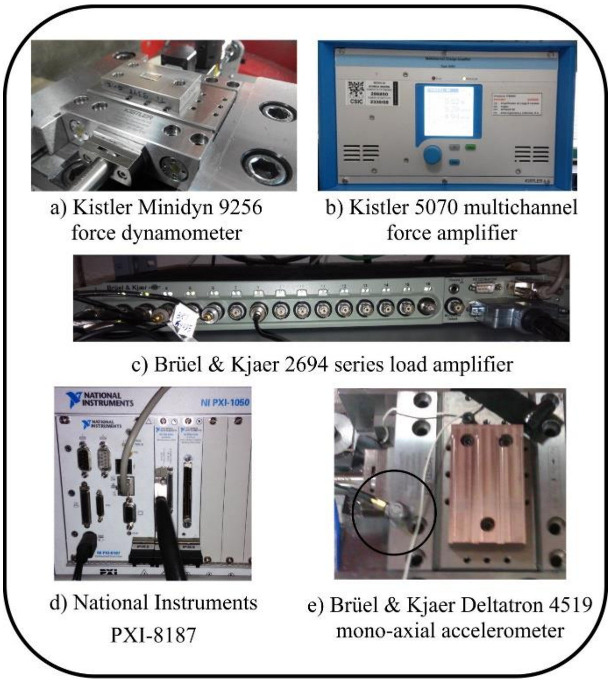
Table 2.3 shows the parameters of the laser sensoring system.

Table 2.3 Technical data of the laser measuring device

Parameters	*Manufacturer specifications*
Laser safety classification	Class 2 acc. to IEC60825-1. 21 CFR 1040.10
Laser type	Visible red light laser |630. . .700 nm| < 1 mW
Repeatability	0.1µm 2σ
Minimum tool diameter	Standard: 15µm
Test speed (spindle)	Up to 200,000 rpm
Operating temperature	–10ºC. . .+70ºC | +5ºC. . .+45ºC

2.2.2 Real-time platform and sensory equipment

The machine tool is also equipped with a dynamometer for capturing the three components of the force signal (*Fx*, *Fy*, *Fz*) and accelerometers for measuring the vibration on three axes (*Vx*, *Vy*, *Vz*). The tool position (*x*, *y*, *z*) is obtained through the Ethernet interface of the machine-tool. Figure 2.9 shows the dynamometer and accelerometers for measuring force and vibration signals, respectively.



a) Kistler Minidyn 9256 force dynamometer

b) Kistler 5070 multichannel force amplifier

c) Brüel & Kjaer 2694 series load amplifier

d) National Instruments PXI-8187

e) Brüel & Kjaer Deltatron 4519 mono-axial accelerometer

Figure 2.9 Dynamometer, accelerometers, amplifier and Real-time platform

The Kistler Minidyn 9256 piezoelectric dynamometer measures force on three axes. A measurement ranges of 250 N for 3-axis force and 8Nm for 2-axis torque measurement can be used. Natural frequencies above 5 kHz allow the measurement of signals with engagement frequencies of up to about 2 kHz, which correlate with a rotational speed of 60,000 rpm for a two edge tool.

By the other hand, three Brüel & Kjaer Deltatron 4519 mono-axial accelerometer were installed for measuring vibration on the three-axes. This sensor has a sensitivity of 103.7 mV/g and a measurement range of up to 20 kHz. All the accelerometers were connected to a Brüel & Kjaer 2694 series load amplifier. All data signals were fed into a NI 6251 National Instruments data acquisition card, with a sampling frequency of 50 kHz. A National Instruments high-performance PXI-8187 embedded controller processes the signals.

Finally, the sampling frequencies are shown in Table 2.4.

Table 2.4 Sampling frequencies for the measured signals

Variable	*Unit*	*Sampling frequency (s^{-1})*
Position in x-axis, x_{coord}	mm	795
Position in y-axis, y_{coord}	mm	795
Position in z-axis, z_{coord}	mm	795
Vibration in x-axis, *Vx*	mV	50000
Vibration in y-axis, *Vy*	mV	50000
Vibration in z-axis, *Vz*	mV	50000
Force in x-axis, *Fx*	N	50000
Force in y-axis, *Fy*	N	50000
Force in z-axis, *Fz*	N	50000
Moment in z-axis, *Mz*	Nm	50000

2.3 Micro-drilling processes

2.3.1 Forces and vibrations estimation models in micro-drilling operations

During the research period the first results were published in [263]. The study on the signals of a micro-drilling process in order to extract relevant process features is presented. These patterns are correlated with the cutting tool condition, providing the foundations for further developments of indirect cutting tool monitoring systems. Forces and vibrations were recorded when micro-drilling of a tungsten-cooper alloy with TiAlN-coated tools. Three tools with diameters, *D*, of 0.1, 0.5 and 1.0 mm, respectively, were used and five consecutive holes were elaborated with each tool as shown in Table 2.5. The nominal cutting conditions were represented by spindle rotation speed, n_{rpm}; feed rate, f_{rate} and the drilling depth, h_d.

Table 2.5 Nominal condition of the micro-drilling process

D	n_{rpm}	f_{rate} *(mm/min)*		h_{d1}	h_{d2}
(mm)	*(rev/min)*	f_{rate1}	f_{rate2}	*(mm)*	*(mm)*
1.0	20 000	440	352	1.0	2.0
0.5	40 000	440	352	0.5	1.0
0.1	48 000	530	424	0.1	0.2

Measured signals were processed by using time-domain statistics, Fast Fourier transform and Hilbert-Huang transform for extracting features. Figure 2.10 shows the correlation analyses between the obtained features and the number of elaborated holes (N_{hole}). Correlation analyses between the obtained features and the number of elaborated holes were then carried out in order to identify which of these features can be used for estimating the tool condition.

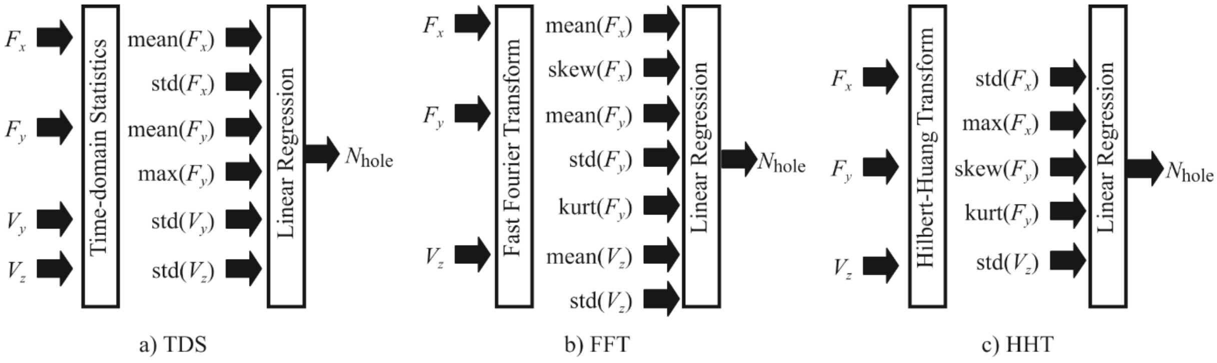


Figure 2.10 Correlation analyses between the obtained features and the number of elaborated holes

The maximum, mean value, standard deviation, skewness and kurtosis are computed for each signal Linear regression analysis to identify the possible dependence of the number of elaborated holes with any of these parameters are then carried out. The following Eqs. (2.4), (2.5) and (2.6) describe the lineal regression models obtained for the time-domain statistics, Fast Fourier transform and Hilbert-Huang transform.

$$N_{hole} = 41.86 + 40.95 \bullet F_{x\_mean} - 102.80 \bullet F_{x\_std} - 49.33 \bullet F_{y\_mean} + \ldots \\ + 22.06 \bullet F_{y\_max} - 0.11 \cdot V_{y\_std} + 210.1 \bullet V_{z\_std} \tag{2.4}$$

$$N_{hole} = 3.08 - 1.18 \bullet F_{x\_mean} - 0.02 \bullet F_{x\_skew} - 0.17 \bullet F_{y\_mean} + 0.04 \bullet F_{y\_std} + \ldots \\ + 0.001 \bullet F_{y\_kurt} + 0.75 \bullet V_{z\_mean} - 0.47 \bullet V_{z\_std} \tag{2.5}$$

$$N = 1.93 + 6.29 \bullet F_{x\_std} - 5.14 \bullet F_{x\_max} + 1.28 \bullet F_{y\_skew} + 0.36 \bullet F_{y\_kurt} + 3.91 \bullet V_{z\_std} \tag{2.6}$$

Time-domain statistics features did not show remarkable correlation with the tool usage level. On the contrary, FFT and HHT yielded very interesting outcomes, because some of the analyzed features showed a clear relationship with the number of elaborated holes. Table 2.6 shows a comparison between the correlation coefficients (R^2) obtained for each model with 95% confidence the parameters that have a significant relationship, with regard to the number of elaborated holes. After that, a micro-drilling process was experimentally studied. The analysis with three TiAlN-coated drills (diameters 0.1 mm; 0.5 mm and 1.0 mm) and a workpiece of tungsten–copper alloy was reported in [193]. Variations in tool dimensions were measured after the completion of each hole, while force and vibration signals were measured throughout the cutting process. The behavior of force and vibration for each tool diameter is depicted in Figure 2.11. For the sake of clarity, only the signals from the first hole are shown.

Table 2.6 Correlation coefficients of the models

Models	*Correlation coefficients, R^2*
Time-domain statistics	0.2439
Fast Fourier transform	0.5668
Hilbert Huang transform	0.5307

However, some interesting results can be inferred from Figure 2.11. The magnitude and amplitude of force signals are only relevant on the z-axis, which is consistent with the micro-drilling operation. Another interesting issue is how the maximum force on the *z*-axis decreases from 42.7 N in the 1 mm diameter tool to 2.14 N in the 0.1 mm diameter tool.

This is more than a 20-fold reduction of force magnitude in the 0.1 mm tool diameter. Therefore, the influence of disturbances such as the offset of the dynamometer and air-coolant on force signals clearly limits the suitability of this sensorial information at a micro-scale, at least by means of time domain-based performance indices. Likewise, the magnitude of the vibration signals on the *x* and *z*-axes and the oscillations complicate direct use of this information in the time domain.

Other features were extracted from the signals by using time-domain statistics, fast Fourier transform, wavelet transform, and Hilbert–Huang transform. These features were related with the number of drilled holes by using three modeling techniques: statistical regressions, neural networks and neuro-fuzzy systems.

In each case, the parameters obtained from the corresponding feature extraction technique were used as inputs of the model. When time-domain statistics were used (Figure 2.12 a)), the mean value, *mean*(•), standard deviation, *std*(•), and root mean squared value, *rms*(•), of each measured signal were entered as inputs. When the feature extraction was based on FFT (Figure 2.12 b)), the inputs of the models were the energies, E_i(•), and frequencies, ν_i(•), of the five (i = 1,…, 5) most energetic values of the transform of each component. In a WT-based feature extraction (Figure 2.12 c)), the model inputs were the energy values of the four components, E_i(•), i = 1,…,4, for each measured signal. Finally, when the feature extraction was based on

HHT (Figure 2.12 d)), the inputs were the energy value, $E_i(\bullet)$, and the number of peaks, n_{peak}, of the five (i = 1,…,5) most energetic IMFs. Moreover, the feed rate and the tool diameter were used as inputs in all the cases. The model output is the number of drilled holes. This variable was selected because it can be easily related to a threshold that represents the maximum allowable number of holes per drill. For training or fitting the models, each signal was divided into eight sections, so that, overall 600 samples of data were obtained. These data were divided into a training set of 570 elements and a validation set of 30 elements.

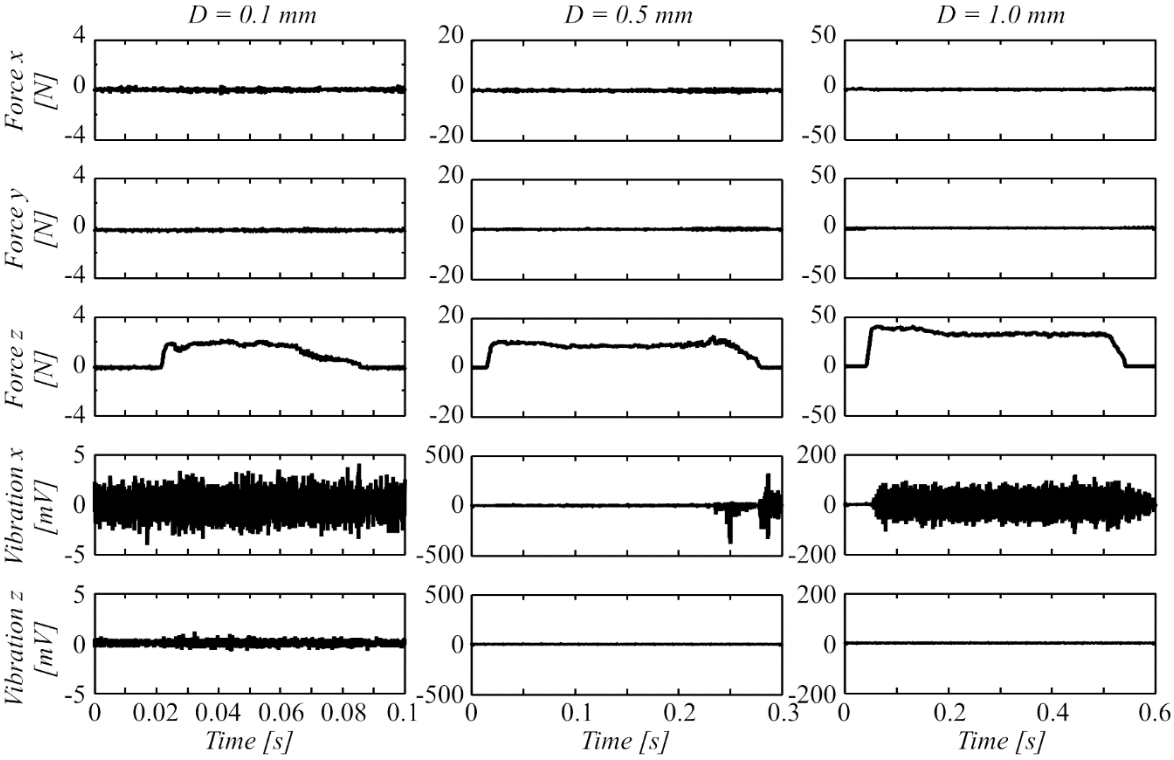
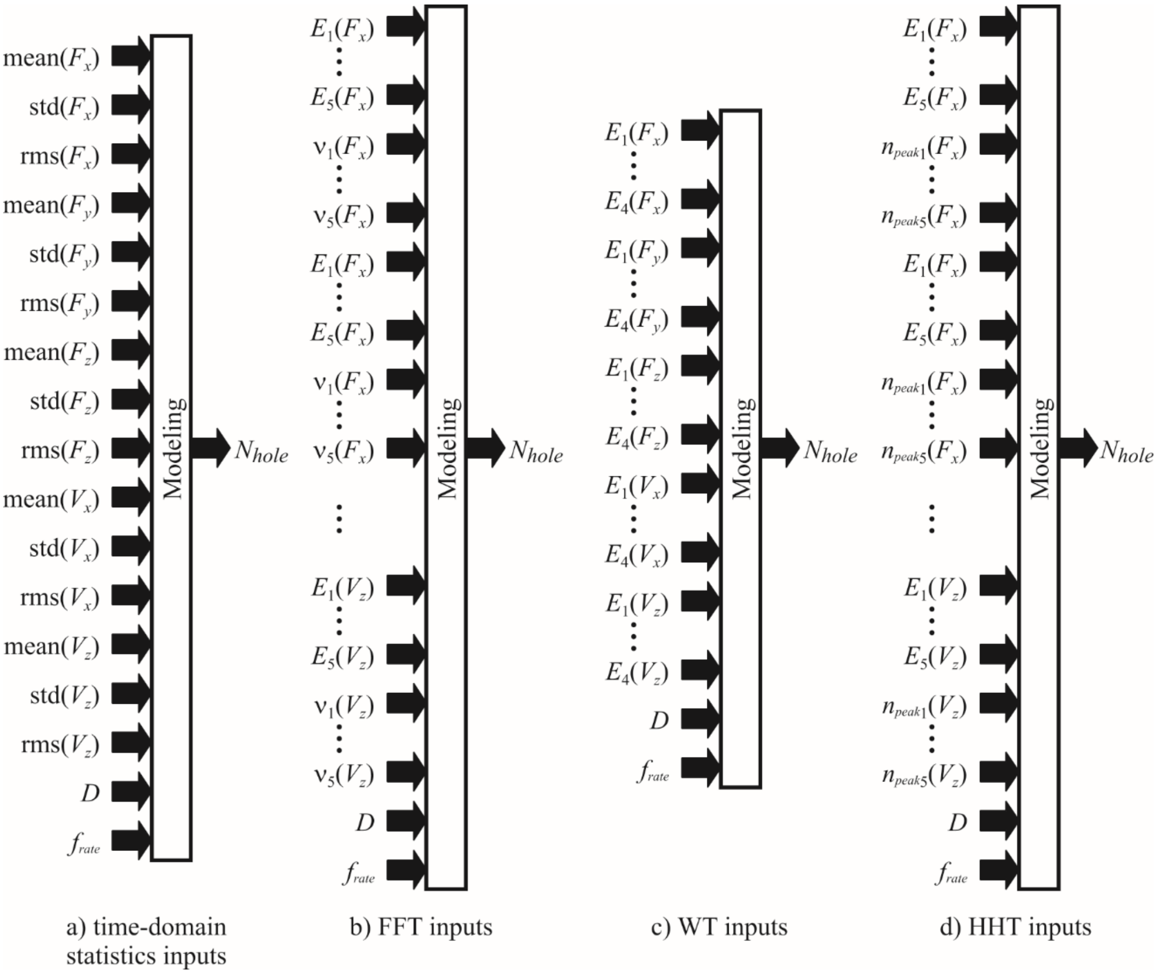


Figure 2.11 Graphical representations of the measured signals

A feed-forward layered network of perceptron with twenty nodes that is used for data processing was used as second modeling technique. This is the most widely applied topology of the MLP, with only one hidden layer that guarantees its performance as a universal approximator, giving the MLP good modeling capabilities [264]. The training inputs as shown in the Figure 2.12 showed the same sequence before and once again, the output was tool usage.

Finally, an adaptive neuro-fuzzy inference system (ANFIS) was used. ANFIS implements the Takagi-Sugeno model for the structure of the fuzzy system If-Then rules [265, 266]. The ANFIS architecture has five layers and the error back propagation is used as the learning strategy to determine the premise parameters of the rules. The parameters of each consequent are estimated using the least squares method. In the first step or *forward pass*, the input models are propagated and the optimum consequents are estimated with an iterative procedure of least squares, whereas the premises remain fixed. In the second step or *backward pass* the back propagation procedure is used to modify the premise parameters and the consequents remain constant. This procedure is repeated until the stop condition is satisfied (error criterion). When the values of the premise parameters are set, the general system output is expressed as a linear combination of the consequents.



a) time-domain statistics inputs

b) FFT inputs

c) WT inputs

d) HHT inputs

Figure 2.12 Models of the feature extraction-techniques

The training parameters are 100 iterations (a higher number of iterations causes overtraining and, as a result of this, undesired peaks in the system output), using a hybrid training mode such that error back propagation did not reach the output value desired, and a step size of 0.05. The increase in step size did not produce a significant improvement in the output; however, it did increase operation computation time.

In order to compare the results obtained during the modeling step, the R^2 coefficient for each model is shown in Table 2.7. This value represents part of the variability of the modeled parameter, which is explained by the fitted model, which is usually taken as a measure of model quality. The statistical regression-based models have very poor prediction capabilities. On the contrary, ANN-based and ANFIS-based models show better correlations, but their values are relatively low. This is due to high non-linearity and noise caused by the complex nature of the cutting phenomena in micromechanical machining.

Table 2.7 Comparative evaluation of model performance

	Correlation, R^2			*Generalization capabilities*		
	Regr.	*ANN*	*ANFIS*	*Regr.*	*ANN*	*ANFIS*
Time domain	0.0000	0.8501	0.1846	0.2594	0.0491	0.6784
FFT	0.0000	0.2265	0.2818	0.7498	0.1358	0.0598
WT	0.0000	0.5195	0.3854	0.7515	0.7306	0.7319
HHT	0.0000	0.5525	0.2727	0.7348	0.0346	0.8543

Likewise, Table 2.5 also depicts the generalization capabilities of the models. This generalization capability was determined by using the probability value of the comparison of the means of the fitting and validation set residuals. If there is no statistically significant difference between both means, it can be concluded that this model has good generalization capabilities. It can be noted from the table, that the ANN-based model has higher correlation values, but lower generalization capabilities than the ANFIS-based model. Indeed, ANFIS is not only a pioneering work, but also the simplest computationally and the most viable for real-time applications. One of the advantages of ANFIS is that it combines the semantic transparency and intrinsic robustness of fuzzy systems with the learning ability of neural networks. The good generalization capability of ANFIS is supported on its main principle based on extracting fuzzy rules in each level of a neural network. Once the rules have been obtained, they provide the necessary information on the global behavior of the system.

Figure 2.13 show a graphical representation of these models, plotting the relationship between predicted and observed values for the training set and the validation set. These figures show the mean values with the respective confidence intervals, considering a statistical significance of 95%. The most convenient combinations of feature extraction technique and modeling strategy are WT+ANN (Figure 2.13a)), WT+ANFIS (Figure 2.13b)) and HHT+ANFIS (Figure 2.13c)). This result is consistent with the feature extraction step that demonstrates the good capability of HHT. Nevertheless, wavelets-based feature (in time-frequency domain technique alone does not provide good enough results) in combination with ANFIS and ANN yields good results.

From the scientific viewpoint, the combination of WT and ANN provides better tradeoff between appropriate correlation and good generalization capability. However, from the technical standpoint the implementation of the monitoring system at industrial scale has severe constraints. For instance, the computational cost (computation time, signal processing time, etc.) is lower for wavelets and ANFIS combination. This result makes WT+ANFIS a good candidate for the final implementation of the monitoring system at industrial scale.

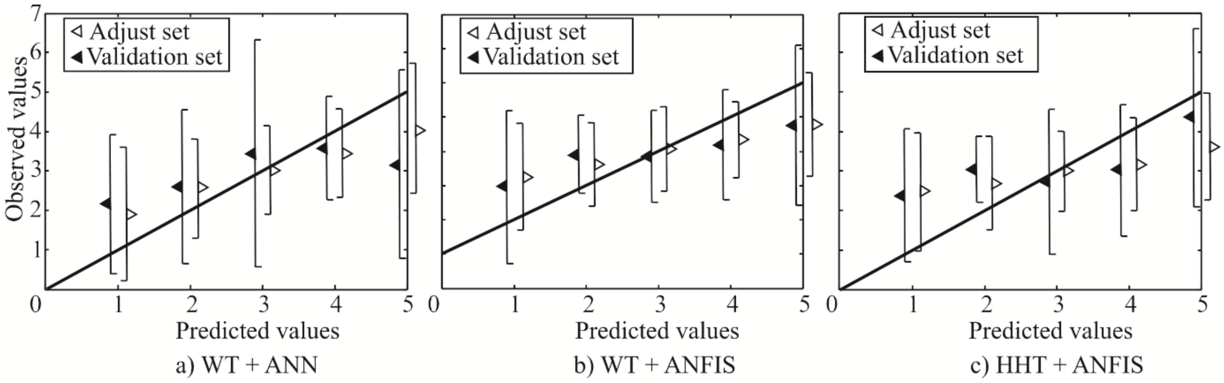


Figure 2.13 Relationship between predicted and observed values for most convenient combinations

An additional advantage of the wavelet in the feature extraction stage is that it offers the possibility of filtering the signal in the various frequency bands and of separating the interesting frequency from the noise frequency, which considerably reduces the area for searching in the subsequent stage, achieving a better final prediction.

Table 2.8 Factor levels for experimental factors

Material	*Levels*	*V*c *(m/min)*	*f_{rate}* *(mm/min)*	*step* *(mm)*	*h_d* *(mm)*
	min	31.4	200	0.05	0.00
$W_{78}Cu_{22}$	mean	47.1	300	0.10	1.25
	max	62.8	400	0.15	2.50
	min	11.0	30	0.05	0.00
Ti_6Al_4V	mean	16.5	45	0.10	1.25
	max	22.0	60	0.15	2.50
	min	11.0	30	0.05	0.00
Ti grade 2	mean	16.5	45	0.10	1.25
	max	22.0	60	0.15	2.50
	min	31.4	240	0.05	0.00
Invar	mean	47.1	360	0.10	1.25
	max	62.8	480	0.15	2.50
	min	31.4	700	0.05	0.00
Al7075	mean	47.1	800	0.10	1.25
	max	62.8	900	0.15	2.50

An upgraded model was presented in [254], being the main contribution the possibility to generalize the model developed before for a single material to a multiple material scenario. Furthermore, in this investigation one strategy for optimization was introduced. Experimental work has been carried out for measuring the thrust force for five different commonly used alloys, under several cutting conditions. In the experimental study, a micro-drilling process was carried out on five commonly used alloys: a sintered tungsten-copper alloy ($W_{78}Cu_{22}$), two titanium-aluminum-vanadium alloys (B348 grade 2 and Ti_6Al_4V), an aluminum alloy (Al7075) and Invar. 0.5 mm-diameter drills were applied in all case studies.

Three cutting parameters were considered in the processes: cutting speed, V_c; feed rate and drill step, *step*. Also, the drilling depth was taken into account as another factor. Three levels (minimum, mean and maximum) were considered for the experimental factors (see Table 2.8).

A three-level orthogonal array was used as experimental design. One of the main contributions was the use of every material, mechanical and thermal properties (yield tensile, σ_Y; hardness, *HRB*; Young's modulus, *E*; elongation, δm; mass density, ρm; thermal conductivity, *kc*; and heat capacity, c_h (see Table 2.9) to be included as model inputs.

Table 2.9 Properties of the workpiece materials

Parameters	*Materials*				
	$W_{78}Cu_{22}$	*Ti_6Al_4V*	*Ti grade 2*	*Invar*	*Al7075*
σ_Y (MPa)	240	830	276	679	503
HRB	90	108	80	109	87
E (GPa)	240	114	103	148	71.7
δm(%)	8	10	20	5.5	9
ρm *(*g/cm^3)	15.12	4.43	4.51	8.05	2.81
*kc (*W/m·K)	198	6.7	16.4	10.15	130
c_h (kJ/kg·K)	0.21	0.53	0.52	0.52	0.96

Subsequently, a MLP neural network-based was chosen for modeling the behavior of the thrust force. It was composed by an eleven inputs layer (V_c, f_{rate}, *step*, h_d, σ_Y, *HRB*, *E*, δm, ρm, *kc*, c_h), 80 neurons in the hidden layers and a single output neuron, which offers the predicted value of the force. Hidden neurons use sigmoid transfer function while the output neuron uses linear transfer function. The available experimental data was randomly divided into a training set containing 1 973 samples (corresponding to the 80%) and a validation set with the remaining 493 samples. All the inputs and output were normalized in the interval from zero to one, by using linear interpolation.

The training process was carried out through the error backpropagation algorithm, with adaptive learning rate and momentum. The following training parameters were selected: initial learning rate, 0.001; ratio to increase learning rate, 1.05; ratio to decrease learning rate, 0.70; momentum constant, 0.9; and minimum performance gradient, 10–10. The stop condition was established after the 300 000 epochs. The behavior of the sum of squared errors through the training process was shown in the Figure 2.14.

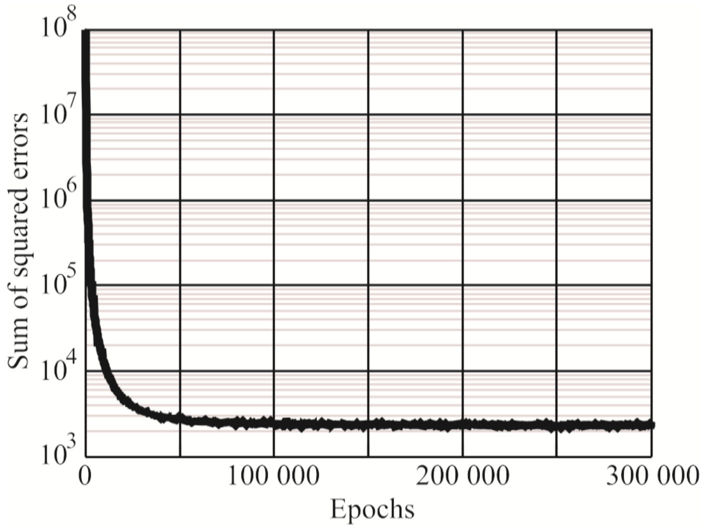


Figure 2.14 Sum of squared errors through the training process

The obtained neural model has a correlation coefficient, R^2, of 0.8969, so, it explains the 89% of the variability in the response variable. The standard error of the estimations was 1.084 and the mean absolute error was 0.7340. The ANOVA of the model (see Table 2.10) shows that there is a statistically significant relationship between the variables at the 95% confidence level.

Table 2.10 ANOVA of the thrust force regression model

Source	*Sum of Squares*	*Degrees of Freedom*	*Mean Square*	*F-Ratio*	*p-Value*
Model	20172	1040	19.40	7.796	0.0000
Residuals	2319	932	2.49		
Total	22491	1972			

In order to determine the generalization capability of the model, the residuals coming from the training and validation sets were analyzed. If the model is able to generalize its predictions, the both residuals sets should have a similar normal distribution (i.e., a normal distribution with the same mean and standard deviation). Therefore, the mean values and standard deviations of both sets were compared, through a *t*-Student and an *F* tests, respectively. These tests gave associated probability values of 0.6125 and 0.1832, therefore, the null hypothesis (both means are equal and both standard deviations are equal) cannot be rejected with at the 95% confidence level. It can be concluded that both residuals sets come from the same distribution and the model has a good generalization capability. The points are located near the ideal model line and there is no indication of heteroscedasticity. Moreover, residuals coming from the training and validation sets show a similar distribution. Figure 2.15 shows a graphical representation of the obtained thrust force for the mean values of steps and initial drilling depth.

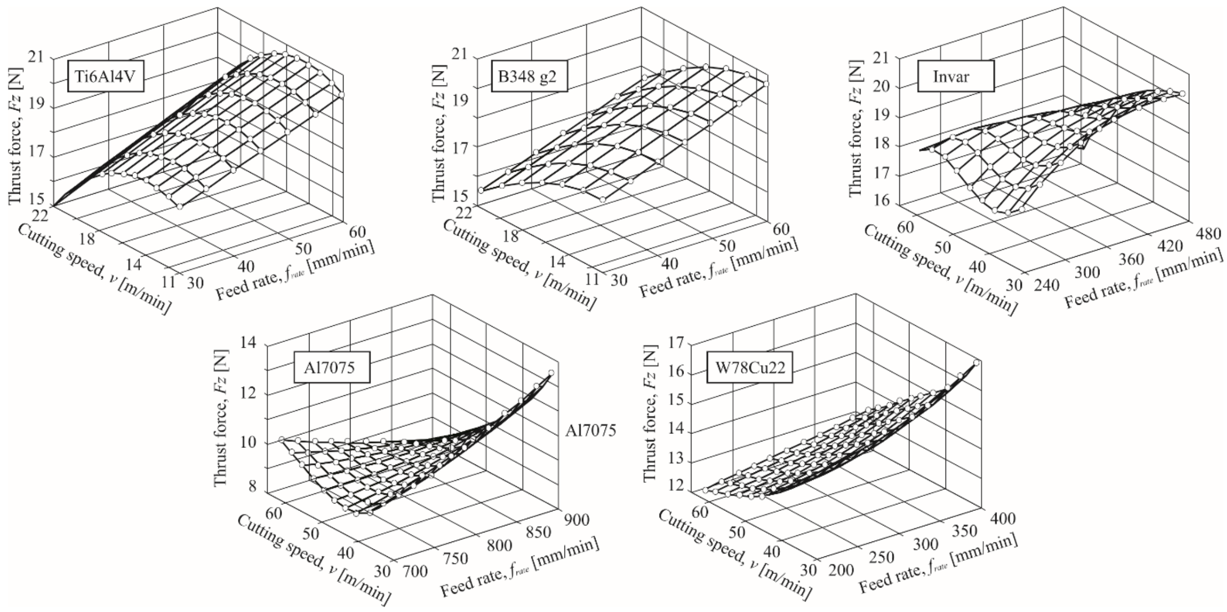


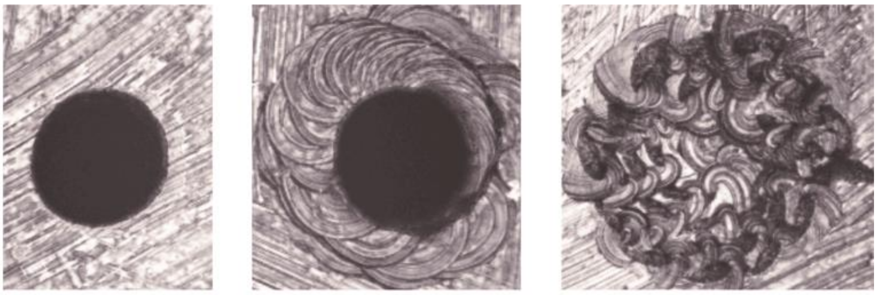
Figure 2.15 Graphical representation of the thrust force model for mean values of steps and initial drilling depth

Finally, during the investigation an optimization process was executed by considered two different and conflicting objectives: the unit machining time and the thrust force (based on the previously obtained model). A multi-objective genetic algorithm was used for solving the optimization problem and a set of non-dominated solutions was obtained. The Pareto's front representation was depicted and used for assisting the decision making process. These results were not included in the writing of this chapter, because it only considered the modeling techniques. In the following chapter (see Chapter 3), an optimization method based on cross entropy will be refined and applied to a micromachining process.

2.3.2 Run-out and holes quality prediction models in micro-drilling operations

Nowadays, monitoring systems are becoming a main component of the modern industry. The relevance of monitoring systems is even more evident in manufacturing processes from simple operations to overall manufacturing plant in order to produce high-quality products in a very short time [267]. In micro-manufacturing processes, the need of monitoring systems is growing up very fast, because of the small scale of processes, which make almost impossible to detect any failure by simple visual inspection.

One of the more common and undesired phenomenon, which must be detected by the monitoring systems in micro-drilling processes, is the so-called run-out (see Figure 2.16). It consists in an eccentric motion of the drill caused by the excessive centrifugal force due to the high rotational speeds [268]. Run-out can cause not only the damage of the surrounding surface but also the breakage of the cutting tool [269].

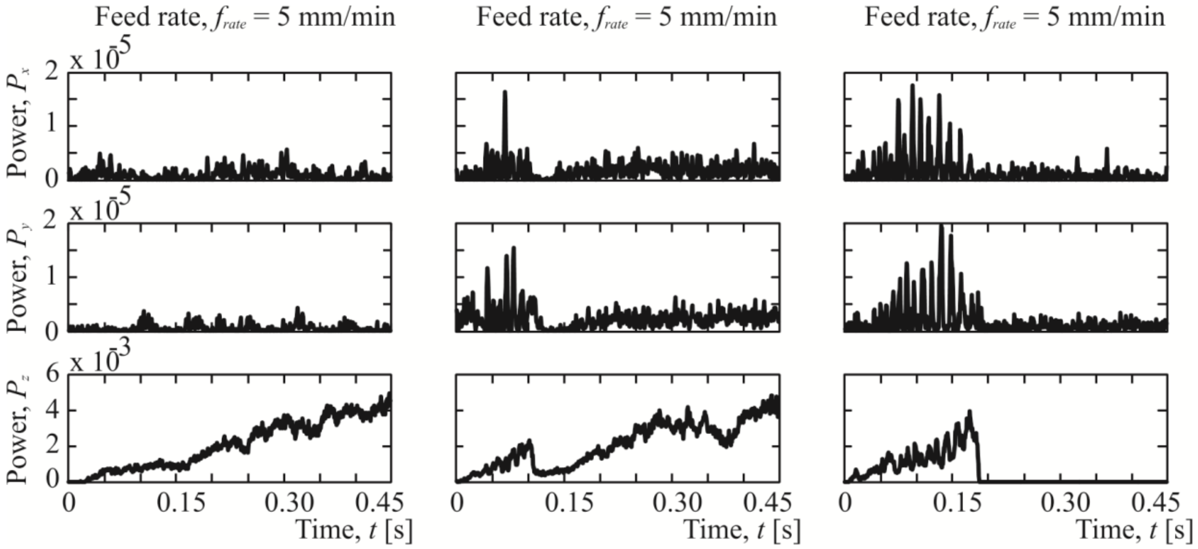


a) Normal hole b) Run-out (without breakage) c) Run-out and breakage

Figure 2.16 Three possible conditions of the drilled holes

A two-step monitoring system for run-out detection is reported in [254]. The first step uses the fast Fourier transform for extracting features from the online measured force signals. The final goal was the design of a real-time run-out detection system. Therefore, it is very important to observe the transient behavior of the signals. For this purpose, a short-time Fourier transform (STFT) was chosen, which is used for determining the sinusoidal frequency and phase content of local sections of a signal as it changes over time. In the discrete time case, data could be broken up into chunks or frames. Each chunk is Fourier transformed, and the complex result is added to a matrix, which records magnitude and phase for each point in time and frequency.

The typical behavior of the force signal power, in the selected band, for three different drilling processes (normal, run-out without breakage, and run-out with breakage) is shown for Ti_6Al_4V (see Figure 2.17) and $W_{78}Cu_{22}$ (see Figure 2.18). It was considered a time interval of 0.45 s according to the corresponding feed rate (i.e., 5 mm/min).



a) Normal Hole b) Run-out (without breakage) c) Run-out and breakage

Figure 2.17 Power of the (50…200)-Hz band in the Ti6Al4V force signals

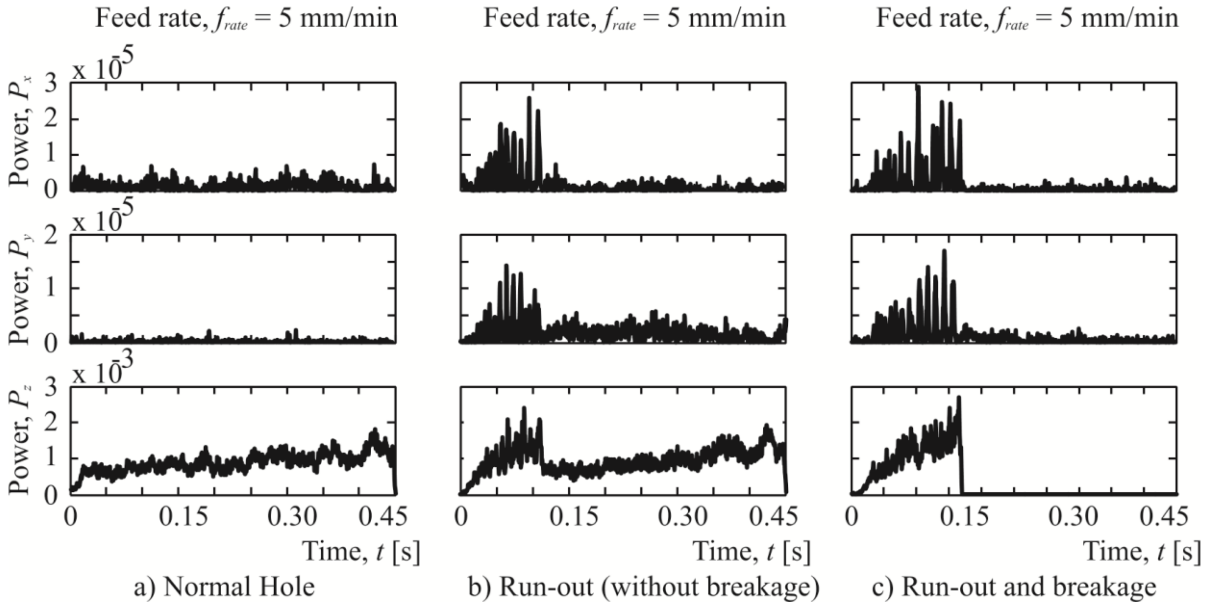


Figure 2.18 Power of the (50…200)-Hz band in the W78Cu22 force signals

When run-out takes place, the three components of the forces increase their values. This increment is especially noticeable in the *x*- and *y*-components. This behavior can be explained by the drill buckling. On the other hand, when the run-out causes the breakage of the drill, there is a clear decrease in the *z*-component of the force. The x- and y-components also decrease, but as the value of the *z*-component is higher, the decrease is more evident.

Another interesting issue is the presence of a lower frequency harmonic through the run-out (see Figure 2.19). The eccentric motion of the drill can be one of the possible causes. The relevant information is constrained to the first part of the signal because the run-out takes place during the tool entrance, when the drill must overcome the cohesive forces of the workpiece surface.

In the second step, MLP-based model predicts the process condition from the previously obtained features. This is a feed-forward neural network, which is trained by using the so-called back-propagation algorithm. This technique is a variant of the gradient descendant method applied in two steps. The neural network was trained by using the Levenberg-Marquardt algorithm, because it is faster than other back propagation algorithms, although it requires a larger amount of memory. The training process was carried out with an initial value of μ_d = 10^{-3}; the μ_d decrease factor was 0.1, the μ increase factor was 10, the maximum value of μ_d was 10^{10}, and the minimum performance gradient was 10^{-7}. The process was stopped after 5×10^5 epochs.

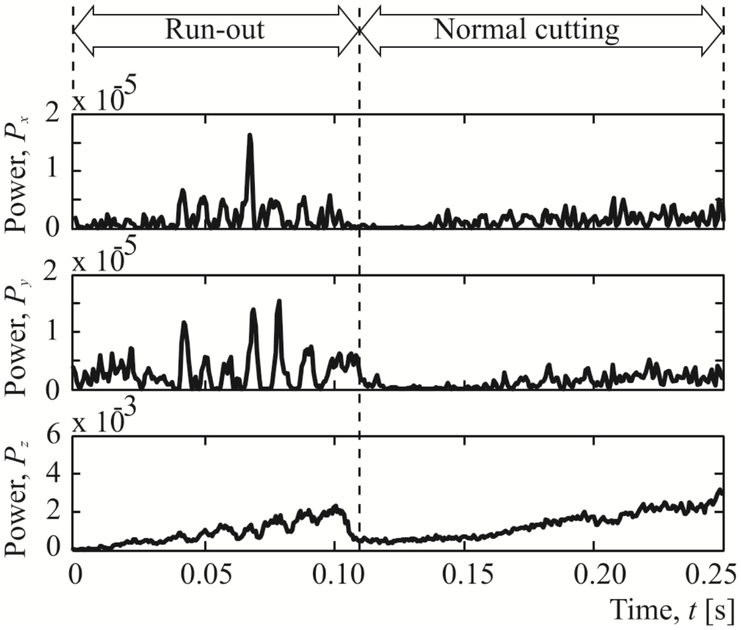


Figure 2.19 Power of the force signals (detail) when run-out takes place

The selected network model (see Figure 2.20) has eight input neurons ($\iota_1 \ldots \iota_8$); six of them corresponding to the two first FFT spectrum values of each force signal (F_x, F_y, and F_z) and the other two to the material (codified as 0 = Ti_6Al_4V and 1 = $W_{78}Cu_{22}$) and the used feed rate. All inputs were normalized in the interval [0, 1] by using linear interpolation. The hidden layer was composed of 20 neurons ($\eta_1 \ldots \eta_{20}$) using sigmoid activation functions. Finally, an output neuron (*status*) brings the predicted cutting status.

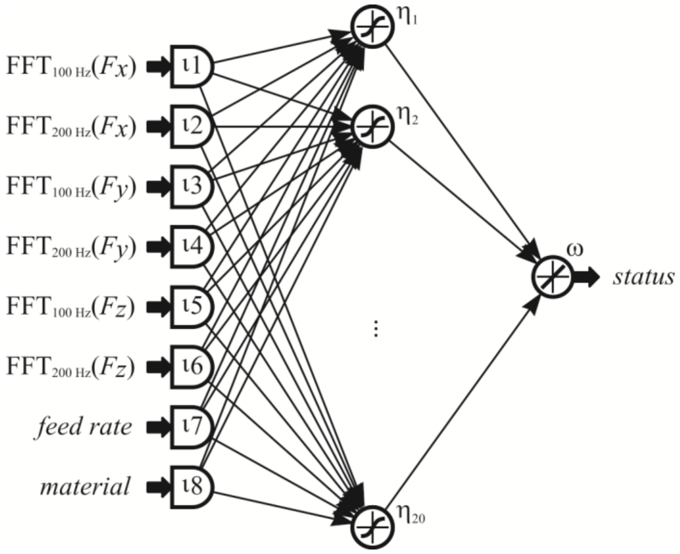


Figure 2.20 Neural network-based model to predict run-out in micro-drilling processes

Overall, 46 experiments were used for this purpose, distributed as shown in Table 2.11. This system takes 500 items per time from the force signals data. As each signal was divided into sets of 500 points, a total of 1734 sets were obtained; 1449 of them correspond to *normal* cutting conditions, 112 to *run-out* situations, and 173 to *breakage* conditions. Features are extracted from these 500 points by using FFT analysis. Only the two first values of the transformed signal (100 and 200 Hz) are taken because, as it was shown before, the run-out and tool breakage are clearly revealed in the (50…200 Hz) band. The total amount of data was divided into a training dataset (70 % of the samples), which was used for fitting the neural network model, and a validation dataset (30 % of the samples), which was used to test the generalization capabilities of the previously fitted model.

Table 2.11 Distribution of the training data

Materials	f_{rate}	*Drilling result*			*Totals*		
	(mm/min)	*Normal*	*Run-out*	*Breakage*			
Ti_6Al_4V	5	16	2	1	19	24	46
	10	1	0	2	3		
	15	1	0	1	2		
$W_{78}Cu_{22}$	5	6	5	1	12	22	
	8	6	1	1	8		
	10	1	0	1	2		

The adjusted model had a R^2, equal to 0.9586, indicating that the model as fitted explains 95% of the variability in the dependent variable. The standard error of estimate is 0.0899, and the mean absolute error is 0.0266. From the analysis of variance (ANOVA) of the model (see Table 2.12), it is evident that the significant relationship between the variables is at the 99 % confidence level.

Table 2.12 Analysis of variance of the neural network model for run-out detection

Source	*Sum of Squares*	*Degrees of Freedom*	*Mean Square*	*F-Ratio*	*p-Value*
Model	189.30	200	0.9465	117.2	0.0000
Residuals	8.17	1012	0.0081		
Total	197.47	1212			

In order to test the generalization capabilities of the model, the obtained predictions for the training and validation sets were compared (see Figure 2.21). The difference between the mean values of the predictions of the training and validation sets is neglected. In addition, the mean predicted values of both sets are very similar to the observed values.

Nevertheless, the spread of the predictions for the training set is remarkably less than the spread for the validation set. Consequently, the predictions of the model, under different conditions, are not as reliable as those obtained for data used in the training process.

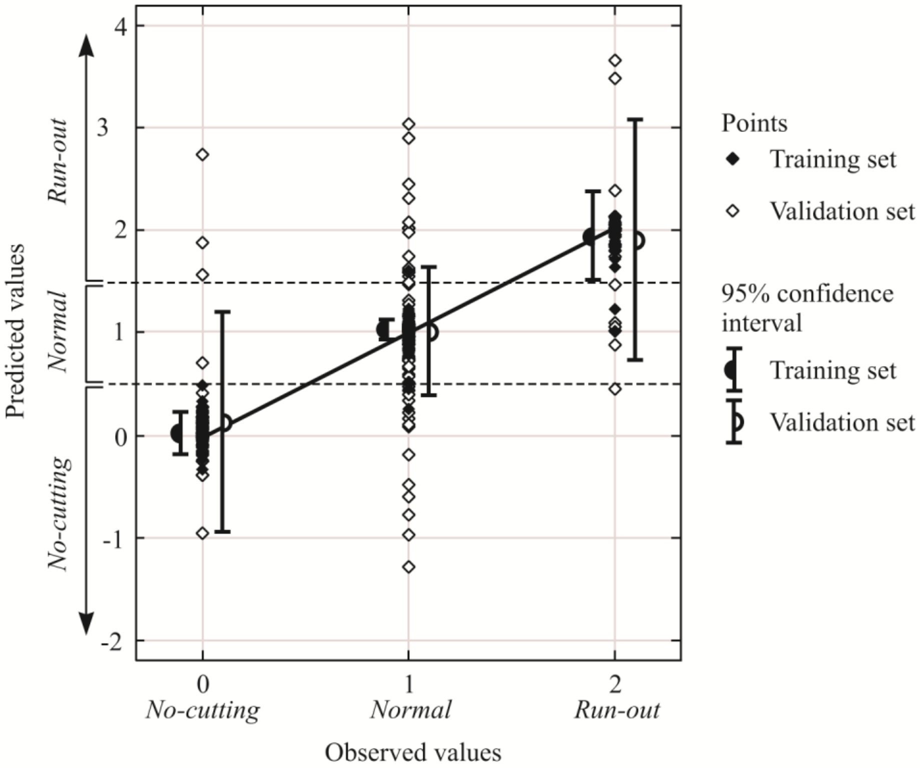


Figure 2.21 Predictions of the model for the training and validation sets

Table 5 shows the probability of predicting a condition from some observed data. As it can be seen from these outcomes, the probability of misunderstanding some data is lower than 30% for any condition. Specially, the probability of wrongly classifying some data coming from a *run-out* condition (false acceptances) is 25.2%, while the probability of identifying wrongly a *run-out* from data coming from other condition (false rejects) is only 6.1%. In this validation, the system was able to detect more than 70% of the run-out conditions with less than 10 % of false detections. For micro-drills, detecting and reducing run-out can yield considerable gains in tool life and productivity.

Table 2.13 Probability of predictions for the validation set

		Predicted condition		
		No cutting	*Normal*	*Run-out*
Observed condition	*No cutting*	75.5%	23.9%	0.6%
	Normal	6.1%	88.4%	5.5%
	Run-out	0.9%	24.3%	74.8%

By the other hand, holes quality errors are an undesired but unavoidable consequence in drilling operations. Due to the small dimensions involved in the micro-drilling processes, quality measurement and control must be carried out offline, by using microscopy or other high precision measurement devices. The study about the correlation between the holes quality and the force signals in the micro-drilling process of 0.1 mm and 0.5 mm-diameter holes in a sintered tungsten-copper alloy was presented in [231].

For analysis purposes, each signal was divided in four segments (see Figure 2.22):

- Empty motion: corresponds to the period where the tool is not cutting yet;
- Tool entrance: includes from the instant where the tool tip touches the metal surface until the tool tip is completely inside the metal;
- Forward feed: comprises the rest of the downward motion of the tool;
- Backward feed: includes from the upward motion of the tool until leaving the hole.

This classification was carried out because each segment has its own physical characteristics and the corresponding force signals are, consequently, of different magnitude. The empty motion signals were used as a reference in order to differentiate from the other ones, the component actually created by the cutting process from what is just noise.

The holes' quality error at the entrance of the holes was measured through a three-dimensional scanning of the top surface of the part. This scanning was carried out by a Zygo NewView 600s white-light interferometry microscope, having a camera resolution of 0.55 μm and a measurement array of 640×480 points, corresponding to a field of view of (352×264) mm, provided by a 20X Mirau objective. The surface coordinate, *z*, was measured in the interval (–7 … 7) μm; outside these bounds, the coordinate values cannot be determined and, consequently, it can be considered that these areas correspond to the damaged zone.

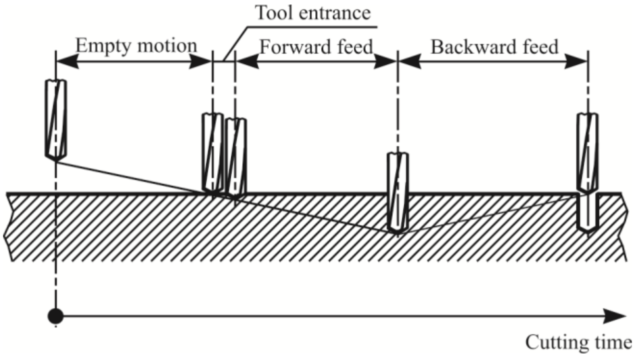
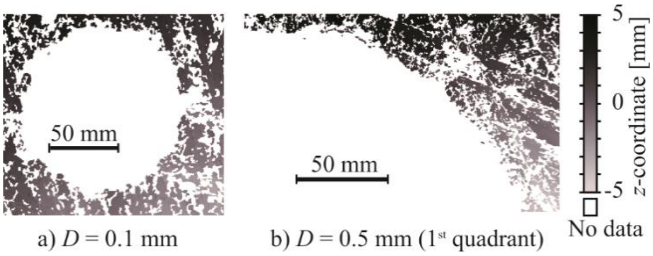


Figure 2.22 Segments of the cutting operation

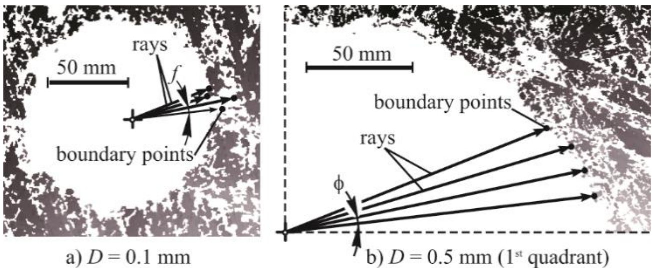
Figure 2.23 shows the whole circumference scanned for a 0.1 mm-diameter hole. On the contrary, for 0.5 mm-diameter holes, the whole circumference is greater than the field of view of the microscope and, then, approximately a quadrant can be scanned at once. For obtaining the holes' quality error, the first step is to identify the boundary of the holes' entrance. This was done by taking some point as a provisionally center for referencing purposes. This provisionary reference center is located at the middle point of the area, if the whole circumference is included in the available data (see Figure 2.24 a)). On the contrary, if only a quadrant is included into the available data, the eventual center is located at the corner nearest to the actual center (see Figure 2.24 b)).



a) $D = 0.1$ mm b) $D = 0.5$ mm (1st quadrant)

Figure 2.23 Scanned surfaces of the holes entrance

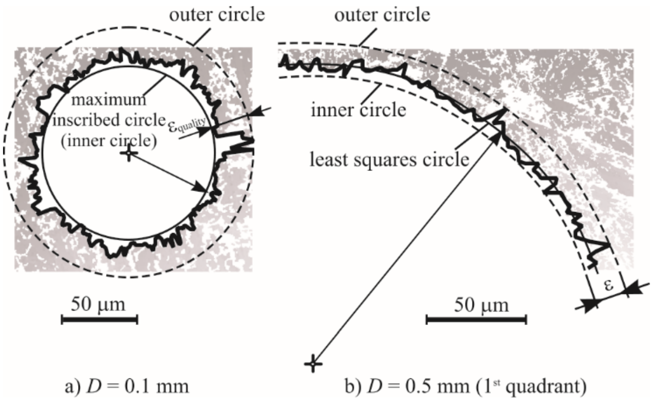
From this provisionally center, the considered angle q_{cen} (the entire circle, $q_{cen} = 2\pi$, or just a quadrant, $q_{cen} = \pi/2$, respectively) is divided by *i-th* rays, separated by an angle $\phi = q_{cen}/i$. Then, for each ray, the place where it intersects the point having some value of surface measurement is taken as a boundary point (see Fig. 8). For this problem, an angle $\phi = 1°$ was used.



a) $D = 0.1$ mm b) $D = 0.5$ mm (1st quadrant)

Figure 2.24 Determination of the holes entrance boundary

Depending on the amount of available data, the holes' quality error is determined by using different methods. When the whole circumference is available, the maximum inscribed circle (see Figure 2.25 a)) is determined because this method is more accurate for irregular boundaries. Nevertheless, this method requires the availability of the whole boundary; therefore, when only a sector of the boundary is available, the least square circle is computed (see Figure 2.25 b)).



a) $D = 0.1$ mm b) $D = 0.5$ mm (1st quadrant)

Figure 2.25 Determination of the holes quality error

This approach is based on the minimization of the expression [270]:

$$\sum_{i=1}^{nb}\left[\left(x_c - x_i\right)^2 + \left(y_c - x_i\right)^2 - radius^2\right]^2 \tag{2.7}$$

where, *nb* is the number of boundary points; x_{i} and y_{i} are the coordinates of the *i-th* boundary points; x_{C} and y_{C} are he coordinates of the center and *radius* is the average diameter of the circle. After determining the center and the radius for the average circle, the maximum and minimum radius, $radius_{\max}$ and $radius_{\min}$, are computed as the higher and lower distances from the center to any boundary point. The difference between these radiuses is the, so-called, roundness error (see Figure 2.25), that in our case corresponds to the quality error:

$$\varepsilon_{quality} = radius_{\max} - radius_{\min} \tag{2.8}$$

From the determined circles, i.e., maximum inscribed circle and least squares circles, depending on the case, the inner and outer circles are obtained for each hole and, therefore, the hole quality error, ε, is computed. Table 2.14 shows for each hole the obtained inner, outer and mean radiuses and the holes quality error. The three components of the forces were measured during the drilling process. The behavior of these signals, in three different intervals (tool entrance, forward motion and backward motion), was analyzed using a wavelet toolbox package. A Haar wavelet-based 6th order decomposition was used. The plotted frequency interval was limited by the cutoff frequency of the used force sensors (approx. 5kHz). The power spectra for the three components of the force in the 0.1 mm-diameters holes have a noticeable increase in the band near of 2.5 kHz. The Figure 2.26 analyzes the signal increase behavior and to relate this behavior with the holes quality error. The power of the *x*- and *y*-components growths at high frequencies (around 4.0 kHz), while in the z-component increases at the lower frequencies.

Table 2.14 Inner, mean and outer radiuses and holes quality error

D (mm)	*No.*	*Inner radius, $radius_{in}$ (μm)*	*Outer radius, $radius_{out}$ (μm)*	*Mean radius, $radius_{mean}$ (μm)*	*Hole quality error, $\varepsilon_{quality}$ (μm)*
0.1	1	56.7	81.1	63.9	24.3
	2	56.9	82.2	63.2	25.2
	3	56.7	84.0	63.6	27.3
	4	56.3	113.4	67.6	57.2
	5	55.7	88.8	63.8	33.1
0.5	1	254.9	293.0	281.7	38.1
	2	239.6	267.4	252.5	27.9
	3	234.6	272.9	264.4	38.4
	4	248.0	286.1	297.4	38.1
	5	303.2	334.8	325.2	31.6

This behavior can be detected even before the cutting start (during the free motion), so it can be expected that it should be closely related to the natural frequencies of the machine-tool-workpiece system. However, after the beginning of the cutting processes, the increase of the power in these frequency bands is significantly higher.

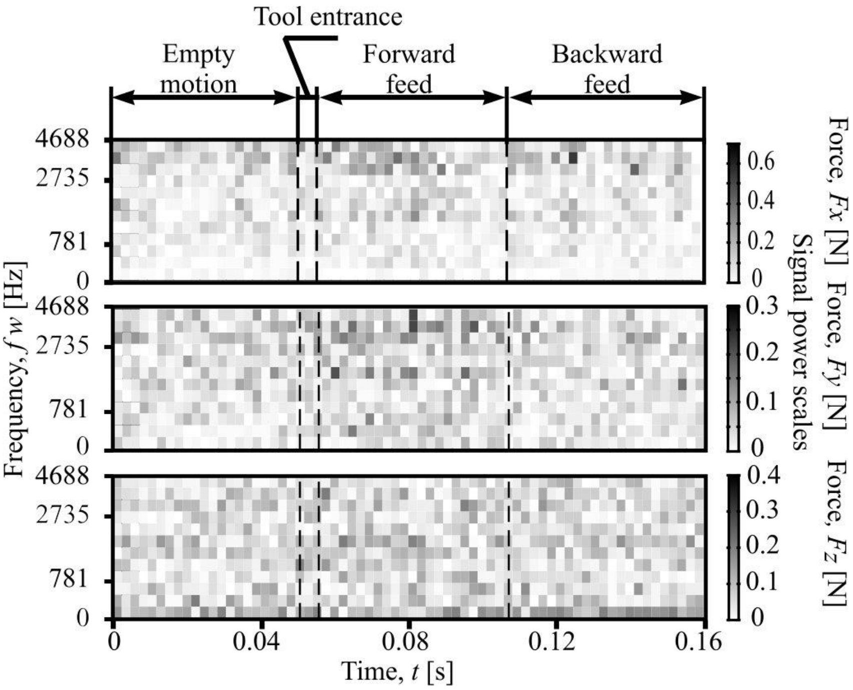


Figure 2.26 Wavelet package spectra for the first 0.1 mm-diameter hole

In the 0.5 mm-diameter power spectra (see Figure 2.27), the change of the signal power after the beginning of the cutting process is more evident due to the higher signal-to-noise ratio. The behavior of the *x*- and *y*-components of the force is very similar to the 0.1 mm-diameter signals, showing a raise in the neighborhood of the 2.5 kHz and 4.0 kHz bands, especially noticeable in the forward feed interval. On the contrary, the increasing in the lower band for the *z*-component of the force is not present in the 0.5 mm-diameter signals.

In order to obtain key signatures associated to the signal behavior, each frequency band of each segment (i.e., tool tip entrance, forward feed and backward feed) was independently analyzed. The differences between the mean power value of every component of the force and the corresponding mean values of the empty motion signals were computed. This difference keeps only the information from the cutting process disregarding the contribution from the empty motion.

The computed parameters representing the power spectra change (36 for each component of the force, i.e., three for each cutting interval and 12 for each frequency band obtained according to the used method for wavelet decomposition) are shown in Figure 2.28, for the first hole of each diameter. The power spectra for the force components show an irregular behavior for both diameters. However, the 0.5 mm-diameter signals show mostly positive values. That is in full correspondence with the higher signal-to-noise ratio of the 0.5 mm-diameter drilling process.

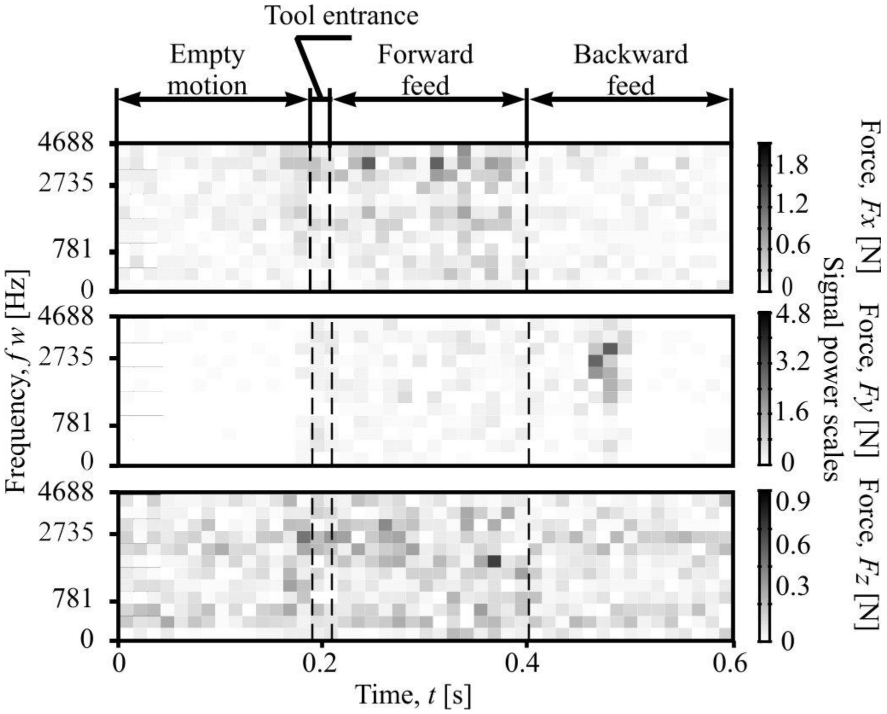


Figure 2.27 Wavelet package spectra for the 0.5 mm-diameter holes

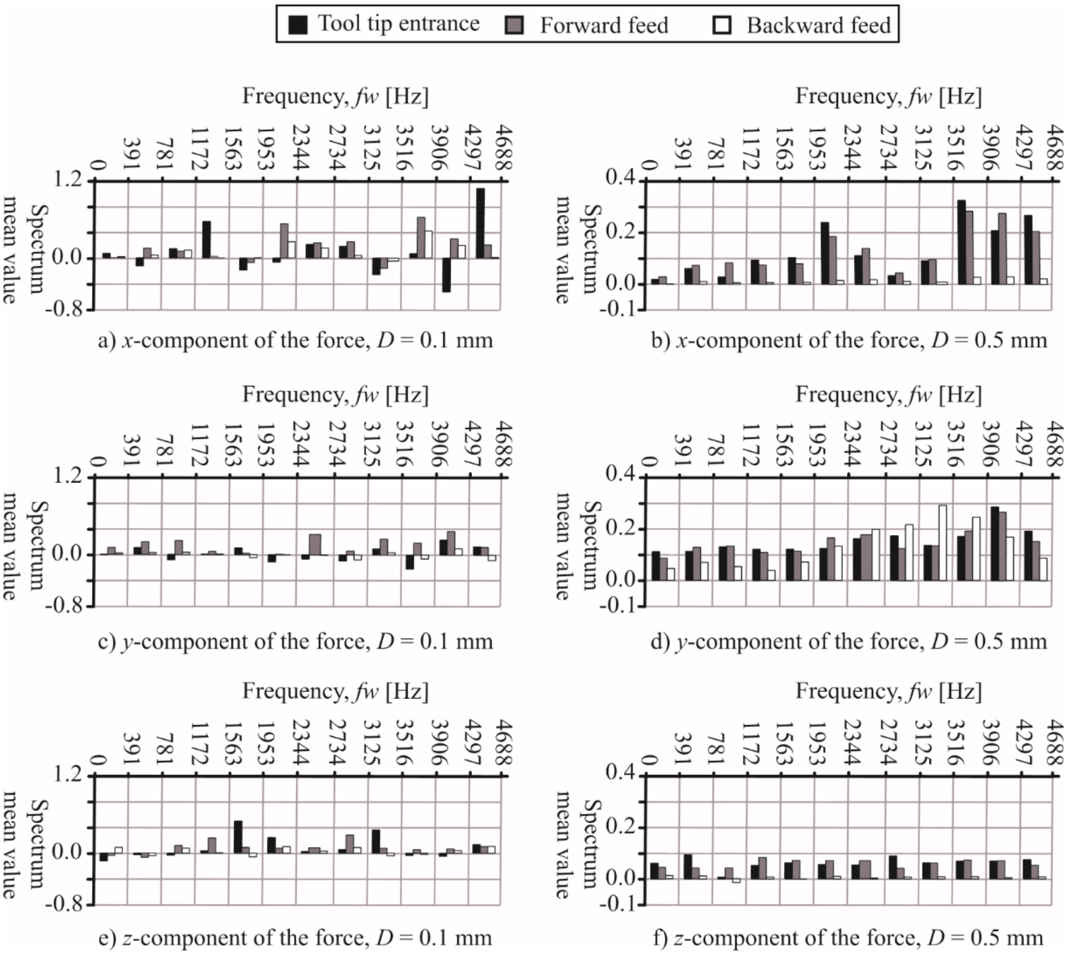
In order to determine the features that are more closely related to the holes' quality error, linear regression analyses were carried out between the selected features and hole diameter (as independent variables) and the holes' quality error (as dependent variable). Another interesting fact is that those features with a higher correlation only belong to the signal segments in the tool tip entrance and the backward feed. This can be explained if the damage is mainly produced at the tool tip entrance and, later on, this damage affects the chip exit during the backward motion. In addition, the frequency bands of the higher correlated features, corresponds to the natural frequencies of the systems, as was shown in the previous section.

The final model, for representing the main force features, was also obtained by multiple regressions, but by considering only the previously selected features and the holes diameter as independent variables (see Table 2.15).

The final model was obtained by removing the dependent variables from the model, one at once, until to maintain only those terms being statistically significant at 95% of confidence. The obtained model can be expressed by the equation:

$$\varepsilon_{quality} = 32.4 - 350 \bullet F_{z,\mathrm{back}}^{(0\ldots391)} + 1.05 \bullet F_{z,\mathrm{back}}^{(3\,906\ldots4\,297)} \tag{2.9}$$

having a correlation coefficient, R^2, equal to 0.8238, meaning that the model as fitted explains the 82% of the variability in the observed data. From the analysis of the variance of the model (see Table 2.15), it can be concluded that there is a statistically significant relationship between the independent and dependent variables at a 99% of the confidence level.



a) *x*-component of the force, *D* = 0.1 mm

b) *x*-component of the force, *D* = 0.5 mm

c) *y*-component of the force, *D* = 0.1 mm

d) *y*-component of the force, *D* = 0.5 mm

e) *z*-component of the force, *D* = 0.1 mm

f) *z*-component of the force, *D* = 0.5 mm

Figure 2.28 Mean values of the force signals power spectra change during the cutting process, for the first holes of each diameter

There are some interesting results derived from the obtained model. Firstly, the only signal having features that are highly correlated with the holes' quality error is the *z*-component of the force. This may be explained by the changes in the friction force between the tool, the chip and the holes' entrance profile.

Evidently, these changes seem to affect more heavily the variations of the through force (*z*-component) than the cross-section forces (*x*- and *y*-components). Another important aspect is the comparison between the different cutting intervals. The greater correlation appeared in the backward feed motion. It would be due to the friction between the tool, the chip and the holes' profile. Finally, the frequency bands included in the model are also noteworthy. The first one is the frequency band from 3 906 Hz to 4 297 Hz, which has a direct relationship with the hole quality error. On the contrary, the lower frequency band (up to 391 Hz) has an inverse relationship with the holes quality error (see Figure 2.29).

Table 2.15 Values of the variables used in the multiple regression model

D (mm)	$F_{z,\text{entrance}}^{(3516\ldots3906)}$	$F_{z,\text{back}}^{(0\ldots391)}$	$F_{z,\text{back}}^{(3906\ldots4297)}$	$F_{z,\text{entrance}}^{(2344\ldots2734)}$	$\varepsilon\ (\mu m)$
0.1	–0.00346	0.00850	0.00411	0.00190	24.3
	0.00164	–0.00452	–0.00727	–0.06509	25.2
	–0.01215	0.01316	–0.00279	–0.05434	27.3
	0.03612	–0.04069	0.00950	0.04113	57.2
	0.00183	0.00000	–0.00136	0.00827	33.1
0.5	0.06944	0.01260	0.00424	0.05464	38.1
	0.04414	–0.00324	–0.00414	0.05305	27.9
	0.04532	0.00577	0.00534	0.05671	38.4
	0.06223	–0.00605	0.00230	0.05435	38.1
	0.06381	0.00527	0.00367	0.03474	31.6

Table 2.16 ANOVA of the hole quality model

Source	*Sum of Squares*	*Degrees of Freedom*	*Mean Square*	*F-Ratio*	*p-Value*
Model	701.3	2	350.7	10.37	0.002
Residuals	150.0	7	21.4		
Total	851.3	9			

The rationale of this behavior relies on the difference between the vibration characteristics of the machine-tool-part system (see Figure 2.30). The increase in the holes' quality error causes two simultaneous but essentially different phenomena: the increase in the gap between the tool and the holes' surface, and the increase in the irregularities of the holes entrance. The first one produces a decrease in the pressure on the chip while the second one raises the frequency of the change in this pressure. It must be noted that there is not relationship between the gap and the pressure frequency or between the irregularities and the pressure mean value.

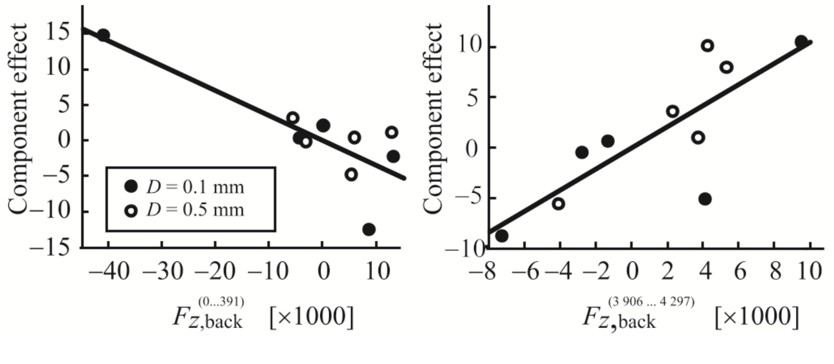


Figure 2.29 Components effect on the model

The main effect of the mean value of the pressure is the chip deformation, which, in turn, causes a rising in the damping with the consequent decrease in the low frequencies amplitudes. On the other hand, the increase in the frequency of the pressure variation creates a high frequency excitation force that increases the high frequency vibrations. The main outcomes of this study are the basement for obtaining reliable models for monitoring systems in micro-drilling operations.

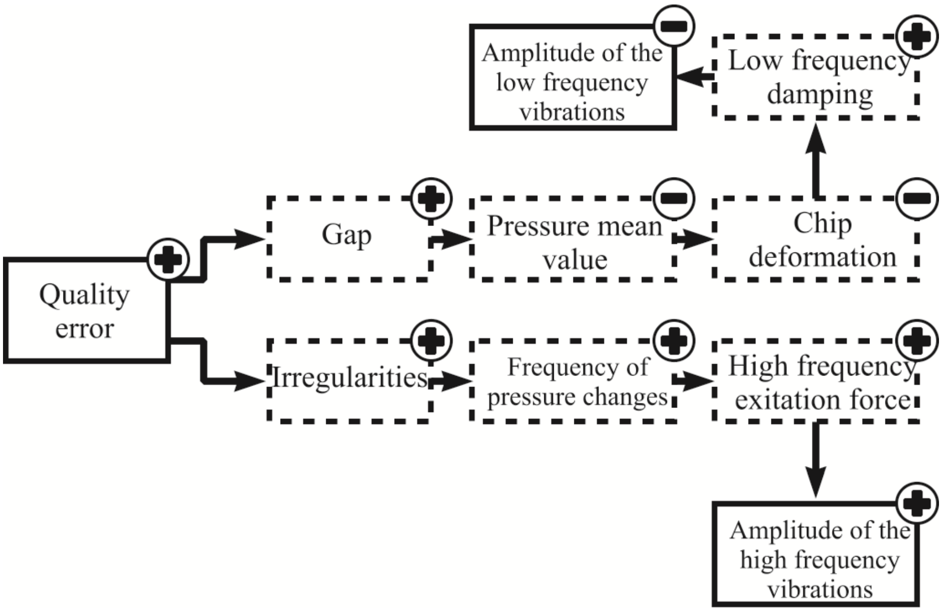

Amplitude of the low frequency vibrations
Low frequency damping
Quality error
Gap
Pressure mean value
Chip deformation
Irregularities
Frequency of pressure changes
High frequency exitation force
Amplitude of the high frequency vibrations

Figure 2.30 Phenomenological relationships between the model variables

2.4 Micro-milling processes

2.4.1 Surface roughness prediction models in micro-milling operations

Nowadays, the micrometric and nanometric dimensional precision of industrial components is a common feature of micro-milling manufacturing processes. Hence, a great importance is given to key issues such as online metrology and real-time monitoring systems for accurate control of surface roughness and dimensional quality. A real-time monitoring system is proposed in [255] in order to predict surface roughness with an estimation error of 9.5%.

In the experimental setup, the z-axis component vibration is measured using two different diameters (0.5 mm and 1 mm-diameter) under several cutting conditions on a sintered tungsten–copper alloy ($W_{78}Cu_{22}$). Then, an ANFIS model is implemented for modeling surface roughness, yielding a high goodness of fit indices and a good generalization capability.

A Multiple Linear Regression (MLR) technique was used to obtain the first model. MLR is a generalization of linear regression by considering more than one independent variable, and a specific case of general linear models formed by restricting the number of dependent variables to one. In this particular case, the relationship between the dependent variable logarithm average surface roughness (*Ra*) and the logarithms of the independent variables: cutting speed (V_c), feed rate per tooth (f_{tooth}) and axial cutting deep (a_p) were employed to generate an exponential model, as described in Eq. (2.10). The available experimental data was divided into a training set, composed of 79 data samples (corresponding to 70% of the total dataset), and a validation dataset composed of the remaining 35 data samples.

$$Ra = e^{71.74} \bullet V_c^{0.4267} \bullet f_{tooth}^{0.2955} \bullet a_P^{0.5613} \tag{2.10}$$

The regression model has a correlation coefficient, R^2, of 0.8956, which explains 89% of the variability in the response variable. Despite the good correlation coefficient obtained with the regression model, its principal problem was a generalization capability of below 30%.

This generalization capability was determined by using the probability value of the comparison of the means of the fitting and the validation set residuals. If there is a statistically significant difference between both means, therefore the model has bad generalization capabilities. The ANOVA of the model is shown in Table 2.17. There is a statistically significant relationship between the variables dependents and independent at a confidence level of 95%.

In micro-milling processes, the vibration signals captured during cutting time are very noisy and with many frequency bands (servo-motors noise, machine stiffness, etc.). It is therefore very difficult to obtain a regression model with a good generalization capability. An alternative approach is to apply Artificial Intelligence techniques, to filter uncertainty, background noise, and the time-variant behavior of micro-milling processes. An Adaptive Network-based Fuzzy Inference System was selected to create the model due to its computational simplicity and suitability for real-time applications [271]. In this particular case, Gaussian membership functions are used. Once defined, the rules are normalized according to their importance. The next step is to calculate the consequent parameters, i.e., the Takagi–Sugeno function for each fuzzy rule. Finally, defuzzification is performed in the last layer as shown in Fig. 3. The signal was filtered before the modeling step, using a Finite Impulse Response filter, to calculate the Root Mean Square. The window size of the filter, selected using the sampling frequency (*fs*), was 50 kHz, in all cases. The work frequency (*fw*) as a function of the rpm of the spindle was 0.75 kHz for a diameter of 0.5 mm and 0.67 kHz for a diameter of 1.0 mm.

Table 2.17 ANOVA of the *Ra* regression model

Source	*Sum of Squares*	*Degrees of Freedom*	*Mean Square*	*F-Ratio*	*p-Value*
Model	21.35	3	7.12	205.77	0.0000
Residuals	3.80	110	0.03		
Total	25.15	113			

The model was used for estimating *Ra*, but, in this case, it included the vibration of the z-axis (*Vz*) as an independent variable. The main difference with Eq. (2.10) is the use of z-axis vibrations as the main input of the model to predict average surface roughness. The same dataset and distribution, previously described in this chapter, were used to create the model.

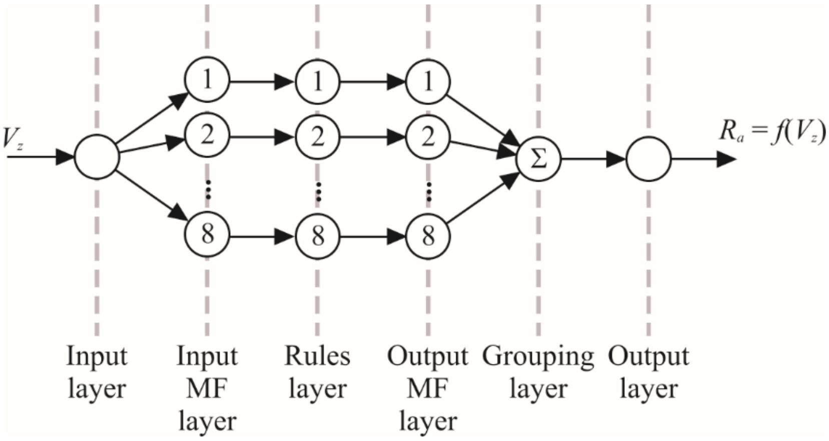


Figure 2.31 Surface roughness ANFIS model architecture

ANFIS model has a correlation coefficient, R^2, of 0.8735, and a generalization capability of 0.9315. The residuals from the training and validation sets were analyzed through a comparison of the mean values and standard deviations of both sets, employing a *t-Student* and an *F-test*, respectively. Therefore, the null hypothesis (both means are equal and both standard deviations are equal) cannot be rejected at the 95% confidence level. Therefore, both sets of residuals may be found in the same distribution and the model has a good generalization capability. In conclusion, ANFIS was the model selected to correlate the influence of the cutting parameters on the surface roughness.

Finally, the optimization task is executed by considering two contradictory objectives: unit machining time and surface roughness. A multi-objective genetic algorithm is also applied to solve the optimization problem, obtaining a set of non-dominated solutions. Pareto front representation is a useful decision-making tool for operators and technicians in the micro-milling process. An example of the Pareto front utility-based approach that selects two points close to both extreme ends of the frontier, in the first case (point 1), machine time is of greater importance, and in the second case (point 2), importance is attached to surface roughness. In general terms, users can select different combinations, at all times moving along the Pareto front. The Pareto front and the behavior of the vibration are shown in Figure 2.32. As it was remarked before, the main objective of this chapter is to address the modeling of micromachining processes. The next chapter will be entirely focused on multi-objective optimization applied to micromachining processes.

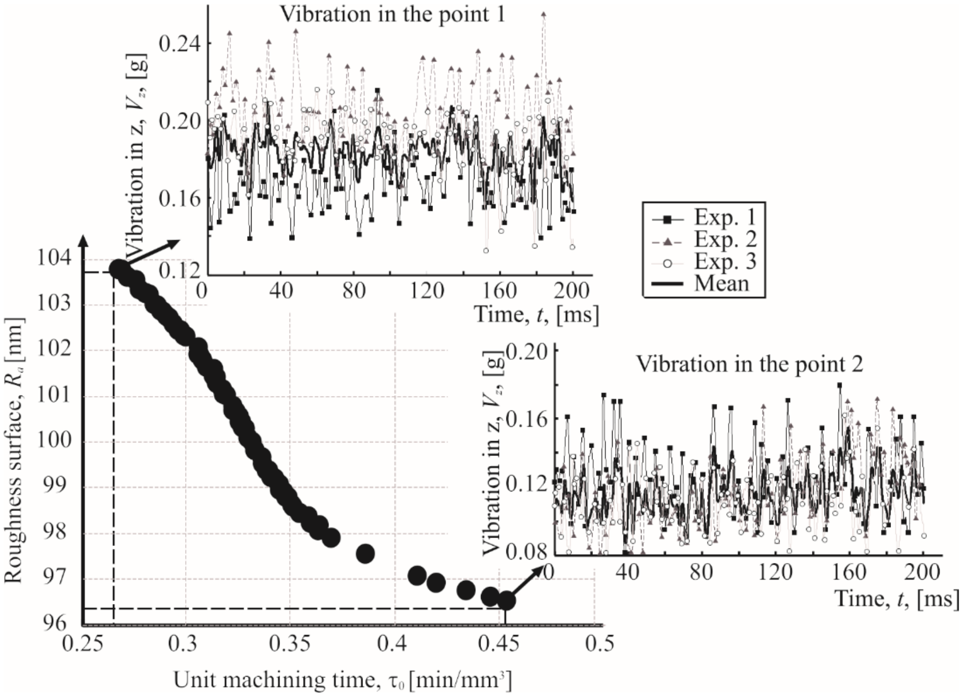


Figure 2.32 z-axis vibration behavior at points 1 and 2 of the Pareto's front

2.5 Models summary for micromachining processes

In this section, a summary of the principal models obtained during the research period are shown in Table 2.18. All the abbreviations presented in the summary are described along the chapter, also it is explained in the glossary of terms (see annex I).

Table 2.18 Model summary obtained during the research period

Process	*Inputs*	*Output*	*Feature Extraction*	*Modeling techniques*	*Optimization*	R^2	G_c	*Reference*
Micro-drilling	F_x, F_y, V_y, V_z	N_{hole}	Stats, FFT, HHT	Regression	-	-	-	[263]
Micro-drilling	F_x, F_y, F_z, V_y, V_z	N_{hole}	Stats, FFT, WT, HHT	Regression, MLP, ANFIS	-	0.75, 0.85, 0.38	~0, 0.73, 0.85	[193]
Micro-drilling	V_c, f_{rate}, $step$, h_d, σ_Y, HRB, δm, E, ρm, kc, c_h	F_z	FIR	MLP	Genetic Algorithm	0.89	0.61	[254]

Process	*Inputs*	*Output*	*Feature Extraction*	*Modeling techniques*	*Optimization*	R^2	G_c	*Reference*
Micro-drilling	*Material*, f_{rate}, F_x, F_y, F_z	*status*	FFT	MLP	-	0.96	-	[254]
Micro-drilling	F_z	ε	WT	Regression	-	0.83	-	[231]
Micro-milling	V_c, f_{tooth}, a_p	*Ra*	-	Regression	Genetic Algorithm	0.89	~.30	[255]
Micro-milling	V_z	*Ra*	RMS	ANFIS	Genetic Algorithm	0.87	0.93	[255]
Micro-drilling	V_c, f_{rate}, *step*, h_d, *HRB*, *E*, ρm, *kc*	F_z	FIR	Regression, ANFIS	Genetic Algorithm	0.84, 0.86	~0, 0.93	[260]
Micro-drilling	V_c, f_{rate}, *step*, h_d, *HRB*, *E*, ρm, *kc*	F_z	FIR	Regression, MLP, ANFIS	-	0.78, 0.73, 0.88	0.65, 0.97, 0.62	[272]
Micro-drilling	V_c, f_{rate}, *step*, h_d, *HRB*, *E*, ρm, *k*	V_p	RMS	Regression, MLP, ANFIS	-	0.27, 0.25, 0.72	0.67, 0.85, 0.34	[272]
Micro-milling	f_{tooth}, V_p	*Ra*	RMS	HIM	Simulated Annealing	-	-	[273]
Micro-drilling	V_c, f_{rate}, *step*	F_z	FIR	MLP	Cross Entropy	-	-	[274]
Micro-drilling	V_c, f_{rate}, *step*	V_p	FIR	MLP	Cross Entropy	-	-	[274]

2.6 Conclusions

The micro-scale processes are non-linear, time-variant processes that are difficult to represent with precise mathematical equations. Throughout this chapter, different modeling techniques for micromachining processes are presented to address the representation of important phenomena and characteristics such as run-out, hole quality and surface roughness.

Firstly, a comparative study of different feature extraction techniques in time and frequency domains as well as traditional (i.e., linear regression) and Artificial Intelligence-based strategies (multilayer perceptron, ANFIS) was presented. Among the extraction techniques under analysis, wavelet transform and HHT showed the best performance from the scientific viewpoint. On the other hand, multilayer perceptron yielded the best modeling results. The combination of wavelet transforms and multilayer perceptron was especially suitable, showing not only a high goodness of fit but also good generalization capabilities. Following, the use of artificial neural networks for modeling the behavior of the thrust force, in a micro-drilling process showed the advantages of using such a hybrid approaches. The neural network was capable of dealing with the complex, non-linear and noisy experimental data, acquired from real-time experiments in this study.

Secondly, two modeling techniques were presented: a two-step monitoring systems for detecting the occurrence of run-out in micro-drilling processes of Ti_6Al_4V and $W_{78}Cu_{22}$ alloys and a relationship between the thrust force signals and the holes' quality error in the micro-drilling process of a tungsten-copper alloy was derived. The monitoring system was able to identify more than 70 % of the run out conditions, with less than 10 % of false run out detections. Due to the intolerance of micro-drills to run out, the proposed approach can yield considerable savings in tools usage and increment in productivity. By the other hand, the wavelet spectrum toolbox was used to obtain the signal power distribution in the time-frequency domain, showing the behavior of the signal during three intervals of the cutting process: the tool tip entrance, the forward feed motion and the backward feed motion. It was found that the higher correlation exists in the frequency bands 0~391 Hz and 3906~4297 Hz, during the backward feed motion. An explanation was proposed for this fact, based on the simultaneous rising of the mean value and the frequency changes of the pressure on the chip.

Thirdly, a surface roughness modeling was carried out in this chapter for micro-milling operations. Nowadays, the micrometric and nanometric dimensional precision of industrial components is a common feature of micro-milling manufacturing processes. A real-time monitoring system based on a neuro-fuzzy model has been proposed as an alternative tool to predict surface roughness during micro-milling processes. This monitoring system was created to provide a reliable and economic procedure for predicting surface roughness. In this case, it combines a uniaxial vibration sensor and an AI-based procedure in a real-time monitoring platform. Furthermore, an ANFIS model was created to estimate surface roughness with a very good generalization capability, 0.93, and an estimation error of 9.5%. Finally, a summary of the principal models obtained during the research period was developed.

The next chapter is focused on a multi-objective optimization strategy based on estimation of distribution algorithm (EDA). Some modifications are introduced in the cross-entropy multi-objective optimization algorithm. The Simple Multi-Objective Cross-Entropy method (SMOCE) is proposed on the basis of a new procedure for addressing constraints, i.e., the use of variable cutoff values for selecting the elitist population and filtering of the elitist population after each epoch are some of the modifications presented in the algorithm.

Chapter 3
CROSS ENTROPY MULTI-OBJECTVE OPTIMIZATION ALGORITHM

This chapter presents two set of modifications respect to the cross entropy multi-objective optimization algorithm (MOCE) introduced by Bekker and Aldrich [114]. First, a group of modifications are introduced in the cross entropy multi-objective optimization algorithm, also called (MOCE+), based on a new procedure for addressing constraints: (i) the use of variable cutoff values for selecting the elitist population; and, (ii) filtering of the elitist population after each epoch. The second and final modifications packages are introduced in the Simple Multi-Objective Cross Entropy method (SMOCE), based on only four parameters (epoch number, working population size, histogram interval number, and elite fraction) stored in the algorithm, in order to facilitate the tuning process. The final proposed method (SMOCE) is evaluated using different test suites. Furthermore, a comparison with some other well-known optimization methods is carried out. The comparative study demonstrates the good figures of merit of the SMOCE method in complex test suites. Finally, the proposed method is validated in the multi-objective optimization of a micro-drilling process. Two conflicting targets are considered: total drilling time and vibrations on the plane that is perpendicular to the drilling axis. The Pareto front, obtained through the optimization process, is analyzed through quality metrics and the available options in the decision-making process.

This chapter consists of five sections. The first section explains the basic cross-entropy concepts and the modified (two set of modifications) introduced to the multi-objective cross-entropy method (MOCE+ in subsection 3.1.1 and SMOCE in subsection 3.1.2). In the second section, the sensibility of the algorithm to the remaining parameters is experimentally studied. Below, a comparative study with all available gradient-free methods is beyond the scope in the section 3.3. Subsequently, some solutions are presented that use both methods and the experimental study in which the empirical data was obtained from the micro-drilling process is described. Finally, the chapter conclusions are presented in the section 3.5.

3.1 Algorithm description

3.1.1 Multi-Objective Cross-Entropy algorithm (MOCE+)

The Cross-Entropy method is inspired by an adaptive variance minimization algorithm for estimating the probabilities of rare events on stochastic networks [109]. The main rationale of CE is the construction of a random sequence of solutions which converges probabilistically to an optimal or a near-optimal solution in two iterative stages [275]. In the first iteration, a sample of random data is generated according to a specified random mechanism. A better sample is produced in the next iteration and the parameters of the random mechanisms (i.e. parameters of the probabilistic density functions) are updated with the corresponding data [276] and [277]. Let *X* be a random variable on a space *X*, P_X is its probability density function (PDF), and let the score *f* be a real function in *X*. The CE method aims to find the minimum of *f* over *X*, and the corresponding states *x** that satisfy this minimum:

$$y^* = f(x^*) = \min_{x=X} f(x) \tag{3.1}$$

The CE method provides a methodology for creating a sequence of $x_0, x_1, \ldots, x_N$ and levels $y_0, y_1, \ldots, y_N$ such that y converges to y^* and *x* converges to x^*.

We are concerned with estimating the probability $\Omega(y)$ of an event $E_\upsilon = \{x \in X \mid f(x) \geq y\}, y \geq \mathbb{R}^+$. Defining a collection of functions I_υ for $x \in X, y \in \mathbb{R}^+$ [106].

$$I_\upsilon(x,y) = I_{\{f(x_i \geq y)\}} = \begin{cases} 1 if \ f(x) \leq y \\ 0 \quad otherwise \end{cases} \tag{3.2}$$

Let $\rho(-, \upsilon)$ a family of probability density functions on X parametrized by a real valued vector υ and $\rho(x, \upsilon)| \upsilon \in \Lambda$.

$$\Omega(y) = P_\upsilon(f(x) \geq y) = E_\upsilon \bullet I_\upsilon(x, \upsilon) \tag{3.3}$$

where E_υ denotes the corresponding expectation operator.

In this manner Eq. (3.3) converts the optimization problem into an associated stochastic problem with very small probability using a variance minimization technique such as importance sampling where the random sample is drawn from an a priori appropriate probability density function *h*. Taking a random sample x_0, x_1, … x_N from an importance sampling (different) density *h on X* and evaluating:

$$\hat{\Omega} = \max \frac{1}{N}\sum_{i=1}^{N} I_{\{f(x_i)\geq y\}} \bullet W(x_i) \tag{3.4}$$

where $\hat{\Omega}$ is called the importance sampling and $W(x) = \rho(x,\upsilon)/h(x)$ is called the likelihood ratio.

Searching the optimal sampling density $h^*(x)$ is problematic, since determination of $h^*(x)$ requires $\hat{\Omega}$, called the importance sampling, to be known.

$$h^*(x) = \frac{I_{\{f(x_i)\geq y\}} \bullet \rho(x,\upsilon)}{\hat{\Omega}} \tag{3.5}$$

Thus the parameter vector, called the referenced parameter or tilting parameter υ^*, should be chosen such that the distance between h^* and $\rho(x, \upsilon)$ is minimal, reducing the problem to a scalar case.

A measure of distance between two densities ρ and h is the Kullback-Leibler distance, also called cross-entropy between ρ and h:

$$\Gamma(\rho,h) = \int \rho(x) \bullet \ln \rho(x)dx - \int h^*(x) \bullet \ln \rho(x)dx \tag{3.6}$$

Minimizing $\Gamma(\rho(x, \upsilon), h^*)$ is equivalent to maximizing $\int h^*(x) \bullet \ln \rho(x)dx$ which implies:

$$\max_{\upsilon} \Gamma(\upsilon) = \max_{\upsilon} Ep(I_{\{f(x_i)\geq y\}} \bullet \ln \rho(x,\upsilon)) \tag{3.7}$$

Using again the importance sampling, we can rewrite Eq. (3.6) to compute the expectation in Eq. (3.7). Therefore we can draw a sample x_0, x_1, … x_N from ρ and estimate the maximum (or minimum) of $\hat{D}(\upsilon)$:

$$\max_{\upsilon} \hat{\Gamma}(\upsilon) = \max_{\upsilon} \frac{1}{N}\sum_{i=1}^{N} I_{\{f(x_i)\geq y\}} \bullet \frac{p_x(x_i)}{h(x_i)} \bullet \ln \rho(x,\upsilon) \tag{3.8}$$

However h is still unknown in Eq. (3.8). The CE algorithm tries to overcome this difficulty by adaptively constructing a sequence of parameters $\left(y_t \middle| t \geq 1\right)$.

The proposed algorithm, called MOCE+, is inspired by the multi-objective cross-entropy approach, proposed by [114] (see Figure 3.1). All the functions of the MOCE+ algorithm described below are available from the following repository: http://gamhe.eu/downloads/CEMOO/MOCE+/.

The algorithm is focused on solving a multi-objective optimization problem, given by:

$$\min \mathbf{y} = \mathbf{f}(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{y} \in \mathbb{R}^m \tag{3.9}$$

where:

$$l_i \le x_i \le u_i, \qquad i = 1 \ldots n \tag{3.10}$$

constrained by:

$$g_i(x) \le 0, \qquad i = 1 \ldots p \tag{3.11}$$

It starts by creating an empty elite population, $\left\{ x_{ij}^{*} \right\} = \left\{ \varnothing \right\}$, source which will store the best solutions throughout the execution time of the algorithm.

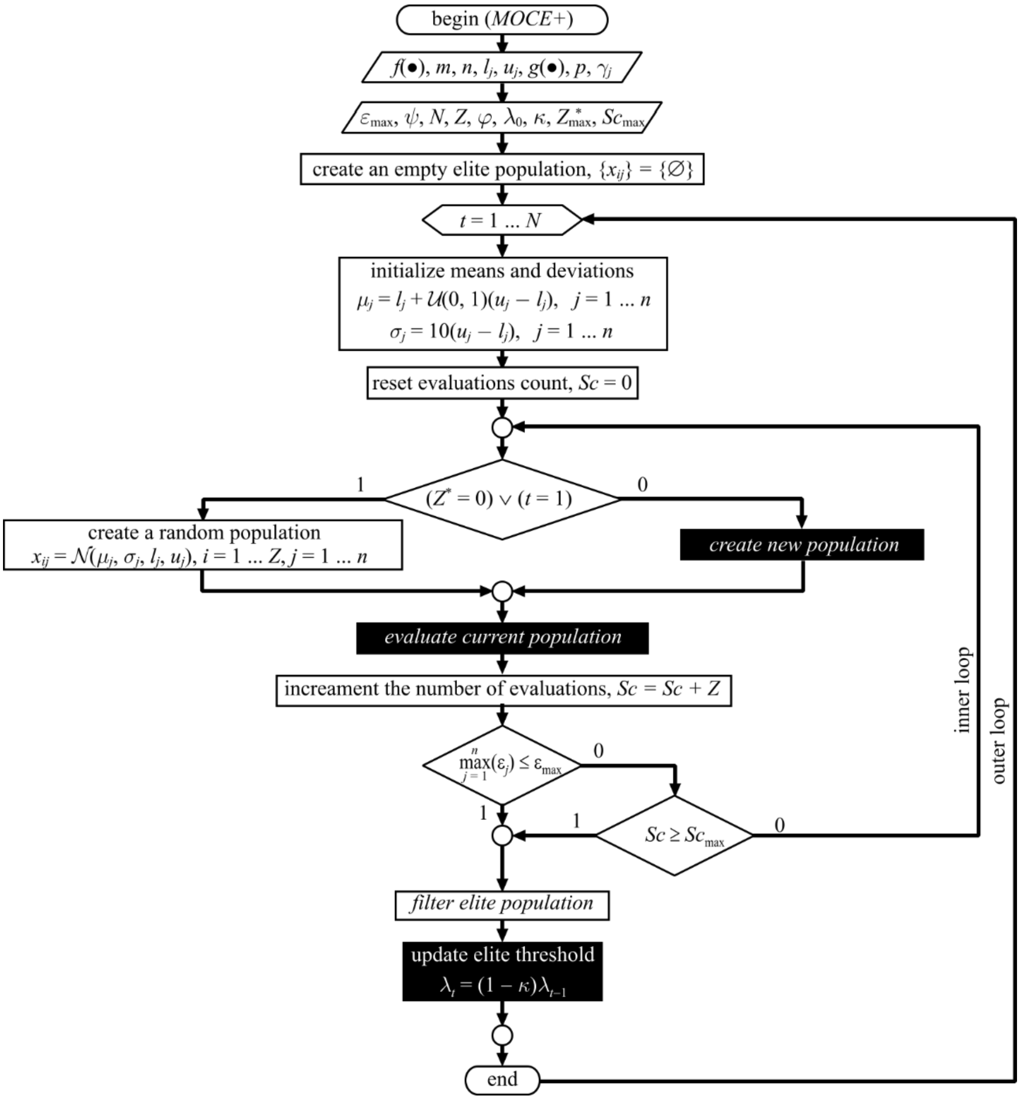

begin (*MOCE+*)
$f(\bullet)$, m, n, l_j, u_j, $g(\bullet)$, p, γ_j
$\varepsilon_{\max}$, ψ, N, Z, φ, λ_0, κ, $Z^*_{\max}$, $Sc_{\max}$
create an empty elite population, $\{x_{ij}\} = \{\varnothing\}$
$t = 1 \ldots N$
initialize means and deviations
$\mu_j = l_j + \mathcal{U}(0, 1)(u_j - l_j), \; j = 1 \ldots n$
$\sigma_j = 10(u_j - l_j), \; j = 1 \ldots n$
reset evaluations count, $Sc = 0$
1
$(Z^* = 0) \vee (t = 1)$
0
create a random population
$x_{ij} = \mathcal{N}(\mu_j, \sigma_j, l_j, u_j), i = 1 \ldots Z, j = 1 \ldots n$
create new population
evaluate current population
increament the number of evaluations, $Sc = Sc + Z$
$\max_{j=1}^{n}(\varepsilon_j) \le \varepsilon_{\max}$
0
1
1
$Sc \ge Sc_{\max}$
0
filter elite population
update elite threshold
$\lambda_t = (1 - \kappa)\lambda_{t-1}$
end
inner loop
outer loop

Figure 3.1 General algorithm (black background elements represent introduced improvements)

Then, a loop (outer loop) of *N* iterations is performed, where the values of the means and standard deviations, μ_i and σ_i : *i*=1 ... n, for each decision variable are computed from the corresponding intervals, by:

$$\mu_i = l_i + u(0,1)(u_i - l_i), \quad i = 1 \ldots n \tag{3.12}$$

$$\sigma_i = 10(u_i - l_i), \quad i = 1 \ldots n \tag{3.13}$$

Note that while the means are randomly chosen from the interval, the standard deviation is deterministically computed as 10 times the length of the interval. Then, a nested loop (inner loop) takes place until it reaches some stopping conditions. It starts by creating the working population, $\{x_{\mathrm{ij}}, i = 1 \ldots Z, j = 1 \ldots n\}$, which can be done in two different ways. If the elite population is empty or if it is the first iteration in the loop, the working population is generated from normal random distributions, with means μ_i and deviations σ_i, truncated in the intervals $[l_i, u_i]$, for each decision variable. Otherwise, the working population is created from the elite population (see Figure 3.2).

During the investigation the first result were introduced in [274]. A new creation of the working population from the elite solutions is carried out in a different way, as suggested by [114]. In the original approach, a frequency histogram, with $r = t + 2$ intervals (where *t* is the epoch) is constructed for each variable, from the solutions contained in the elite population. Then, with some likelihood, $0 \le \varphi \le 1$, the frequencies of the histogram are inverted, in order to avoid convergence to a narrow sector of the Pareto front. Finally, the mean and standard deviations are computed for each histogram interval and a subset of new solutions are created from a truncated normal random distribution; each subset contains a number of solutions that are proportional to the frequency of the corresponding histogram interval. As the number of elements in the subset is an integer, round errors may occur. In such cases, some elements are added to the last interval until the prescribed working population size, *Z*, is reached.

Two improvements were introduced in this section of the algorithm: the number of histogram intervals, *r*, was computed as the minimum of $t + 2$ and a fifth of the elite population size, *Z**, avoiding an excessive number of intervals for small elite populations. A further improvement was integrated in which the added solutions compensated the rounded errors. These solutions were incorporated into randomly selected subsets, instead of into the subset. As modifications, they were intended to avoid inappropriate trends in the creation of those solutions.

When the new working population was created, each solution was evaluated according to the procedure diagram shown in Figure 3.3, obtaining the corresponding set of objective vectors, $\{y_{\mathrm{ij}} = f_{\mathrm{j}}(x_{\mathrm{i1}}, \ldots, x_{\mathrm{in}}), i = 1 \ldots Z, j = 1 \ldots m\}$. After this step, a further improvement was added, in order to deal with constraints. This variation relies on the penalty approach that modifies the values of the objective functions through the expression:

$$y_{\text{ij}} = y_{\text{ij}} + \sum (\gamma_{\text{k}} \max(0, g_{\text{k}}(x_{\text{i1}}, \ldots, x_{\text{in}}))), \qquad i = 1 \ldots N, \qquad j = 1 \ldots m \tag{3.14}$$

where, $\gamma_{\text{k}}, k = 1 \ldots p$, are the prescribed weights for each constraint.

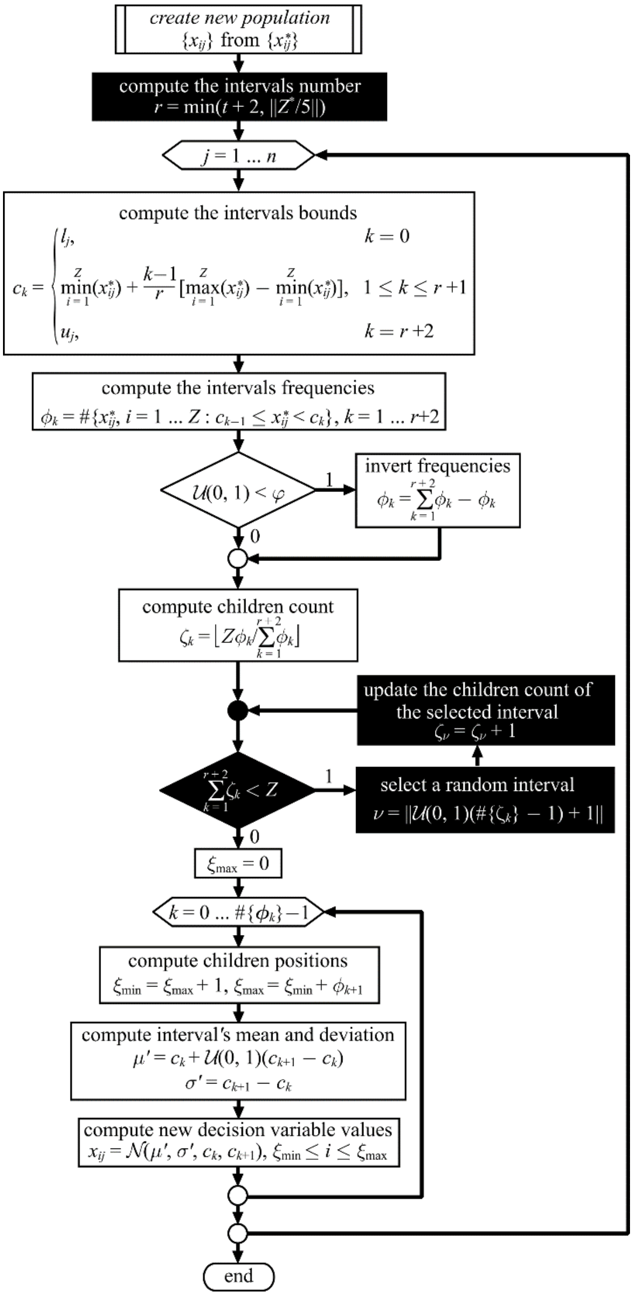

create new population
$\{x_{ij}\}$ from $\{x^*_{ij}\}$
compute the intervals number
$r = \min(t + 2, \|Z^*/5\|)$
$j = 1 \ldots n$
compute the intervals bounds
$c_k = \begin{cases} l_j, & k = 0 \\ \min_{i=1}^{Z}(x^*_{ij}) + \frac{k-1}{r}[\max_{i=1}^{Z}(x^*_{ij}) - \min_{i=1}^{Z}(x^*_{ij})], & 1 \le k \le r+1 \\ u_j, & k = r+2 \end{cases}$
compute the intervals frequencies
$\phi_k = \#\{x^*_{ij}, i = 1 \ldots Z : c_{k-1} \le x^*_{ij} < c_k\}, k = 1 \ldots r+2$
$\mathcal{U}(0, 1) < \varphi$
1
0
invert frequencies
$\phi_k = \sum_{k=1}^{r+2} \phi_k - \phi_k$
compute children count
$\zeta_k = \lfloor Z\phi_k / \sum_{k=1}^{r+2} \phi_k \rfloor$
update the children count of the selected interval
$\zeta_\nu = \zeta_\nu + 1$
$\sum_{k=1}^{r+2} \zeta_k < Z$
1
0
select a random interval
$\nu = \|\mathcal{U}(0, 1)(\#\{\zeta_k\} - 1) + 1\|$
$\xi_{\max} = 0$
$k = 0 \ldots \#\{\phi_k\} - 1$
compute children positions
$\xi_{\min} = \xi_{\max} + 1, \xi_{\max} = \xi_{\min} + \phi_{k+1}$
compute interval′s mean and deviation
$\mu' = c_k + \mathcal{U}(0, 1)(c_{k+1} - c_k)$
$\sigma' = c_{k+1} - c_k$
compute new decision variable values
$x_{ij} = \mathcal{N}(\mu', \sigma', c_k, c_{k+1}), \xi_{\min} \le i \le \xi_{\max}$
end

Figure 3.2 Creating new working population

Then, each solution in the working population was ranked by its Pareto dominance: in other words, by considering the number of other solutions which dominate it. The elite population, $\{x^*_{\text{ij}}\}$ source was increased, by adding all the solutions with a lower Pareto dominance than some threshold value, λ_{t}.

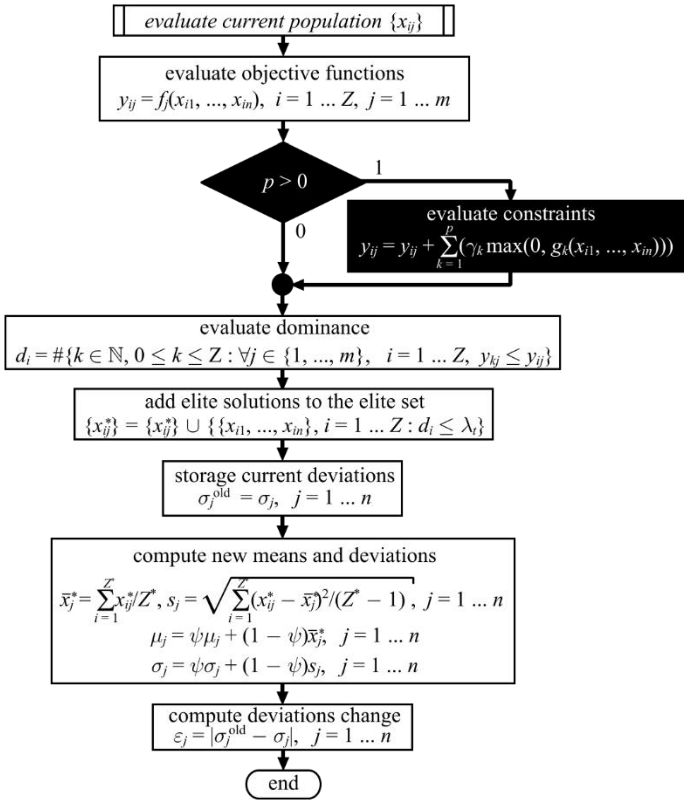

evaluate current population $\{x_{ij}\}$
evaluate objective functions
$y_{ij} = f_j(x_{i1}, ..., x_{in}),\ i = 1 ... Z,\ j = 1 ... m$
$p > 0$
1
0
evaluate constraints
$y_{ij} = y_{ij} + \sum_{k=1}^{p}(\gamma_k \max(0, g_k(x_{i1}, ..., x_{in})))$
evaluate dominance
$d_i = \#\{k \in \mathbb{N}, 0 \le k \le Z : \forall j \in \{1, ..., m\},\ i = 1 ... Z,\ y_{kj} \le y_{ij}\}$
add elite solutions to the elite set
$\{x^*_{ij}\} = \{x^*_{ij}\} \cup \{\{x_{i1}, ..., x_{in}\}, i = 1 ... Z : d_i \le \lambda_t\}$
storage current deviations
$\sigma_j^{old} = \sigma_j,\ j = 1 ... n$
compute new means and deviations
$\bar{x}^*_j = \sum_{i=1}^{Z} x^*_{ij}/Z^*,\ s_j = \sqrt{\sum_{i=1}^{Z}(x^*_{ij} - \bar{x}^*_j)^2/(Z^* - 1)},\ j = 1 ... n$
$\mu_j = \psi\mu_j + (1 - \psi)\bar{x}^*_j,\ j = 1 ... n$
$\sigma_j = \psi\sigma_j + (1 - \psi)s_j,\ j = 1 ... n$
compute deviations change
$\varepsilon_j = |\sigma_j^{old} - \sigma_j|,\ j = 1 ... n$
end

Figure 3.3 Evaluating population

In the next step, the means and the standard deviation of each variable were updated from the values of the elite population by using a smoothing factor, ψ:

$$\mu_i^t = \psi\mu_i^{(t-1)} + (1-\psi)mean(x_i^*), \qquad i = 1\ldots n \tag{3.15}$$

$$\sigma_i^t = \psi\sigma_i^{(t-1)} + (1-\psi)stddev(x_i^*), \qquad i = 1\ldots n \tag{3.16}$$

Three stopping conditions were evaluated. Firstly, the change in the standard deviation was computed for each variable:

$$\varepsilon_j = \left|\sigma_j - \sigma_j^{old}\right| \tag{3.17}$$

A condition that is fulfilled when all the changes are lower than or equal to a prescribed value, ε_{max}:

$$\max_{j=1}^{n}(\varepsilon_j) \le \varepsilon_{max} \tag{3.18}$$

The second condition is fulfilled, if the number of evaluations, *Sc*, which is incremented by *Z* in each iteration of the inner loop, is greater than or equal to a prescribed value, Sc_{max}:

$$Sc \ge Sc_{max} \tag{3.19}$$

Finally, if any of the previously described conditions are fulfilled, the inner loop stops and the elite population is ranked and filtered to store only $Z^*_{\max}$ source solutions. This procedure is depicted in Figure 3.4.

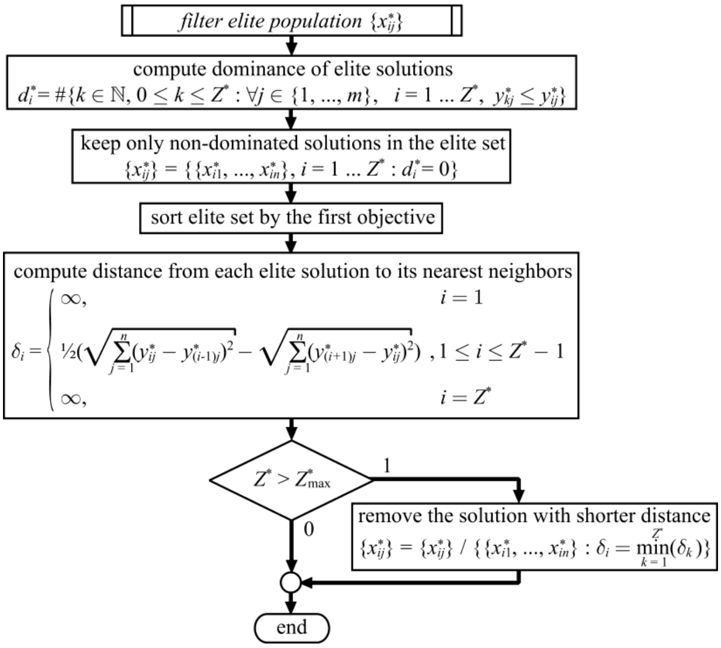

filter elite population $\{x^*_{ij}\}$
compute dominance of elite solutions
$d^*_i = \#\{k \in \mathbb{N}, 0 \le k \le Z^* : \forall j \in \{1, ..., m\}, \ i = 1 ... Z^*, \ y^*_{kj} \le y^*_{ij}\}$
keep only non-dominated solutions in the elite set
$\{x^*_{ij}\} = \{\{x^*_{i1}, ..., x^*_{in}\}, i = 1 ... Z^* : d^*_i = 0\}$
sort elite set by the first objective
compute distance from each elite solution to its nearest neighbors
$\delta_i = \begin{cases} \infty, & i = 1 \\ \frac{1}{2}\left(\sqrt{\sum_{j=1}^{n}(y^*_{ij} - y^*_{(i-1)j})^2} - \sqrt{\sum_{j=1}^{n}(y^*_{(i+1)j} - y^*_{ij})^2}\right), & 1 \le i \le Z^* - 1 \\ \infty, & i = Z^* \end{cases}$
$Z^* > Z^*_{\max}$
1
0
remove the solution with shorter distance
$\{x^*_{ij}\} = \{x^*_{ij}\} / \{\{x^*_{i1}, ..., x^*_{in}\} : \delta_i = \min_{k=1}^{Z^*}(\delta_k)\}$
end

Figure 3.4 MOCE+: Filtering elite population

At this point, another modification was introduced, by decreasing the threshold value, λ_t, by a factor, κ:

$$\lambda_{\mathrm{t}} = (1-\kappa)\lambda_{\mathrm{t-1}} \tag{3.20}$$

This change meant that we used broader threshold levels for the first iterations (where preservation of diversity is important) and finer threshold levels for the final iterations (yielding an elite population closer to the actual Pareto front). All the functions and flowcharts of the MOCE+ algorithm described below are available from the following repository [278].

3.1.2 Simple Multi-Objective Cross Entropy method (SMOCE)

A new set of modifications was introduced in the Simple Multi-Objective Cross Entropy method (SMOCE) (see Figure 3.5) algorithm. It is based on the previous results presented in the MOCE+ algorithm [274] but it includes several important improvements. In the first place, only one loop remains in the algorithm, which is executed *N* times.

The other previously considered stopping conditions, such as evaluation number or convergence criterion, are not taken into account in this new version. This modification allows, on one hand, reducing the number of objective function evaluations and, on the other hand, removing algorithm parameters such as the maximum evaluation number and the convergence limit.

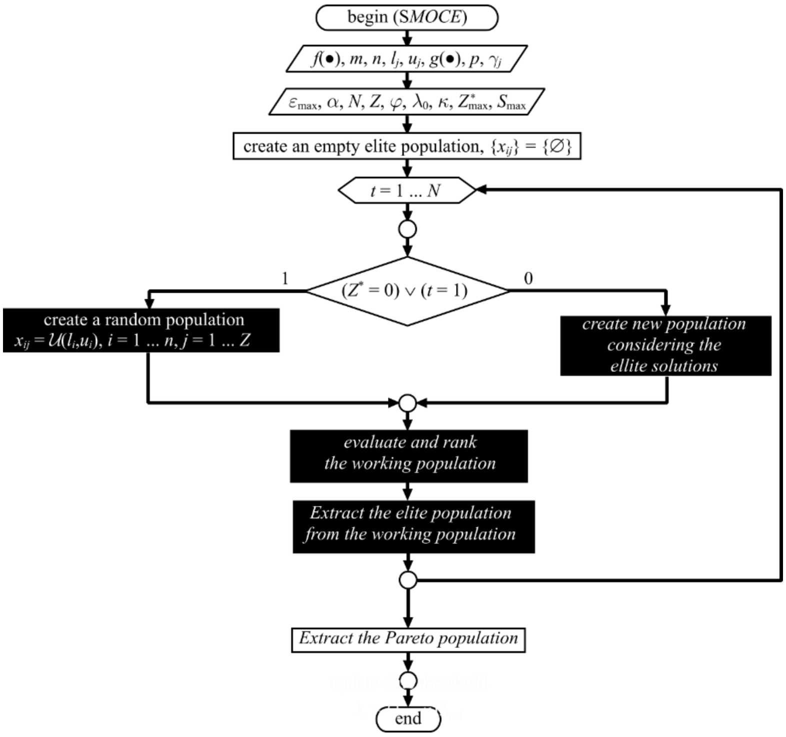

begin (SMOCE)
$f(\bullet)$, m, n, l_j, u_j, $g(\bullet)$, p, γ_j
$\varepsilon_{\max}$, α, N, Z, φ, λ_0, κ, $Z^*_{\max}$, $S_{\max}$
create an empty elite population, $\{x_{ij}\} = \{\varnothing\}$
$t = 1 \dots N$
1
$(Z^* = 0) \vee (t = 1)$
0
create a random population $x_{ij} = \mathcal{U}(l_i,u_i)$, $i = 1 \dots n$, $j = 1 \dots Z$
create new population considering the ellite solutions
evaluate and rank the working population
Extract the elite population from the working population
Extract the Pareto population
end

Figure 3.5 Block diagram representing the simple multi-objective cross entropy algorithm (SMOCE)

As in any multi-objective optimization technique, SMOCE aims to solve the following problem:

$$\min \mathbf{y} = f(\mathbf{x}): \quad \mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^m \tag{3.21}$$

where:

$$l_i \le x_i \le u_i, \quad i = 1 \ldots n \tag{3.22}$$

which is constrained by:

$$g(\mathbf{x}) \le 0, \quad i = 1 \ldots p \tag{3.23}$$

The core of the SMOCE is the working population, at the epoch *t*:

$$\{(x_{i,k}, y_{j,k})^{(t)}, \quad i = 1 \ldots n, j = 1 \ldots m, k = 1 \ldots Z\} \tag{3.24}$$

composed of the *Z* solutions $\mathbf{x}_k = [x_{1,k}, \ldots, x_{n,k}]$ and their respective evaluated objective functions, $\mathbf{y}_k = [y_{1,k} = f_1(\mathbf{x}_k), \ldots, y_{m,k} = \mathrm{f}_m(\mathbf{x}_k)]$.

The evolutionary process takes place on a loop with a unique ending condition: arrival at the epoch number, *N*. These are two of the main differences between SMOCE and MOCE+, where two nested loops and three different ending conditions are considered. These modifications permit, on one hand, a reduction of objective function evaluations and, on the other hand, the removal of algorithm parameters such as the maximum evaluation number and the convergence limit.

In the first epoch, an initial working population is randomly created; in the following epochs, a new population $[(x_{i,k}, y_{j,k})^{t}]$ is created from the previous one $[(x_{i,k}, y_{j,k})^{t-1}]$. The corresponding values of the objective function are evaluated for each solution after creating the population.

Considering the elite solutions:

$$[(\xi_{i,k}, \upsilon_{j,k})^{(t)}, \quad i=1\ldots n, j=1\ldots m, k=1\ldots E] \tag{3.25}$$

where, the elite solution number, E = ψZ, and a parameter, ψ, called the elite fraction, are extracted from the current working population. These elite solutions are included in the next epoch population, which introduces elitism in SMOCE (another improvement with respect to MOCE+).

In the following subsections, the main steps of the SMOCE algorithms are explained in further detail.

3.1.2.1 Creating initial population

Another difference between SMOCE and MOCE+ is the creation of the initial population. In the MOCE+ algorithm, the initial population is created by using a normal random distribution, while a uniform random distribution is used in SMOCE, i.e.:

$$x_{i,j} = \mathcal{U}(l_i, u_i), \quad i=1\ldots n, j=1\ldots Z \tag{3.26}$$

where, $x_{i,j}$ is the value of the *i*-th decision variable in the *j*-th solution; and $\mathcal{U}$(a, b) is the uniform random distribution in the interval [*a*, *b*].

This change means we can generate uniformly distributed individuals throughout the decision variables domain, which is especially convenient for dealing with problems where the Pareto solutions are concentrated in a small section of the whole domain.

3.1.2.2 Creating a new population

The creation of the new working population, from the current one at each epoch, is another important difference between the two methods. In the new approach, the elite solutions, $[(\xi, \upsilon)_1, \ldots, (\xi, \upsilon)_{Epop}]$, are clustered by using the histogram of the objective functions, instead of the histogram of the decision variables.

This approach appears to be the most appropriate as, in most of the problems, there is no correspondence between the variable domain and the objective domain; consequently, individuals with similar decision variables values, could have very different objective function values.

Δ intervals are created in each dimension of the objective space, to establish the histogram:

$$\left[\underline{c}_{i,k}, \overline{c}_{i,k}\right], \quad i = 1\ldots m, k = 1\ldots\Delta \tag{3.27}$$

Where the lower and upper bounds for each interval are:

$$\underline{c}_{i,k} = b_i^{\min} + \frac{(k-1)}{\Delta}(b_i^{\max} - b_i^{\min}) \tag{3.28}$$

and;

$$\overline{c}_{i,k} = b_i^{\min} + \frac{k}{\Delta}(b_i^{\max} - b_i^{\min}) \tag{3.29}$$

where:

$$b_i^{\min} = \min(\{\upsilon_{i,1}, \ldots \upsilon_{i,E}\}), \quad i = 1\ldots m \tag{3.30}$$

and;

$$b_i^{\max} = \max(\{\upsilon_{i,1}, \ldots \upsilon_{i,E}\}), \quad i = 1\ldots m \tag{3.31}$$

are the minimum and maximum values of the *i*-th objective in the elite population.

The Δ intervals, obtained by this way, are then combined with the *m* variables, to obtain Δ^{am} classes, and then all the ψZ elitist solutions are arranged into these classes. Thereafter, the mean value and the standard deviations are computed from the solutions of each class for each objective function:

$$\mu_{i,k} = \frac{1}{\#[\xi^k]}\sum_{j=1}^{\#[\xi^k]} \xi_{i,j}^k, \quad i = 1\ldots m, \quad k = 1\ldots\Delta^m \tag{3.32}$$

$$\sigma_{i,k} = \sqrt{\frac{\sum_{j=1}^{\#[\upsilon^k]}(\xi_{i,j}^k - \mu_{i,k})^2}{\#[\xi^k]-1}}, \quad i = 1\ldots m, \quad k = 1\ldots\Delta^m \tag{3.33}$$

where, $[(\xi, \upsilon)_k]$ is the subset of the elite population belonging to the k-th class, i.e.:

$$[(\xi,\upsilon)_k] \subset [(\xi,\upsilon)] : \underline{c}_{i,k} \le \upsilon_i \le \overline{c}_{i,k}, \forall i = 1\ldots m \tag{3.34}$$

Lastly, the new working population is composed of the E_{pop} elite population and $\#[\upsilon_k](Z - E_{pop})/E_{pop}$, $k = 1\ldots\Delta^m$, which are sets of new solutions for each class, created by using a normal random distribution with mean $\mu_{i,k}$, and standard deviation $\sigma_{i,k}$, and truncated to the interval $[l_i, u_i]$, for $i = 1\ldots m$:

$$x_{i,j}^{(t)} = \mathcal{N}(\mu_{i,k}, \sigma_{i,k}, l_i, u_i), \quad i = 1 \ldots n \tag{3.35}$$

3.1.2.3 Evaluating the population

The evaluation of the solutions implies not only computing the values of the objective functions, but also the constraints, which are considered by the penalty method. Therefore, the constrained objective functions take the following form:

$$y_{i,k}^{(t)} = f_i(\mathbf{x}_k^{(t)}) + \sum\nolimits_{j=1}^{p} \gamma_j g_j(\mathbf{x}_k^{(t)}), \quad i = 1 \ldots n, k = 1 \ldots Z \tag{3.36}$$

where, $\gamma_j \geq 0$ are the penalty coefficients assigned to each constraint.

3.1.2.4 Extracting the elitist population

Elitism is considered in SMOCE through including the elite solutions in the working population of the next epoch.

Selection of the elitist population is also different in the new approach. In MOCE+, elite population includes all the individuals with rank lower than some threshold value (which decreases from some initial value to zero, through the algorithm execution). On the contrary, in SMOCE some prescribed fraction of the working population, including the individuals with the lower rank, is selected as the elite population in each epoch.

This selection is carried out through the Pareto ranking criterion, which is based on the concepts of vector dominance.

A vector, **v** $\in \mathbb{R}^m$, dominates another vector, **u** $\in$ ℝm, (denoted as $\mathbf{v} \preceq \mathbf{u}$) , if and only if all the components of v are less or equal than the respective components of **u**, and exists at least one component of **v** which is strictly less than the respective component of **u**:

$$\mathbf{u} \preceq \mathbf{v} \Leftrightarrow (\forall i \in \{1 \ldots m\}, v_i \leq u_i) \wedge (\exists j \in \{1 \ldots m\}, v_j < u_j) \tag{3.37}$$

In a vector set $\Omega = [\mathbf{v}_1 \ldots \mathbf{v}_n] \subset \mathbb{R}^m$, a vector $\mathbf{v}^* \in \Omega$ is said to be non-dominated if and only if:

$$\neg\exists k \in \{1 \ldots n\} : \mathbf{v}_k \preceq \mathbf{v}^* \tag{3.38}$$

By using these concepts, the Pareto ranking of a vector, in a vector set, is the number of other vectors which dominate it. The elite population is composed by the E_{front} solutions with the lower Pareto rank.

3.2 Analysis of the sensibility to the algorithm parameters values

3.2.1 Screening

The first step for selecting the most convenient values of the SMOCE algorithm parameters is to determine their influence on the Pareto front quality and on the execution time (i.e., on the efficacy and efficiency of the algorithm). In order to do that, a screening study was carried out. The four parameters of the SMOCE (epoch number, *N*; working population size, *Z*; histogram intervals number, Γ; and elite fraction, ψ) were considered in the intervals shown in Table 3.1. A 2-level half-fraction with center point design of experiments was selected for the simulations. Five replicates were executed for each experimental point.

Thirteen commonly used test problems, coming from three suites conventionally known as MOP [279], ZDT [103] and WFG [102], were considered in the simulations. All of them have two objectives, but are very different in the other features (see Table 3.2).

Four metrics were used for evaluating the quality of the obtained Pareto fronts: the hyperarea ratio, *HR*; the generational distance, *GD*; the convergence, *CV*; and the spacing, *SP*. While the generational distance and convergence measure the front convergence (i.e., how closed is the obtained front to the true Pareto front), the spacing reflects its diversity (i.e., how uniformly are the solutions distributed through the obtained Pareto front), and the hyperarea ratio combines both criteria.

Table 3.1 Levels of the Algorithm Parameters for the Screening

Algorithm parameter	*Level values*		
	Low	*Medium*	*High*
Epoch number, *N*	10	2505	5000
Working population size, *Z*	50	525	1000
Histogram intervals number, *D*	5	15	25
Elite fraction, α	0.10	0.35	0.60

The influence of each parameter is evaluated through a multiple regression, by considering the t-Student test of the corresponding coefficients. The relationship is considered to be significant at a 95% confidence level. Figure 3.6 represents the obtained relationships.

As can be seen, the epoch number has a direct relationship with the execution time in all the considered test problems. This fact can be expected from the same structure of the SMOCE algorithm, as this parameter is the only included ending condition. An increase of the epoch number, also improves the front diversity (expressed by an inverse relationship with the spacing) in most of the test problems (except in MOP1 and MOP2) but it does not have a significant relationship with the other metrics.

Table 3.2 Levels of the Algorithm Parameters for the Screening

Problem	*Reference*	*Variables*	*Geometry*	*Modality*
MOP1	[279]	1	Convex	Unimodal/Unimodal
MOP2		3	Concave	Unimodal/Unimodal
MOP3		2	Disconnected	Multimodal/Unimodal
MOP4		3	Disconnected	Multimodal/Unimodal
MOP6		2	Disconnected	Unimodal/Multimodal
ZDT1	[103]	30	Convex	Multimodal/Unimodal
ZDT2		30	Concave	Multimodal/Unimodal
ZDT3		30	Disconnected	Unimodal/Multimodal
WFG2	[102]	32	Disconnected	Unimodal/Multimodal
WFG3		32	Degenerated	Unimodal/Unimodal
WFG4		32	Concave	Multimodal
WFG5		32	Concave	Deceptive
WFG6		32	Concave	Unimodal
WFG7		32	Concave	Unimodal
WFG8		32	Concave	Unimodal
WFG9		32	Concave	Multimodal, deceptive

A rise in the population size also increases the execution time for all of the test problems. It also improves the quality (both, the convergence and diversity) of the obtained Pareto front for most of the test problems.

The execution time is not affected by the histogram interval number. A rise in this parameter causes an improvement in the front convergence (showed in the direct relationship with the convergence metric and the inverse relationship with the generational distance for most of the test problems), and in the front diversity (indicated by the inverse relationship with the spacing). There is also a direct relationship with the hyperarea for most of the problems.

Finally, the elite ratio has no influence neither in the execution time nor in the quality metrics, except in the spacing for problems in sets ZDT and WFG, where and improvement takes place.

By taking into account the previous analyses, the values of the histogram interval number and the elite fraction are chosen in their respective higher levels (i.e., Γ = 25 and ψ = 0.65), because they correspond to the better Pareto front quality without worsening the execution time.

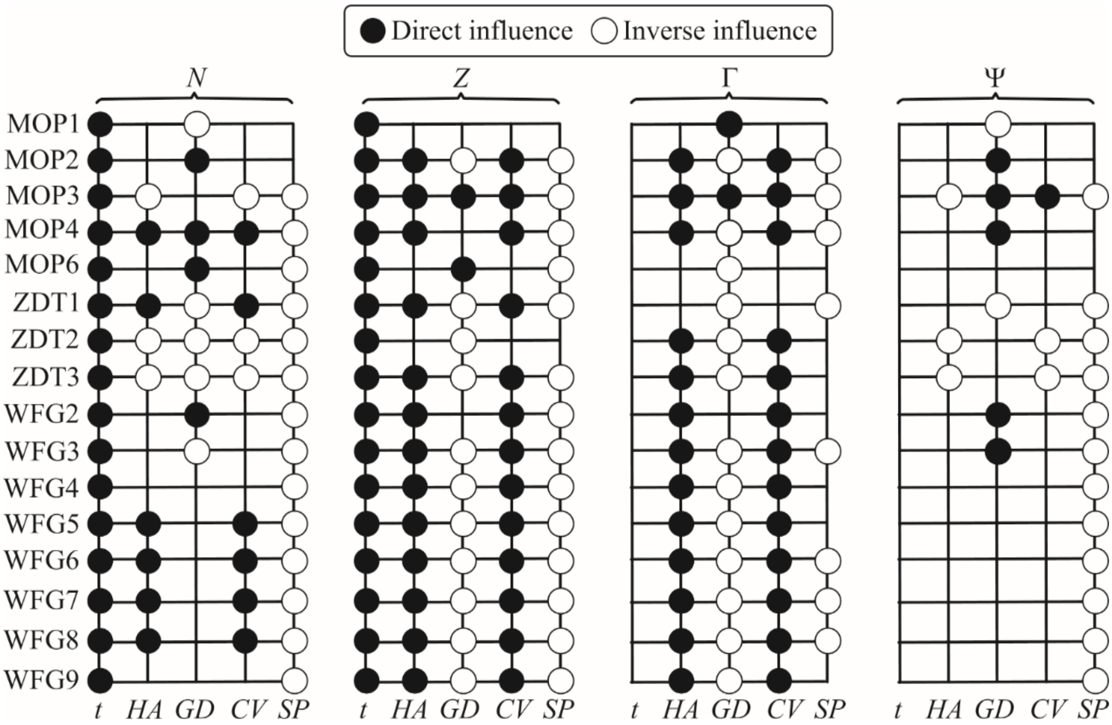


Figure 3.6 Relationships resulting from the screening analysis

3.2.2 Response surface

For obtaining the relationship between the epoch number and the population size with the execution time and the front quality, the following response surface analysis was carried out. For doing that, a full 3-levels experimental design was selected. Both factors (population size, Z, and epoch number, N) were kept in their respective levels used for screening (see Table 3.2. Twenty replications were carried out for each experimental point. For analyzing the front quality, the hyperarea ratio was selected because this parameter characterizes both the convergence and diversity. Figure 3.7 shows the graphical representation of the execution time and hyperarea ratio obtained for each experimental level for the MOP suite.

As it is shown in Figure 3.7, both experimental factors have a direct influence on the execution time. On the contrary, they increase the front quality only up to medium levels. After this point, there is not a significant change in the front quality. Therefore, the most convenient values for the experimental factors are those corresponding (or near) to the middle levels, i.e., Z = 525, N = 2505. It can be pointed out that the hyperarea ratio values obtained for this suit are very high. Except for MOP6 problem, their higher values are close to one. This means that the obtained Pareto front is very close to the theoretical solution.

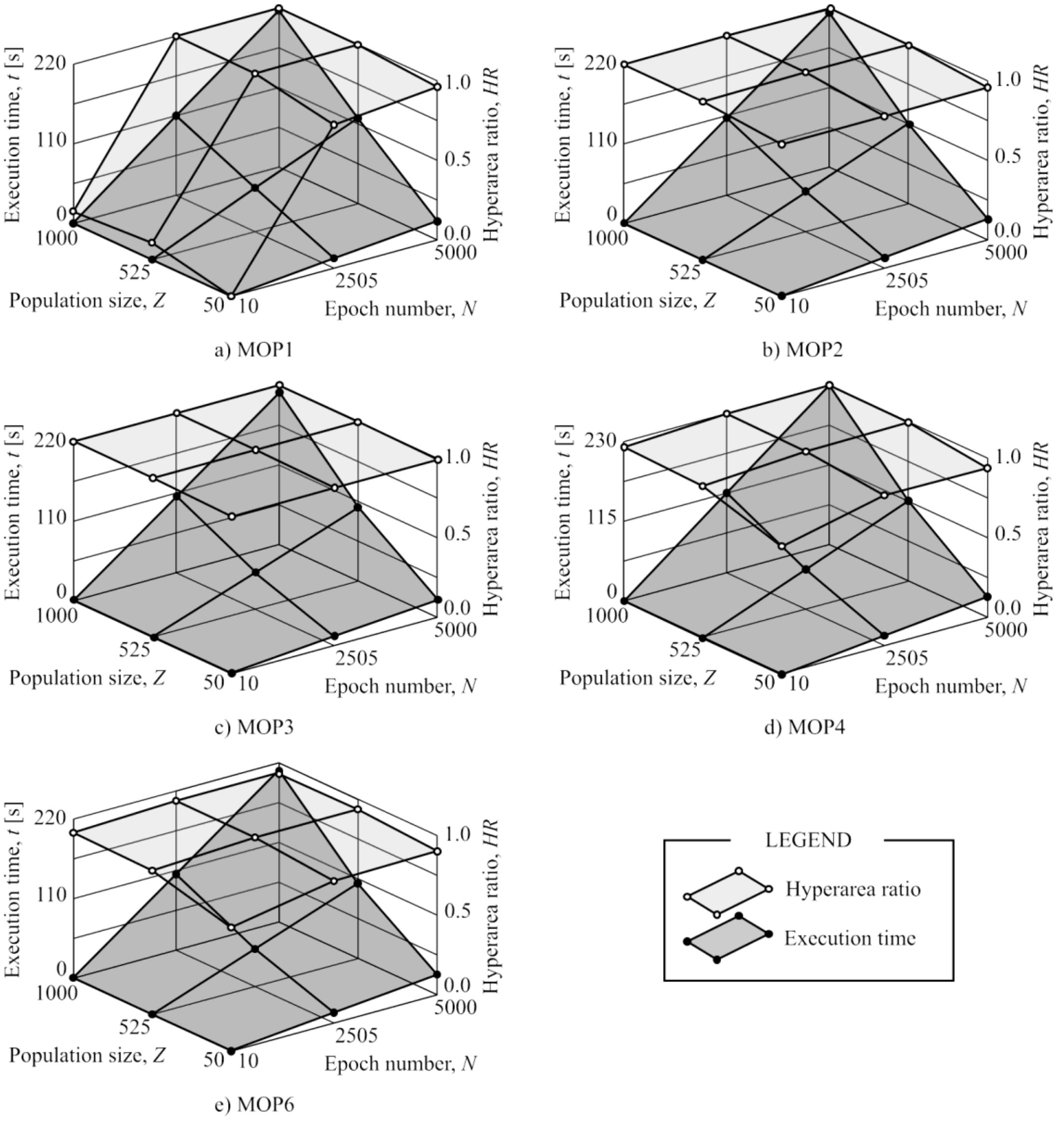
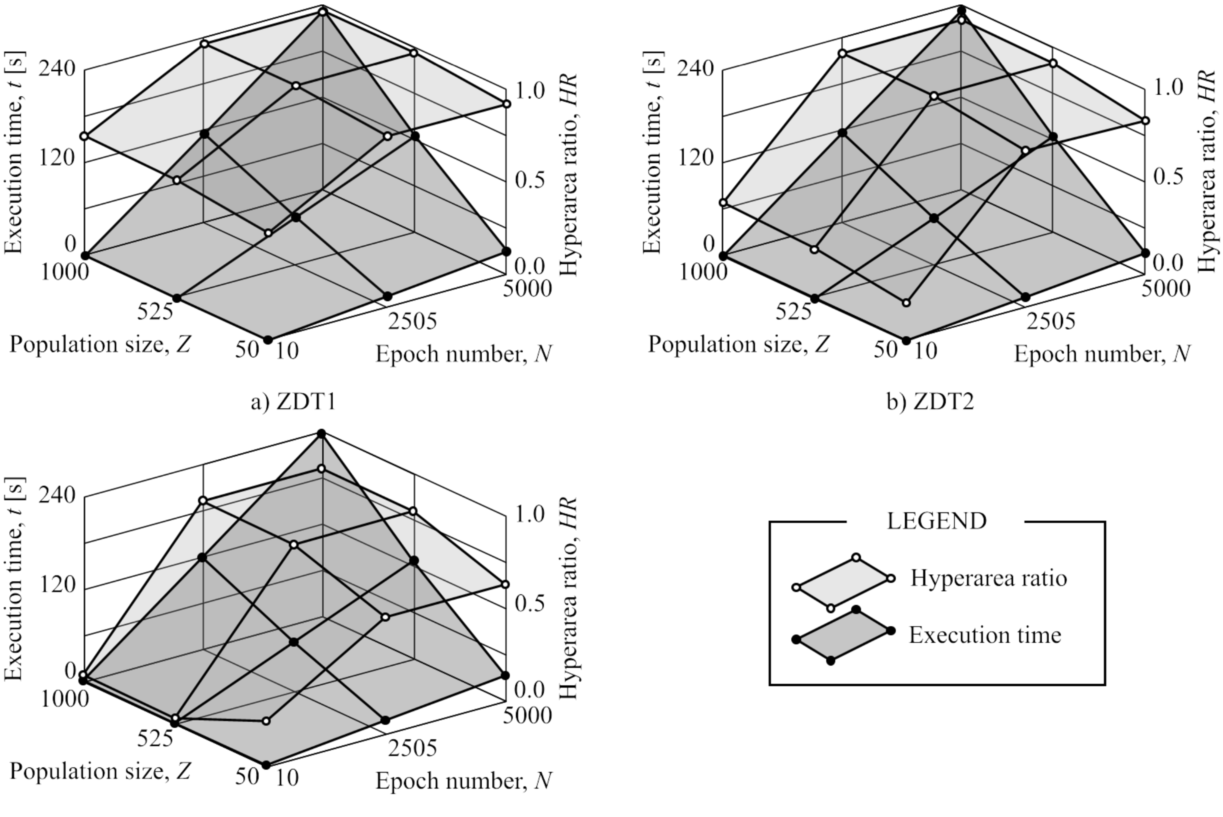


Figure 3.7 Behavior of the execution time and hyperarea ratio vs. population size and epoch number for test problem suite MOP

Figure 3.8 and Figure 3.9 show the obtained results for the suite ZDT and WFG, respectively. In both, the behavior is similar not only for the execution time but also for the hyperarea ratio. In all the cases, the combination of medium levels (Z = 525, N = 2505) allows to obtain near-optimal front quality with reasonable execution time.

The most noticeable difference is given by the lower values of hyperarea ratio in problem of the two last suites compared with MOP. This can be due to the higher complexity of this test suit.

However, in spite of this fact, the differences in the execution time are neglected. This indicates that the execution time depends on the number of evaluation but not on the complexity of the optimization problem (number of variables, characteristics of the target function, etc.).



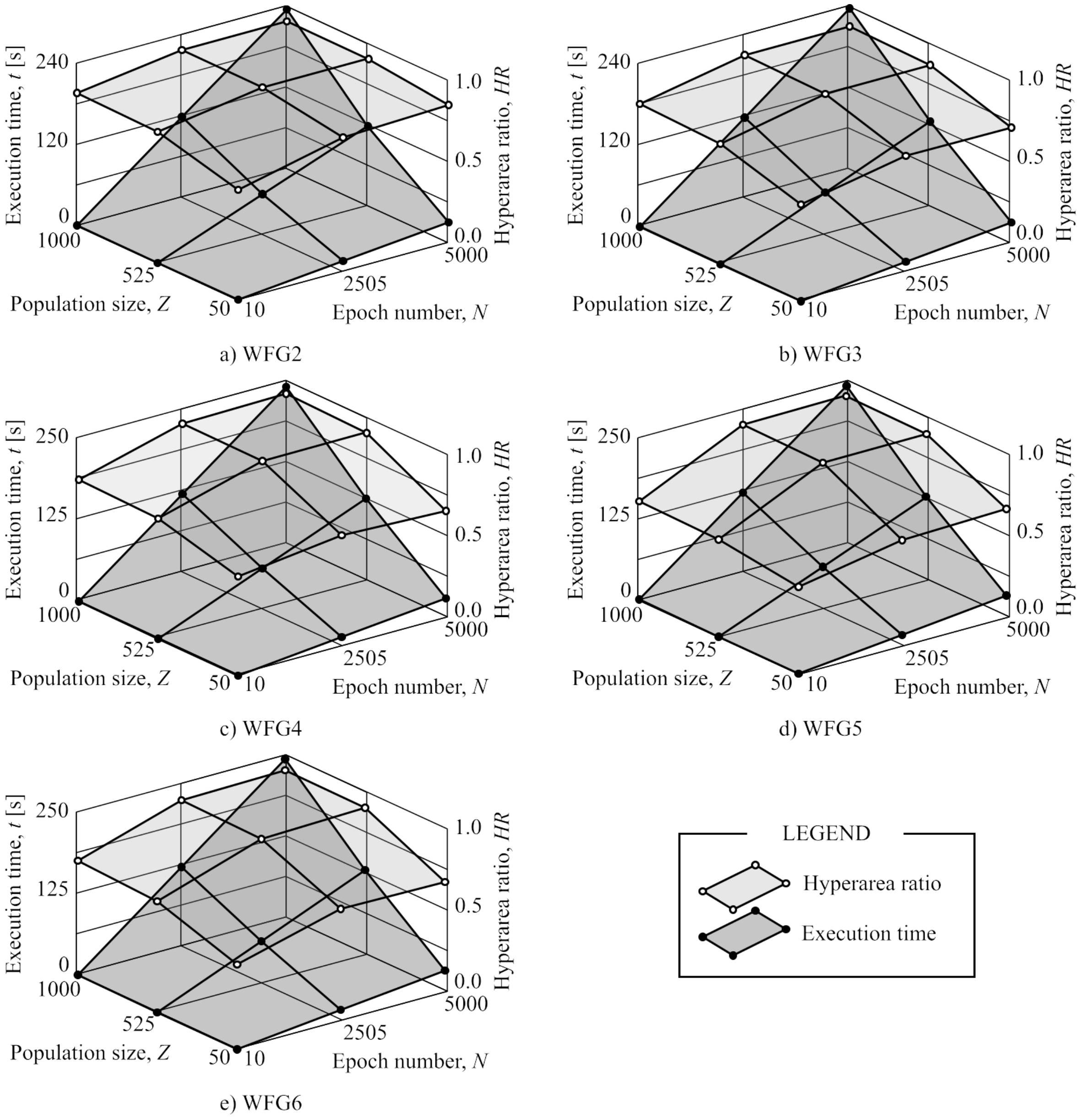
a) ZDT1

b) ZDT2

c) ZDT3

Figure 3.8 Behavior of the execution time and hyperarea ratio vs. population size and epoch number for test problem suite ZDT

The quality of the obtained Pareto front does not increase beyond some points by incrementing the parameters (population size and epoch number), at least, on the basis of the considered intervals.



a) WFG2

b) WFG3

c) WFG4

d) WFG5

e) WFG6

Figure 3.9 Behavior of the execution time and hyperarea ratio vs. population size and epoch number for test problem suite WFG

3.3 Comparative study. Advantages and drawbacks.

A comparative study was performed on the basis of other approaches reported in the literature, to analyze the main drawbacks and advantages of the proposed algorithm. So, only some of the most well-established algorithms of similar scope are included in this study. Firstly, a comparison between the two algorithms described in the previous sections was performed (see Table 3.4). For this, series of widely used multi-objective optimization test problems (see Table 3.2) were evaluated.

MOP and ZDT problems were addressed by using the parameters shown in Table 3.3. These values were obtained by trial and error and by considering those reported in various studies. The tradeoff between the Pareto front quality was also taken into account for setting these parameters.

Table 3.3 Parameters values for solving problems MOP's and ZDT's

Parameter	*Value*
Convergence limit, $\varepsilon_{\max}$	0.01
Smoothing factor, ψ	0.9
Epochs number, N	100
Population size, Z	50
Frequency inversion likelihood, ϕ	0.3
Initial elite threshold, λ_0	2
Decreasing threshold factor, κ	0.05
Maximum elite population size, $Z^{*}_{\max}$	160
Maximum evaluation number, $Sc_{\max}$	500

The quality of the Pareto front was evaluated by using four metrics [114]. While generational distance, GD, and convergence, CV, measure front convergence (i.e., how close the obtained front is to the true Pareto front), and spacing, SP, measures diversity (i.e., how uniformly distributed the solutions are); the maximum Pareto front error, ME, combines both criteria.

A total of 50 replications were performed for each problem. The mean values and confidence intervals (at a 95% confidence level) of each metric were shown in Table 3.4. The generational distances, GD, were worse than those generated by MOCE for MOPs, but the values were still low. There was no significant difference for ZDT2, but for ZDT1 and ZDT3 the values offered by SMOCE were remarkably better. Convergence, CV, was lower for all the problems, except for MOP1 where no significant difference was found. Spacing, SP, was also low, except in MOP4. For MOP1, ZDT1 and ZDT2 the values from SMOCE was slightly better than those from MOCE, while they were slightly worse for both MOP2 and ZDT3. Finally, the maximum Pareto front error, ME, was significantly lower for MOP2, MOP4 and ZDT3; it was notably higher in MOP1; while there were no significant differences in ZDT1 and ZDT2.

Table 3.4 Comparison between MOCE and SMOCE for solving MOP and ZDT problems

Problem	*Reference*	*Metrics* [×10^{-3}]			
		GD	*CV*	*SP*	*ME*
MOP1	MOCE	0.0	6.8	20.23	3.8
	SMOCE	0.6 ± 0.0	6.8 ± 0.0	8.7 ± 0.4	10.4 ± 2.8
MOP2	MOCE	0.0	11.2	0.6	12.8
	SMOCE	1.0 ± 0.0	10.2 ± 0.0	2.2 ± 0.4	1.4 ± 0.2
MOP4	MOCE	0.2	95.7	15.7	290.3
	SMOCE	0.8 ± 0.1	5.5 ± 0.4	66.9 ± 3.6	6.2 ± 0.7
ZDT1	MOCE	1.2	71.6	3.9	23.5
	SMOCE	0.8 ± 0.0	8.0 ± 0.2	2.4 ± 0.0	23.1 ± 0.9
ZDT2	MOCE	1.2	254.5	2.7	23.5
	SMOCE	1.2 ± 0.1	12.6 ± 1.3	2.5 ± 0.1	25.2 ± 3.3
ZDT3	MOCE	3.9	28.9	2.4	130.0
	SMOCE	0.6 ± 0.0	5.8 ± 0.1	4.5 ± 0.2	17.4 ± 0.8

A total of 50 replications were performed for each problem. The mean values and confidence intervals (at a 95% confidence level) of each metric were shown in Table 3.4. The generational distances, GD, were worse than those generated by MOCE for MOPs, but the values were still low. There was no significant difference for ZDT2, but for ZDT1 and ZDT3 the values offered by SMOCE were remarkably better. Convergence, CV, was lower for all the problems, except for MOP1 where no significant difference was found. Spacing, SP, was also low, except in MOP4. For MOP1, ZDT1 and ZDT2 the values from SMOCE was slightly better than those from MOCE, while they were slightly worse for both MOP2 and ZDT3. Finally, the maximum Pareto front error, ME, was significantly lower for MOP2, MOP4 and ZDT3; it was notably higher in MOP1; while there were no significant differences in ZDT1 and ZDT2.

Summarizing, it can be concluded that the proposed algorithm, SMOCE, performed in a similar way to MOCE for the MOP and the ZDT test suites. It should be noted that its behavior was better for ZDT problems than for MOP problems.

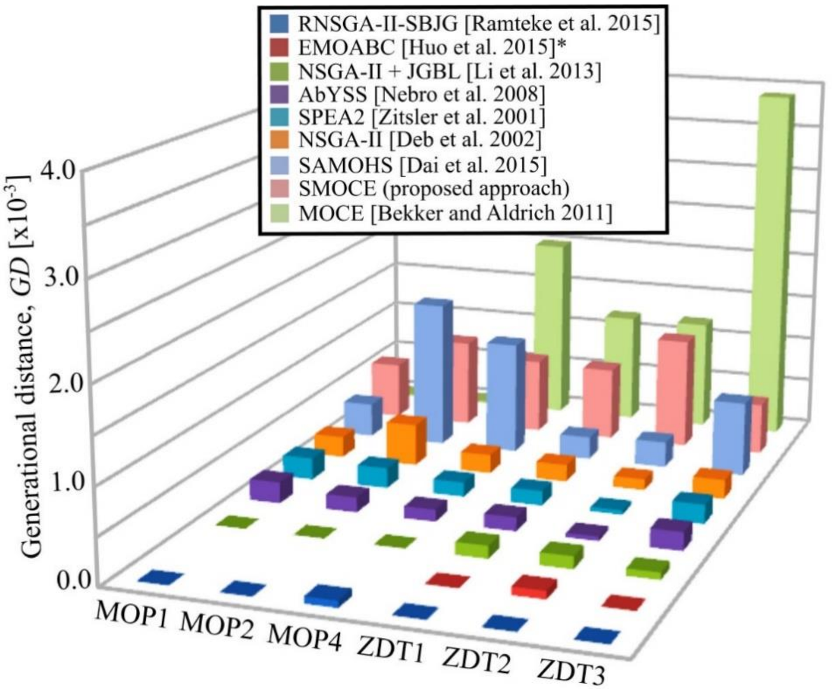


Figure 3.10 Comparison of different approaches for MOP’s and ZDT’s problems

***Only for the ZDT’s test suite functions**

A comparison with other multi-objective algorithms, including the archive-based hybrid scatter search (AbYSS) [280], the strength Pareto evolutionary algorithm (SPEA2) [281], the non-dominated sorting genetic algorithm II (NSGAII) [282], the non-dominated sorting genetic algorithm II + learning paradigm based on jumping genes (NSGAII+JGBL) [283], the elite-guided multi-objective artificial bee colony (EMOABC) [284], a self-adaptive multi-objective harmony search (SAMOHS) [285], a real-coded NSGA-II with simulated binary jumping gene operators (RNSGA-II-SBJG) [286] and the previously mentioned multi-objective cross entropy (MOCE) method [114], are all shown in Figure 3.10. It may be noted that the generational distance of the solutions provided by SMOCE are better than those provided by MOCE (except for MOP1 and MOP2), but they are worse than those provided by the other evolutionary approaches. Nevertheless, the generational distance values are very low in all the problems; therefore, the solutions are acceptable for many applications.

Quite different results were obtained when attempting to solve more complex problems, especially those from the WFG test suite (see Table 3.2). These problems were addressed by using the parameters shown in Table 3.5. The increases in the epoch numbers, *N*, and the population size, Z, were notable, while the convergence limit, ε, and the maximum evaluation number, $\eta_{\max}$, all decreased.

Table 3.5 Optimization problems solved by MOCE and SMOCE

Parameter	*Value*
Convergence limit, $\varepsilon_{\max}$	0.005
Smoothing factor, ψ	0.1
Epochs number, N	2500
Population size, Z	160
Frequency inversion likelihood, ϕ	0.1
Initial elite threshold, λ_0	4
Decreasing threshold factor, κ	0.5
Maximum elite population size, $Z^*_{\max}$	160
Maximum evaluation number, $Sc_{\max}$	130

Quality metrics for the WFG problems are shown in Table 3.6.

A comparison of the results, for the WFG test suite, with multi-objective cross-entropy (MACE) [7], multi-objective cross-entropy optimization using generalized decomposition (MACE-gD) [7], multi-objective evolutionary algorithms based on decomposition (MOEA/D) [287], real-coded NSGA-II with simulated binary jumping gene operators [286] and regularity model-based estimation of distribution algorithms (RM-MEDA) [288], were shown in Figure 3.11. This figure shows a much better performance of SMOCE in relation to the WFG test suite and in comparison with the other approaches.

Even though the performance of SMOCE for simple problems was no better than other previously published approaches, it is evident that SMOCE performed notably better for complex problems. The main rationale to explain this behavior is the complexity of SMOCE, which is a powerful tool for solving complex problems, but which requires more computational resources to be effective at solving simple problems. The main disadvantage of SMOCE is linked to problem-dependence parametrization. The optimal setting of most gradient-free multi-objective algorithms is a challenging and unsolved problem. However, the fast convergence of SMOCE and the parameter settings serve as guidelines to help overcome this limitation.

Table 3.6 Comparison between MOCE and SMOCE for solving MOP and ZDT problems

Problem	*Metrics* [×10^{-3}]			
	GD	*CV*	*SP*	*ME*
WFG2	2.5 ± 0.3	6.8 ± 0.7	117.9 ± 10.2	28.7 ± 4.4
WFG3	1.5 ± 0.2	6.8 ± 0.3	119.2 ± 13.6	20.3 ± 1.8
WFG4	1.5 ± 0.4	10.3 ± 1.7	125.2 ± 7.9	18.7 ± 6.0
WFG5	3.3 ± 0.3	8.7 ± 1.2	96.3 ± 10.0	45.9 ± 4.1
WFG6	6.2 ± 0.8	39.2 ± 8.7	721.0 ± 32.0	52.7 ± 4.9
WFG7	2.4 ± 0.2	7.4 ± 0.7	59.2 ± 11.7	33.7 ± 3.6
WFG8	8.3 ± 1.1	36.7 ± 39.7	333.0 ± 68.3	96.9 ± 10.9
WFG9	1.8 ± 0.4	8.7 ± 1.7	80.9 ± 22.0	22.6 ± 3.8

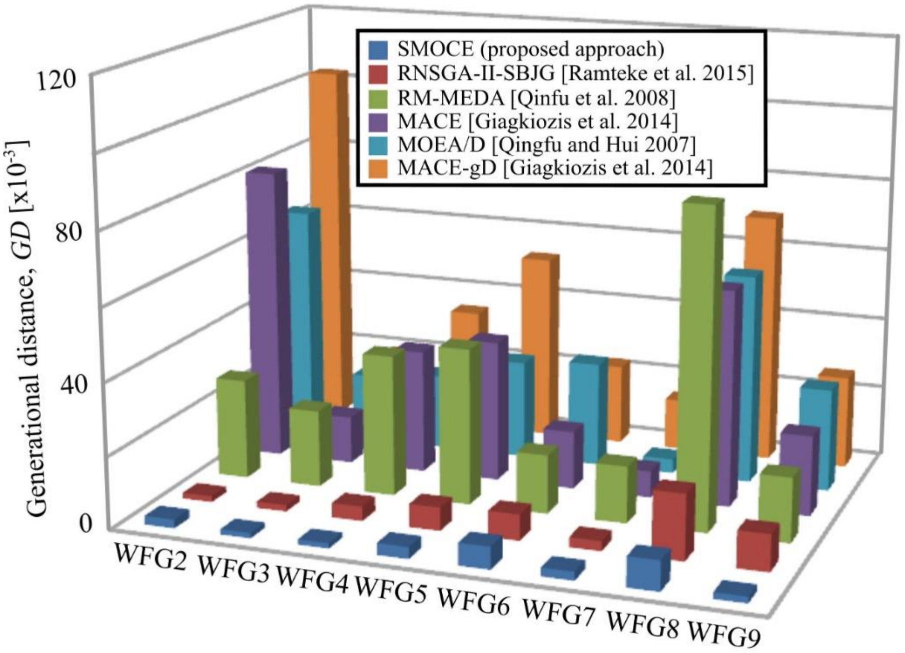


Figure 3.11 Comparison of different approaches for WFG's problems

3.4 SMOCE Application to micro-manufacturing processes

3.4.1 Current techniques and procedures for the optimization of machining processes

Selection of the optimal cutting conditions is a very important task in the design of machining processes [289]. The optimization of cutting conditions involves two different challenges: on the one hand, obtaining accurate robust models linking the objectives and constraints to the decision variables; on the other hand, developing effective and efficient strategies that can negotiate the optimization problems that have been formulated [290]. Both tasks are of high physical and mathematical complexity.

Firstly, chip formation occurs over geometrically complex domains that change quickly over time [291]. Moreover, metal-cutting processes involve complex mechanical and thermal phenomena (such as, thermo-visco-plasticity, friction and fracture). Most of these issues have yet to be fully understood and there are, therefore, no reliable phenomenological models for these processes. Consequently, in spite of some partially successful approaches, such as Oxley's predictive theory [292], there is at present no reliable analytical model of machining processes that can be used for industrial applications.

Another commonly used approach is the finite-element method, which provides approximate solutions for problems where analytical solutions cannot be obtained [293]. Nevertheless, the accuracy of the finite-element-based model is not good enough for use in real machining processes. Moreover, the solutions of these models are computationally expensive, a key issue in optimization, where multiple evaluations of the model are required. Consequently, empirical models, either based on statistical regressions or based on fuzzy modeling [185] and artificial intelligence tools [294] are currently the most reliable choice in cutting-process modeling [295].

The optimization of the cutting processes involves the use of the above-mentioned models, represented by functions that do not fulfill the conditions of continuity, differentiability and unimodality, usually required for conventional analytical and numerical techniques. One alternative is the use of optimization heuristics, based on the use of soft-computing tools. These techniques are especially useful in multi-objective optimization, where several different, but often connected objectives are considered.

Simulated annealing [296], the ant-colony algorithm [297] and particle swarm optimization [298] may all be mentioned among the most widely used heuristics for the optimization of machining processes. Nevertheless, genetic algorithms are still the most popular technique for a posteriori multi-objective optimization, due not only to their robustness and efficacy, but also to their ability to obtain the so-called Pareto front in a single processing run [299]. Several approaches have been proposed for this target; the most widely reported are perhaps the multi-objective genetic algorithm (MOGA), the fast non-dominated sorting genetic algorithm and the niched Pareto genetic algorithm (NPGA) [300], which have been widely applied to the optimization of cutting processes [105].

3.4.2 Cross-entropy-based multi-objective optimization: a micro-drilling process as case study

The cross-entropy method requires a precise or an approximate mathematical model of the physical process yielded from input-output data. In the previous sections, the improved cross-entropy method for multi-objective optimization was assessed using well-known benchmarks with mathematical models. Micro-scale processes are non-linear, time-variant processes that are difficult to represent with precise mathematical equations. Therefore, two models of force and acceleration were obtained using a well-established method inspired in Artificial Intelligence.

For the sake of clarity, a feed-forward neural network was considered for modeling purposes. A multilayer perceptron was specifically selected for modeling the micro-drilling process. The neural network was composed of four input layers (each input corresponding to one independent variables), 5 neurons in the hidden layers and a single output neuron, which offer the predicted value of the force. Hidden neurons used a sigmoid transfer function while the output neuron used a linear transfer function. The training process was performed by means of the error backpropagation algorithm, with an adaptive learning rate and momentum. The following training parameters were selected: initial learning rate, 0.001; ratio to increase learning rate, 1.05; ratio to decrease learning rate, 0.70; momentum constant, 0.9; and minimum performance gradient, 10^{-10}. The stop condition was established after 500000 epochs.

The first model was obtained for representing the relationship between the thrust force, F_z, and the cutting parameters: cutting speed (v_c), feed-rate per tooth, (f_{tooth}), peck drilling step (*step*) and hole depth (h_d). The available experimental data was divided into a training set, composed of 768 data samples (corresponding to the 80% of the total dataset), which was used for fitting the models, and a validation set composed of the remaining 192 data samples, which was used for testing their generalization capabilities (see Table 2.18).

The second model was generated to represent the vibration on the plane perpendicular to the drilling axis (xy-plane), $\ddot{r}_p$, source which also depends on the cutting parameters (v_c, f_{tooth}, *step* and h_d). This parameter was selected because it is closely related to the quality of the drilled holes (see Table 2.18).

Therefore, the main goal of the optimization process was to select the most convenient cutting parameters (i.e., speed, v_c, feed rate, f_{tooth}, and *step*) for carrying out the drilling process of a 0.8 mm-depth hole. All the implemented functions for the micro-drilling case study are available at the following repository [278]. All these variables were considered in their respective experimental intervals:

$$\begin{aligned}
&9.4m / \min \leq v_{\mathrm{c}} \leq 27.6m / \min \\
&10mm / \min \leq f_{\mathtt{tooth}} \leq 300mm / \min \\
&0.02mm \leq step \leq 0.06mm
\end{aligned} \tag{3.39}$$

The experiment was performed on a titanium–aluminum–vanadium alloy (Ti_6Al_4V), with 0.2 mm-diameter drills.

A Kern-Evo high-precision machining center (see Figure 2.8) was used in the experimental setup, equipped with a Kistler Minidyn 9256 piezoelectric force dynamometer for capturing the three components of the force signal (F_x, F_y, F_z).

Before the modeling step, both signals were filtered using a finite impulse response filter. With regard to thrust force, the mean value of each peak was computed to characterize the signal at each peak, while the root mean square (RMS) was used in the case of vibrations for the same purpose. Two objectives were simultaneously considered. The first was drilling time, τ, which can be computed with the equation:

$$\tau = \frac{step}{f_{tooth\_0}}\left[\left(\frac{h_{\mathrm{d}}}{step}\right)^{2} + \left(\frac{h_{\mathrm{d}}}{step}\right) - 2\right] + \frac{h_{\mathrm{d}}}{f_{\mathtt{tooth}}}\lim_{x\to\infty} \tag{3.40}$$

where, $f_{\mathrm{tooth\_0}}$ is the fast-feed rate used for the backward motion. The second objective was the amplitude of the vibrations on the plane perpendicular to the drilling axis, which was modeled, in the previous section, by using a neural network:

$$\ddot{r}_{\mathrm{P}} = \phi_{\mathrm{NN}}(v_{\mathrm{c}}, f_{\mathtt{tooth}}, step) \tag{3.41}$$

Indeed, both objectives must be minimized. With this combination, a high-productivity drilling process that guarantees high-hole quality (closely related with the perpendicular vibrations) may be expected.

Furthermore, certain constraints must be fulfilled. The thrust force, Fz, should be lower than the allowable thrust force, F_{z}^{al} , which is pre-established to avoid buckling-based breakage of the tool. This constraint can be expressed by:

$$\frac{F_{\mathrm{Z}}}{F_{\mathrm{Z}}^{al}} - 1 \leq 0 \tag{3.42}$$

while the thrust force is computed with the previously obtained neural network-based model:

$$F_{\mathrm{z}} = \phi_{\mathrm{NN}}(v_{\mathrm{c}}, f_{\mathrm{tooth}}, step) \tag{3.43}$$

and the allowable force can be determined with Euler’s equation:

$$F_{\mathrm{z}}^{al} = \eta_{\mathrm{f}} \frac{\pi^{2} E\, \mathrm{I}_{\mathrm{min}}}{\mu_{\mathrm{b}} L} \tag{3.44}$$

where, E = 650 MPa is the Young’s modulus of the tool material, $I_{\mathrm{min}} = 9.54 \times 10^{-6}$ mm^{-4} is the minimum area moment of inertia of the drill cross-section, μ_{b} = 2 is the Poisson’s coefficient, which takes into account the boundary conditions, L = 2.5 mm is the length of the drill flute, and η_{f} = 0.5 is a security factor.

The optimization process was carried out with the parameters shown in Table 3.7. These parameters were selected based on the previously obtained results for standard test problems. It can be noted that the convergence limit was reduced to zero, keeping only the maximum evaluation number as the stopping criterion.

Table 3.7 Parameters values for solving the drilling optimization problem

Parameter	*Value*
Convergence limit, $\varepsilon_{\max}$	0
Smoothing factor, ψ	0.5
Epochs number, N	100
Population size, Z	5000
Frequency inversion likelihood, ϕ	0.1
Initial elite threshold, λ_0	4
Decreasing threshold factor, κ	0.1
Maximum elite population size, $Z^{*}_{\max}$	500
Maximum evaluation number, $Sc_{\max}$	10^5

Altogether, 25 runs were performed to increase the reliability of the optimization results. Figure 3.12 depicts the Pareto fronts that were obtained. Three zones can be easily noted in the graph. Zone I includes those solutions with lower vibration amplitudes and longer drilling times. Hence, it shows cutting parameters that will obtain high quality holes, but over excessive operating times. Zone I comprises parameters in the intervals: 10.6m/mim≤v_c≤24.8m/min; 280mm/min≤ f_{tooth}≤300mm/min; 0.025mm≤*step*≤0.060mm.

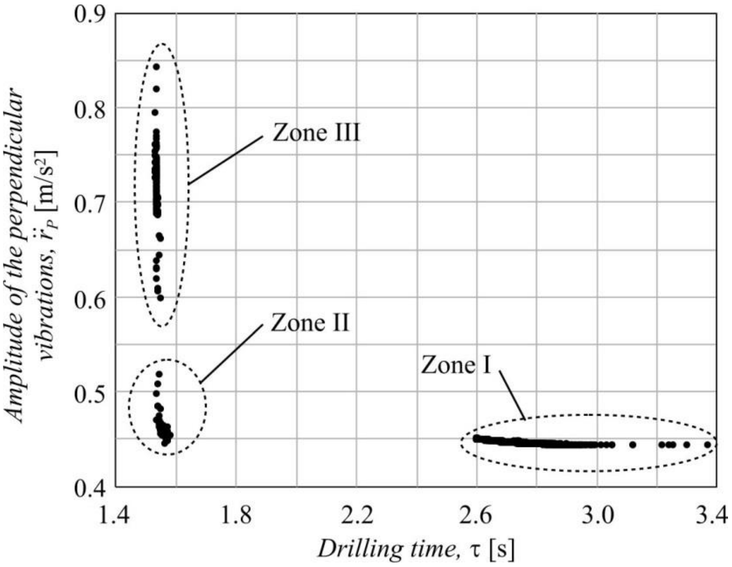


Figure 3.12 Pareto front of the drilling process

Table 3.8 Outcomes of the drilling process optimization

Parameter		*Confidence interval*
Number of solutions		280 ± 57
Execution time [s]		763.4 ± 30.4
Hyperarea		1.8733 ± 0.0026
Spacing		0.0117 ± 0.0078
Minimum value	$f_{Z1} \equiv \tau$	1.5292 ± 0.0001
	$f_{Z2} \equiv \ddot{r}_p$	0.4444 ± 0.0002
Maximum value	$f_{Z1} \equiv \tau$	3.0900 ± 0.0761
	$f_{Z2} \equiv \ddot{r}_p$	0.7637 ± 0.0146

Zone II contains the best combination of solutions in both objectives, so they are the most convenient parameters for most of the operations, giving reasonably good hole quality within low drilling times. The cutting parameters in this zone are the intervals: 9.4m/mim≤v_c≤23.2m/min; 245mm/min≤f_{tooth}≤299mm/min; 0.031mm≤*step*≤0.060mm.

Finally, Zone III involves solutions with shorter execution times, but with higher vibration amplitudes. Consequently, these are only solutions for holes where quality is not an important requirement. Solutions in this zone have cutting parameters within the following intervals: 10.0m/mim$\leq v_c \leq$25.5m/min; 276mm/min$\leq f_{\text{tooth}} \leq$300mm/min; 0.025mm$\leq$*step*$\leq$0.060mm.

Regardless of their proper characteristics, the cutting parameters of the three zones share some features. The Pareto front solutions, in all the zones, cover almost all of the cutting speed intervals, v_{c}, and steps, *s*. On the contrary, the values of the feed rates, f_{rate}, are higher than $245mm/\min$ in all of the zones. Table 3.8 displayed 95% confidence intervals for the execution time and some characteristics and quality metrics of the Pareto fronts that were obtained. As may be noted, there is a good coincidence between most of the parameters, especially in the quality metrics (hyperarea and spacing), where the variability is lower than 1%. The minimum values of the objectives in the Pareto fronts are also of low variability, but the maximum values are a little more widespread.

3.5 Conclusions

A modified cross-entropy method to address simple multi-objective optimization problems is presented in this chapter. The proposed algorithm (SMOCE) is a simplification of a previous multi-objective cross-entropy method (MOCE+), where only four parameters (epoch number, working population size, histogram interval number, and elite fraction) are stored in the procedure, in order to facilitate the tuning process.

In the analysis of the relationship of these parameters on the algorithm performance, both the histogram interval number, and the elite fraction showed not significant influence, so they were removed from the study. On the contrary, the epoch number and the working population size had a remarkable influence on the execution time. Both influenced the Pareto front quality, up to some level, but as from this value, the quality stopped improving. It should be noted that this behavior was similar for all the problems under consideration.

Subsequently, a comparison of algorithm performance linked to SMOCE, MOCE+ and NSGA-II, showed no better results for the MOP and the ZDT suites. Nevertheless, SMOCE performed notably better than the other two approaches for problems from the WFG suite. As a result, it can be concluded that SMOCE, is especially suitable for highly complex optimization problems, particularly when the algorithm parameters are properly selected.

Finally, the SMOCE is validated in an industrial study case (micro-drilling process). The resulting Pareto front demonstrated the suitability of the proposed approach for solving practical problems, including those involving constraints and using neural network-based models. The algorithm (SMOCE) developed in this chapter is one of the core element of the cognitive level. In the next chapter will be described the Artificial Cognitive Architecture.

Chapter 4
ARTIFICIAL COGNITIVE ARCHITECTURE. DESIGN AND IMPLEMENTATION

Nowadays, even though artificial cognitive architectures represent an emerging field of research, there are many constraints on the broad application of artificial cognitive control at an industrial level and very few systematic approaches truly inspired in biological processes, from the perspective of control engineering. One way to address the bio inspiration is the emulation of human socio-cognitive skills and to formalize this approach from the viewpoint of control engineering facing actual industrial problems.

In this chapter, an artificial cognitive control architecture is proposed. It is based on the shared circuit model of socio-cognitive skills taking into account paradigms from Computer Sciences, Neuroscience and Systems engineering. The design and implementation of artificial cognitive control architecture is focused on four key areas: (i) self-optimization and self-leaning capabilities by estimation of distribution and reinforcement-learning mechanisms; (ii) portability and scalability based on low-cost computing platforms; (iii) connectivity based on middleware; and (iv) model-driven approaches. The results of simulation and real-time application to force control of micro-manufacturing processes are presented as a proof of concept. The proof of concept of force control yields good transient responses, short settling times and acceptable steady-state error. The artificial cognitive control architecture built into a low-cost computing platform demonstrates the suitability of its implementation in an industrial setup.

The chapter in divided in six sections. The bases from layer to module architectures and module to modes architecture are introduced in the section 4.1. Subsequently, the self-capabilities proposed in the cognitive architecture are described in section 4.2. Following, section 4.3 describes the foundation of the artificial cognitive control architecture. After that, the design and development are presented in section 4.4. The simulation and the real time application to monitoring and control a micro-drilling processes are presented in section 4.5. Finally, some concluding remarks on the experimental results are outlined in section 4.6.

4.1 Bases and general description

The bio-inspiration roots of the Artificial Cognitive Architecture is based on the shared circuits model approach [57]. SCM approach serves as the foundation for designing an artificial cognitive system where imitation, deliberation, and mindreading processes are emulated through computational efficient algorithms in a computational architecture. Hurley's approach suggests that these capacities can be achieved just by having control mechanisms, other-action mirroring, and simulation. An artificial cognitive system should incorporate these capacities and therefore it would be capable of responding efficiently and robustly to nonlinearities, disturbances and uncertainties.

4.1.1 From layer-based approach to module-based approach.

A computational architecture for an artificial cognitive system is underpinned by the modified shared circuits model. Therefore, it is necessary to enrich SCM approach from a computational science viewpoint. To develop a complex cognitive agent, it is necessary to make a global structure that would be a collection of information processing elements, linked by information forwarding elements layered atop physical/information interfaces [301].

This section explains the modifications introduced to SCM to enrich and improve its capacities, taking into account the suggestions reported in the state-of-the-art and the main constraints of the SCM approach. Since a layer-based model was incorporated in a computational architecture, five modules were constructed made up of one or more processes performed by the above-described layers. Moreover, some limitations of SCM approach and some modifications to enrich and improve the SCM approach are introduced. A functional parallelism between layers in SCM and modules in are established.

The neural block control for synchronous generators presented by Felix, et al. [25] shows a cognitive solution for this problem, due to it has to face with nonlinearities in the synchronous generator dynamics. The nonlinearities at the control problem are present at MSCM control, but the neural block control uses a specific mathematical model developed to control a synchronous generator.

SCM approach lies within the scope of philosophy, psychology, and neuroscience, so it does not consider essential aspects of the problems that must be addressed to represent and incorporate SCM to a computational architecture. The layer 1 (feedback control) is vaguely described in Hurley's paper [57], as a flow of information supplied by the effect of the actual output on the environment, taking account of the actual state of the environment, i.e., the exogenous signal. Therefore, SCM approach does not clearly justify how the agent learns from observing others, i.e., how successful instrumental of other agents in its behavior associations are incorporated.

From the perspective on System Theory and Computational Science, module 1 of MSCM presented by [45] is equivalent to layer 1 in SCM introduced by [57]. Module 1 is represented by a controller *C* and an optimization/adjustment process for this controller. This controller performs the instrumental association between input and output, similar to the description in SCM and very similar to closed-loop control systems widely used since the early 20th century. For the sake of clarity, it is assumed that the feedback (y') is the process output with noise and disturbances (y^{**}). Thus, the proposed system partakes of the enactive nature underlined by SCM. The feedback can be inhibited to benefit the output prediction (y_2) generated by module 2, as shown later. So, similar to layer 1 of SCM, the inputs are a reference signal r, according with the objectives, and the system output y''. The control signal u'' is the output of this module.

In order to harmonize all components of the module 1, unlike SCM, an external module in charge of objectives management is proposed to run at the executive level. Moreover, the instrumental association-making process is equipped with an initial knowledge base of instrumental associations, which undergoes modification as agents learn from their environment.

Apart from this own knowledge base, there is another knowledge base, a sort of common repository, which is enriched and modified by the successful input/output relationships the agents observe. This knowledge base is managed by module 3. In SCM, this process is equivalent to learning, and it is where observation is referred to, since an architecture made up of agents is proposed. In this work, there is a space to hold common, shared knowledge modified by all the agents that have the same role or belong to the same type. Inter-agent communications are thus eliminated, and the architecture is accordingly simplified. Nevertheless, this solution implies that a mechanism for controlling access to the knowledge base has to be included.

Using these two knowledge bases, an agent learns through what modules 3 and 1 do. The knowledge base managed by module 3 collects and manifests the behavior of others on the basis of what is envisaged in the common repository, and it is in the second knowledge base managed by module 1, where this new knowledge is incorporated, as we shall see later. If neither the actions described in the instrumental associations used by module 1 nor any other imitable actions are successful, module 1 will be the module in charge of carrying out a new action resulting from a controller optimization/adjustment. Module 1 also contains an optimization/adjustment process.

An optimization/adjustment procedure is introduced in MSCM. This procedure uses a set of inverse models $\underline{M}$ that module 2 handles, as shown later. The minimization or maximization criterion in optimization is determined by performance index J (see *Module a: Performance Index or Figures of Merit Computation*). Also participating in this module is an anticipative stage *C'* that attempts to speed up the control process by anticipating changes in the system reference.The system utilizes a performance index or a figure of merit J to assess its own behavior. Therefore, a performance index J is calculated by weighting the figures of merit J_i selected by selector J_s according with the actual objectives and goals. These performance indices are basically error-based criteria (i.e., deviation of the process output with respect to the reference).

It is important to note that SCM does not manage objectives and goals, therefore the management of objectives are not adequately addressed. Thus, it is necessary to include a module that carries out this task. The main role of module 0 is to supply a set of reference signals that module 1 uses to achieve the eventual objectives. MSCM can handle multi-objectives by technical, production, economic, and other objectives into references r_i and the corresponding figures of merit J_i at the system's executive level.

The user sets the objectives into the objective manager or module 0 which translates these input into a reference signal r to the basic adaptive feedback control implemented to the module 1 (see *Module 1: Basic Adaptive Feedback Control*) and into a performance index switch J_s (see Figure 4.1). The performance index switch selects a figure of merit that evaluates how well the objectives are being achieved (see *Module A: Performance Index or Figures of Merit Computation*).

Layer 2 of SCM is the layer in charge of simulating the effects of a possible action on the future input signal anticipating and thus avoiding some negative effects of the feedback process (see Figure 4.1). However, it is too soon to predict the effect of an action on the environment when the action has never been observed before, as SCM approach assumed. Makino [59] also remarks that SCM does not specify when the operation of layer 2 should be inhibited. He also enunciates the *self-observation principle* (*SOP*) and establishes that, in order to enact the property of mindreading (one of the properties whose enactment SCM describes), one needs to develop a predictive model on the basis of observation of one's own movements. In this sense, Llinás and Roy [56] suggest that it is needed that the nervous system evolves a set of strategic and tactical rules to optimize prediction in order to generate intelligent motricity.

The *SOP* principle serves to adapt some modification basically in layer 2 of SCM. In module 2 there is a set of forward models $\underline{M}$ on whose basis, given a control signal, a future output is generated. In order to perform this task, it is also relevant to take into account the actual characteristics of the environment, i.e., the exogenous input and/or the influence of noise and disturbances. This input differs from the output of the actuator system because it consists of external events to the agent independent of the agent's action.

In order to be more specific, module 2 of this proposal runs before the action is performed in order to evaluate/deliberate about different action possibilities, depending on whether the agent's criteria (module 1) are successful or not. However, it is always functioning with the object of its output's being compared to the output of the actual process, so that module 1 can learn (see *Module Interaction*).

SCM approach establishes that, when the actual input and the predicted input do not match, the actual input is used as module 1's input. This rule is applied in this work at module 2 of the MSCM since the meaning of the deviations of the process models is the presence of noise or unknown behavior y^*. An artificial mechanism can be introduced to update the model and to use the module 1 using a threshold for the level of noise. This mechanism is carried out by a process that observes the new effects and learns to incorporate these new effects into the model $\underline{M}$.

In layer 3 of SCM, the property of imitation is enacted. The agent carries out an action that mirrors or copies behavior observed in others. Hurley [57] underlines this fact as an important one in the learning process. However, as Carpendale and Lewis [58] critically observe, mere mirroring of an action does not lead to understanding of that action, as shown in the examples given by the authors. They identify the cause of this error as the interrelationship between layers in SCM, whose description evades the distinction between information and knowledge. In addition, SCM adopts imitation as conjunction, i.e., a phenomenon that conjugates at the same time:

- Learning of an instrumental relationship between a body movement and its effect, and thus;
- a way of carrying out such movement.

In addition to the literature in this field, empirical evidence and daily experience support the idea that both learning through observation and imitation can occur independently. Under this philosophy, we tend to forget that the copying process often requires the observer to establish the necessary relationship between the visual information gained from observing the action and motor output, under conditions where it is not obvious how the necessary information for this sort of mapping has been acquired. So MSCM proposal enriches SCM work addressing learned knowledge as that knowledge that has been incorporated into the set of instrumental associations, that module 1 handles in MSCM. In the meantime, the knowledge that expresses imitation, used in layer 3 in SCM, remains in the set managed by module 3 in MSCM.

Therefore, on the basis of error signal *e* the output generates action, mirroring the behavior of others, as depicted in Figure 4.1. The input for module 3 similarly to the input for module 1 may be determined either by the system input plus feedback or by the effects simulated by module 2, depending on whether module 4 is inhibiting module 2 or not. The module 3 supports a set of inverse models $\underline{M}^{-1}$ that obtains an imitative output action taking the error *e* as input (see Figure 4.1).

Module 3 of MSCM is used to enact the capacity of imitation of a common knowledge base (discussed in detail in the description of module 1), a sort of repository shared by the agents, which is enriched and modified by the agent-observed relationships that prove successful.

Layer 4 is the layer in charge of inhibiting the capacities for evaluating different possibilities of action according with SCM. However, SCM approach does not identify when to inhibit. In addition, the conceptual scheme of SCM does not clarify the influence of the action of layer 4 on the interaction among layers 2 and 3. If layer 4 is not inhibiting the input simulation in layer 2, layer 3 may have as inputs the actual input or the input simulated by layer 2. Likewise, if layer 4 is not inhibiting the output simulation in layer 3, layer 2 may have as its input this simulated output or the actual output.

For this reason, from the perspective of computational science, it is necessary to define and implement some mechanisms to decide whether or not to perform imitation. That is why the decision whether to enact deliberation (which occurs through the operation of module 2) or imitation (which is enacted in layer 3) is proposed to be done in another module. The module that is in charge of managing any orders of this sort issued to SCM layers 2 and 3. So, very close to SCM layer 4, MSCM module 4, therefore performs executive functions within the system, as it is the module in charge of managing the aforesaid orders (see Figure 4.1).

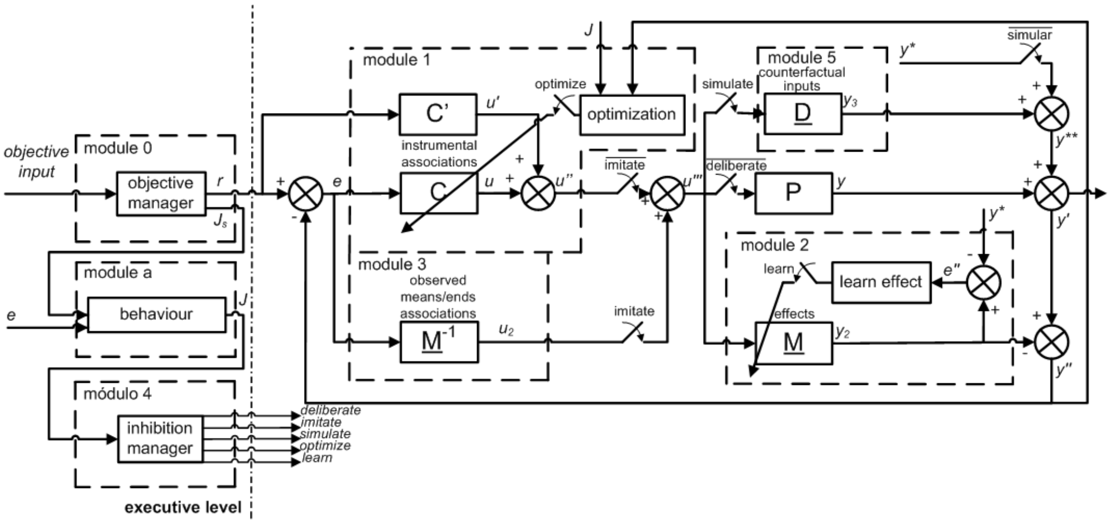


Figure 4.1 Expanded block diagram [45]

Thus, as Makino [59] proposes concerning SCM approach, inhibition depends on failure monitoring. What kind of inhibition there is characterizes the deliberation, action, and imitation cycle, as described in *Module Interaction.* We might stress that, where there is a certain amount of disturbance in the environment, there must be no imitation. The use of a performance index *J* is used as a figure of merit in module 4 is introduced in MSCM approach.

The issue of the development of mindreading capacities is not convincingly addressed in the SCM. One possibility is to observe the behavior of others when layer 4 is not inhibiting layer 3, in order to acquire the observed input/output relationships. In this work, since there is a common knowledge base where each agent contributes, no observation entailing an exchange of messages among agents is necessary.

A key issue is when module 1 is optimized or when a new learning is enabled at module 2, because it depends on the system's status (deliberating, acting, etc.), forming part of the system's operational cycle. Therefore, module 4 is in charge of checking if the system shifts from one state to another (see *Module Interaction*); accordingly, module 4 also decides when to perform optimization and learning.

SCM describes a fifth layer in charge of simulating the effects of the behavior of other processes on the basis of self's own behavior. In the computational system, this translates into a module in charge of simulating effects while running offline. The authors believe that, unlike as it is proposed in SCM, the decision to inhibit/not inhibit the operation of module 5 is not a self-decision, but may be made by module 4. The activation of one module alone or combined modules (modules 2, 3 and 5) enables one capacity or another; accordingly, it is advisable for the decision to be centralized in a single module. So deliberation on how a possible action would influence others or external noise (exogenous input) is made if module 2 and 5 are running at once. Deliberation on how a possible imitated action would influence others or external noise (exogenous input) is made if module 2, 3 and 5 are functioning together.

4.1.2 Module Interaction

The relationships and interactions among modules make possible to artificially emulate the cognitive capacities of deliberation, imitation, and mindreading. In order to develop a computational framework aiming at control system design, it is necessary to address module interactions. Layers interactions is one of the main weaknesses of the SCM approach. SCM approach just outlined these interactions, neglecting the temporal issues underneath how they operate. This temporal pattern is essential to clarify how and when the system acts, and therefore it is necessary to set a method or strategy to establish a sequence of actions for each module. In this section, one of the possible operating sequences of the different modules is explored, in view of the results already reported in the literature.

SCM overlooks a crucial component for imitation: motivation. Whether or not imitation takes place, as we have seen, depends on whether or not module 3 has been inhibited from functioning. Under Makino's assertions, the necessary motivation can be provided by failure monitoring: if the actual action is not successful, the imitation mechanisms are triggered. In addition, SCM suggests that first self is related with actions of other, as evidenced by observation of newborns' imitation of the facial gestures of others. And afterwards, those self is related bidirectionally with the mental states of self in a certain way, through learning.

The above-mentioned arguments lead to introduce new mechanisms in the MSCM, as follows. If the result of evaluating the set of possible innate or already-acquired actions is not satisfactory, the action leading to the sought-after objective is imitated. If this action is satisfactory, the agent learns, incorporating the action into the set of instrumental associations that the module 1 control uses. Otherwise, module 1 acquires new knowledge by means of an optimization process. The optimization process is the last mechanism, because it is more costly than copying an observed association.

The following knowledge bases, according with the described models, set upon the type of involved models, are proposed in this work:

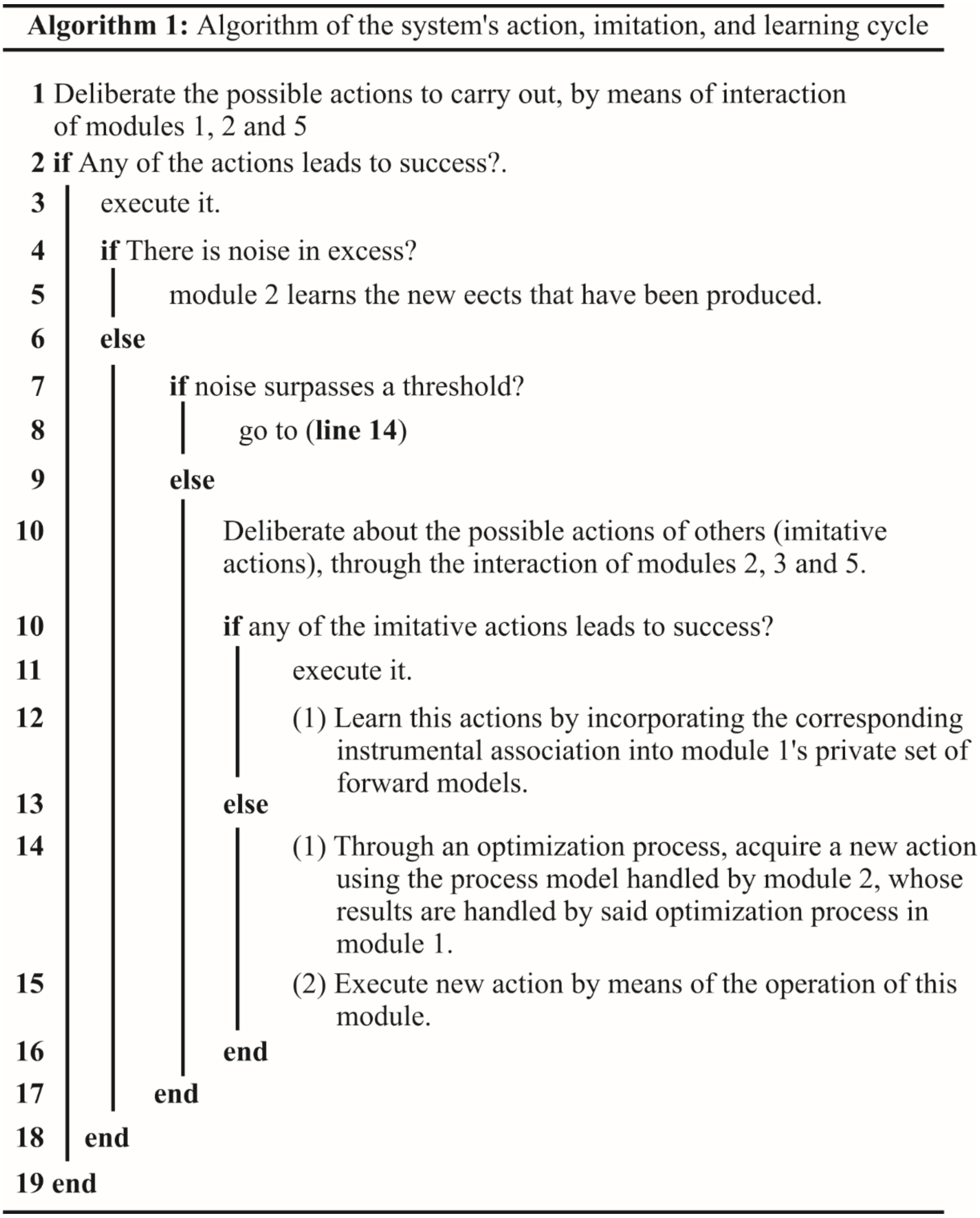
- Own set of forward models $\underline{D}$, handled by module 5 to simulate the counterfactual effects of actions taken by the system. This set replaces process $\underline{P}$ if this simulation is required
- Own set of forward models for effects $\underline{M}$, available for the module 2
- Set of inverse models for imitation $\underline{M^{-1}}$, available for the module 3

In this approach the strategic deliberation that performs module 5 must participate in all deliberation phases of the procedure shown in figure 2, together with module 2. The rationale is that it is important to roughly know the eventual disturbances that could appear in deliberation to anticipate the influence of disturbance on the process.

An artificial cognitive control system is designed according the method described in Figure 4.2. In this iterative procedure, the evaluation of whether or not there is an excessive noise is determined by observing whether noise surpasses a certain threshold.

We can point out that, when a decision's success is evaluated in the action, imitation, and learning procedure, success will depend directly on the error found for the decision's implementation.

In order to carry out this process with the described modules, the modules are connected as shown in Figure 4.1. The main feature to stress here is that now module 4 is inhibiting module operation by acting on the switches to choose between the output calculated by modules 1, 2 and/or 3. Thus, module 4 is in charge of governing the action, imitation, and learning cycle.



Algorithm 1: Algorithm of the system's action, imitation, and learning cycle

```
1 Deliberate the possible actions to carry out, by means of interaction
  of modules 1, 2 and 5
2 if Any of the actions leads to success?.
3     execute it.
4     if There is noise in excess?
5         module 2 learns the new eects that have been produced.
6     else
7         if noise surpasses a threshold?
8             go to (line 14)
9         else
10            Deliberate about the possible actions of others (imitative
              actions), through the interaction of modules 2, 3 and 5.
10            if any of the imitative actions leads to success?
11                execute it.
12                (1) Learn this actions by incorporating the corresponding
                      instrumental association into module 1's private set of
                      forward models.
13            else
14                (1) Through an optimization process, acquire a new action
                      using the process model handled by module 2, whose
                      results are handled by said optimization process in
                      module 1.
15                (2) Execute new action by means of the operation of this
                      module.
16            end
17        end
18    end
19 end
```

Figure 4.2 Algorithm of the system’s action, imitation, and learning cycle

The comparator at the output of module 2 enables to discern whether the predicted input is similar to the actual input; if it is not, the actual input is used to calculate next control actions. The actuator system receives a signal and acts on the process input $\underline{P}$. Module 2 is always operating, because it handles a representation of the process (model). The process model is necessary to enable the optimization/adjustment in module 1 from the results of comparing the process output and model output in the module 2. This optimization/adjustment is not the learning process to incorporate successful imitative knowledge of module 3 to module 1. The optimization/adjustment process is carried out when the imitative action is not successful at the deliberation stage, and it is necessary that module 1 performs a new action enabled for this optimization/adjustment.

For the sake of simplicity, the signal governing deliberation and module 5 can be treated as a single signal, since they act jointly on the suggested operating mechanism. However, this is an operating sequence procedure introduced in this work, and the two cases are considered separately aiming at further research on their activity independently. The interaction between modules is represented in detail in the expanded block diagram shown in the Figure 4.1.

For example, if observations were under deliberation, module 4 would activate the *imitate* signal, not inhibiting the operation of module 3 (which covers the observations or instrumental associations to be imitated). Moreover, it would activate to *simulate* signal; and it would deactivate the no-deliberate signal, i.e., enabling feedback on the basis of the output of module 2 without taking into account actual process *P*.

4.1.3 From module-based to operating mode-based concept: model-driven approach. Drawbacks and challenges.

The architecture presented in this Dissertation is inspired in neuroscience [302] in conjunction with control engineering strategies and methods [45]. It is widely accepted that the cerebellum acquires and maintain internal models for motor control. Recent reports corroborate by neuroimaging and clinical studies the cerebellar role in performance monitoring with focus on sensory prediction, error and conflict processing, response inhibition, and feedback learning [303]. Moreover, the characteristic input-output organization of the cerebral-cerebellum and how it may contribute to forward models for non-motor higher brain functions is also recently presented in a seminal paper [304].

In chapter 1, we showed how micro-scale manufacturing is a clear example of a dynamic system operating in an environment characterized by continuous change, being a perfect stage to proof new cognitive control strategy. In this scenario, one of the main objectives is the development of technologies and algorithms that enable faster, self-organized, self-optimized behavior process control systems. These manufacturing processes are characterized by the presence of nonlinear and time-variant dynamics that emerge from the behavior of temperature, forces, torques and other representative variables; characteristics that increase the functional complexity of micro-manufacturing and the functional requirements and precision of sensors, actuators and computing resources [9, 78].

In this case-study, we describe artificial cognitive control architecture with self-optimization and self-leaning capabilities and its simulation and real-time application to the force control of micro manufacturing processes as a proof of concept. The architecture, based on the model of socio-cognitive skills, overcomes the limitations of the neuroscientific approach [58-60] and takes the principles of simplicity and scalability into account. A further challenge is to implement the architecture in a portable programming language for its assessment and validation in simulated and real micro-manufacturing processes.

In this chapter, the design and implementation of artificial cognitive control architecture is focused on four key areas:

(i) self-optimization and self-leaning capabilities by estimation of distribution and reinforcement-learning mechanisms;
(ii) portability and scalability based on low-cost computing platforms;
(iii) connectivity based on middleware; and
(iv) model-driven approaches.

4.2 Self-capabilities

4.2.1 Self-Learning

Reinforcement learning belongs to a category of unsupervised learning techniques [127]. It is a learning paradigm by rewards/penalties with some interesting applications for controlling complex systems, so as to maximize numerical performance measures that express a long-term objective. The analysis of all available reinforcement learning methods is beyond the scope of this paper. Although [305] offers a fairly comprehensive catalog of learning problems with a description of an important number of state-of-the-art algorithms.

Q-learning is one of the most intensively used reinforcement learning techniques, frequently used to find an optimal policy for Markov decision processes. The main rationale behind this choice is the simplicity of its approach, its model-free feature and the good results of this algorithm reported in the literature. It performs by learning an action-value function that ultimately generates the expected utility of taking a given action in a given state and then it follows the optimal policy. When such an action-value function is learned, the optimal policy can be constructed by simply selecting the action with the highest value in each state. Additionally, Q-learning can handle problems with stochastic transitions and rewards, with no further adaptation.

Before learning started, Q can return any single value, chosen by the designer of the problem. Then, each time the agent selects an action; it receives its rewards and enters the new state. The core of the algorithm is a simple iteration to update values. It takes the old value and makes a correction based on the new information. Normally, Q-learning is periodically executed where an episode ends when state s_{t+1} achieved the final state (see Eq. (1.5)). However, Q-learning can also learn in non-episodic tasks. It may be noted that Q-learning does not specify a method to select the action to perform in each state. However, there are several policies to select an action, i.e., the well-known $\in$ -greedy or softmax policies.

It is considered that the state is a set of parameters of the model or models, thus each state is identified unequivocally with a set of parameters. Thereby, the actions to change from one state to another are those that change at least one parameter of the set $(K_1^{(t)}, K_2^{(t)}, \ldots, K_n^{(t)})$. Thus, the Q-values function is $Q(s_t, a_t) = Q(s_t)$. The continuous space of the variables is discretized for simplicity as already reported in [306]. Therefore, each parameter K_i has its own bounds $\left[K_i^{\min}, K_i^{\max}\right]$ given by the model's parameters. Then, if there are M possible values of each parameter, the range of values are:

$$K_{i_1} = K_i^{\min}, K_{i_2} = K_{i_1} + \frac{K_i^{\min} - K_i^{\max}}{M-1}, \ldots, K_{i_M} = K_i^{\max} \tag{4.1}$$

From the equation (1.5) is clear the that the space of states with a dimension of M^N is finite. Due to the restrictions of a real environment, we cannot increase the step for specific parameter at a given time.

For a given state $s_t \leftrightarrow (K_1^{(t)}, K_2^{(t)}, \ldots, K_n^{(t)})$ its available actions will be those that change s_t to $s_{t+1} \leftrightarrow (K_1^{(t+1)}, K_2^{(t+1)}, \ldots, K_N^{(t+1)})$ where:

$$K_i^{(t+1)} \in [\max(K_i^{\min}, K_i^{(t)} + step), \min(K_i^{\max}, K_i^{(t)} + step)] \tag{4.2}$$

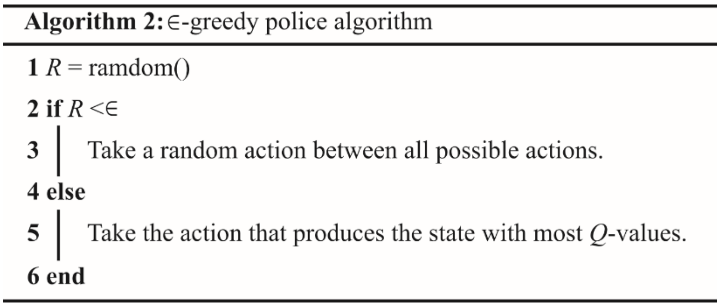
In artificial cognitive control architectures, as in any hierarchical approach there are different time scales (bandwidths). The learning procedure runs at a lower frequency than the control mechanism, which resembles a cascade concept, because the process has to run for a sufficient length of time, in order to guarantee that appropriate learning takes place. If the control mechanism has a sampling time of p_control, the learning has to be performed at least ten times slower than the control, i.e., $p_\mathrm{learning} = \delta_{learn} * p_\mathrm{control}$, where $\delta_{learn} \in \mathbb{N}, \delta_{learn} \geq 10$ [82]. The reward function can be defined as:

$$R = \begin{cases} +500 & if \quad \phi(t) \leq 0.05 \\ +100 & if \quad 0.05 \leq \phi(t) \leq 0.1 \\ -100 & if \quad 0.1 \leq \phi(t) \end{cases} \tag{4.3}$$

where the performance index associated with the action that is taken, $\phi(t)$, has the following expression:

$$\phi(t) = \left(\sum_{i=1}^{\delta} \frac{f_i - y_i^{(t)}}{f_i} \right)^2 \tag{4.4}$$

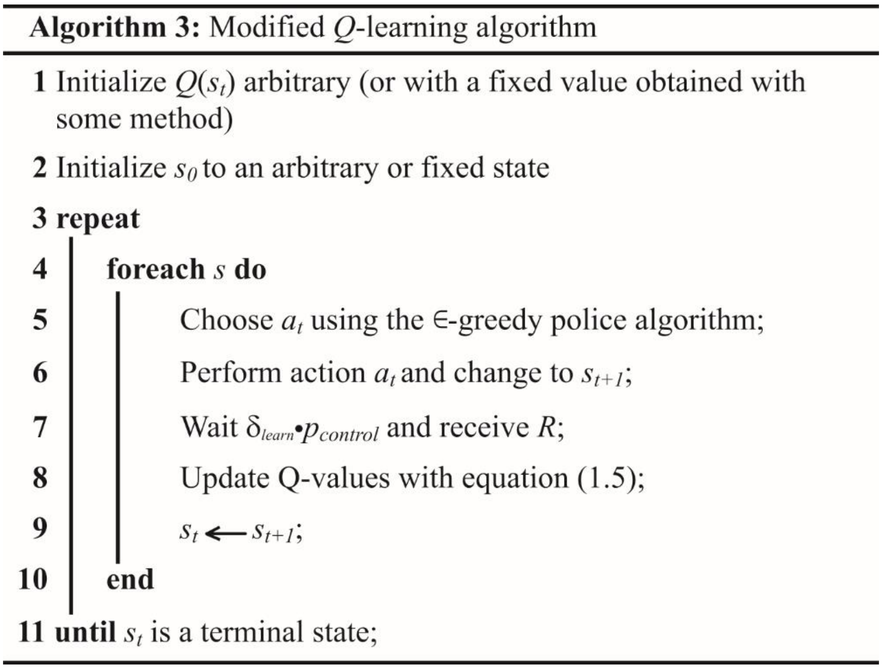
in which, ϕ_t is the reference value in time $t+i\bullet p_{control}$ and $y_i^{(t)}$ is the output of the process in time $t+i\bullet p_{control}$ with the parameter set $(K_1^{(t)},K_2^{(t)},\ldots,K_n^{(t)})$. As it can see, ϕ(t) is the mean square error evaluated in $\left[t,t+i\bullet p_{control}\right]$. For the sake of simplicity, we used the $\in$-greedy policy in this first approach, to choose an action, because it is sufficient in almost all scenarios. The $\in$-greedy policy algorithm is shown in Figure 4.3.



```
Algorithm 2: ∈-greedy police algorithm
$R$ = ramdom()
if $R$ <∈
    Take a random action between all possible actions.
else
    Take the action that produces the state with most $Q$-values.
end
```

Figure 4.3 Algorithm for $\in$-greedy policy

All the steps in the modified Q-learning algorithm are presented in Figure 4.4.



```
Algorithm 3: Modified $Q$-learning algorithm
Initialize $Q(s_t)$ arbitrary (or with a fixed value obtained with
some method)
Initialize $s_0$ to an arbitrary or fixed state
repeat
    foreach $s$ do
        Choose $a_t$ using the ∈-greedy police algorithm;
        Perform action $a_t$ and change to $s_{t+1}$;
        Wait $\delta_{learn}\bullet p_{control}$ and receive $R$;
        Update Q-values with equation (1.5);
        $s_t \longleftarrow s_{t+1}$;
    end
until $s_t$ is a terminal state;
```

Figure 4.4 Modified Q-learning algorithm

4.2.2 Self-optimization

As it was explained in the section 1.2, the literature is very rich of deterministic and stochastic methods for solving optimization problems [99, 100]. In EDAs, a probabilistic model is built, based on elite individuals who are subsequently sampled to produce a new population of better individuals. A positive aspect of EDAs is that the fusion of prior information into the optimization procedure is straightforward, thereby reducing convergence time when such information is available. From a computational cost viewpoint, the amount of heuristics compared with other gradient-free optimization methods is reduced, which means that, in practice, many heuristic optimization methods are not suitable [104].

For all the above reasons, the attention is focused on so-called Cross Entropy method [109], as the main self-optimization algorithm for the artificial cognitive control architecture. The most attractive feature of cross entropy is that, for a certain family of instrumental densities, the updating rules can be analytically calculated, making them extremely efficient and fast. The method can be described as:

Let *X* be a random variable defined in the space χ and $f : \chi \rightarrow \mathbb{R}$ a score function. The CE method seeks to find x' such that:

$$f(x') = \min_{x \in \chi} f(x) \tag{4.5}$$

The algorithm transforms this problem into an associated stochastic problem by defining a family of random variables with density function $g(x, v)$, $v \in \Gamma$ and solving it as the simulation of a rare event, where the event is sampling around the optimum of *f*. The algorithm can be summarized as follows:

1. Initialize v_0.
2. Generate a sample of size $N, (x_i^t)_{1 \le i \le N}$, from the density function $g(x, v_t)$. Let $f_1 \ge f_2 \ge \cdots \ge f_N$, .
3. Update *v0* to

$$v_t + 1 = \arg\min_{v} \frac{1}{N} \sum_{i=0}^{N} I_{\{y \le \gamma t\}} (f(x_i^t)) \bullet \ln g(x_i^t, v_t) \tag{4.6}$$

4. Repeat from step 2 until convergence or ending criterion.
5. Assuming that convergence has been reached at $t = t_0$, the random variable defined by the density function $g(x, v_{t'})$ should have all of its mass concentrated on x'.

Step 3 is performed using the best Z* samples, also called elite samples. The sampling density function needed in the 2nd step is usually unknown, but in most cases it can be assumed to be a normal distribution function. In this case, v represents the mean μ and the standard deviation of the normal distribution is σ. The solution of the equation is simply the sample mean μ_t and sample deviation σ_t of the elite samples. It also follows that the mean should converge to x' and the deviation should converge to zero. A smoothing parameter ψ for the mean vector and dynamic smoothing β_t for the deviation are applied, in order to prevent the occurrences of 0 s and 1 s in the parameter vectors.

$$\begin{aligned} \mu_{t+1} &= \alpha \bullet \mu_{t+1} + (1-\alpha) \bullet \mu_t \\ \sigma_{t+1} &= \beta_t \bullet \sigma_{t+1} + (1-\beta_t) \bullet \sigma_{t+1} \\ \beta_t &= \beta - \beta \bullet \left(1 - \frac{1}{t}\right)^q \end{aligned} \tag{4.7}$$

where $0.4 \le \psi \le 0.9; 0.6 \le \beta \le 0.9; 2 \le q \le 7$.

Finally, constrained optimization problems can be addressed from an engineering viewpoint, which therefore means to boundaries on the distribution function for the generation of samples, to ensure that sampling is from within the appropriate region. More details about the evolutionary algorithms, especially, on cross-entropy method are done in Chapter 3 .

4.3 General design

The architecture consists of cognitive and executive levels. The main differences in the implementation with regard to the MSCM architecture [45] developed in previous investigations by the GAMHE group are described below. MSCM embodies a computational infrastructure that is plausible from a neuroscience and psychological viewpoint. The overall diagram of the architecture is shown in Figure 4.5.

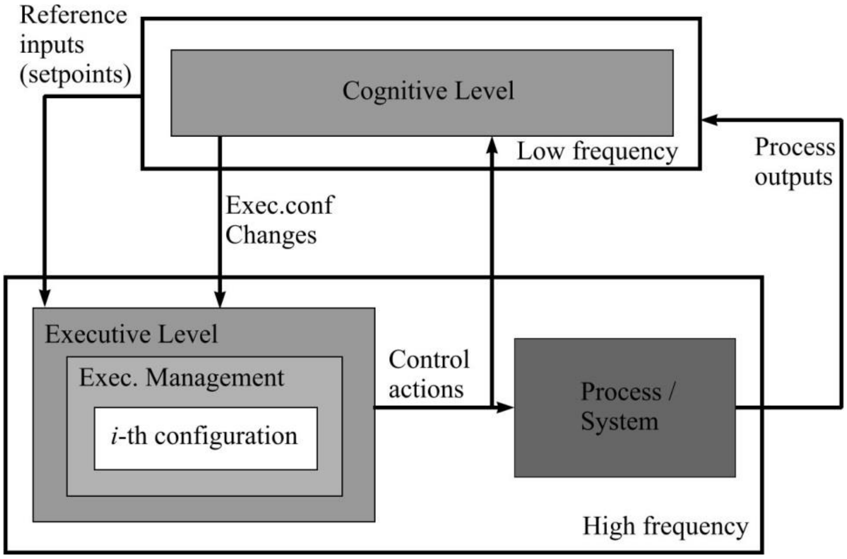


Figure 4.5 Overall diagram of the artificial cognitive architecture

A detailed scheme of the cognitive and the executive levels is shown in Figure 4.6 to gain a better understanding of the interconnection between the different parts of the architecture. There is a mirror of the execution level at the cognitive level (see Figure III.2), necessary for organization, adaptation, and learning mechanisms that make use of real-time simulations. Otherwise, these mechanisms would have to use the processing time of the execution unit, thereby limiting real-time performance.

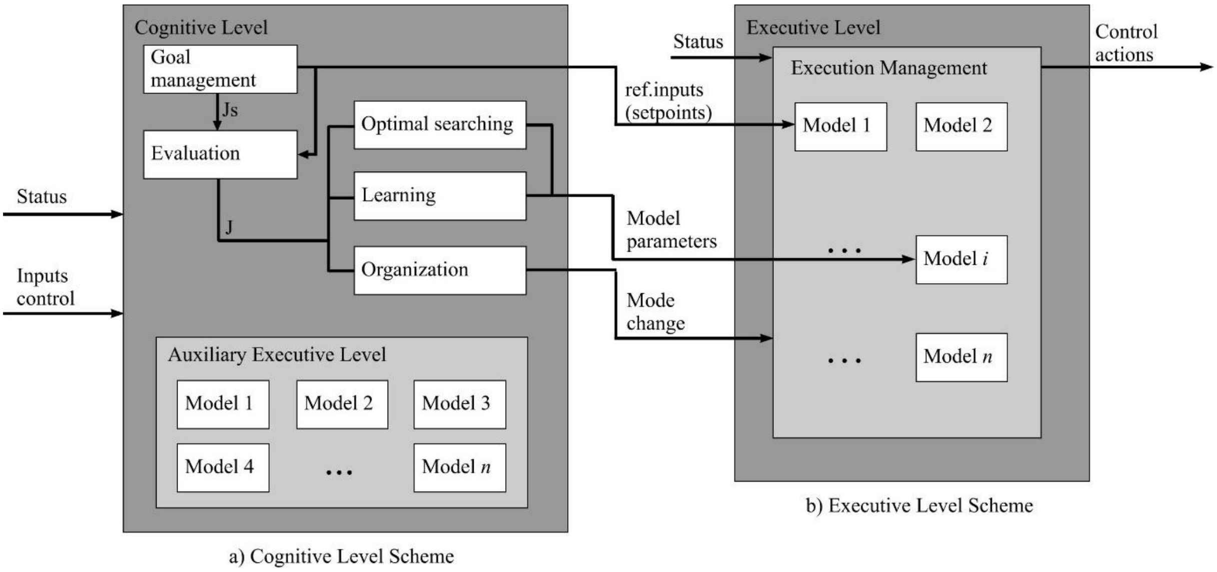


Figure 4.6 Cognitive and executive levels of the artificial cognitive architecture

On the other hand, in the executive unit (see Figure III.3), the most important input is the *exec.conf.changes* that serve to introduce modifications both in configuration and parameters. Parameter modifications change the parameters of a particular model (model *i*) from the learning mechanism and changes to its configuration can be of two types: new assignment of a model to a specific mode or switching between modes.

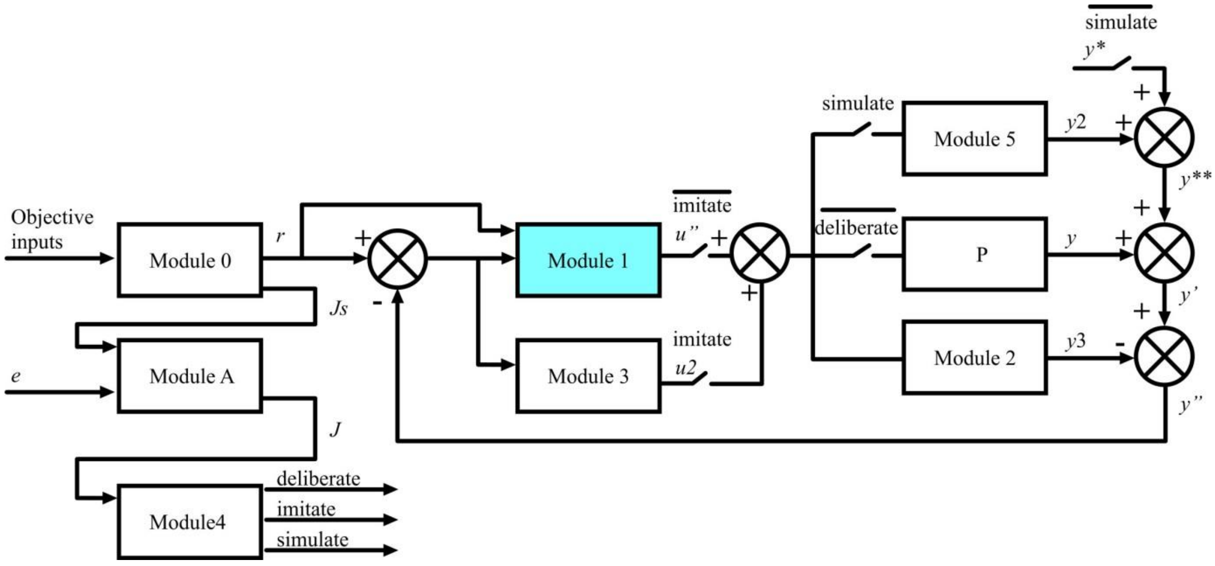


Figure 4.7 Configuration diagram by Single Loop

Particularly, a simplified version of this architecture, focusing on three special configurations of the blocks of MSCM is proposed in this Thesis. The first configuration is composed exclusively of module 1 of MSCM, due to the nature of module 1, are going to call this configuration *Single loop* (see Figure 4.7).

The second configuration is composed essentially of module 3. This module stores inverse models so, activating only this module, we achieve a configuration called *Anticipation* (see Figure 4.8).

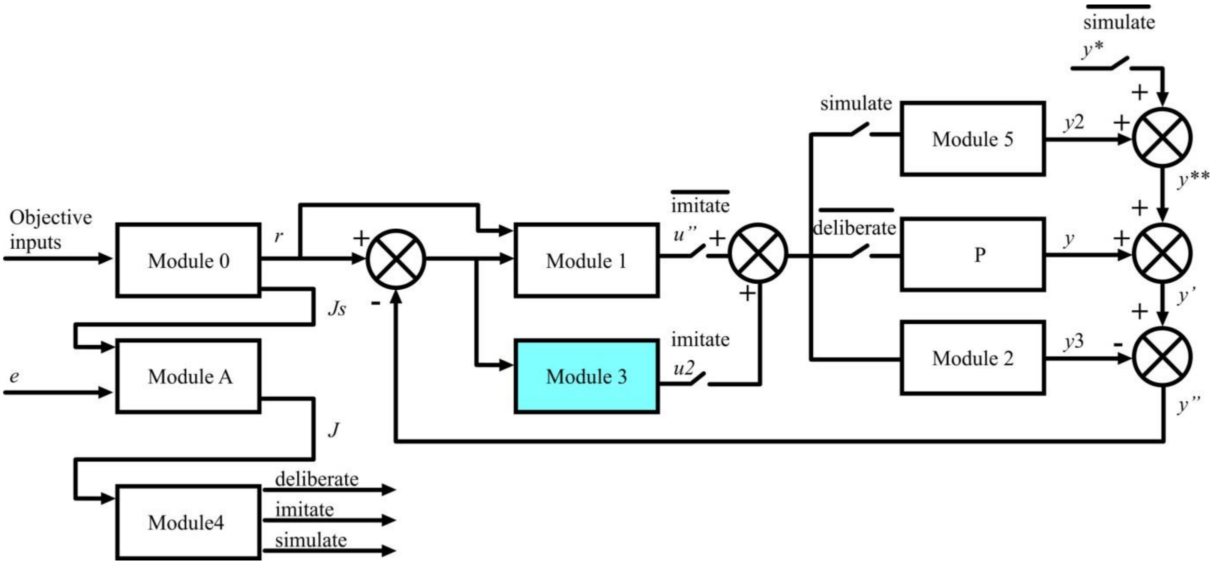


Figure 4.8 Configuration diagram by Anticipation

The last configuration is called *Anticipation + Mirroring*. Its name comes from the activation of module 2 and 3 enabling such mirroring skill (see Figure 4.9).

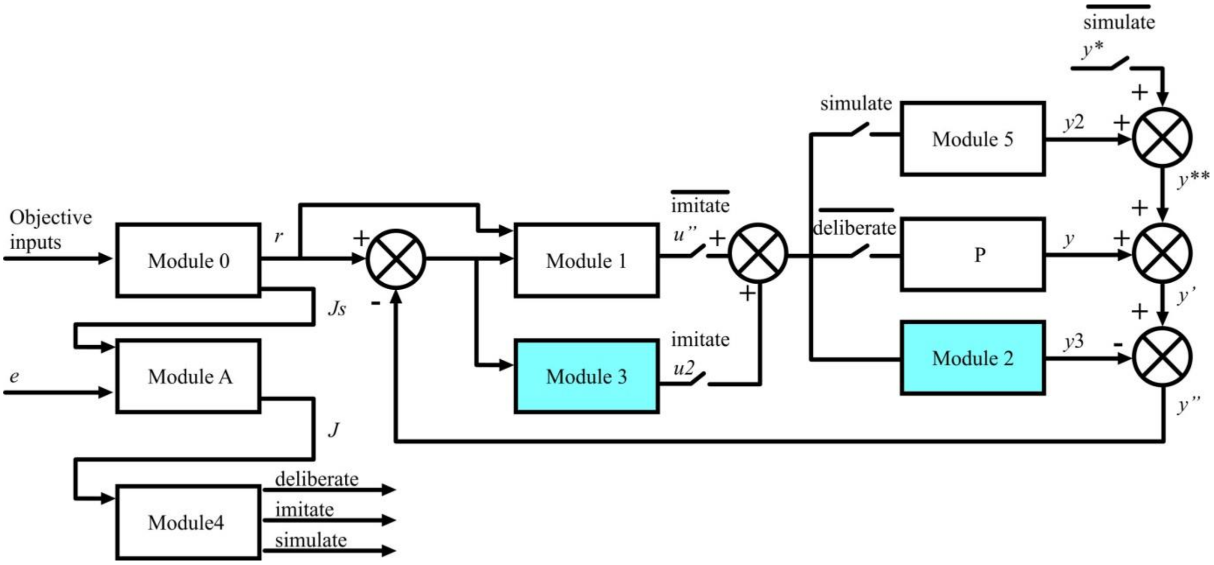


Figure 4.9 Configuration diagram by Anticipation + Mirroring

Therefore, the architecture is designed so that the user can easily add more types of models, which can lead also to other types of functioning modes. Now, it is necessary to address the concept of executing modes that is the mechanism for defining how to connect the different types of models for a certain operating mode of architecture. For the sake of clarity, only three operating modes are considered which are single feedback control, inverse control and internal model control with the novelty of incorporating self-optimization (see section 4.2.2) and self-learning (see section 4.2.1). Similarly, Fatemi and Haykin [50] have also considered three main operating modes. The implementation of the above architecture is coded in Java, which guarantees greater portability between different operating systems.

Note that the type of model is only described in the operating modes and not the model itself. The adaptation is in charge of choosing the type of model for each mode depending on the main process characteristics, know-how and available models. The organization mechanism is responsible for switching between execution modes without deciding on the type of model used for each mode. Therefore, adaptation is responsible for the transition between running modes and Execution Configuration, which is the concept that permits the interconnection of models in Execution Management. Learning is performed in MSCM through Module 2, while here it is done by a specific component that enables learning in all models of the architecture. A further component of the architecture, Optimal Searching, is in charge of optimization on the basis of simulation-type models (e.g., any computational representation of the controlled system that resembles well-known output error models). Therefore, the ability to emulate Module 5 can be carried out by simulation-type models.

The initialization procedure starts with the optimal setting of parameters of each model (inverse model, forward model, single loop model) on the basis of the cross entropy method introduced in Chapter 3 and the simulation model of the process. Indeed, self-optimization is a basic step that can serve to carry out other tasks beyond this one. The use of an error-based performance index and a rough model of the process is enough to perform this task as it is shown later in the proof of concept of the artificial cognitive control system in section 4.5.

Moreover, necessary variables for learning and organization will be initialized such as the Q-learning tables and the performance index table, in order to be later applied to set the initial conditions in the organization as well as to choose the appropriate execution mode. The performance index table is also updated using a forgetting factor.

The self-organization procedure during each execution cycle consists of eight main steps. First, the system checks if the control signal (action) has yielded the expected results according to the performance index, if so, the artificial cognitive control remains in this execution mode (e.g., single loop control). If the execution mode does not achieve Artificial Cognitive Architectures appropriate results it is necessary to verify if the learning is enabled, otherwise, it is activated. The duration of the learning (set by the user) should be checked in the third step.

The performance index table is updated with the real-time computed value with a forgetting factor. All possible control signals (actions) are then computed and the control signal that produces the best behavior (i.e., the best performance index) is taken from the table. During the execution (with learning enabled), if the mode (controller) present in the executive level obtain a worse performance index than the other controllers, in the next iteration the cognitive level send a new mode to overwrite the previous one, also the old controller is penalized and the full performance index tables is updated.

4.4 Implementation

The development of the artificial cognitive architecture is described in this section. The artificial cognitive architecture was implemented in low-cost computing platforms. The analysis of the best available middleware to enable networked, transparent, portable and reliable communication, and low-cost computing platforms is a key step before selecting the most appropriate ones for the instantiation. All the results presented in this section have been partially reported in [82, 307].

4.4.1 Requirement analysis

The artificial cognitive architecture should comply with both functional (FR) and non-functional (NFR) requirements. The main functional requirements can be summarized as follows.

- **FR1 Control architecture**: the main function of this architecture is to monitor and control processes; the implementation of the architecture permits to assign a process to the architecture and prepare the architecture to perform those tasks.
- **FR2 Models**: the architecture has several models that serve to monitoring and control a process with different procedures. There are four types of models: single loop models, direct and inverse models, and simulation models. The configuration of direct and inverse models resembles the internal model control paradigm well known in the Control Engineering community, but also claimed as the main explanation and rationale behind brain-cerebellum interaction [21].
- **FR3 Modes**: the architecture must run in different modes. A mode is defined by a preset configuration of the different elements of the architecture (models, reference values and process entity) to control a process. A mechanism enables the application to switch between modes while running. Switching can be smoothed out by a first-order filter to guarantee a seamless transition from one mode to another one.

- **FR4 Adaptation**: the application must provide a component for selecting models required by a specific mode. The choice of component may be to accomplish different objectives.
- **FR5 Optimization**: the architecture must provide functionality for optimal setting of control models on the basis of a simulation model of the physical process. With this action, the architecture will be able to improve its behavior while running different processes.
- **FR6 Online learning**: similar to optimization, the architecture should provide a mechanism to execute a learning algorithm during the regulation process. Once again, this mechanism will improve the behavior of the overall system.
- **FR7 Objectives**: the architecture must ensure that the user provides the objectives to be achieved, e.g., productivity, performance, etc.
- **FR8 Data types**: the architecture must allow different data types, such as integer, double or string. The main non-functional requirements are described as follows.
- **NFR1 Middleware**: the architecture should be quite generic and flexible to enable the use of any middleware. For instance, the user may wish to use the architecture to control a process in a different place, i.e., to distribute the architecture to control remote process.
- **NFR2 Extensibility**: the architecture shall be designed to ease the tasks of adding models, control algorithms, optimization and learning procedures, etc.

4.4.2 Libraries and classes description

In order to comply with the above-mentioned requirements, an object-oriented library was designed. Along with the general classes and interfaces, some classes are provided to ease the instantiation tasks of the architecture. The following will briefly explain the most important concepts of this library with different functions [308] as it can see in Table 4.1.

As it is expected the most important packages are: **app** (see Figure III.1), **data** (see Figure III.4), **model** (see Figure III.6) and **process** (see Figure III.7). Below, the packages are explained. In this package are the classes necessary to represent a set of variables in the architecture. Firstly, *VariableInfo* represents the information of a given variable, namely, its type and its name. With this information a variable is uniquely determined, so a *VariableInfo* is perfect to identify variables (see Table 4.2). Using this class, a *VariableSet* is built. This class provides different methods to create and work with sets of variables, such as sums two sets, multiply by a scalar, etc. As it is supposed each variable inside the set is identified by its name and type, i.e., its *VariableInfo*. With this **data** model the **RF8 is considered**, i.e., it is possible to use different data types in the architecture.

Table 4.1 Main packages of the designed library

Name	*Package Descriptions*
app	contains all the classes representing the components of the articial cognitive architecture as well as classes that represent the application to control processes.
data	contains the classes that represent a variable set in the architecture.
exceptions	contains all the exception thrown in the architecture.
model	contains an interface that represents a model in the architecture an auxiliary class to ease the implementation.
process	contains interfaces that represents a process in the architecture and a process observer as well as some auxiliary class to ease the implementation.
utils	contains a Log class to show information while running.

The **model** interface provides the sufficient methods to do almost everything it can need to do with a model (see Table 4.3). The *ModelType* represents the possible types of model that the architecture accepts: single loop, mirroring, also named forward models, simulation and anticipation, named inverse models. The *AbstractModel* class implements a template method common for all the models used in the architecture.

Table 4.2 Overview of the data package

Name	*Data Package Descriptions*
VariableInfo	represents the information of a variable.
VariableSet	represents a set of variables.

The process to be controlled by the architecture is represented by the **process** interface (see Table 4.4). As it will see in an example, each process to be controlled has to be encapsulated in a class that implements this interface. The *ProcessObserver* interface represents a process observer, this interface has to be implemented by those classes which needs to know the state of the process as well as its outputs. In order to ease the instantiation of a process an *AbstractProcess* class is provided. This class implements almost all the methods of **process** and run the process's logic in a different thread. This logic is represented by the *ProcessLogic* interface.

Table 4.3 Overview of the model package

Name	*Model Package Descriptions*
AbstractModel	auxiliary class to ease the task of instantiation.
Model	represents a model of the architecture.
ModelType	represents the possible types of a model.

The **app** package contains two interfaces, *App* and *AppObserver*; an auxiliary class, *AbstractApp;* and two more packages, Executive Level and Cognitive Level. An overview of the two packages is shown in Table 4.5 and Table 4.6, respectively.

Table 4.4 Overview of the process package

Name	*Process Package Descriptions*
AbstractProcess	auxiliary class to ease the task of instantiation.
Process	represents the process in the architecture.
ProcessLogic	auxiliary class used by AbstractProcess
ProcessObserver	represent a process observer.

The **app** interface represents the control application itself. Namely, it is the object to interconnect all the components of the architecture and the coordinates to control a process. It will be the access point to work with the artificial cognitive architecture. The *AppObserver* is an interface that represent an observer subscribed to the application, normally it will be extended for classes that need to know what is happened in the architecture, e.g., when a control action is sent to the process. The *AbstractApp* implements several simple methods of the App interface.

For a better understanding of the architecture, we begin explaining the Executive Level (see Table 4.5). In this package we can find three main interfaces, *Module*, *ExecutionConfiguration* and *ExecutionManagement*.

The *Module* represents, mainly, a container of models, so the modules are very important in the architecture. However, the power of a module relies on the models it contains, i.e., an empty module is nothing for the architecture. Each model has a type and a module is not restricted to contain only models of one type, e.g., a module can contain two forward models and one inverse model. Due to, mainly, a *Module* is only a repository of models an implementation is provided in class *SimpleModule*. This class implements all the necessary methods and ensures that each execution has an independent copy of each model that it needs. This interface along the **model** interface satisfies the **RF2**.

Table 4.5 Overview of the app package for the Executive Level

Name	*App Package Descriptions*
AbstractExecutionManagement	Auxiliary class to ease the task of instantiation.
ExecutionConfiguration	represents an execution configuration of the architecture.
ExecutionManagement	represents the execution management of the architecture
Module	represents a module of the architecture.
SimpleModule	an useful implementation of module to be used directly.

The second interface, the *ExecutionManagement*, represents the component that maintains the execution threads of the architecture. These threads are independent of the other components of the architecture and always try to run in the same frequency of the process to provide a better control. Then, the function of the *ExecutionManagement* is to interpret the *ExecutionConfiguration* and execute the models of each configuration in the right order with its correct inputs. In order to ease the implementation of the *ExecutionManagement* an abstract class, *AbstractExecutionManagement*, is provided. This class implements some simple methods of its interface in order the user can focus on the important methods.

Finally, the *ExecutionConfiguration* is an auxiliary class that the *ExecutionManagement* uses to retrieve the information of how the models are connected in a given mode. The cognitive level is the recipient of the self-capabilities that will be implemented (see Table 4.6). Before describing these components, we will explain the modes of the artificial cognitive architecture.

A mode in the architecture is simply a topology, or a pattern to interconnect some models. With this conception, a mode can be represented graphically by a graph like the common single loop of the control theory is represented (see Figure 4.10). This conception permits to interconnect different models to achieve different results in order to cover the objective. The class that represents a mode in the architecture is Mode. It is important to note that a mode does not specify a concrete model, it only specifies the type of models that are connected. All these considerations address the **RF3**.

The *Organization* in the architecture has the function of loading the modes in the *ExecutionManagement* and switching between modes when it is necessary. To switch between modes, the *Organization* has to notify to the *ExecutionManagement* this change. However, the *Organization* does not know what models have to use in each mode, this task corresponds to the *Adaptation*. In this manner, the mission of the Adaptation is ideally choosing the best models for each mode dependent of circumstances, e.g., the process status. This translation in the architecture corresponds to change between *Mode* and *ExecutionConfiguration*.

Table 4.6 Overview of the app package for the Cognitive Level

Name	*App Package Descriptions*
AbstractAdaptation	auxiliary class to ease the task of instantiation.
AbstractLearning	auxiliary class to ease the task of instantiation.
AbstractOptimalSearching	auxiliary class to ease the task of instantiation.
AbstractOrganization	auxiliary class to ease the task of instantiation.
Adaptation	represents the adaptation of the architecture.
Evaluation	represents the evaluation of the architecture.
GoalManagement	represents the goal management of the architecture.
Learning	represents the learning of the architecture.
Mode	represents a mode of the architecture.
OptimalSearching	represents the optimal searching of the architecture.
Organization	represents the organization of the architecture.
PerformanceIndex	represents a performance index that will be used by the Evaluation.
TrackValue	represents a set of ideal values that the GoalManagement has to track.

In addition, the *Adaptation* is prepared to run on-line to receive information in real-time and be able to execute one algorithm of self-adaptation to change the model parameters or, if it is necessary, change the model itself. With these two components along with the Mode class, the requirements **RF3** y **RF4** are achieved.

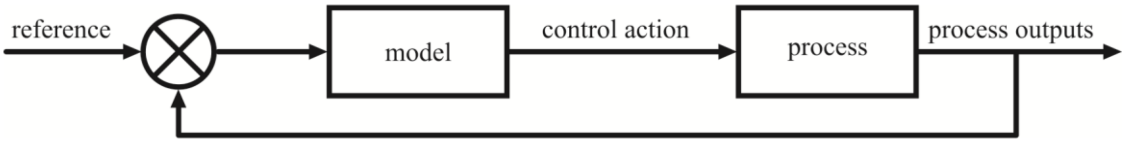


Figure 4.10 General single loop graph

In order to provide the capability of running an optimal searching to find the best parameters for a given mode the *OptimalSearching* interface is provided. It presents all the necessary methods to implements an algorithm of off-line optimization. By this way, the **RF5** is fulfilled.

The Learning interface makes possible the implementation of a learning algorithm in the architecture. This interface presents the methods required to implements an algorithm of on-line learning, thus achieving the **RF6**.

In order to aid in the tasks that these components perform, a component that measures the behavior of the system according to different goals is the *Evaluation*. This component provides methods to facilitate the computation of a performance index that indicates the behavior of the system. Thanks to this index the other components can know if they have to act or not. To provide a common interface to easily use different performance indices in the architecture, the class *PerformanceIndex* is created.

The *GoalManagement* is obtained based on the user goals and performance indices. The main role of this component is to parse the user goals and translates them, using some algorithm, into a combination of one or more performance indices. It is important to note that this is a very complex component and its correct implementation is out of the scope, due to that it is possible that in future versions of the architecture its interface may change. For now, its interface provides methods that permit a user with technical knowledge fix some objectives such as setting the set point to a fixed value. To represent this tracking objectives the *TrackValue* interface is created.

The features of the *GoalManagement* make it possible to achieve the **RF7**. In order to ease the instantiation of the architecture four abstract classes are provided:

- *AbstractAdaptation*
- *AbstractLearning*
- *AbstractOptimalSearching*
- *AbstractOrganization*

For a more complete insight, the whole class diagram can be found in ANNEX III. CLASS DIAGRAMS. Furthermore, a simplified graphical representation of the whole architecture was shown in Figure 4.5. All these considerations are achieved in the **RF1**, i.e., with this architecture design we are capable to control an industrial process.

4.4.3 Controllers

The control strategies summarized in Table 4.7 were selected to perform the proof of concept of the artificial cognitive control. Firstly, the Fuzzy Control Language (FCL) defined in the IEC 61131[309] was selected to implement a single loop controller. For the single loop mode, the inputs selected were the error (ε) and change of this error in time (Δε). It does not mean that other inputs for the controller cannot be selected. The user can define and configure models and control strategies in the artificial cognitive control architecture. The main rationale for using fuzzy and neuro-fuzzy approaches can be briefly summarized as follows.

Table 4.7 Algorithms implemented for each execution mode [82]

Implemented algorithms	*Single loop*	*Forward model*	*Inverse model*
Algorithm	Fuzzy Logic	ANFIS	ANFIS
System	Two-input/single output	Single-input/single output	Single-input/single output
Inputs	Error (ε) and change in Error (Δε)	Feed rate (f_{rate})	Force (F_z)
Outputs	Feed rate (f_{rate})	Force (F_z)	Feed rate (f_{rate})
Membership functions type	Triangular-shaped	Gaussian	Gaussian
Number of membership functions	7	3	3
Inference system	Mamdani	Takagi-Sugeno	Takagi-Sugeno
Number of rules	49	9	9
Defuzzification	Center of area	Weighted average	Weighted average
Iterations	–	100	100
Learning rate	–	0.01	0.01
Training algorithms	–	GENFIS 2	GENFIS 2
Training data set	–	62 samples	62 samples
Validation data set	–	82 samples	82 samples

Neuro-fuzzy systems combine their ability to accurately model any nonlinear function, an excellent learning capacity, and an ability to represent human thought and robustness in the presence of noise and process uncertainty. These characteristics are essential to deal with uncertainties, nonlinearities, and time varying behavior. Some neuro-fuzzy systems cannot be applied to real-time process control, mainly because of the computational cost involved and because they are unable to meet the control system's requirements.

Furthermore, a graphical representation of the different control modes proposed in the cognitive architecture are shown in the Figure 4.11. As it can be seen, in all the cases the references signal (set point) is the force values (F_{ref}) preset to control the process. In the particular case of the single loop mode the feedback signal is the z-component of the force measured ($F_{process}$) in process. Finally, a combination between direct and inverse models is presented in the internal control mode. In this case, the feedback is the algebraically sum of the z-component of the force measured in the process (gross turning) and the z-component of the force estimated (F_{est}) in the mirroring model (fine turning). The error signal is defined as:

$$\varepsilon = F_{ref} - F_{process} \tag{4.8}$$

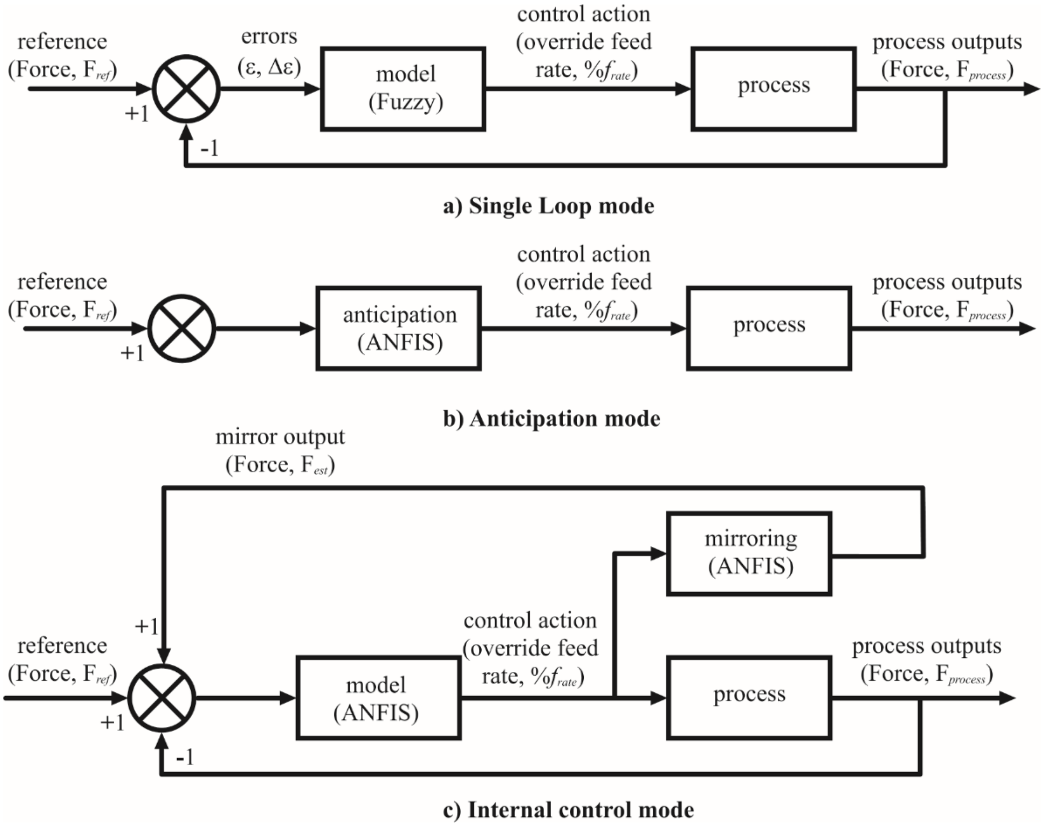


Figure 4.11 Schemes of control modes

4.4.4 Middleware

The scientific community is currently working to connect and to integrate sensors with other devices that will improve factory production. One key issue in the endeavor is the long-distance monitoring and control of complex plants, which requires synergetic strategies that link smart devices and communication technologies with advanced computational methods [310]. The solution chosen for this work employs distributed object computing middleware, which enables common network programming tasks to be automated, regardless of other considerations such as what communication protocols and networks are used to interconnect the distributed objects.

The first middleware option analyzed was Common Object Request Broker Architecture (CORBA). CORBA technology provides a clear opportunity for process monitoring and strategic process control. In real-time CORBA (RT-CORBA) specification, mechanisms and policies are defined to control processor resources, communication resources and memory resources to support the real-time distributed requirements of the application fields [311]. The second option was ZeroC Ice that provides a simple and easy-to-understand communication solution. Yet, despite its simplicity, Ice is flexible enough to accommodate even the most demanding and mission-critical applications. A comparison with other popular distributed computing solutions can be found here [312]. One of the most important features of Ice is its enhanced set of services, such as event distribution, firewall transversal with authentication and filtering, automatic persistence, automatic application deployment and monitoring, and automatic software distribution and patching. All services can be replicated for fault tolerance, so as to avoid the introduction of any single point of failure. The use of these services greatly reduces development time, because they eliminate the need to create distribution infrastructure as part of the application development. The third solution that was explored, Java Remote Method Invocation (RMI), enables the programmer to create distributed Java technology based applications, in which the methods of remote Java objects can be invoked from other Java virtual machines, possibly on different hosts. RMI uses object serialization to marshal and unmarshal parameters and does not truncate types, supporting true object-oriented polymorphism.

Although the design and application of artificial cognitive control architecture based on middleware is essential, because the middleware facilitates communication between different hosts, the design of the architecture and its development is done in a middleware-free manner; independent of the middleware chosen to enable communication. ZeroC Ice was selected for the implementation, on the basis of good results previously reported in the literature, its advantages in relation to CORBA, its versatility and ease of use [313]. Three distributed units were considered in the deployment of the artificial cognitive control architecture, on the basis of the design and specificities of the case study with regard to proprietary software.

- **Cognitive level**: This unit contains the components of the cognitive level: learning and optimization mechanisms, organization logic, execution logic and the application itself. It is expected that this unit will be deployed in a low-cost computational hardware.
- **Executive level**: This unit contains the models that are used in the single loop, direct, inverse and simulation modes of the architecture. Once again, the unit is expected to be deployed in a low-cost computational hardware.
- **Process unit**: A distributed process is needed, because that is one of the objectives, i.e., the control of a physical process. The Ice specification file is programmed with this description and parses it with the slice2java program. This program will generate several auxiliary classes needed by ZeroC Ice for communication over the net. In addition to the common use of Ice, IceGrid, is an important service that enables clients to discover the corresponding servers. Acting in an intermediary role,

IceGrid decouples clients from their servers and is intended to improve the performance and reliability of applications through support for replication, load balancing and automatic fail over. The program needed to execute the IceGrid registry process, IceGrid registry, and the program to run a server, IceGrid node are both provided with the Ice installation package.

4.4.5 Auxiliary tools for the implementation

UML is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard means of visualizing the system design. It was applied to redesign and to implement the artificial cognitive architecture [314]. Java was selected as the programming language, because of its universal portability in different environments. In addition, the Real Time Specification for Java (RTSJ) provides an advantage to extend the simulation results to a full real time environment [315].

Having completed the UML-based design, the next step was the implementation of the architecture using Eclipse, as the integrated development environment (IDE) that facilitates work with Java. We also used SWIG [316], an interface compiler that connects programs written in C and C++ with several languages such as Perl, Python or Java. SWIG permits the re-use of models programmed in C/C++ and performs its tasks through the Java Native Interface (JNI) framework that enables assembly and communication between the Java Virtual Machine (JVM) and programs written in C, C++.

4.4.6 Low-cost computing platforms

Various state-of-the-art low-cost computing frameworks were analyzed, to choose the most suitable one to meet the objectives for deployment in a low-cost computational platform with artificial cognitive control. We reviewed three of the most popular low-cost computing platforms reported in the literature: Raspberry Pi 2 Model B, HummingBoard-i2 and BeagleBone Black. These computing platforms also have forums which share posts from the community of users that help others to make efficient use of the hardware.

Raspberry Pi 2 is a low cost, credit-card sized computer capable of everything expected from a desktop computer, from browsing the Internet and playing high-definition videos, to the use of spreadsheets, word-processing, and gaming [317]. It is a small versatile low-cost device with a 900 MHz Single-Core ARM v7 and 1 GB SDRAM that enables users of all ages to explore computing, and to learn programming languages such as Scratch and Python.

Raspberry Pi 2 can interact with the environment and devices in a wide range of digital maker projects, from gaming machines to open-source voice computing. The HummingBoard-i2 and Raspberry Pi 2 both share a very similar layout and configuration, making transition projects between both of them very easy. The former represents a good choice too, with a 1.0 GHz Dual-Core ARM v7 and 1GB SDRAM, but over twice the cost of Raspberry PI 2.

BeagleBone Black is suitable for users looking for a little more power that Raspberry Pi, an easier set up, easier commercialization, or users who have a need to interface with many external sensors. Its configuration consists of an AM335X 1 GHz ARM Cortex-A8 and 512 MB of DDR3 RAM at a similar cost to Raspberry PI 2.

The main criteria for selecting the low-cost computing platform were the world-wide support for a low-cost computing platform to facilitate implementation and to solve deployment problems and, less importantly, the cost of the platform. A large user community provides ample support for trouble shooting when using the platform. We finally chose Raspberry Pi 2 Model B, partly because it is the most popular platform with an active user community.

4.5 Application of the artificial cognitive control architecture in micromachining. Final validation

4.5.1 Architecture setup for micromachining processes

In order to show a real example to validate the architecture, a particular instantiation have been carried out. The instantiation is a simplified version of the architecture which will be distributed in order to perform monitoring and control tasks. It shows the potential of the architecture as well as its flexibility and scalability. The principals machine features used for the implementation and configuration of the artificial cognitive architecture are shown in Table 4.8.

In order to achieve this task some general steps to instantiate the architecture must be followed:

1. Class implementation in Java: it is recommended to extend from the abstract classes presented in the architecture. The number of classes or their complex depends of the tasks it wants to realize. In this particular case, the different steps to follow:
   (a) Implementation of the inference models using the technology SWIG to take advantage of the models implemented in C++ by GAHME research group.
   (b) Implementation of a self-optimization algorithm based on the cross-entropy method described in the Chapter 3 .
   (c) Implementation of an on-line learning algorithm based on the Q-learning algorithm (see section 4.2.1).

2. Implementation of the Ice classes needed to make the units of our architecture distributed. Particularly, it has followed the delegation pattern to make this task easier.
3. Deployment of the cognitive unit in the Raspberry Pi 2 ($P_{\text{i\_cog}}$).
4. Deployment of the executive unit in the Raspberry Pi 2 ($P_{\text{i\_exe}}$).
5. Deployment of the process unit in the process host.
6. Adjusting and tuning models' parameters with simulation.
7. Adjusting and tuning models' parameters with experimentation and tests in a real manufacturing environment.

Table 4.8 Machines features

Type	**Process host**
Operating System	Microsoft Windows XP Professional (version 2002), Service Pack 3
CPU	Intel(R) Pentium(R) 4 CPU 3.20Ghz 3.19Ghz
RAM	2GB
Type	**Raspberry Pi 2 model B**
Operating System	Raspbian
CPU	quad-core ARM Cortex A7, 900 MHz
RAM	1GB
Type	**Registry host**
Operating System	Ubuntu 12.04 LTS
CPU	Intel(R) Core(TMT) 2 CPU 6400, 2.13Ghz
RAM	2GB

The deployment of all components in the architecture is shown in Figure 4.12 a) and b). Raspberry Pi 2 (Pi 1) and Raspberry Pi 2 (Pi 2) run the cognitive and executive part of the architecture, respectively.

Process host has the mission of retrieving the process outputs from the KERN Evo machine (see section 2.2.1) and sending them to the architecture via ZeroC Ice, as well as receiving the action control from the architecture and sending it to the KERN Evo machine. As shown in the picture, communication between Process host and KERN Evo is done over the Ethernet. The Icegrid registry program is permanently running on the Registry host to enable the different hosts to identify each other. Finally, the Client host can use the developed graphical user interface to interact with the different components.

In order to simplify the implementation, it has taken several considerations that it is necessary to be explained. For the sake of clarity, we are grouping these considerations in different parts:

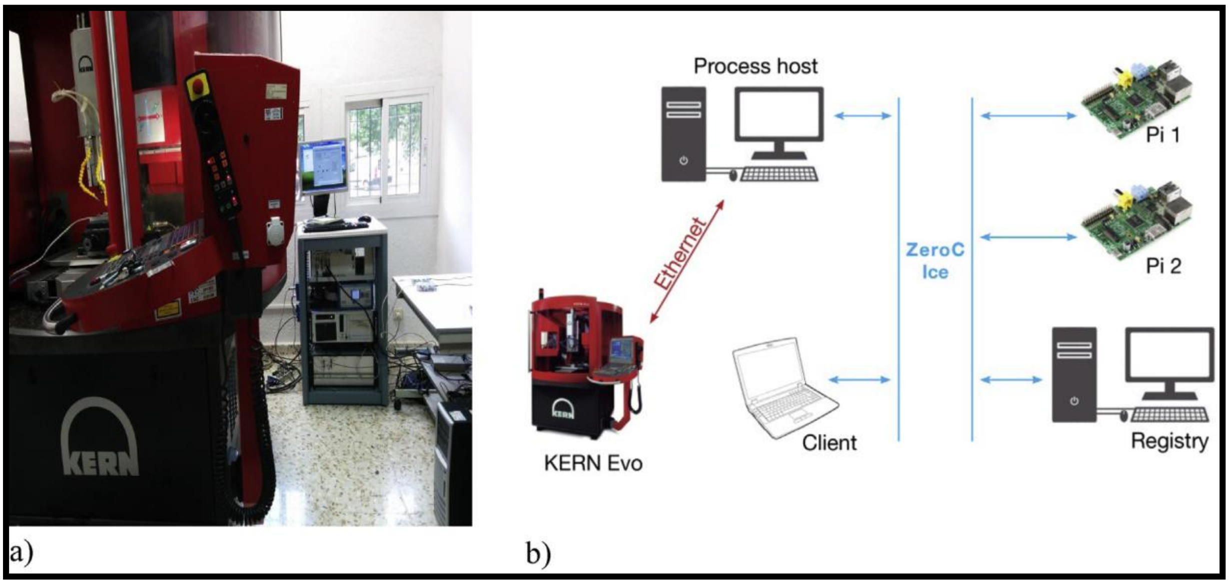


Figure 4.12 a) Overall view of the industrial setup, b) Schematic diagram of the architecture of artificial cognitive control

4.5.1.1 Models

In this instantiation, it will use several models. Three of them are coded in C/C++. For this reason, the SWIG tool was used to be able to handle them in Java. The models that are exported from C/C++ are a fuzzy controller, an ANFIS forward model and an ANFIS inverse model. In this manner, the implementation of the models simply used them in a class that extends *AbstractModel*, an example of this can be seen in Figure IV.4. The others model was described in Chapter 2 .

4.5.1.2 Process

Two processes are implemented for testing scenarios, one for the simulation case and other for the real case. The connection between the machine and the real process was done via Dynamic Data Exchange (DDE) and some steps are needed to configure the machine before start it. A simplified snippet of code can be found in Figure IV.5.

4.5.1.3 Modes

Three modes of the MSCM were used. A simplified graph-shape version of these modes are depicted in Figure 4.11. The implementations are shown in the Figure IV.1; Figure IV.2 and Figure IV.3, respectively.

4.5.1.4 Evaluation

In this instantiation only one performance index was considered, namely the relative Mean Squared Error (MSE). We choose this index because is a common criteria measurement used in both control theory and manufacturing processes.

4.5.1.5 Learning, Optimal Searching and Organization

Reinforcement learning mechanisms are essential to emulate actual functionalities of cognitive systems. The research is focused on a suitable reinforcement learning strategy to improve the overall performance of the artificial cognitive architecture. The design of a self-learning mechanism can bring advantages with regard to other biologically-inspired control architectures. The deployment of a self-learning capability can facilitate further designs and implementation of other self-capabilities such as self-organization, self-optimization and self-adaptation.

In order to provide the instantiation with an on-line learning algorithm it has implemented the Q-learning algorithm described in section 4.2.1. The off-line optimization algorithm based on the cross-entropy method is presented in Chapter 3 . Finally, the organization algorithm is described in section 4.2.2.

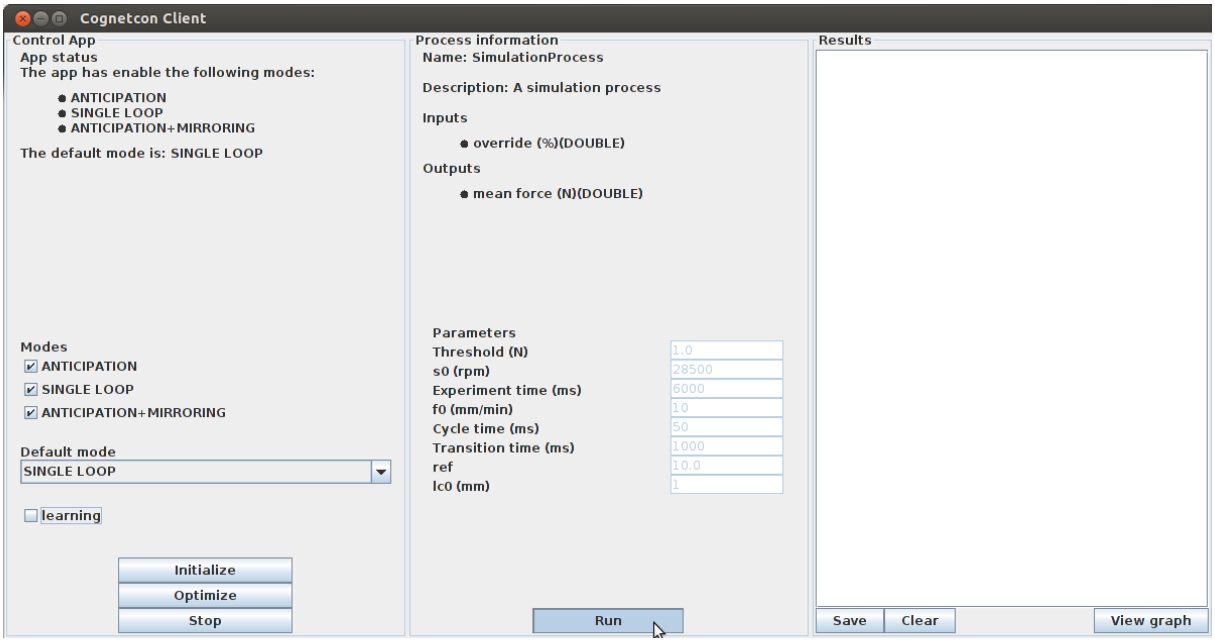
4.5.1.6 Execution Management

It has done an implementation of *ExecutionManagement* that maintains each execution configuration running in a different thread. For each configuration the management gathers the inputs of each model, executes it and, if the configuration is the main configuration, the control action is sent to the process directly.

4.5.1.7 Application

Finally, the implementation of the Application is simply an organizer that receives the process's output and distributes it with the reference to the adaptation, organization and learning components. An interesting detail to be noted is that the process's output and the reference is distributed between the components if and only if the process's output exceeds a threshold, given by one process's parameter.

It is important to note that this instantiation has used, when it is possible, the abstract classes that are provided with the library and has coded in order to be able to run correctly in both non-distributed and distributed environments. In order to ease the implementation in the distributed environment we have used the ZeroC - Ice technology. To use this technology, it has written the specification of the three units we want to deploy remotely and generate the Ice-classes. To use IceGrid it is necessary to create some configuration files in order to set up the main the connection with the registry machine and to set up some needed parameters. To conclude the user interface of the artificial cognitive control architecture was shown in the Figure 4.13.



App status
The app has enable the following modes:

- ANTICIPATION
- SINGLE LOOP
- ANTICIPATION+MIRRORING

The default mode is: SINGLE LOOP

Modes
☑ ANTICIPATION
☑ SINGLE LOOP
☑ ANTICIPATION+MIRRORING

Default mode
SINGLE LOOP

□ learning

Name: SimulationProcess

Description: A simulation process

Inputs

- override (%)(DOUBLE)

Outputs

- mean force (N)(DOUBLE)

Parameters	
Threshold (N)	1.0
s0 (rpm)	28500
Experiment time (ms)	6000
f0 (mm/min)	10
Cycle time (ms)	50
Transition time (ms)	1000
ref	10.0
lc0 (mm)	1

Figure 4.13 User interface of the artificial cognitive control architecture

4.5.2 Validation on a micromachining process

Due to the small dimensions involved in the micro-drilling processes, the control is very difficult to carry out online. For this kind of processes, the use of online indirect monitoring is particularly important. Measurable process signals such as forces, vibration, acoustic emission and motor current have been often used for this purpose. Online quality control systems can provide real-time information, which can be supplied as a feedback to CNC for online adjusting cutting parameters. These kind of systems has been proposed for conventional machining. Monitoring and control systems have been also proposed in order to adaptively modify the cutting parameters in real-time for guaranteeing the geometric quality [231].

In the experimental study, the instantiation was tested in an industrial environment. All cutting operations were done on a Kern Evo Ultra-Precision Machine Centre, equipped with a Heidenhain iTNC540 CNC. The experimental platform included a cutting force sensor on three axes, two vibration sensors for y, z axes and a laser sensor for measuring variations in tool length and radius. The measurement of cutting force signals was done with a multi-component dynamometer (see section 2.2).

This experimental setup is reported in [82, 83]. Several experiments were performed to validate the design and implementation of the artificial cognitive architecture. The experiment consisted of single and consecutive (10 holes) 0.5 mm drilling operations at a spindle speed of 10,000 rpm and a feed rate of 100 mm/min. First, in order to assess each control mode (i.e., single loop, inverse control and internal model control) only one drilling action was carried out.

In order to validate the architecture 437 experiments were run (see Figure 4.14). The workpiece material was a tungsten-aluminum-vanadium alloy (Ti_6Al_4V) widely used in biomedical applications due to its excellent biocompatibility. In all the cases, a 0.5-diameter micro-drill was used. Figure 4.14 shows experiment matrix developed in the experimental test.

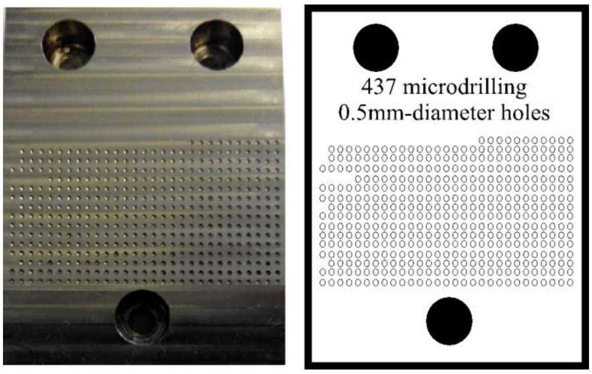


Figure 4.14 Experiment matrix made with 0.5mm-diameter micro-drills [83]

Figure 4.15 a) depicts the results of the simulation of each control mode. In order to carry out this simulation, the model of the micro-drilling process represented in Eq. (4.9) is considered as well the influence of noise represented by Eq. (4.9). Based on the technical knowledge of the process an input and output system is considered using the following variables: input is the feed rate (f_{rate}) and output, the cutting force (F_z). For the study an approximate representation of the process behavior was used. The linear model represented via difference equation is expressed accordingly:

$$\begin{aligned} F_z(k) = a_1 \bullet f_{rate}(k) + a_2 \bullet f_{rate}(k-1) + a_3 \bullet f_{rate}(k-2) + a_4 \bullet f_{rate}(k-3)... \\ - b_1 \bullet F_z(k) + b_2 \bullet F_z(k-1) + b_3 \bullet F_z(k-2) \end{aligned} \tag{4.9}$$

where f_{rate}(k) is the feed rate and F_z(k) is the cutting force at k-instant. The coefficients of the difference equation are $a_1 = 0.0043$, $a_2 = 0.0246$, $a_3 = 0.0086$, $a_4 = 0.0000217$, $b_1 = -2.447$, $b_2 = 1.993$ and $b_3 = -0.541$.

This model roughly describes the dynamic behavior of the drilling process and it has been verified experimentally. However, model parameters depend on the workpiece material, cutting conditions, and tool wear. Therefore, model coefficients are time variant and variables of cutting conditions, workpiece material and tool wear. A disturbance d(t) (Eq. (4.10)) or noise input is included, in order to better replicate a real industrial process:

$$d(t) = Ap \bullet (\sin(2\overline{\omega}t) + \sin(3\overline{\omega}t) + \sin(4\overline{\omega}t) + \sin(5\overline{\omega}t)) \tag{4.10}$$

Where $\overline{\omega}$ = 7.61 rad/s. This frequency corresponds to the greatest frequency of the poles of the third-order system model of the drilling process given in Eq. (4.9). The amplitude of the disturbance is Ap = 10 (about 10% of additive noise).

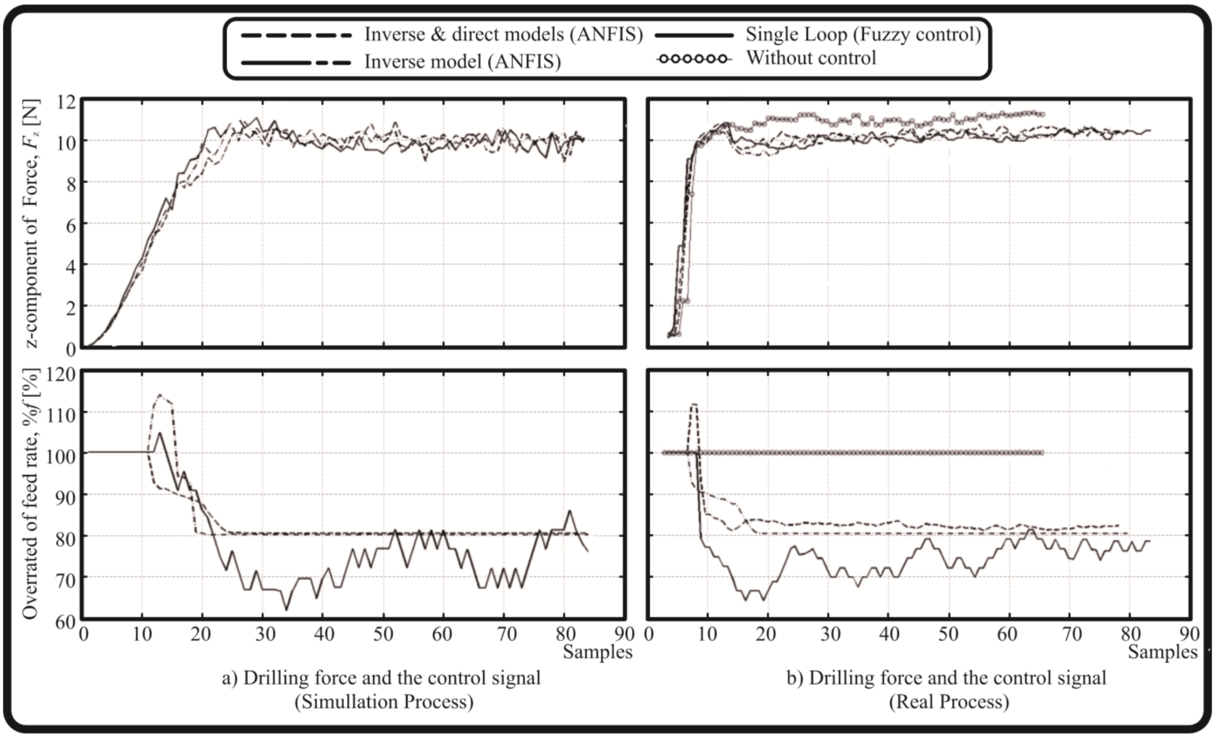


Figure 4.15 Behavior of the a) drilling force and the control signal (overrated feed rate) for each execution mode isolated on the basis of the simulation; b) drilling force and the control signal for each execution mode isolated from real time experiments

Once the force model was defined the optimal searching algorithms (using SMOCE) is executed to fix the initial parameters for the controllers (see Table 4.9). For the single loop model, two gains are introduced for the error (*KE*) and the change of the error (*KDE*) and a gain (K_{out}) for the output of the controller. In the case of the forward and inverse models, the first one, the input ($\%f_{rate}$) is multiplied for the gain K_{for} and the output (F_{ref}) for the output gain K_{for_out}; in the second occurs the opposite, being the input (F_{ref}) is multiplied for the gain K_{inv} and the output (%*f*) for the output gain K_{inv_out}. Furthermore, the initial process configuration for the override feed rate and the set point (reference force) and the learning rate, discount factor, rewards and penalties are also defined in Table 4.9.

Figure 4.15 b) shows the behavior of drilling force in real time for three modes (single loop, inverse control and internal model control) of operations as well as the behavior of the drilling force without control (i.e., at constant feed rate). Each control mode is running isolated in each micro-drilling operation. For the sake of space, the experimental results depicted in Figure 4.15 b) are running after the optimal setting of parameters on the basis of the cross entropy method. The dynamic response for the three operating modes is appropriate after running the initialization of the artificial cognitive control system. The optimal setting of parameters for each control modes is performed using the rough model of the process (Eq.(4.9)) and the mean square error performance index.

Table 4.9 Controller and learning configuration after the optimal searching

Process			
Parameters	**Default values**	**Range or types**	
F_{ref}	10N	---	
f	100%	10%	150%
Controller models			
Mode	Single loop		
KE	9.3783	-30	30
KDE	0.1181	-30	30
K_{out}	100	10	150
Mode	Inverse model		
K_{inv}	1.2480	0	10
K_{inv_out}	35.80	10	150
	Forward model		
K_{for}	80	0	100
K_{for_out}	7	5	20
Learning			
α	0.1	---	
γ	0.85	---	
reward	0.9 (if the actual mode obtains the best performance index)		
penalty	-0.1 (if the actual mode receives a performance index higher than 0.7)		

Table 4.10 shows a comparative study of the single loop operating mode with the inverse model (i.e., inverse control) and the inverse and forward models (i.e., internal model control). The integral of square error (*ISE*), the average of absolute error (*AAE*), the mean square error (*MSE*) and the overshoot (*Ovt*) are used. The error performance indices of the single loop controllers are better than the other controllers whereas the inverse control (inverse model) yields the better overshoot. It is important to remark that the micro-drilling process is non-linear and time variant process, and therefore the performance of each control mode may deteriorate due to nonlinearities, uncertainty and time-variant behavior of the process.

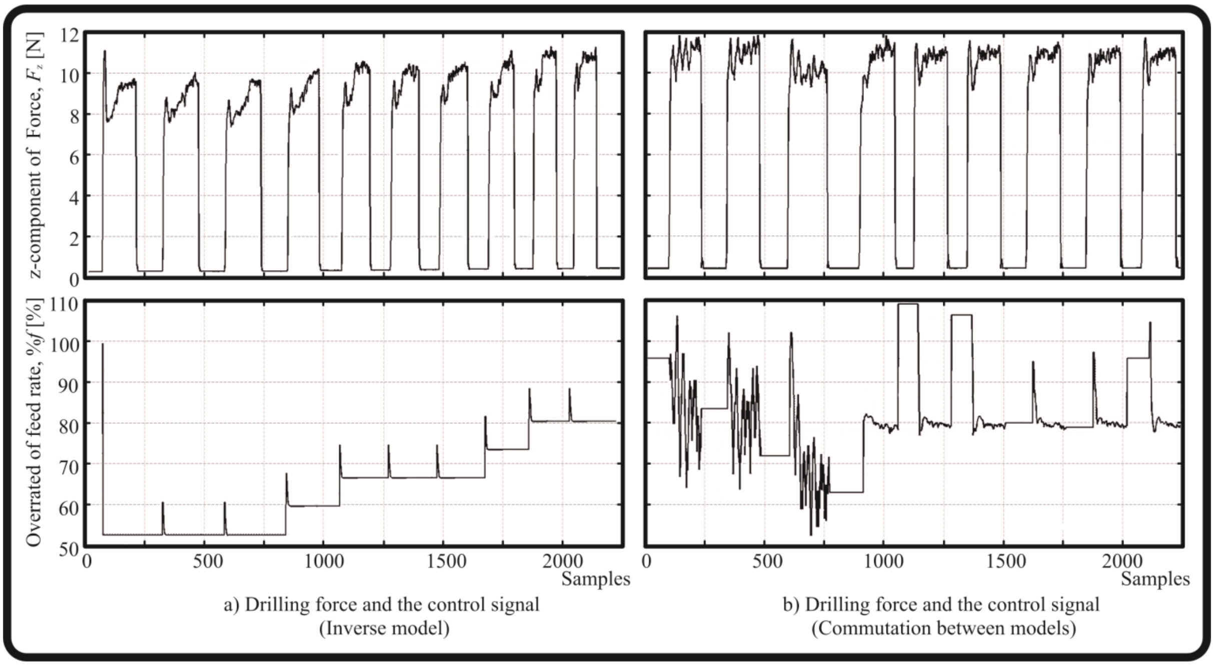
Figure 4.16 a) shows the behavior of the drilling force when the learning is activated for the inverse model (i.e., inverse control mode). The drilling of 10 holes is performed, in order to show the influence of reinforcement learning on improving the performance of the inverse model. Initially the response is very poor and the system cannot reach the set point. The behavior of the drilling force is quite good from the 7th hole due to reinforcement learning.

Table 4.10 Performance indices for real time experiments

Controllers	*ISE*	*AAE*	*MSE*	*Ovt.(%)*
CNC alone (without control)	559.79	1207.39	25.66	12.15
Single loop (fuzzy control)	338.45	628.23	17.62	7.16
Inverse model (inverse control)	380.40	706.48	19.13	5.71
Inverse and forward models (IMC control)	418.81	742.48	20.26	6.74

The single loop mode (i.e., the fuzzy control) is functioning in the first three holes. After that, the poor performance index motivates the change to the internal model control where direct and inverse models are activated (see Figure 4.16 b)). This is a clear case study where the single loop is deteriorating due to the influence of disturbance such as tool wear and the artificial cognitive control tries to find an adequate solution by changing the execution mode. The dynamic response and the performance index are then better in the new execution mode and the system remains in this mode.

There are some issues to be analyzed. The offsets in the dynamic response (see Figure 4.16) is the result of the influence of air for tool refrigeration on measured force. This negative effect is not easy to be removed because it is not a constant value. The second issue is the parametrization of the threshold in the performance index to change from one mode to another one, which depends mainly on the application. The changing between the different models can be appreciated during the experiment running in the Figure 4.17.



a) Drilling force and the control signal (Inverse model)

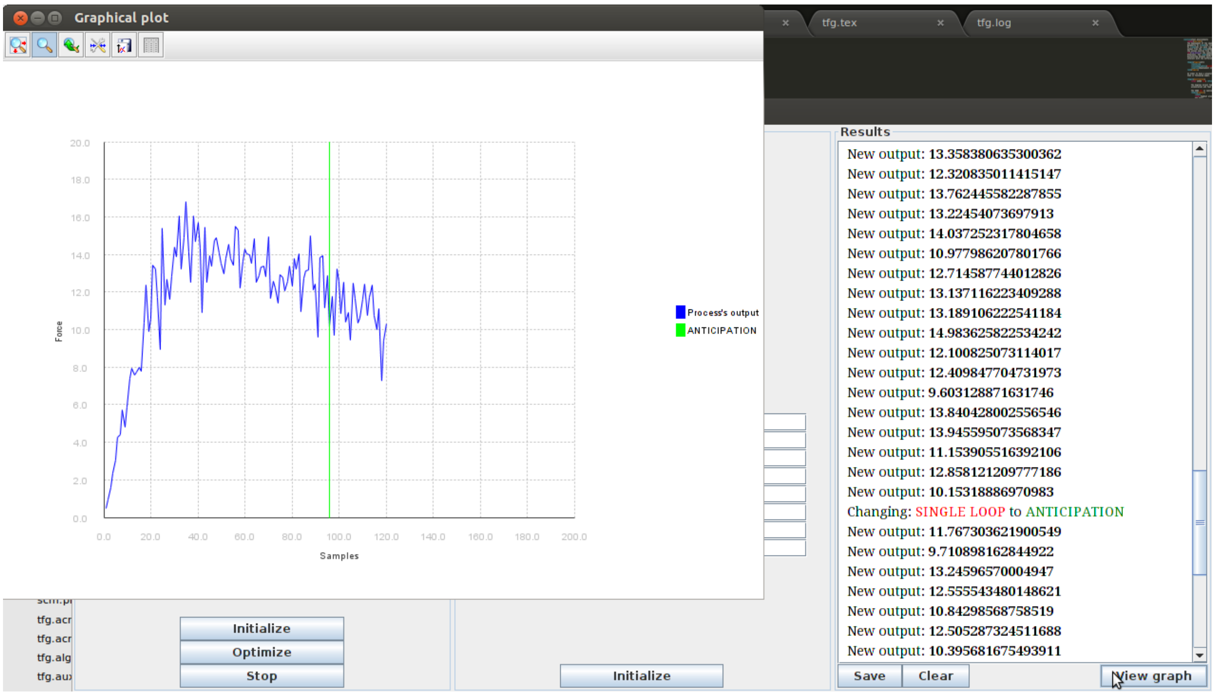
b) Drilling force and the control signal (Commutation between models)

Figure 4.16 a) Reinforcement learning for the inverse model and b) Commutation between models in 0.5mm real-time micro-drilling

Finally, in order to validate the performance of self-learning strategy in the frame of the cognitive control architecture, a set of micro-drilling experiments was run [83]. In the particular study case, the real-time constraints are the short cutting time and the delay in the communications. The cutting time per hole is about 5s with the considered cutting conditions. For this reason, it is difficult to control the process in real-time and exist a short delay between the machine center and the cognitive architecture.

Firstly, the monitoring of the process without the cognitive control architecture was performed when drilling three holes. Later, the anticipation control and the single loop control modes were executed without the learning algorithm. The set point for the force control was 10N. Finally, arrays of ten drills were used to assess the learning algorithm.

The results of these experiments are depicted in Figure 4.18. Figure 4.18 a) shows the behavior of the force without control, with an evident increase beyond 10N. Despite the design and tuning of the control modes, the initial results were not encouraged due to the influence of uncertainty, noise, time-variant and nonlinear behavior of the micro-drilling process on deteriorating the real-time performance. In one scenario, with the anticipation control mode (see Figure 4.18 b)), the control signal cannot reach the set point meanwhile in the other, with the single loop mode (see Figure 4.18 c)) the behavior of the force surpasses for a long time the set point even beyond the transient state.



```
New output: 13.358380635300362
New output: 12.320835011415147
New output: 13.762445582287855
New output: 13.22454073697913
New output: 14.037252317804658
New output: 10.977986207801766
New output: 12.714587744012826
New output: 13.137116223409288
New output: 13.189106222541184
New output: 14.983625822534242
New output: 12.100825073114017
New output: 12.409847704731973
New output: 9.603128871631746
New output: 13.840428002556546
New output: 13.945595073568347
New output: 11.153905516392106
New output: 12.858121209777186
New output: 10.15318886970983
Changing: SINGLE LOOP to ANTICIPATION
New output: 11.767303621900549
New output: 9.710898162844922
New output: 13.24596570004947
New output: 12.555543480148621
New output: 10.84298568758519
New output: 12.505287324511688
New output: 10.395681675493911
```

Figure 4.17 Commutation between the different modes

On the contrary, if the learning algorithm is activated the control signal is modified per iteration and the overall performance of the artificial cognitive control is improved. Different figures of merit or performance indices are calculated and depicted in Table 4.11. The results corroborate how the Q-learning algorithm contributes to a better response and reduces the error-based performance indices.

Table 4.11 Performance indices for real time experiments

Modes	*Iterations*	*ISE*	*AAE*	*MSE*	*Ovt.(%)*
Anticipation Mode	1	601.53	1869.50	25.03	-3.27
	2	503.04	1513.61	22.89	-3.44
Anticipation Mode + Learning	5	359.10	854.78	21.19	5.00
	10	257.20	599.35	19.17	6.28
Single Loop Mode	1	283.79	714.09	23.14	14.04
	2	251.76	667.47	21.79	13.90
Single Loop Mode + Learning	5	173.48	450.72	17.29	12.34
	10	127.62	433.52	14.71	11.43

Figure 4.18 shows how the anticipation mode + learning (see Figure 4.18 d)) and single loop + learning (see Figure 4.18 e)) keep the cutting force value over the force point set (10N). This result corroborates the suitability of the learning algorithm in the cognitive architecture and the potential for industrial processes.

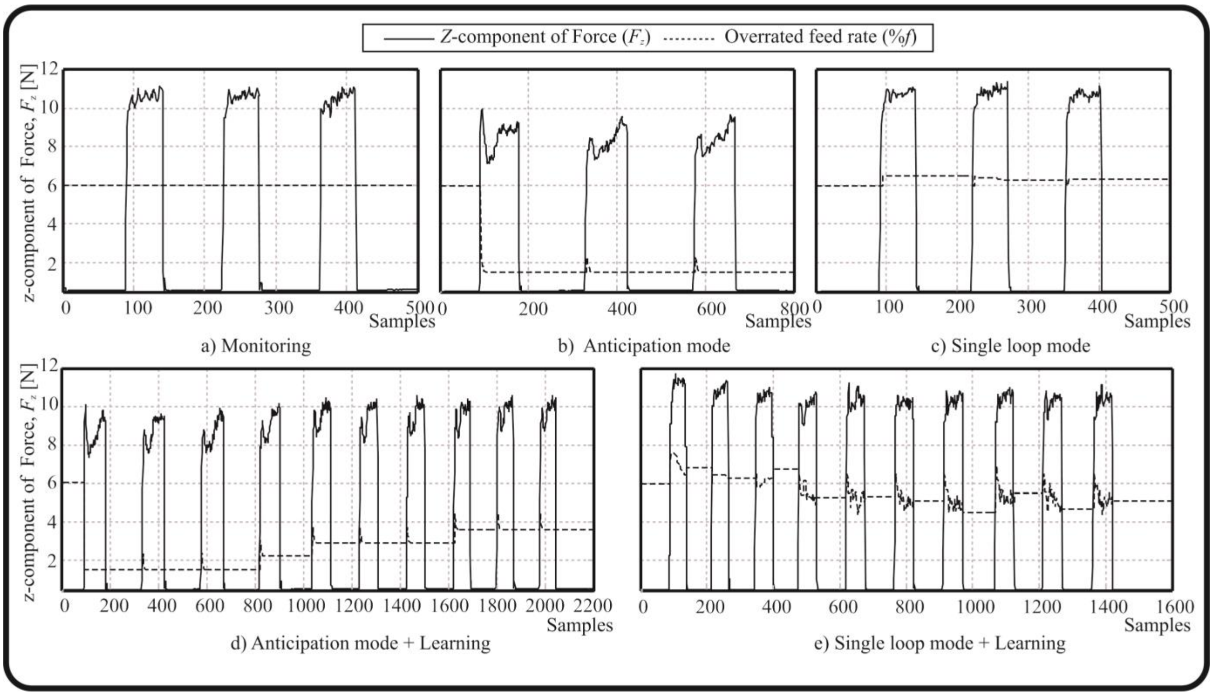


Figure 4.18 Behavior of the force for the single loop and anticipation mode with and without learning

4.6 Conclusions

The shared circuit model approach is enriched and improved using the state-of-the-art on this field. Moreover, relevant reports on this issue as well as the contributions of the authors are also outlined. The MSCM is postulated from the viewpoint of System Theory and Computer Science. Another contribution of the modified SCM is to enable more efficient and faster manufacturing through cooperative, self-organized, self-optimized behavior by process control systems. This work leads to a progress on artificial intelligent systems from the imitation of self-human mind evolution. From a theoretical point of view, MSCM provides an alternative conceptual framework to perform control tasks in an efficient fashion that characterizes human cognitive processes.

The design and implementation of an artificial cognitive control system in a low-cost computing platform with self-optimization and self-learning capabilities presented in this chapter involves a transition from module-based implementation to an operating mode-based implementation. This change in the artificial cognitive architecture provides a new group of functionalities for the reconfiguration and the self-adaptability, changing the connection between the different elements (modes and models) to achieve a goal.

In addition, self-capabilities, represented by the optimization procedure using cross-entropy algorithm and the online learning mechanism on the basis of a Q-learning algorithm, were designed and implemented in the architecture. Furthermore, different classes were developed that provide this instantiation with the ability to run in a distributed manner. The overall assessment of the instantiation was performed by simulating, and a real manufacturing environment either with very promising results. Beyond the case-study on force control for micro-drilling processes and the results that have been presented, the artificial cognitive control architecture built on a low-cost platform hardware has demonstrated the suitability of the implementation in an industrial setup. The functional and nonfunctional requirements are fully satisfied by means of a simple instantiation configured with middleware.

This research work has provided an important starting point to address the main challenge of an artificial cognitive approach embedded in low-cost hardware industrial computing on the basis of low-cost hardware.

Conclusions and future work of the Doctoral Thesis are presented in the next chapter. Fully aware of the difficulties to transfer of knowledge the artificial cognitive control architecture at industrial levels, the subsequent objectives in the near future are as follows:

(i) to introduce actual concepts such as cloud computing, big data and cyber physical system to create a global mode (plant level);

(ii) to add further models to the repository for more complex tests, and

(iii) to improve the way in which the instantiation can execute the components for improved performance on this new low-cost computing platform.

CONCLUSIONS

This Doctoral Thesis is based on a scientific methodology that combine theoretical and experimental research methods, dealing with the design, development and implementation of an artificial cognitive architecture for monitoring and control of complex processes. The roots of proposed bio-inspired architecture are the shared circuits model for emulating socio-cognitive skills. The use of computational intelligence techniques for modeling, optimization and control in the different operating modes is key issue. The design and implementation of a modular and reconfigurable architecture is validated in several micromachining operations. During this multidisciplinary and interdisciplinary research, many scientific and technical challenges in different areas of knowledge are addressed, from mechanical engineering and micro-manufacturing to control and industrial informatics.

The proposed artificial cognitive architecture has taken into account computational intelligence methods and paradigms from the Computer Sciences to provide a computationally efficient solution. A wide range of contemporary methods for signal processing, filtering, feature extraction, pattern identifications, modeling, optimization and learning has been considered to develop the proposed artificial cognitive architecture.

Furthermore, the reconfiguration and the portability capability of the architecture has been assessed in low-cost computing platform using learning and optimization strategies in monitoring and control tasks. In addition, two micromachining processes have been selected to validate the cognitive and executive levels of the architecture. The selection of micro-manufacturing processes as case studies is justified for the great impact on the economy, production and services this century.

Following, a summary of the main results reported in each chapter are presented:

Review of the state of the art. Challenges and opportunities

Firstly, the review of the state of the art is concentrated on methods and techniques in the conjunction of Artificial Intelligence, Systems Engineering and Computer Sciences. A special attention is focused on the theoretical foundations for modeling, optimization and machine learning methods prioritizing control and monitoring of industrial applications. In this sense, the review has focused on soft-computing techniques, analyzing current trend for monitoring and control of micro-scale manufacturing processes. This review is not only necessary to know the evolution of these techniques, but also to identify good candidates able to provide better performance in architectures for monitoring and control complex physical processes.

Due to the multidisciplinary, interdisciplinary and heterogeneous nature of this doctoral thesis, the state of the art has been focused in order to generate scientific and technical contributions in each and every one of the fields of application, areas of knowledge and topics addressed during the investigation. In order to achieve the main objective, as it was described before, three main areas (modeling, optimization and learning) are specifically considered during the development of the artificial cognitive architecture. The most relevant results in each knowledge area are outlined.

Signal processing and modeling.

From the scientific viewpoint, the simulation and experimental study demonstrates that the combination of WT and ANN provides the best tradeoff between appropriate correlation and good generalization capability, reported in [193]. However, from the technical standpoint, the implementation of such a monitoring system at industrial scale has severe constraints. For instance, the computational cost (computation cost, signal processing time, etc.) is lower for the wavelets and ANFIS combination. This result means that WT + ANFIS is a good candidate for final implementation of the monitoring system at industrial scale. In all the cases a statistical significance of 95% was considered. An additional advantage of the wavelet in the feature extraction stage is that it offers the possibility of filtering the signal in the various frequency bands and of separating the interesting frequency from the noise frequency, which considerably reduces the area for searching in the subsequent stage, achieving a better final prediction.

Another interesting conclusion is the high correlation only belongs to the signal segments in the tool tip entrance and the backward feed with holes quality error in micro-drilling processes, shown in [231]. This can be explained if the damage is mainly produced at the tool tip entrance and, later on, this damage affects the chip exit during the backward motion. In addition, the frequency bands of the higher correlated features, corresponds to the natural frequencies of the systems. The greater correlation was found in the backward feed motion of the tool, equal to 0.8238, meaning that the model as fitted explains the 82% of the variability in the observed data with a 99% of the confidence level. It would be explained by the friction between the tool, the chip and the holes' profile. Finally, the frequency bands included in the model are also noteworthy. The first one is the frequency band from 3906 to 4297 Hz, which has a direct relationship with the hole quality error. On the contrary, the lower frequency band (up to 391 Hz) has an inverse relationship with the holes' quality error.

A two-step monitoring systems for detecting the occurrence of run out in microdilling processes of Ti_6Al_4V and $W_{78}Cu_{22}$ alloys was presented in [254]. It uses a FFT transform (first step) for extracting features from the online measured force data and a multilayer perceptron neural network (second step) for predicting the process condition from the previously obtained features. The monitoring system was able to identify more than 70 % of the run out conditions, with less than 10 % of false run out detections. Due to the intolerance of micro-drills to run out, the proposed approach can yield considerable gains in tool life and productivity. The adjusted model had a coefficient of determination, equal to 0.9586, indicating that the model as fitted explains 95%of the variability in the dependent variable. The standard error of estimate is 0.0899, and the mean absolute error is 0.0266. In order to test the generalization capabilities of the model, the obtained predictions for the training and validation sets were compared. The difference between the mean values of the predictions of the training and validation sets is neglected. In addition, the mean predicted values of both sets are very similar to the observed values. Nevertheless, the spread of the predictions for the training set is remarkably less than the spread for the validation set. Consequently, the predictions of the model, under different conditions, are not as reliable as those obtained for data used in the training process.

An artificial neural network model based on mechanical and thermal properties for five materials was reported in [318]. The available experimental data was randomly divided into a training set containing 1 973 samples (corresponding to the 80%) and a validation set with the remaining 493 samples. The training process was carried out on the basis of backpropagation algorithm, with adaptive learning rate and momentum. The following training parameters were selected: initial learning rate, 0.001; ratio to increase learning rate, 1.05; ratio to decrease learning rate, 0.70; momentum constant, 0.9; and minimum performance gradient, 10^{-10}. The stop condition was established after the 300 000 epochs. The obtained neural model has a correlation coefficient of 0.8969, so, it explains the 89% of the variability in the response variable. The standard error of the estimations was 1.084 and the mean absolute error was 0.7340. In order to assess the generalization capability of the model, the residuals coming from the training and validation sets were analyzed. It motivated to carry the comparison between mean values and standard deviations of both sets using t-Student and an F tests, respectively. These tests yielded associated probability values of 0.6125 and 0.1832, respectively. Therefore, the null hypothesis (both means are equal and both standard deviations are equal) cannot be rejected with at the 95% confidence level. It can be concluded that both residuals sets come from the same distribution and the model has a good generalization capability.

Self-Optimization.

Thirteen non-dominated solutions for a two objective optimization problem were reported in [318]. The minimization of the unit machining time (inverse of the material removal rate, M), and the thrust force were the targets based on the optimal set of cutting parameters in the micro-drilling process. The selection of the most convenient solution depends on the specific conditions of the productions. For example, if the productivity is the most important issue, solution No. 13 should be selected, as it produces the lower unit machining time in spite of the relatively high thrust force value. On the contrary, if the tool cost is determinant, solution No. 1 should be chosen, because its lowest force minimizes the risk of tool breakage. Finally, the other solutions may be suitable for other halfway situations. The Pareto's front obtained demonstrated the useful of the heuristic technique to solve industrial problems, in this particular case, in micro-manufacturing of especial alloys, using this studies, the operator can choose a fast decision in function of accelerate the production or decrease the tool spending.

The non-dominated sorting genetic algorithm was selected in [255] to approach multi-objective optimization problem in a micro-milling process. Two objective functions were defined, the first function, surface roughness and the second one, the unit machining time. As in the previous example, the selection of the most convenient solution depends on the specific conditions of the productions. Two possible industrial decisions were analyzed: productivity was the most important issue in the first one and the surface roughness requirement was penalized and, in the second, the best surface roughness parameters were obtained with lower productivity.

Furthermore, the behavior of the vibration signal, *V*z, for the two points was described. The cutting parameter values for first scenario were a cutting speed of 125.2 m/min, a feed rate of 1.9l m/tooth, and an axial deep of cut of 0.1 mm. The range of the z-axis vibration of between 0.12g and 0.26g, with a mean of between 0.14g and 0.20g. For the second scenario, the cutting parameter values were a cutting speed of 71.4 m/min, a feed rate of 2.6 lm/tooth and an axial depth of cut of 0.1 mm with a range of the z-axis vibration at between 0.08g and 0.18g, with a mean of between 0.09g and 0.16g.

The surface roughness prediction was based on the initial cutting parameters, an on-line feedback of the signal during the cutting process is necessary. The proposed real-time system, using the ANFIS models, is capable of predicting surface roughness as a function of the z-axis vibration captured during the cutting process. This system can provide a surface roughness estimate with an error reading of 9.5% in an industrial setup, depending on the cutting parameters.

The core of the contribution is the modified Multi-Objective Cross-Entropy (SMOCE) method that was reported in [274]. First, a comparison using MOP, ZDT, WFG test suits with other multi-objective algorithms was done. In general terms, the proposed method provides a better solution (except for MOP1 and MOP2), but they are worse than those provided by the other evolutionary approaches. Nevertheless, the generational distance values are very low in all the problems; therefore, the solutions are acceptable for many applications.

Once defined the improved cross-entropy method for multi-objective optimization, a micro-drilling process was selected to validate the implementation in a real industrial scenario. For the optimization, two process representative models of thrust force and vibration on the plane perpendicular to the drilling axis were obtained using a well-established method inspired in Artificial Intelligence. Below, two objectives were simultaneously considered: the drilling time and the amplitude of the vibrations on the plane perpendicular to the drilling axis. Furthermore, certain constraints must be fulfilled, defining the thrust force lower than the allowable thrust force is pre-established to avoid buckling-based breakage of the tool.

Altogether, 25 runs were performed to increase the reliability of the optimization results. Three zones can be easily noted (see Figure 3.12): Zone I shown cutting parameters that will obtain high quality holes, but over excessive operating times; Zone II contained the best combination of solutions in both objectives, so they are the most convenient parameters for most of the operations, giving reasonably good hole quality within low drilling times and Zone III involved solutions with shorter execution times, but with higher vibration amplitudes. Consequently, these are only solutions for holes where quality is not an important requirement. The Pareto front solutions, in all the zones, cover almost all of the cutting speed intervals and steps. On the contrary, the values of the feed rates are higher than source in all of the zones. In all the cases, a 95% confidence intervals for the execution time and some characteristics and quality metrics of the Pareto fronts that were obtained.

Self-learning

In order to validate the performance of self-learning strategy in the frame of the cognitive architecture, a group of micro-drilling experiments were executed. The study was reported in [83]. Firstly, the monitoring of the process without control was performed. Later, the anticipation control and single loop control modes were executed without the learning algorithm. The set point for the force control was 10N. Finally, arrays of ten drills were used to assess the learning algorithm.

Despite the appropriate design of the control modes, the initial results were not too good due to the influence of uncertainty, noise, time-variant and nonlinear behavior of the micro-drilling process on deteriorating the real-time performance. In one scenario, with the anticipation control mode, the control signal cannot reach the set point meanwhile in the other, with the single loop mode the behavior of the force surpasses for a long time the set point even beyond the transient state.

On the contrary, if the learning algorithm is activated the control signal is modified per iteration and the overall performance of the artificial cognitive control is improved. The results corroborate how the Q-learning algorithm contributes to a better response and reduces the error-based performance indices.

The behavior of the drilling force when the learning is activated for the inverse model (i.e., anticipation mode) is reported in [82]. The drilling of 10 holes is performed, in order to corroborate the influence of reinforcement learning on improving the performance of the inverse model. Initially the response is very poor and the system cannot reach the set point. The behavior of the drilling force is quite good from the 4th hole due to reinforcement learning.

The single loop mode is functioning in the first three holes. After that, the poor performance index motivates the change to the internal model control where direct and inverse models are activated. This is a clear case study where the single loop is deteriorating due to the influence of disturbance such as tool wear and the artificial cognitive control tries to find an adequate solution by changing the execution mode. The dynamic response and the performance index are then better in the new execution mode and the system remains in this mode. Finally, the combinations of anticipation mode + learning and single loop + learning keep the cutting force value over the force point set. This result validates the utility of the learning algorithm inside the cognitive control architecture and the utility of this in industrial processes.

In order to conclude the main contributions, from the scientific and technical point of view, achieved during the realization of the present doctoral thesis are summarized as follows:

1. A library of models was developed to correlate the cutting parameters and physical-mechanical material properties with the forces and vibration signals captured during the elaboration of micro-drilling and micro-milling operations.
2. Two methodologies were developed for detecting run-out and predicting the holes' quality in micro-drilling processes. In addition, a methodology was proposed for the estimation of surface roughness in micro-milling processes. In all cases, the prediction models are based on soft-computing techniques such as neural networks and neuro-fuzzy algorithms.
3. Some modifications of a multi-objective cross-entropy method were introduced, where four parameters (epoch number, working population size, histogram interval number,

and elite fraction) are stored in the algorithm, in order to facilitate the tuning process. In the analysis of the relationship of these parameters on the algorithm performance, both the histogram interval number, and the elite fraction showed not significant influence, so they were removed from the study. On the contrary, the epoch number and the working population size had a remarkable influence on the execution time. Both influenced the Pareto front quality, up to some level, but as from this value, the quality stopped improving.

4. A reinforcement learning technique based on Q-learning algorithm was implemented. Due to the nature of the Q-learning algorithm some modifications and considerations were introduced to facilitate the deployment in the definition of the concepts of state and action, as well as the reward function.
5. An artificial cognitive architecture was designed and implemented for the monitoring and control of micromachining processes. The present architecture is divided in two main levels: the cognitive level, which includes all the functionalities (modeling, optimization, learning, ...) developed during the doctoral thesis; the executive level, which execute the monitoring and control actions in the processes, using an optimal configuration for each iteration (self-optimization algorithm) in function of the experience accumulated (self-learning algorithm) during the previous iteration between the cognitive level and the information captured from the process. Furthermore, several modes were developed to provide the modular and reconfigurable capabilities able to adapt to the non-linearity and non-stationarity behaviors of micro-manufacturing processes.
6. Finally, the proposed architecture was validated monitoring and controlling the force value in micro-drilling operations. During this stage, more than 400 tests were performed, combining all the controller modes (monitoring, single loop, anticipation, single loop + learning and anticipation + learning) described in the doctoral thesis, validating the utility of the learning and optimization algorithm inside the artificial cognitive architecture for the monitoring and control industrial processes.

List of contributions

Following, the list of the scientific and technical contributions put into consideration in the scientific community is shown. During the research period different results have been published in journals (*Science Citation Index,* SCI), international and national conferences. Furthermore, two foreign research stays, lectures, master thesis degree direction and reviewer activities have become part of the formation process.

Scientific publications in Journals (SCI):

- **Beruvides, G.**; Castaño, F.; Quiza, R.; Haber, R. (2016) "*Surface Roughness Modeling and Optimization of Tungsten-copper Alloys in Micro-milling Processes*". In: Measurement, Elsevier, 86, pp.246-252, doi: doi:10.1016/j.measurement.2016.03.002.
- **Beruvides, G.**; Quiza, R.; Haber, R. (2016) "*Multi-objective optimization based on an improved cross-entropy method. A case study of a micro-scale manufacturing process*". In: Information Sciences, Springer, 334-335, 9, pp. 161-173, doi: 10.1016/j.ins.2015.11.040.
- Castaño, F; del Toro, R; Haber, R; **Beruvides, G.** (2015) "*Conductance sensing for monitoring micromechanical machining of conductive materials*" In: Sensors and Actuators A: Physical, Elsevier, 232, pp.163-171, doi: 10.1016/j.sna.2015.05.015
- Haber, R; Juanes, C; **Beruvides, G.** (2015) "*Artificial cognitive control with self-x capabilities: a case study of a micro-manufacturing process*", in: Computers in Industry, 74, pp.135-150, doi: 10.1016/j.compind.2015.05.001.
- **Beruvides, G.**; Quiza, R.; Rivas, M.; Castaño, F; Haber, R. (2014) "*Online Detection of Run Out in Microdrilling of Tungsten and Titanium Alloys*". In: The International Journal of Advanced Manufacturing Technology, Springer, 74, 9, pp. 1567-1575, doi: 10.1007/s00170-014-6091-1.
- **Beruvides, G.**; Quiza, R.; del Toro, R.; Castaño, F; Haber, R. (2014) "*Correlation of the Holes Quality with the Force Signals in a Microdrilling Process of a Sintered Tungsten-Copper Alloy*". In: International Journal of Precision Engineering and Manufacturing, Springer, 15, 9, pp. 1801-1808, doi: 10.1007/s12541-014-0532-5.
- **Beruvides, G.**; Quiza, R.; del Toro, R.; Haber, R. (2013). "*Sensoring System and Signals Analysis for Tool use Monitoring in Microdrilling of a Sintered Tungsten-Copper Composite Material*". In: Sensors and Actuators A: Physical, Elsevier, 199, pp. 165-175, doi: 10.1016/j.sna.2013.05.021.
- **Beruvides, G.**; Quiza, R.; Haber, R.; del Toro, R. (2013). "*Extracción de Rasgos de las Señales para la Monitorización Indirecta de la Herramienta en el Microtaladrado*". In: Revista Dyna, 88, 4, pp. 405-413. Bilbao, Spain, doi: 10.6036/DYNAII.

Book chapters:

- Quiza, R.; **Beruvides, G.**; Davim, J.P. (2014) "*Modeling and optimization of mechanical systems and processes*" (Chapter 8). In: Davim, J.P. (ed.), Modern mechanical engineering. Berlin (Germany): Springer, ISBN 978-3-642-45175-1, pp. 169-198.

International conferences:

- Beruvides, G.; Villalonga, A.; Franciosa, P.; Quiza, R.; Rivas, M.; Castaño, F; Haber, R. (July 19-21th, 2017) "Self-adaptive Architecture for Pattern Recognition on Multi-Stage Assembly Processes" CIRP - 11th CIRP Conference on Intelligent Computation in Manufacturing Engineering (ICME), Naples, Italy.
- La Fe, I.; Quiza, R.; **Beruvides G.**; Haber, R. E. (November 21-25th, 2016) "*Modelalción Empírica de la Fuerza de Corte en el Proceso de Microfresado de Ti6Al4V*", In: II Simposio Internacional de Modelación Aplicada a la Ingeniería (MAI), La Habana, Cuba. ISBN: 978-959-261-533-5.
- Castaño, F.; Del Toro, R. M.; Haber, R. E.; **Beruvides, G.**, (March 14-17th, 2016) "*Monitoring tool usage on the basis of sensory information in microdrilling operations*", IEEE International Conference on Industrial Technology (ICIT), Taipei, Taiwan.
- **Beruvides, G.**; Juanes, C; Castaño, F; Haber, R. (July 22-24th, 2015) "*A self-learning strategy for artificial cognitive control systems*", IEEE International Conference on Industrial Informatics – INDIN15, Cambridge, UK pp. 1180-1185, doi: 10.1109/INDIN.2015.7281903.
- Cannavacciuolo, C; Rivas, M; Quiza, R; Haber, R; **Beruvides, G.** (April 6-10th, 2015) "*Efectividad del diseño ortogonal y los algoritmos genéticos en la optimización multi-objetivo del proceso de torneado*" CIUM 2015. Matanzas, Cuba. ISBN: 978-959-16-2442-0.
- Castaño, F; Haber, R; del Toro, R; **Beruvides, G.** (March 17-19th, 2015) "*Conductance Sensor for Micromachining. A Case Study on Monitoring Tool-Workpiece Contact*", IEEE International Conference of Industrial Technology – ICIT, Sevilla, Spain, pp. 1422-1426. doi: 10.1109/ICIT.2015.7125296
- **Beruvides, G.**; Quiza, R.; Rivas, M.; Castaño, F; Haber, R. (December 9-12th, 2014) "*Artificial intelligence-based modeling and optimization of microdrilling processes*", IEEE Symposium Series on Computational Intelligence- SSCI 2014, Orlando, USA, pp.49-53. doi: 10.1109/CIES.2014.7011830.
- Castaño, F; Haber, R; del Toro, R; **Beruvides, G.** (December 9-12th, 2014) "*Application of hybrid incremental modeling strategy for surface roughness estimation in micromachining processes*", IEEE Symposium Series on Computational Intelligence- SSCI 2014, Orlando, USA, pp. 54-59. doi: 10.1109/CIES.2014.7011831.
- **Beruvides, G.**; Quiza, R.; Rivas, M.; Castaño, F; Haber, R. (November 10-12th, 2014) "*Intelligent Models for Predicting the Force and Perpendicular Vibrations in*

Microdrilling Processes", IEEE 26th International Conference on Tools with Artificial Intelligence- ICTAI 2014, Limassol, Cyprus, pp. 506-511. doi: 10.1109/ICTAI.2014.82.

- **Beruvides, G.**; Quiza, R.; Rivas, M.; Castaño, F; Haber, R. (October 29th – November 1st, 2014) "*A Fuzzy-Genetic System to Predict the Cutting Force in Microdrilling Processes*", 40th annual conference of IEEE industrial electronics society-IECON 2014, Dallas, USA, pp. 34-37. doi: 10.1109/IECON.2014.7048473.

National conferences:

- Castaño, F.; Haber, R. E.; **Beruvides, G.**, (September 07-09th, 2016) "Inteligencia Computacional Embebida Para La Supervisión De Procesos De Microfabricación", In: XXXVII Jornadas de Automática, Madrid, España.
- **Beruvides, G.**; Castaño, F; Haber, R. (June 22-24th, 2016) "*Surface Quality Prediction using Hybrid Incremental Modeling*", In: XII Simposio CEA de Control Inteligente, Gijón, España. ISBN: 978-84-16664-18-4
- **Beruvides, G.**; Juanes, C; Haber, R; Castaño, F. (June 24-26th, 2015) "*Arquitectura de Control Cognitivo Artificial usando una plataforma computacional de bajo costo*", In: XI Simposio CEA de Control Inteligente, Badajoz, España. ISBN: 978-84-606-9052-8.
- **Beruvides, G.**; Haber, R; del Toro, R; Castaño, F. (June 10-12th, 2015) "*On-line Artificial Cognitive Control for Micromanufacturing Processes*", In: Congreso máquinas-herramienta y tecnologías de fabricación, San Sebastián, España.
- **Beruvides, G.**; Quiza, R.; Rivas, M.; Castaño, F; Haber, R. (June, 2014) "*Modelo Neuronal de la Fuerza de Corte en el Microtaladrado de Aleaciones Especiales*". In: X Simposio CEA de Control Inteligente, Segovia, España.
- Castaño, F; Haber, R; del Toro, R; **Beruvides, G.** (June, 2014) "*Monitorización inteligente en tiempo real del acabado superficial de micro-piezas basado en modelado híbrido incremental*". In: X Simposio CEA de Control Inteligente, Segovia, España.

Pre-doctoral stays in other countries:

- Institute for Production Systems and Design Technology, IPK (2015), with the professor Ph.D. Eckart Uhlmann from August 30th to November 28th, Fraunhofer, Berlin, Germany
- International Manufacturing Centre (2016) with professor Ph.D. Derek Ceglarek from August 28th to December 23th, Warwick Manufacturing Group, University of Warwick, Coventry, United Kingdom.

Thesis director & Lectures

Master degree thesis:

- La Fe Perdomo, Iván (January, 2017) "Sistema de experto para la selección de parámetros óptimos en el microfresado de ranuras", Universidad de Matanzas, Cuba.
- Villalonga Jaén, Alberto (January, 2017) "Sistema de monitoreo de bajo costo para procesos y sistemas mecánicos", Universidad de Matanzas, Cuba.

Lectures:

- "*Procesos de Manufactura Actuales. Nociones Generales*", (June, 2015). In: Universidad de Matanzas, Matanzas, Cuba.

Reviewer activities (JCR)

- IEEE Transitions on Industrial Informatics
- Engineering Applications of Artificial Intelligence
- Journal of Intelligent Manufacturing
- Expert Systems with Applications
- Fuzzy Information and Engineering
- Measurement
- Dyna
- Materials Research Innovations
- Journal of Precision Engineering and Manufacturing
- Revista Iberoamericana de Automática e Informática Industrial RIAI

Future works

The present doctoral thesis has opened new research lines to be addressed in future works.

From the scientific-technical point of view, a new version of proposed architecture will be developed. The objective is integration of advanced techniques from the paradigms of Cloud Computing, Cyber-Physical Systems, Big Data and Internet of Things within the actual implementation. This objective is supported with the new trends in the principal researches lines in Europe to develop the Factory of the Future. Besides, another concept to support the new research lines is the known as *Industry 4.0* (Europe), also called *Smart Factory* (USA, Japan, etc.) based on strategy for the strong customization of products under the conditions of highly flexibility in mass production. The required automation technology is improved by the introduction of methods of self-optimization, self-configuration, self-diagnosis, cognition and intelligent support of workers in their increasingly complex work.

In this sense, the first steps were taken in the framework of the research project CONMICRO [319]. The main objective was embedded self-optimized controllers and cognitive capacities in the real-time, modular, network and reconfigurable platform. Furthermore, the ability to interact with middleware systems for global distributed monitoring systems, providing the basis for the development of intelligent distributed control systems based on cognitive capacities at the highest level.

Secondly, a real time device will be developed to predict phenomena such as: chatter and run-out using vibration signals captured in micro-milling processes with a relation of diameters between 30μm-500μm. At present, a series geometrical, wear and surface roughness measures are done with two high resolution measure techniques: a 3D micro coordinate measurement machine and surface roughness measurement device. The main objective is correlated the frequency-amplitude variations in real-time with the vibration signals captured during milling and micro-milling process. Based on these results, it is currently working on the phases of design of the hardware configuration and implementation of the different modules to compose the device for predictive tasks. This investigation line is carried out between three research groups: GAMHE (Centre for Automation and Robotic, Spain), CEFAS (University of Matanzas, Cuba) and IPK (Institute for Production Systems and Design Technology Fraunhofer, Germany) supporting the interaction during the research period with multi-disciplinary team from different latitudes.

Thirdly, based on the same multi-disciplinary interaction during the formation period, but in this case applied to a different research line are the bases of collaboration between GAMHE; CEFAS and DLM (Warwick Manufacturing Group, UK). The main objective is to bring the artificial cognitive architecture developed in the present doctoral thesis to address pattern recognition strategies during part assembly processes for the automobile industry. The car industry is a complex scenario with multiple layers or station during the assembly process, interacting machines, robots, measurement devices, etc. at the same time. The new proposal includes an adaptive artificial cognitive architecture able to control in real-time the key characteristic based on specific key indicators measured multi-level assembly process using data classification, modeling, optimization and decision making. The new design includes Big Data, Cloud Computing and CPS techniques supporting to current concepts introduced in the Factory of the Future program.

Finally, a two level (machine level (local mode) and cloud server (global mode)) architecture to predictive maintenance strategies is introduced in the project AM4G [320]. This condition-based monitoring system is also focused on challenges of cyber-physical systems applied to manufacturing networks. Furthermore, a modular and configurable event and alarm register is developed. The main objective is to detect which local actions on the machine must be carried out immediately (emergency stop, programmed shutdown, slow speed ...) in close coordination with the machine PLC/CNC system. Furthermore, the signals, alarm records, etc. storage during a window time are interchanged with the cloud server. Finally, in the global mode, information of the same machine family is used to obtain a best algorithm coefficient configuration for each machine connected in the manufacturing network, updating the parameters configuration (local model), in the next connection machine-cloud server.

CONCLUSIONES

La presente tesis doctoral se basa en una metodología científica que combinas métodos de investigación teórica y experimental, abordando el diseño, desarrollo e implementación de una arquitectura cognitiva artificial para la monitorización y el control en procesos complejos. El método propuesto un conjunto de técnicas de inteligencia computacional como base para el diseño e implementación de una arquitectura modular y reconfigurable, que ha sido validada en varias operaciones de micromecanizado. Durante la investigación el carácter multidisciplinario e interdisciplinario está presente en varios desafíos científicos y técnicos en diferentes áreas del conocimiento, desde la ingeniería mecánica y la microfabricación hasta el control y la ingeniería informática.

La hipótesis de partida de este trabajo se ha materializado en una implementación en Java sobre un computador de bajo coste de la arquitectura cognitiva artificial. De este modo el fundamento que nace en el nexo entre el paradigma del control por modelo interno y la conectividad cerebro-cerebelo como base de la inteligencia humana, se ha materializado en una biblioteca de modelos directos e inversos como veremos más adelante. El segundo principio que ha sido considerado en la arquitectura basada en modelos y modos de operación, es el modelo de los circuitos compartidos y la emulación de las capacidades y experiencias socio-cognitivas de los seres humanos.

Durante la investigación se ha re-elaborado y aplicado una amplia gama de métodos contemporáneos para el procesamiento de señales, filtrado, extracción de características, identificación de patrones, modelado, optimización y aprendizaje para dotar a la arquitectura cognitiva artificial de las capacidades necesarias.

Por otra parte, la capacidad de reconfiguración y portabilidad de la arquitectura se ha basado en el acondicionamiento de los diferentes modos, utilizando estrategias de aprendizaje y optimización. Además, en la tesis doctoral se han seleccionado dos procesos de micromecanizado para validar los niveles cognitivo y ejecutivo de la arquitectura. La selección de dichos procesos de fabricación como casos de estudio se justifica por el gran impacto que tienen en la economía, producción, servicios y estrategias sociales en el siglo XXI.

A continuación, se presentan un resumen por capítulos de los principales resultados obtenidos durante la presente tesis doctoral:

Revisión del estado del arte. Retos y oportunidades

En primer lugar, se ha realizado una revisión del estado en los paradigmas de la teoría general de los sistemas, la teoría del control, las técnicas de computación blanda y la ciencia de la computación. Se ha prestado especial atención a la base teórica de los algoritmos de modelización, optimización y aprendizaje de máquinas enfocados en aplicaciones industriales desde las perspectivas de monitoreo y control. En este sentido, la revisión se ha centrado en las técnicas de *soft-computing*, analizando su evolución desde sus inicios hasta las tendencias actuales. Esta revisión no sólo permite conocer la evolución de estas técnicas, sino que también se identificaron un grupo de técnicas para proporcionar un mejor desempeño en las nuevas generaciones de arquitecturas para la supervisión y control de procesos físicos complejos.

Debido a la naturaleza multidisciplinaria, interdisciplinaria y heterogénea de esta Tesis Doctoral, la revisión del estado del arte se ha centrado en generar contribuciones científicas y técnicas en todos y cada uno de los campos de aplicación, áreas de conocimiento y temas abordados durante la investigación. Para lograr el objetivo principal, como se ha descrito anteriormente, se abordan tres cuestiones específicas, a saber: modelado, auto-optimización y auto-aprendizaje para dotar a la arquitectura cognitiva artificial de las funcionalidades necesarias. A continuación, se presentan los resultados más relevantes.

Principales resultados durante las etapas de procesamiento y modelado

Desde el punto de vista científico se demostró en [193], que la combinación de ondeletas o wavelets (WT) y ANN proporciona un mejor equilibrio entre el coeficiente de correlación y capacidad de generalización. Sin embargo, desde el punto de vista técnico, la implementación de un sistema de monitoreo a escala industrial tiene severas limitaciones. Por ejemplo, el coste computacional (tiempo de cálculo, tiempo de procesamiento de la señal, etc.) es menor para las combinaciones WT y ANFIS. Este resultado significa que WT + ANFIS es un buen candidato para la implementación final del sistema de monitoreo a escala industrial. En todos los casos se consideró una significación estadística del 95%. Una ventaja adicional de la ondeleta en la etapa de extracción de características es que ofrece la posibilidad de filtrar la señal en las diversas bandas de frecuencia y de separar la frecuencia interesante de la frecuencia de ruido, lo que reduce considerablemente el área de búsqueda en la etapa siguiente y una mejor predicción final.

Otro hecho interesante fue que la correlación más alta con errores de calidad de los agujeros en los procesos de microtaladrado sólo guardan relación a los segmentos de señal en la entrada y la salida de la herramienta [231]. Esto se explica si el daño se produce principalmente en la entrada de la punta de la herramienta y, posteriormente, este daño afecta la salida de la viruta durante el movimiento hacia atrás. Además, las bandas de frecuencia de las características correlacionadas más altas, corresponde a las frecuencias naturales de los sistemas. Durante el modelado, la mayor correlación se encontró en el movimiento de salida de la herramienta, igual a 0.8238, lo que significa que el modelo presenta un 82% de la variabilidad en los datos observados, con un 99% del nivel del intervalo de confianza. Dicho fenómeno se explica por la fricción entre la herramienta, la viruta y el perfil de los agujeros. Por último, también son dignas de mención las bandas de frecuencias incluidas en el modelo. El primero es la banda de frecuencia de 3906 a 4297 Hz, que tiene una relación directa con el error de calidad de agujero. Por el contrario, la banda de frecuencia más baja (hasta 391 Hz) tiene una relación inversa con el error de calidad de los agujeros.

Otro resultado interesante fue el propuesto en [254]. En este se desarrolló un sistema de monitoreo en dos etapas para detectar la ocurrencia de agotamiento en los procesos de microtaladrado de las aleaciones Ti_6Al_4V y $W_{78}Cu_{22}$. El mismo, utiliza una transformada FFT (primer paso) para extraer características de los datos de fuerza medidos en línea y una red neuronal perceptron multicapas (segundo paso) para predecir la condición del proceso a partir de las características obtenidas previamente. El sistema de monitoreo fue capaz de identificar más del 70% de las condiciones de descentrado de la herramienta, con menos del 10% de falsas detecciones. Debido a la predicción de perdida de tolerancia dimensionales en los microtaladros, el enfoque propuesto puede producir ganancias considerables en la vida de la herramienta y la productividad. El modelo ajustado tuvo un coeficiente de determinación, igual a 0.9586, lo que indica que el modelo presenta un 95% de variabilidad en la variable dependiente. El error estándar de estimación es 0.0899, y el error absoluto medio es 0.0266. Con el fin de probar las capacidades de generalización del modelo, se compararon las predicciones obtenidas para los conjuntos de entrenamiento y validación. La diferencia entre los valores medios de las predicciones de los conjuntos de entrenamiento y validación se desecha. Además, la media de los valores predichos de ambos conjuntos fue muy similares a los valores observados. Sin embargo, la propagación de las predicciones para el conjunto de entrenamiento fue notablemente menor que la propagación para el conjunto de validación. En consecuencia, las predicciones del modelo, en diferentes condiciones, no fueron tan fiables como las obtenidas para los datos utilizados en el proceso de formación.

Por último, se introdujo un modelo de red neuronal artificial basado en propiedades mecánicas y térmicas para cinco materiales [318]. Los datos experimentales disponibles se dividieron aleatoriamente en un conjunto de entrenamiento con 1973 muestras (correspondiente al 80%) y un conjunto de validación con las 493 muestras restantes. El proceso de modelación se llevó a cabo sobre la base del algoritmo de retropropagación, con velocidad de aprendizaje adaptativo y momento. Para el mismo, se seleccionaron los siguientes parámetros de entrenamiento: tasa de aprendizaje inicial, 0.001; ratio para aumentar la tasa de aprendizaje, 1.05; ratio para disminuir la tasa de aprendizaje, 0.70; constante de momento, 0.9 y gradiente de rendimiento mínimo, 10^{-10}. La condición de parada se estableció después de las 300 000 iteraciones. El modelo neural obtenido tiene un coeficiente de correlación de 0.8969, por lo que explica el 89% de la variabilidad en la variable de respuesta. El error estándar de las estimaciones fue 1.084 y el error absoluto medio fue de 0.7340. Con el fin de evaluar la capacidad de generalización del modelo, se analizaron los residuos procedentes de los conjuntos de entrenamiento y validación, llevando a cabo la comparación entre los valores medios y las desviaciones estándar de ambos conjuntos utilizando las pruebas *t-Student* y *F-Fisher*, respectivamente. Estas pruebas arrojaron valores de probabilidad asociados de 0.6125 y 0.1832, respectivamente. Por lo tanto, la hipótesis nula (ambos medios son iguales y ambas desviaciones estándar son iguales) no pudo ser rechazada con un nivel de confianza del 95%. Finalmente, se pudo concluir que ambos conjuntos de residuos provienen de la misma distribución y además el modelo presentó una buena capacidad de generalización.

Auto-optimización

Trece soluciones no dominadas para un problema de optimización de dos objetivos se obtuvieron en [318]. La minimización del tiempo de mecanizado de la unidad (inverso de la tasa de arranque de material, MRR) y la fuerza de empuje fueron los objetivos basados en el conjunto óptimo de parámetros de corte en un proceso de microtaladrado. La selección de la solución más conveniente dependió de las condiciones específicas. Por ejemplo, si la productividad es la cuestión más importante, se debió seleccionar la solución No. 13, ya que produce el menor tiempo de mecanizado unitario a pesar del relativamente alto valor de la fuerza de empuje. Por el contrario, si el coste de la herramienta es determinante, se debió elegir la solución No. 1, debido a la fuerza mínima, se minimiza el riesgo de rotura de la herramienta. Finalmente, las otras soluciones pueden ser adecuadas para otras situaciones intermedias. El frente de Pareto demostró la utilidad de la técnica heurística para resolver problemas industriales, en este caso particular, en micro-fabricación de aleaciones especiales. Basándose en estos estudios, el operador puede tomar una decisión de marera rápida en función de acelerar la producción o disminuir el gasto de herramientas.

En Beruvides, et al. [255], se seleccionó un algoritmos genético de clasificación no dominado (NSGA-II) para abordar el problema de optimización multiobjetivo en un proceso de microfresado. Para este, se definieron dos funciones objetivos, la primera, la rugosidad de la superficie; y la segunda, el tiempo de mecanizado unitario. Como en el ejemplo anterior, la selección de la solución más conveniente depende de las condiciones específicas de las producciones. En este aspecto, se analizaron dos posibles decisiones industriales: la productividad fue la cuestión más importante en el primero, penalizando la calidad superficial; en el segundo, se obtuvieron los mejores parámetros de rugosidad superficial con menor productividad. Además, se describió el comportamiento de la componente de la señal de vibración en el eje *z*, *Vz*, para dos puntos. Los valores de los parámetros de corte seleccionados para el primer escenario fueron una velocidad de corte de 125,2 m/min, una velocidad de avance de 1,9 μm/diente y una profundidad axial de corte de 0,1 mm. El rango de la vibración del eje *z* alcanzado estuvo entre 0,12g y 0,26g, con una media entre los 0,14g y 0,20g. Para el segundo escenario, los valores de los parámetros de corte fueron: una velocidad de corte de 71,4m/min, una velocidad de avance de 2,6 μm/diente y una profundidad axial de corte de 0,1 mm con un rango de la vibración del eje z entre 0,08g y 0,18g, con una media entre 0,09g y 0,16g. El sistema en tiempo real propuesto, está basado en modelos ANFIS, capaces de predecir la rugosidad superficial en función de la vibración del eje z capturada durante el proceso de corte. Este sistema puede proporcionar una estimación de rugosidad superficial con una lectura de error del 9,5% en una configuración industrial, dependiendo de los parámetros de corte.

Por último, un método de optimización multiobjetivo mediante entropía cruzada se introdujo en [274]. En primer lugar, se realizó una comparación utilizando funciones de prueba MOP, ZDT, WFG con otros algoritmos multiobjetivos, obteniendo en términos generales, una mejor solución, excepto para MOP1 y MOP2, donde si fueron peores que algunos de los proporcionados por los otros enfoques evolutivos. Sin embargo, los valores de distancia generacional fueron muy bajos en todos los problemas; por lo tanto, dichas soluciones son aceptables para muchas aplicaciones.

Una vez definido el método, se seleccionó un proceso de micro-taladrado para validar la implementación en un escenario industrial real. Para la optimización, se obtuvieron dos modelos representativos del proceso de fuerza de corte y vibración en el plano perpendicular al eje de taladrado, utilizando un método bien establecido inspirado en técnicas de inteligencia artificial. A continuación, se consideraron simultáneamente dos objetivos: el tiempo de taladrado y la amplitud de las vibraciones en el plano perpendicular al eje de taladrado. Además, se introdujeron una serie de restricciones, definiendo la fuerza de pandeo inferior a la fuerza de empuje admisible, evitando así la rotura por pandeo de la herramienta.

En total, se realizaron 25 ejecuciones para aumentar la fiabilidad de los resultados de optimización. Tres zonas se pueden observar fácilmente: Figure 3.12 se mostraron parámetros de corte que obtendrán agujeros de alta calidad (Zona I), pero durante excesivos tiempos de operación; en la Zona II de la misma figura se encontraron la mejor combinación de soluciones en ambos objetivos, por lo que son los parámetros más convenientes para la mayoría de las operaciones, dando una calidad de agujero razonablemente buena en tiempos de perforación bajos y finalmente, la Zona III se implicaron soluciones con tiempos de ejecución más cortos pero con mayores amplitudes de vibración. En consecuencia, estas son sólo soluciones para agujeros donde la calidad no es un requisito importante. Las soluciones de la frontera Pareto, en todas las zonas, cubrieron casi todos los intervalos y escalones de velocidad de corte. Por el contrario, los valores de las velocidades de avance fueron superiores a la fuente en todas las zonas. En todos los casos se establecieron intervalos de confianza del 95% para el tiempo de ejecución.

Auto-aprendizaje

Con el fin de validar el rendimiento de la estrategia de auto-aprendizaje en el marco de la arquitectura cognitiva, se ejecutaron un grupo de experimentos de microtaladrado [82]. En primer lugar, se realizó el monitoreo del proceso sin control. Posteriormente, los modos de control de anticipativo y de control por lazo simple se ejecutaron sin activar el algoritmo de aprendizaje, fijando como variable de control una fuerza de corte de 10N. Finalmente, se utilizaron matrices de taladros de diez agujeros para la evaluación del algoritmo de aprendizaje.

A pesar del diseño apropiado de los modos de control, los resultados iniciales no fueron demasiado satisfactorios debido a la influencia de la incertidumbre, el ruido, el tiempo y el comportamiento no lineal del proceso de microtaladrado del rendimiento en tiempo real. En un escenario, con el modo de control anticipativo, la señal de control no pudo alcanzar el punto de ajuste, mientras que en el otro, con el modo de control por lazo simple, el comportamiento de la fuerza supera durante mucho tiempo el punto de fijado incluso más allá del estado transitorio.

Por el contrario, si se activa el algoritmo de aprendizaje, la señal de control se modifica por iteración y se mejora el rendimiento global del control cognitivo artificial. Los resultados corroboraron cómo el algoritmo *Q-learning* contribuye a una mejor respuesta y reduce los índices de rendimiento basados en errores.

Los resultados obtenidos en [83] mostraron el comportamiento de la fuerza de taladrado cuando el aprendizaje se activó para el modelo inverso (es decir, el modo de anticipación). Durante el mismo, se realizaron 10 agujeros, con el fin de mostrar la influencia del aprendizaje de refuerzo en la mejora del rendimiento del modelo inverso. Inicialmente la respuesta fue muy pobre y el sistema no pudo alcanzar el punto de fijado (fuerza igual 10N). Posteriormente, el comportamiento de la fuerza fue bueno a partir del 4to agujero debido a la influencia del método de aprendizaje del refuerzo propuesto.

En otro ensayo, inicialmente, el modo de control por lazo simple funcionó bien durante los tres primeros agujeros, después, debido a un pobre índice de desempeño el algoritmo de organización realiza el cambio al control del modelo interno donde se activaron los modelos directo e inverso. Este es un caso de estudio claro en el que el control por lazo simple está desajustado debido a la influencia de perturbaciones tales como: el desgaste de la herramienta o la vibración de la misma, demostrando la utilizad del sistema de control cognitivo artificial propuesto para encontrar una solución adecuada cambiando el modo de ejecución en tiempo real. Finalmente, las combinaciones de modo de anticipación + aprendizaje y lazo simple + aprendizaje mantuvieron el valor de la fuerza de corte sobre el punto de fuerza establecido (10N). Este resultado validó la utilidad del algoritmo de aprendizaje dentro de la arquitectura de control cognitivo y la utilidad de éste en procesos industriales.

Para concluir, las principales contribuciones desde el punto de vista científico y técnico archivadas durante la realización de la presente tesis doctoral se resumen a continuación:

1. Se desarrolló una biblioteca de modelos para correlacionar los parámetros de corte y las propiedades físico-mecánico del material con las fuerzas y señales de vibración capturadas durante la elaboración de las operaciones de microtaladrado y microfresado.
2. Se desarrollaron dos metodologías para la detección del descentrado de la herramienta y la predicción de la calidad de los agujeros en los procesos de microtaladrado. Además, se propuso una metodología para la estimación de la rugosidad superficial en los procesos de microfresado. En todos los casos, los modelos de predicción se basan en técnicas de *soft-computing* tales como: redes neuronales y algoritmos neuroborrosos.
3. Se introdujo un nuevo grupo de modificaciones para un método de optimización multiobjetivo mediante entropía cruzada. Dichas modificaciones se aplicaron a cuatro parámetros claves del algoritmo (número de iteraciones, tamaño de población, número de intervalo de histograma y fracción de elite) para facilitar el proceso convergencia. En el análisis de la relación de estos parámetros en el rendimiento del algoritmo, tanto el número de intervalo de histograma como la fracción de elite mostraron una influencia no significativa, por lo que se retiraron del estudio. Por el contrario, el número de iteraciones y el tamaño de la población trabajadora tuvieron una notable influencia en el tiempo de ejecución. Además, ambos influenciaron la calidad del frente de Pareto, hasta cierto nivel, en el cual la calidad dejó de mejorar.
4. Se implementó una técnica de aprendizaje por refuerzo basada en el algoritmo Q-learning. Debido a la naturaleza del algoritmo Q-learning, se introdujeron algunas modificaciones y consideraciones para facilitar el despliegue en la definición de los conceptos de estado y acción, así como la función de recompensa.

5. Se diseñó e implementó una arquitectura cognitiva artificial para el monitoreo y control de procesos de micromecanizado. La arquitectura se divide en dos niveles principales: el nivel cognitivo, que incluye todas las funcionalidades (modelización, optimización, aprendizaje, ...) desarrolladas durante la tesis doctoral y el nivel ejecutivo que ejecuta las acciones de monitoreo y control en los procesos, utilizando una configuración óptima para cada iteración (algoritmo de auto-optimización) en función de la experiencia acumulada (algoritmo de auto-aprendizaje) durante la interacción previa entre el nivel cognitivo y la información obtenida del proceso. Además, se desarrollaron varios modos para proporcionar las capacidades modulares y reconfigurables capaces de adaptarse a los comportamientos de no-linealidad y no-estacionario de los procesos de microfabricación.
6. Por último, se validó la arquitectura propuesta para la monitorización y control de la fuerza de corte en el microtaladrado. Durante esta etapa, se realizaron más de 400 pruebas, combinando todos los modos de operación (monitorización, lazo simple, anticipación, lazo simple + aprendizaje y anticipación + aprendizaje) descritos en la tesis doctoral, validando la utilidad del algoritmo de aprendizaje y optimización dentro de la arquitectura cognitiva artificial propuesta.

Lista de contribuciones

A continuación, se muestra la lista de las contribuciones científicas y técnicas que se pusieron a consideración de la comunidad científica internacional. Durante el período de investigación se han publicado diferentes resultados en revistas de impacto (*Science Citation Index*, SCI), conferencias internacionales y nacionales. Además, dos estancias de investigación en el extranjero, la dirección de tesis de maestría y las actividades como revisor que han formado parte del proceso formativo.

Revistas de impacto (SCI):

- **Beruvides, G.**; Castaño, F.; Quiza, R.; Haber, R. (2016) "*Surface Roughness Modeling and Optimization of Tungsten-copper Alloys in Micro-milling Processes*". In: Measurement, Elsevier, 86, pp.246-252, doi: doi:10.1016/j.measurement.2016.03.002.
- **Beruvides, G.**; Quiza, R.; Haber, R. (2016) "*Multi-objective optimization based on an improved cross-entropy method. A case study of a micro-scale manufacturing process*". In: Information Sciences, Springer, 334-335, 9, pp. 161-173, doi: 10.1016/j.ins.2015.11.040.

- Castaño, F; del Toro, R; Haber, R; **Beruvides, G.** (2015) "*Conductance sensing for monitoring micromechanical machining of conductive materials*" In: Sensors and Actuators A: Physical, Elsevier, 232, pp.163-171, doi: 10.1016/j.sna.2015.05.015
- Haber, R; Juanes, C; **Beruvides, G.** (2015) "*Artificial cognitive control with self-x capabilities: a case study of a micro-manufacturing process*", in: Computers in Industry, 74, pp.135-150, doi: 10.1016/j.compind.2015.05.001.
- **Beruvides, G.**; Quiza, R.; Rivas, M.; Castaño, F; Haber, R. (2014) "*Online Detection of Run Out in Microdrilling of Tungsten and Titanium Alloys*". In: The International Journal of Advanced Manufacturing Technology, Springer, 74, 9, pp. 1567-1575, doi: 10.1007/s00170-014-6091-1.
- **Beruvides, G.**; Quiza, R.; del Toro, R.; Castaño, F; Haber, R. (2014) "*Correlation of the Holes Quality with the Force Signals in a Microdrilling Process of a Sintered Tungsten-Copper Alloy*". In: International Journal of Precision Engineering and Manufacturing, Springer, 15, 9, pp. 1801-1808, doi: 10.1007/s12541-014-0532-5.
- **Beruvides, G.**; Quiza, R.; del Toro, R.; Haber, R. (2013). "*Sensoring System and Signals Analysis for Tool use Monitoring in Microdrilling of a Sintered Tungsten-Copper Composite Material*". In: Sensors and Actuators A: Physical, Elsevier, 199, pp. 165-175, doi: 10.1016/j.sna.2013.05.021.
- **Beruvides, G.**; Quiza, R.; Haber, R.; del Toro, R. (2013). "*Extracción de Rasgos de las Señales para la Monitorización Indirecta de la Herramienta en el Microtaladrado*". In: Revista Dyna, 88, 4, pp. 405-413. Bilbao, Spain, doi: 10.6036/DYNAII.

Capítulos de libros:

- Quiza, R.; **Beruvides, G.**; Davim, J.P. (2014) "*Modeling and optimization of mechanical systems and processes*" (Chapter 8). In: Davim, J.P. (ed.), Modern mechanical engineering. Berlin (Germany): Springer, ISBN 978-3-642-45175-1, pp. 169-198.

Conferencias Internacionales:

- Beruvides, G.; Villalonga, A.; Franciosa, P.; Quiza, R.; Rivas, M.; Castaño, F; Haber, R. (July 19-21th, 2017) "Self-adaptive Architecture for Pattern Recognition on Multi-Stage Assembly Processes" CIRP - 11th CIRP Conference on Intelligent Computation in Manufacturing Engineering (ICME), Naples, Italy.
- La Fe, I.; Quiza, R.; **Beruvides G.**; Haber, R. E. (November 21-25th, 2016) "*Modelalción Empírica de la Fuerza de Corte en el Proceso de Microfresado de Ti6Al4V*", In: II Simposio Internacional de Modelación Aplicada a la Ingeniería (MAI), La Habana, Cuba. ISBN: 978-959-261-533-5.

- Castaño, F.; Del Toro, R. M.; Haber, R. E.; **Beruvides, G.**, (March 14-17th, 2016) "*Monitoring tool usage on the basis of sensory information in microdrilling operations*", IEEE International Conference on Industrial Technology (ICIT), Taipei, Taiwan.
- **Beruvides, G.**; Juanes, C; Castaño, F; Haber, R. (July 22-24th, 2015) "*A self-learning strategy for artificial cognitive control systems*", IEEE International Conference on Industrial Informatics – INDIN15, Cambridge, UK pp. 1180-1185, doi: 10.1109/INDIN.2015.7281903.
- Cannavacciuolo, C; Rivas, M; Quiza, R; Haber, R; **Beruvides, G.** (April 6-10th, 2015) "*Efectividad del diseño ortogonal y los algoritmos genéticos en la optimización multi-objetivo del proceso de torneado*" CIUM 2015. Matanzas, Cuba. ISBN: 978-959-16-2442-0.
- Castaño, F; Haber, R; del Toro, R; **Beruvides, G.** (March 17-19th, 2015) "*Conductance Sensor for Micromachining. A Case Study on Monitoring Tool-Workpiece Contact*", IEEE International Conference of Industrial Technology – ICIT, Sevilla, Spain, pp. 1422-1426. doi: 10.1109/ICIT.2015.7125296
- **Beruvides, G.**; Quiza, R.; Rivas, M.; Castaño, F; Haber, R. (December 9-12th, 2014) "*Artificial intelligence-based modeling and optimization of microdrilling processes*", IEEE Symposium Series on Computational Intelligence- SSCI 2014, Orlando, USA, pp.49-53. doi: 10.1109/CIES.2014.7011830.
- Castaño, F; Haber, R; del Toro, R; **Beruvides, G.** (December 9-12th, 2014) "*Application of hybrid incremental modeling strategy for surface roughness estimation in micromachining processes*", IEEE Symposium Series on Computational Intelligence-SSCI 2014, Orlando, USA, pp. 54-59. doi: 10.1109/CIES.2014.7011831.
- **Beruvides, G.**; Quiza, R.; Rivas, M.; Castaño, F; Haber, R. (November 10-12th, 2014) "*Intelligent Models for Predicting the Force and Perpendicular Vibrations in Microdrilling Processes*", IEEE 26th International Conference on Tools with Artificial Intelligence- ICTAI 2014, Limassol, Cyprus, pp. 506-511. doi: 10.1109/ICTAI.2014.82.
- **Beruvides, G.**; Quiza, R.; Rivas, M.; Castaño, F; Haber, R. (October 29th – November 1st, 2014) "*A Fuzzy-Genetic System to Predict the Cutting Force in Microdrilling Processes*", 40th annual conference of IEEE industrial electronics society-IECON 2014, Dallas, USA, pp. 34-37. doi: 10.1109/IECON.2014.7048473.

Conferencias nacionales:

- Castaño, F.; Haber, R. E.; **Beruvides, G.**, (September 07-09th, 2016) "Inteligencia Computacional Embebida Para La Supervisión De Procesos De Microfabricación", In: XXXVII Jornadas de Automática, Madrid, España.

- **Beruvides, G.**; Castaño, F; Haber, R. (June 22-24th, 2016) "*Surface Quality Prediction using Hybrid Incremental Modeling*", In: XII Simposio CEA de Control Inteligente, Gijón, España. ISBN: 978-84-16664-18-4
- **Beruvides, G.**; Juanes, C; Haber, R; Castaño, F. (June 24-26th, 2015) "*Arquitectura de Control Cognitivo Artificial usando una plataforma computacional de bajo costo*", In: XI Simposio CEA de Control Inteligente, Badajoz, España. ISBN: 978-84-606-9052-8.
- **Beruvides, G.**; Haber, R; del Toro, R; Castaño, F. (June 10-12th, 2015) "*On-line Artificial Cognitive Control for Micromanufacturing Processes*", In: Congreso máquinas-herramienta y tecnologías de fabricación, San Sebastián, España.
- **Beruvides, G.**; Quiza, R.; Rivas, M.; Castaño, F; Haber, R. (June, 2014) "*Modelo Neuronal de la Fuerza de Corte en el Microtaladrado de Aleaciones Especiales*". In: X Simposio CEA de Control Inteligente, Segovia, España.
- Castaño, F; Haber, R; del Toro, R; **Beruvides, G.** (June, 2014) "*Monitorización inteligente en tiempo real del acabado superficial de micro-piezas basado en modelado híbrido incremental*". In: X Simposio CEA de Control Inteligente, Segovia, España.

Estancias pre-doctorales

- Institute for Production Systems and Design Technology, IPK (2015), with the professor Ph.D. Eckart Uhlmann from August 30th to November 28th, Fraunhofer, Berlin, Germany
- International Manufacturing Centre (2016) with professor Ph.D. Derek Ceglarek from August 28th to December 23th, Warwick Manufacturing Group, University of Warwick, Coventry, United Kingdom.

Dirección de tesis y Lecturas

Tesis de Master:

- La Fe Perdomo, Iván (January, 2017) "Sistema de experto para la selección de parámetros óptimos en el microfresado de ranuras", Universidad de Matanzas, Cuba.
- Villalonga Jaén, Alberto (January, 2017) "Sistema de monitoreo de bajo costo para procesos y sistemas mecánicos", Universidad de Matanzas, Cuba.

Lecturas:

- "*Procesos de Manufactura Actuales. Nociones Generales*", (June, 2015). In: Universidad de Matanzas, Matanzas, Cuba.

Revisor en revistas (SCI)

- IEEE Transitions on Industrial Informatics
- Engineering Applications of Artificial Intelligence
- Journal of Intelligent Manufacturing
- Expert Systems with Applications
- Fuzzy Information and Engineering
- Measurement
- Dyna
- Materials Research Innovations
- Journal of Precision Engineering and Manufacturing
- Revista Iberoamericana de Automática e Informática Industrial RIAI

Trabajos futuros

La presente tesis doctoral ha abierto nuevas líneas de investigación que se abordarán en futuros trabajos:

Desde el punto de vista científico-técnico, se desarrollará una nueva versión de la arquitectura propuesta. El objetivo es la integración de técnicas avanzadas como: Cloud Computing, Sistemas Cibernéticos, Big Data e Internet de Cosa con los modos desarrollados durante la tesis doctoral. Este objetivo se basa en las nuevas tendencias en las principales líneas de investigación en Europa para desarrollar la Fábrica del Futuro (*Factory of Future*, FoF). Además, otro concepto, son las nuevas líneas de investigación conocidas como *Industry 4.0* (Europa), también llamado *Smart Factory* (EE.UU., Japón, etc.) basado en la estrategia para la personalización fuerte de productos en las condiciones de alta flexibilidad para producciones en masa. La tecnología de automatización necesaria se mejora mediante la introducción de métodos de auto-optimización, autoconfiguración, autodiagnóstico, cognición y sistemas inteligentes para apoyar a los trabajadores en entornos de trabajo cada vez más complejo.

En este sentido, los primeros pasos se dieron en el marco del proyecto de investigación CONMICRO [319]. El objetivo principal era incorporar controladores auto-optimizados y capacidades cognitivas en la plataforma en tiempo real, modular, de red y reconfigurable. Además, la capacidad de interactuar con sistemas de middleware para sistemas globales de monitoreo distribuido, proveyendo la base para el desarrollo de sistemas inteligentes de control distribuido basados en capacidades cognitivas al más alto nivel.

En segundo lugar, se desarrollará un dispositivo en tiempo real para predecir fenómenos tales como: vibración y apagado mediante señales de vibración capturadas en procesos de microfresado con una relación de diámetros entre 30μm-500μm. En la actualidad, una serie de medidas de rugosidad geométrica, de desgaste y de superficie se realizan con dos técnicas de medición de alta resolución: una máquina de medición de coordenadas 3D y un dispositivo de medición de la rugosidad de la superficie. El objetivo principal es correlacionar las variaciones frecuencia-amplitud en tiempo real con las señales de vibración capturadas durante el proceso de fresado y microfresado. Basándose en estos resultados, actualmente está trabajando en las fases de diseño de la configuración de hardware y la implementación de los diferentes módulos para componer el dispositivo para tareas predictivas. Esta línea de investigación se desarrolla entre tres grupos de investigación: GAMHE (Madrid, España), CEFAS (Matanzas, Cuba) e IPK (Fraunhofer, Alemania) apoyando la interacción durante el periodo de investigación con equipos multidisciplinarios de diferentes latitudes.

En tercer lugar, en base a la misma interacción multidisciplinaria durante el período de formación, pero en este caso aplicada a una línea de investigación diferente son las bases de colaboración entre GAMHE; CEFAS y DLM (Warwick, Reino Unido). El objetivo principal es traer la arquitectura cognitiva artificial desarrollada en la presente tesis doctoral para abordar las estrategias de reconocimiento de patrones durante los procesos de ensamblaje de piezas para la industria del automóvil. La industria del automóvil es un escenario complejo con múltiples capas o estación durante el proceso de montaje, interactuando con máquinas, robots, dispositivos de medición, etc. al mismo tiempo. La nueva propuesta incluye una arquitectura cognitiva artificial adaptativa capaz de controlar en tiempo real la característica clave basada en indicadores clave específicos medidos en el proceso de ensamblaje de multinivel utilizando la clasificación de datos, la modelización, la optimización y la toma de decisiones. El nuevo diseño incluye técnicas Big Data, Cloud Computing y CPS que soportan los conceptos actuales introducidos en el programa FoF.

Por último, en el proyecto AM4G [320] se introduce una arquitectura de dos niveles (nivel local (modo local) y servidor de nube (modo global)) para estrategias de mantenimiento predictivo. Este sistema de monitoreo basado en la condición también se centra en los retos de los sistemas cibernéticos aplicados a las redes de fabricación. Además, se desarrolla un registro de alarma y evento modular y configurable. El objetivo principal es detectar qué acciones locales en la máquina deben realizarse inmediatamente (parada de emergencia, parada programada, velocidad lenta ...) en estrecha coordinación con el sistema PLC / CNC de la máquina. Además, las señales, los registros de alarma, etc., almacenados durante una ventana, se intercambian con el servidor de la nube. Por último, en el modo global, se utiliza información de la misma familia de máquinas para obtener una mejor configuración de coeficientes de algoritmo para cada máquina conectada en la red de fabricación, actualizando la configuración de parámetros (modelo local) en la conexión máquina-nube.

REFERENCES

[1] F. Nafz, J.-P. Steghöfer, H. Seebach, and W. Reif, "Formal Modeling and Verification of Self-* Systems Based on Observer/Controller-Architectures," in *Assurances for Self-Adaptive Systems: Principles, Models, and Techniques*, J. Cámara, R. de Lemos, C. Ghezzi, and A. Lopes, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 80-111, 2013.

[2] R. de Lemos, "The Conflict Between Self-* Capabilities and Predictability," in *Self-star Properties in Complex Information Systems: Conceptual and Practical Foundations,* O. Babaoglu *et al.*, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 219-228, 2005.

[3] S. Bououden, M. Chadli, and H. R. Karimi, "An ant colony optimization-based fuzzy predictive control approach for nonlinear processes," *Information Sciences,* vol. 299, pp. 143-158, 4/1/ 2015.

[4] K. Z. Gao, P. N. Suganthan, Q. K. Pan, T. J. Chua, C. S. Chong, and T. X. Cai, "An improved artificial bee colony algorithm for flexible job-shop scheduling problem with fuzzy processing time," *Expert Systems with Applications,* vol. 65, pp. 52-67, 12/15/ 2016.

[5] C. Pereira, L. Gonçalves, and M. Ferreira, "Exudate segmentation in fundus images using an ant colony optimization approach," *Information Sciences,* vol. 296, pp. 14-24, 3/1/ 2015.

[6] A. Sungthong and W. Assawinchaichote, "Particle Swam Optimization Based Optimal PID Parameters for Air Heater Temperature Control System," *Procedia Computer Science,* vol. 86, pp. 108-111, // 2016.

[7] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition and cross entropy methods for many-objective optimization," *Information Sciences,* vol. 282, no. 0, pp. 363-387, 10/20/ 2014.

[8] H. Karshenas, R. Santana, C. Bielza, and P. Larranaga, "Multiobjective Estimation of Distribution Algorithm Based on Joint Modeling of Objectives and Variables," *Evolutionary Computation, IEEE Transactions on,* vol. 18, no. 4, pp. 519-542, 2014.

[9] R.-E. Precup, P. Angelov, B. S. J. Costa, and M. Sayed-Mouchaweh, "An overview on fault diagnosis and nature-inspired optimal control of industrial process applications," *Computers in Industry,* vol. 74, pp. 75-94, 12// 2015.

[10] S. Barchinezhad and M. Eftekhari, "A new fuzzy and correlation based feature selection method for multiclass problems," *International Journal of Artificial Intelligence,* Article vol. 12, no. 2, pp. 24-41, 2014.

[11] X. Chen, S. Kar, and D. A. Ralescu, "Cross-entropy measure of uncertain variables," *Information Sciences,* vol. 201, pp. 53-60, 10/15/ 2012.

[12] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development,* vol. 44, no. 1.2, pp. 206-226, 1959.

[13] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction* (no. 1). MIT press Cambridge, 1998.

[14] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," *Advances in neural information processing systems,* vol. 19, p. 1, 2007.

[15] Y. Zou, T. Liu, D. Liu, and F. Sun, "Reinforcement learning-based real-time energy management for a hybrid tracked vehicle," *Applied Energy,* vol. 171, pp. 372-382, 6/1/ 2016.

[16] R. Sanz and J. Gómez, "Vindication of a Rigorous Cognitive Science," *Journal of Mind Theory,* vol. 0, no. 1, pp. 5-9, 2008.

[17] A. Meystel, "On Intelligence Control, Learning and Hierarchies," *IEEE Control Systems Magazine,* vol. 14, pp. 63-74, 1994.

[18] J. Albus, "Toward a Computational Theory of Mind," *Journal of Mind Theory,* vol. 0, no. 1, pp. 1-38, 2008.

[19] D. Vernon, G. Metta, and G. Sandini, "A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents," *IEEE Transactions on Evolutionary Computation,* vol. 11, no. 2, pp. 151-180, 2007.

[20] R. Huerta and T. Nowotny, "Fast and robust learning by reinforcement signals: explorations in the insect brain," *Neural computation,* vol. 21, no. 8, pp. 2123-2151, 2009.

[21] M. Ito, "Control of mental activities by internal models in the cerebellum," *Nature Reviews Neuroscience,* vol. 9, no. 4, pp. 304-313, 2008.

[22] M. I. Rabinovich, P. Varona, A. I. Selverston, and H. D. I. Abarbanel, "Dynamical principles in neuroscience," *Reviews of Modern Physics,* vol. 78, no. 4, 2006.

[23] Z. W. Pylyshyn, "Computation and cognition: Toward a foundation for cognitive science," *Computation and Cognition: Toward a Foundation for Cognitive Science,* 1984.

[24] E. Thelen and L. B. Smith, *A Dynamic Systems Approach to the Development of Cognition and Action* (Bradford Book Series in Cognitive Psychology). Cambridge, 1994.

[25] R. A. Felix, E. N. Sanchez, and A. G. Loukianov, "Neural block control for synchronous generators," *Engineering Applications of Artificial Intelligence,* vol. 22, no. 8, pp. 1159-1166, 2009.

[26] A. Weber and F. J. Varela, "Life after Kant: Natural purposes and the autopoietic foundations of biological individuality," *Phenomenology and the Cognitive Sciences,* vol. 1, no. 2, pp. 97-125, 2002.

[27] T. Froese, "On the role of AI in the ongoing paradigm shift within the cognitive sciences," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 4850 LNAI, ed, pp. 63-75, 2007.

[28] R. Sanz, I. López, M. Rodríguez, and C. Hernández, "Principles for consciousness in integrated cognitive control," *Neural Networks,* vol. 20, no. 9, pp. 938-946, 2007.

[29] D. D. Luxton, "Chapter 1 - An Introduction to Artificial Intelligence in Behavioral and Mental Health Care," in *Artificial Intelligence in Behavioral and Mental Health Care*San Diego: Academic Press, pp. 1-26, 2016.

[30] W. Banzhaf, "Artificial Intelligence: Genetic Programming A2 - Wright, James D," in *International Encyclopedia of the Social & Behavioral Sciences (Second Edition)*Oxford: Elsevier, pp. 41-45, 2015.

[31] J. Bongard, "Biologically Inspired Computing," *Computer,* vol. 42, no. 4, pp. 95-98, 2009.

[32] N. Pillay, A. P. Engelbrecht, A. Abraham, M. C. du Plessis, V. Snášel, and A. K. Muda, *Advances in Nature and Biologically Inspired Computing* (no. 419). Springer International Publishing, 2016.

[33] J. Ladyman, J. Lambert, and K. Wiesner, "What is a complex system?," *European Journal for Philosophy of Science,* journal article vol. 3, no. 1, pp. 33-67, 2012.

[34] L. Wei and D. Luo, "A biologically inspired computational approach to model top-down and bottom-up visual attention," *Optik - International Journal for Light and Electron Optics,* vol. 126, no. 5, pp. 522-529, 3// 2015.

[35] T. Tang and H. Qiao, "Exploring biologically inspired shallow model for visual classification," *Signal Processing,* vol. 105, pp. 1-11, 12// 2014.

[36] J. M. Chein and W. Schneider, "The Brain's Learning and Control Architecture," *Current Directions in Psychological Science,* vol. 21, no. 2, pp. 78-84, April 1, 2012 2012.

[37] J. Laird, "Soar Applications," in *The Soar Cognitive Architecture*: MIT Press, pp. 307-324, 2012.

[38] E. Feigenbaum, "Information Theories of Human Verbal Learning," Ph. D. Thesis, Carnegie Mellon University, 1960.

[39] J. R. Anderson and G. H. Bower, *Human Associative memory*. Washington: Winston and Sons, 1973.

[40] J. R. Anderson, *The Architecture of Cognition*. Taylor & Francis, 2013.

[41] H. Asadi, H. Volos, M. M. Marefat, and T. Bose, "Metacognition and the next generation of cognitive radio engines," *Communications Magazine, IEEE,* vol. 54, no. 1, pp. 76-82, 2016.

[42] A. S. Silva Simoes, E. L. Colombini, and C. H. Costa Ribeiro, "CONAIM: A Conscious Attention-Based Integrated Model for Human-Like Robots," *Systems Journal, IEEE,* vol. PP, no. 99, pp. 1-12, 2016.

[43] O. J. Romero-López, "Self-organized and Evolvable Cognitive Architecture for Intelligent Agents and Multi-Agent Systems," in *Applications of Evolutionary Computation*, vol. 6024, C. Di Chio *et al.*, Eds. (Lecture Notes in Computer Science: Springer Berlin Heidelberg, pp. 392-401, 2010.

[44] A. Bannat *et al.*, "Artificial Cognition in Production Systems," *Automation Science and Engineering, IEEE Transactions on,* vol. 8, no. 1, pp. 148-174, 2011.

[45] A. Sánchez Boza, R. H. Guerra, and A. Gajate, "Artificial cognitive control system based on the shared circuits model of sociocognitive capacities. A first approach," *Engineering Applications of Artificial Intelligence,* vol. 24, no. 2, pp. 209-219, 3// 2011.

[46] M. Bazhenov, R. Huerta, and B. H. Smith, "A Computational Framework for Understanding Decision Making through Integration of Basic Learning Rules," *The Journal of Neuroscience,* vol. 33, no. 13, pp. 5686-5697, March 27, 2013 2013.

[47] M. Khamassi, S. Lallée, P. Enel, E. Procyk, and P. F. Dominey, "Robot Cognitive Control with a Neurophysiologically Inspired Reinforcement Learning Model," *Frontiers in Neurorobotics,* vol. 5, p. 1, 07/12 2011.

[48] D. Bruckner, H. Zeilinger, and D. Dietrich, "Cognitive Automation&#x2014;Survey of Novel Artificial General Intelligence Methods for the Automation of Human Technical Environments," *Industrial Informatics, IEEE Transactions on,* vol. 8, no. 2, pp. 206-215, 2012.

[49] S. Borgo, "An ontological approach for reliable data integration in the industrial domain," *Computers in Industry,* vol. 65, no. 9, pp. 1242-1252, 12// 2014.

[50] M. Fatemi and S. Haykin, "Cognitive Control: Theory and Application," *Access, IEEE,* vol. 2, pp. 698-710, 2014.

[51] J. R. Anderson, D. Bothell, M. D. Byrne, S. Douglass, C. Lebiere, and Y. Qin, "An Integrated Theory of the Mind," *Psychological Review,* vol. 111, no. 4, pp. 1036-1060, 2004.

[52] R. S. Chong and R. E. Wray, "Unified Theories of Cognition," in *Encyclopedia of Cognitive Science*: John Wiley & Sons, Ltd, 2006.

[53] A. Newell, *Unified Theories of Cognition*. Cambridge, MA: Harvard UP, 1994.

[54] P. Langley, J. E. Laird, and S. Rogers, "Cognitive architectures: Research issues and challenges," *Cognitive Systems Research,* vol. 10, no. 2, pp. 141-160, 6// 2009.

[55] M. Rabinovich, I. Tristan, and P. Varona, "Neural Dynamics of Attentional Cross-Modality Control," *PLoS ONE,* vol. 8, no. 5, p. e64406, 2013.

[56] R. R. Llinás and S. Roy, "The ‘prediction imperative’ as the basis for self-awareness," *Philosophical Transactions of the Royal Society of London B: Biological Sciences,* vol. 364, no. 1521, pp. 1301-1307, 2009-05-12 00:00:00 2009.

[57] S. Hurley, "The shared circuits model (SCM): How control, mirroring, and simulation can enable imitation, deliberation, and mindreading," *Behavioral and Brain Sciences,* vol. 31, no. 01, pp. 1-22, 2008.

[58] J. I. M. Carpendale and C. Lewis, "Mirroring cannot account for understanding action," *Behavioral and Brain Sciences,* Note vol. 31, no. 1, pp. 23-24, 2008.

[59] T. Makino, "Failure, instead of inhibition, should be monitored for the distinction of self/other and actual/possible actions," *Behavioral and Brain Sciences,* Note vol. 31, no. 1, pp. 32-33, 2008.

[60] D. Kit, D. H. Ballard, B. Sullivan, and C. A. Rothkopf, "A hierarchical modular architecture for embodied cognition," *Multisensory Research,* Article vol. 26, no. 1-2, pp. 177-204, 2013.

[61] M. E. Bratman, D. J. Israel, and M. E. Pollack, "Plans and resource-bounded practical reasoning," *Computational Intelligence,* vol. 4, no. 3, pp. 349-355, 1988.

[62] J. F. Lehman, J. E. Laird, and P. Rosenbloom, "A gentle introduction to Soar, an architecture for human cognition," *Invitation to cognitive science,* vol. 4, pp. 212-249, 1996.

[63] P. Langley, "Cognitive architectures and general intelligent systems," *AI Magazine,* Review vol. 27, no. 2, pp. 33-34, 2006.

[64] R. Sun, E. Merrill, and T. Peterson, "From implicit skills to explicit knowledge: a bottom-up model of skill learning," *Cognitive Science,* vol. 25, no. 2, pp. 203-244, 2001.

[65] R. Sun and X. Zhang, "Accounting for a variety of reasoning data within a cognitive architecture," *Journal of Experimental & Theoretical Artificial Intelligence,* vol. 18, no. 2, pp. 169-191, 2006/06/01 2006.

[66] Z. Mathews, S. B. i Badia, and P. F. M. J. Verschure, "PASAR: An integrated model of prediction, anticipation, sensation, attention and response for artificial sensorimotor systems," *Information Sciences,* vol. 186, no. 1, pp. 1-19, 3/1/ 2012.

[67] S. Franklin, T. Madl, S. D'Mello, and J. Snaider, "LIDA: A Systems-level Architecture for Cognition, Emotion, and Learning," *Autonomous Mental Development, IEEE Transactions on,* vol. 6, no. 1, pp. 19-41, 2014.

[68] V. Cutsuridis and J. G. Taylor, "A Cognitive Control Architecture for the Perception–Action Cycle in Robots and Agents," *Cognitive Computation,* journal article vol. 5, no. 3, pp. 383-395, 2013.

[69] J. R. Anderson, *Rules of the Mind*. L. Erlbaum Associates, 1993.

[70] A. Sanchez-Boza and R. H. Guerra, "A first approach to artificial cognitive control system implementation based on the shared circuits model of sociocognitive capacities," *ICIC Express Letters,* Article vol. 4, no. 5 B, pp. 1741-1746, 2010.

[71] S. Kopácsi, G. L. Kovács, and J. Nacsa, "Some aspects of dynamic 3D representation and control of industrial processes via the Internet," *Computers in Industry,* vol. 64, no. 9, pp. 1282-1289, 12// 2013.

[72] M. Minsky, *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. Simon & Schuster, 2007.

[73] J. H. Taylor and A. F. Sayda, "An Intelligent Architecture for Integrated Control and Asset Management for Industrial Processes," in *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, 2005, pp. 1397-1404.

[74] H. Wang, F. R. Yu, L. Zhu, T. Tang, and B. Ning, "A Cognitive Control Approach to Communication-Based Train Control Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1676-1689, 2015.

[75] Y. Bo, L. Hai-feng, M. Lin, and W. Xun-da, "Automatic brain cognitive control detection method," in *Software Intelligence Technologies and Applications & International Conference on Frontiers of Internet of Things 2014, International Conference on*, 2014, pp. 256-260.

[76] A. Hussain, "Cognitive computation: A case study in cognitive control of autonomous systems and some future directions," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, 2013, pp. 1-6.

[77] A. D. Nuovo, V. M. D. L. Cruz, and A. Cangelosi, "A Deep Learning Neural Network for Number Cognition: A bi-cultural study with the iCub," in *Development and Learning and Epigenetic Robotics (ICDL-EpiRob), 2015 Joint IEEE International Conference on*, 2015, pp. 320-325.

[78] N. Chungoora *et al.*, "A model-driven ontology approach for manufacturing system interoperability and knowledge sharing," *Computers in Industry,* vol. 64, no. 4, pp. 392-401, 5// 2013.

[79] P. Hong-Seok, R. R. Z. Ur, and T. Ngoc-Hien, "A swarm of cognitive agents for controlling smart manufacturing systems," in *Natural Computation (ICNC), 2015 11th International Conference on*, 2015, pp. 861-867.

[80] A. Fiaschetti, A. Pietrabissa, and F. D. Priscoli, "Towards manufacturing 2.0: An innovative architecture for the Factory of the Future," in *Networks and Communications (EuCNC), 2015 European Conference on*, 2015, pp. 450-454.

[81] R. Teti, "Advanced IT Methods of Signal Processing and Decision Making for Zero Defect Manufacturing in Machining," *Procedia CIRP,* vol. 28, pp. 3-15, // 2015.

[82] R. E. Haber, C. Juanes, R. del Toro, and G. Beruvides, "Artificial cognitive control with self-x capabilities: A case study of a micro-manufacturing process," *Computers in Industry,* 2015.

[83] G. Beruvides, C. Juanes, F. Castano, and R. E. Haber, "A self-learning strategy for artificial cognitive control systems," in *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*, 2015, pp. 1180-1185.

[84] J. Lee, B. Bagheri, and H. A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," (in English), *Manufacturing Letters,* Article vol. 3, pp. 18-23, 2015.

[85] S. Liu, "Multi-objective optimization design method for the machine tool's structural parts based on computer-aided engineering," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 78, no. 5, pp. 1053-1065, 2014.

[86] H. Nakayama, Y. Yun, and M. Yoon, "Basic Concepts of Multi-objective Optimization," in *Sequential Approximate Multiobjective Optimization Using Computational Intelligence*Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1-15, 2009.

[87] J. Christensen and C. Bastien, "Chapter | seven - Heuristic and Meta-Heuristic Optimization Algorithms," in *Nonlinear Optimization of Vehicle Safety Structures*Oxford: Butterworth-Heinemann, pp. 277-314, 2016.

[88] H. M. Pandey and A. Gajendran, "Function Optimization Using Robust Simulated Annealing," in *Information Systems Design and Intelligent Applications: Proceedings of Third International Conference INDIA 2016, Volume 3*, C. S. Satapathy, K. J. Mandal, K. S. Udgata, and V. Bhateja, Eds. New Delhi: Springer India, pp. 347-355, 2016.

[89] M. R. Bonyadi and Z. Michalewicz, "Evolutionary Computation for Real-World Problems," in *Challenges in Computational Statistics and Data Mining*, S. Matwin and J. Mielniczuk, Eds. Cham: Springer International Publishing, pp. 1-24, 2016.

[90] E. Kayacan and M. A. Khanesar, "Chapter 8 - Hybrid Training Method for Type-2 Fuzzy Neural Networks Using Particle Swarm Optimization," in *Fuzzy Neural Networks for Real Time Control Applications*: Butterworth-Heinemann, pp. 133-160, 2016.

[91] I. Giagkiozis and P. J. Fleming, "Methods for multi-objective optimization: An analysis," *Information Sciences,* vol. 293, pp. 338-350, 2/1/ 2015.

[92] S. Mirjalili and A. Lewis, "Novel frameworks for creating robust multi-objective benchmark problems," *Information Sciences,* vol. 300, pp. 158-192, 4/10/ 2015.

[93] S. Bharathi Raja and N. Baskar, "Optimization techniques for machining operations: a retrospective research based on various mathematical models," (in English), *The International Journal of Advanced Manufacturing Technology,* vol. 48, no. 9-12, pp. 1075-1090, 2010/06/01 2010.

[94] M. Chandrasekaran, M. Muralidhar, C. M. Krishna, and U. S. Dixit, "Application of soft computing techniques in machining performance prediction and optimization: A literature review," (in English), *International Journal of Advanced Manufacturing Technology,* vol. 46, no. 5-8, pp. 445-464, 2010.

[95] A. López-Jaimes and C. A. Coello Coello, "Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems," *Information Sciences,* vol. 277, pp. 1-20, 9/1/ 2014.

[96] D. Gao, O. Khamisov, and D. Sidorov, "Editorial for special issue on methods of optimisation and their applications," *International Journal of Artificial Intelligence,* Article vol. 13, no. 1, pp. 120-122, 2015.

[97] M. S. Kiran, H. Hakli, M. Gunduz, and H. Uguz, "Artificial bee colony algorithm with variable search strategy for continuous optimization," *Information Sciences,* vol. 300, pp. 140-157, 4/10/ 2015.

[98] H. Wang, Z. Wu, S. Rahnamayan, H. Sun, Y. Liu, and J.-s. Pan, "Multi-strategy ensemble artificial bee colony algorithm," *Information Sciences,* vol. 279, pp. 587-603, 9/20/ 2014.

[99] Y. Wang and Y. Yang, "Particle swarm optimization with preference order ranking for multi-objective optimization," *Information Sciences,* vol. 179, no. 12, pp. 1944-1959, 5/30/ 2009.

[100] J. Zhang, J. Zhuang, H. Du, and S. a. Wang, "Self-organizing genetic algorithm based tuning of PID controllers," *Information Sciences,* vol. 179, no. 7, pp. 1007-1018, 3/15/ 2009.

[101] H. Jiang, J. Chen, and T. Liu, "Multi-objective design of an FBG sensor network using an improved Strength Pareto Evolutionary Algorithm," *Sensors and Actuators A: Physical,* vol. 220, pp. 230-236, 12/1/ 2014.

[102] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *Evolutionary Computation, IEEE Transactions on,* vol. 10, no. 5, pp. 477-506, 2006.

[103] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173-195, 2000.

[104] T. Chen, K. Tang, G. Chen, and X. Yao, "Analysis of computational time of simple estimation of distribution algorithms," (in English), *IEEE Transactions on Evolutionary Computation,* vol. 14, no. 1, pp. 1-22, 2010, Art. no. 5401139.

[105] I. Mukherjee and P. K. Ray, "A review of optimization techniques in metal cutting processes," (in English), *Computers and Industrial Engineering,* vol. 50, no. 1-2, pp. 15-34, 2006.

[106] R. E. Haber, R. M. Del Toro, and A. Gajate, "Optimal fuzzy control system using the cross-entropy method. A case study of a drilling process," (in English), *Information Sciences,* vol. 180, no. 14, pp. 2777-2792, 2010.

[107] R. E. Precup, R. C. David, E. M. Petriu, M. B. Radac, and S. Preitl, "Adaptive GSA-based optimal tuning of PI controlled servo systems with reduced process parametric sensitivity, robust stability and controller robustness," (in English), *IEEE Transactions on Cybernetics,* vol. 44, no. 11, pp. 1997-2009, 2014, Art. no. 6754137.

[108] R. E. Precup *et al.*, "Fuzzy logic-based adaptive gravitational search algorithm for optimal tuning of fuzzy-controlled servo systems," *Control Theory & Applications, IET,* vol. 7, no. 1, pp. 99-107, 2013.

[109] R. Rubinstein, "Semi-iterative minimum cross-entropy algorithms for rare-events, counting, combinatorial and integer programming," (in English), *Methodology and Computing in Applied Probability,* vol. 10, no. 2, pp. 121-178, 2008.

[110] R. Y. Rubinstein, "A Stochastic Minimum Cross-Entropy Method for Combinatorial Optimization and Rare-event Estimation*," *Methodology and Computing in Applied Probability,* journal article vol. 7, no. 1, pp. 5-50, 2005.

[111] J. Tabor and P. Spurek, "Cross-entropy clustering," *Pattern Recognition,* vol. 47, no. 9, pp. 3046-3059, 2014.

[112] R. E. Precup, M. B. Radac, M. L. Tomescu, E. M. Petriu, and S. Preitl, "Stable and convergent iterative feedback tuning of fuzzy controllers for discrete-time SISO systems," (in English), *Expert Systems with Applications,* vol. 40, no. 1, pp. 188-199, 2013.

[113] C. Fu, M. A. Olivares-Mendez, R. Suarez-Fernandez, and P. Campoy, "Monocular Visual-Inertial SLAM-based collision avoidance strategy for Fail-Safe UAV using Fuzzy Logic Controllers: Comparison of two Cross-Entropy Optimization approaches," (in English), *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 73, no. 1-4, pp. 513-533, 2014.

[114] J. Bekker and C. Aldrich, "The cross-entropy method in multi-objective optimisation: An assessment," *European Journal of Operational Research*, vol. 211, no. 1, pp. 112-121, 5/16/ 2011.

[115] C.-X. You, J.-Q. Huang, and F. Lu, "Recursive reduced kernel based extreme learning machine for aero-engine fault pattern recognition," *Neurocomputing*, vol. 214, pp. 1038-1045, 11/19/ 2016.

[116] L. L. Minku, "Which machine learning method do you need? A2 - Menzies, Tim," in *Perspectives on Data Science for Software Engineering*, L. Williams and T. Zimmermann, Eds. Boston: Morgan Kaufmann, pp. 155-159, 2016.

[117] S. Xu and J. Wang, "Dynamic Extreme Learning Machine for Data Stream Classification," *Neurocomputing*, vol. 238, pp. 433-449, 17/05/2017 2017.

[118] S. Lim, C. S. Tucker, and S. Kumara, "An unsupervised machine learning model for discovering latent infectious diseases using social media data," *Journal of Biomedical Informatics*, vol. 66, pp. 82-94, 2// 2017.

[119] F. Xing and L. Yang, "Chapter 4 - Machine learning and its application in microscopic image analysis A2 - Wu, Guorong," in *Machine Learning and Medical Imaging*, D. Shen and M. R. Sabuncu, Eds.: Academic Press, pp. 97-127, 2016.

[120] V. N. Gudivada, M. T. Irfan, E. Fathi, and D. L. Rao, "Chapter 5 - Cognitive Analytics: Going Beyond Big Data Analytics and Machine Learning," in *Handbook of Statistics*, vol. Volume 35, V. V. R. V. G. Venkat N. Gudivada and C. R. Rao, Eds.: Elsevier, pp. 169-205, 2016.

[121] J.-S. Chou and N.-T. Ngo, "Time series analytics using sliding window metaheuristic optimization-based machine learning system for identifying building energy consumption patterns," *Applied Energy*, vol. 177, pp. 751-770, 9/1/ 2016.

[122] D. A. Forsyth and J. Ponce, "A modern approach," *Computer vision: a modern approach*, pp. 88-101, 2003.

[123] X. Zhang, T. Yu, B. Yang, and L. Cheng, "Accelerating bio-inspired optimizer with transfer reinforcement learning for reactive power optimization," *Knowledge-Based Systems*, vol. 116, pp. 26-38, 1/15/ 2017.

[124] Y. Peng and B.-L. Lu, "Discriminative extreme learning machine with supervised sparsity preserving for image classification," *Neurocomputing*, 2017.

[125] C. Voyant *et al.*, "Machine learning methods for solar radiation forecasting: A review," *Renewable Energy*, vol. 105, pp. 569-582, 5// 2017.

[126] K. Chen, Q. Lv, Y. Lu, and Y. Dou, "Robust regularized extreme learning machine for regression using iteratively reweighted least squares," *Neurocomputing*, vol. 230, pp. 345-358, 3/22/ 2017.

[127] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: a survey," *J. Artif. Int. Res.,* vol. 4, no. 1, pp. 237-285, 1996.

[128] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *NIPS*, 1999, vol. 99, pp. 1057-1063.

[129] T. Jaakkola, S. P. Singh, and M. I. Jordan, "Reinforcement learning algorithm for partially observable Markov decision problems," *Advances in neural information processing systems*, pp. 345-352, 1995.

[130] J. P. Rust, "A comparison of policy iteration methods for solving continuous-state, infinite-horizon Markovian decision problems using random, quasi-random, and deterministic discretizations," 1997.

[131] "Q-Learning," in *Encyclopedia of the Sciences of Learning*, N. M. Seel, Ed. Boston, MA: Springer US, pp. 2741-2741, 2012.

[132] "Greedy Q-Learning," in *Encyclopedia of the Sciences of Learning*, N. M. Seel, Ed. Boston, MA: Springer US, pp. 1388-1388, 2012.

[133] K. Wang, T. Y. Chai, and W.-C. Wong, "Routing, power control and rate adaptation: A Q-learning-based cross-layer design," *Computer Networks,* vol. 102, pp. 20-37, 6/19/ 2016.

[134] J. Kober and J. Peters, "Reinforcement Learning in Robotics: A Survey," in *Learning Motor Skills: From Algorithms to Robot Experiments*Cham: Springer International Publishing, pp. 9-67, 2014.

[135] Á. Serrano-Laguna, I. Martínez-Ortiz, J. Haag, D. Regan, A. Johnson, and B. Fernández-Manjón, "Applying standards to systematize learning analytics in serious games," *Computer Standards & Interfaces,* vol. 50, pp. 116-123, 2// 2017.

[136] S. Wen, X. Chen, C. Ma, H. K. Lam, and S. Hua, "The -learning obstacle avoidance algorithm based on EKF-SLAM for NAO autonomous walking under unknown environments," *Robotics and Autonomous Systems,* vol. 72, pp. 29-36, 10// 2015.

[137] J.-L. Loyer, E. Henriques, M. Fontul, and S. Wiseall, "Comparison of Machine Learning methods applied to the estimation of manufacturing cost of jet engine components," *International Journal of Production Economics,* vol. 178, pp. 109-119, 8// 2016.

[138] Y. Qin, "Chapter 1 - Overview of Micro-Manufacturing," in *Micro-Manufacturing Engineering and Technology*Boston: William Andrew Publishing, pp. 1-23, 2010.

[139] V. K. Jain, *Micromanufacturing Processes*. CRC Press, 2012.

[140] Z. Li, D. Yang, W. Hao, T. Wu, S. Wu, and X. Li, "A novel technique for micro-hole forming on skull with the assistance of ultrasonic vibration," *Journal of the Mechanical Behavior of Biomedical Materials,* vol. 57, pp. 1-13, 4// 2016.

[141] R. Zhao, J. Q. Han, B. B. Liu, and M. Wan, "Interaction of forming temperature and grain size effect in micro/meso-scale plastic deformation of nickel-base superalloy," *Materials & Design,* vol. 94, pp. 195-206, 3/15/ 2016.

[142] H. Zhang, P. D'Angelo Nunes, M. Wilhelm, and K. Rezwan, "Hierarchically ordered micro/meso/macroporous polymer-derived ceramic monoliths fabricated by freeze-casting," *Journal of the European Ceramic Society,* vol. 36, no. 1, pp. 51-58, 1// 2016.

[143] C. Hopmann and T. Fischer, "New plasticising process for increased precision and reduced residence times in injection moulding of micro parts," *CIRP Journal of Manufacturing Science and Technology,* vol. 9, pp. 51-56, 5// 2015.

[144] F. Schmitt and A. Olowinsky, "Chapter 26 - Laser Beam Micro-joining A2 - Qin, Yi," in *Micromanufacturing Engineering and Technology (Second Edition)*Boston: William Andrew Publishing, pp. 613-635, 2015.

[145] S. M. Goushegir, J. F. dos Santos, and S. T. Amancio-Filho, "Failure and fracture micro-mechanisms in metal-composite single lap joints produced by welding-based joining techniques," *Composites Part A: Applied Science and Manufacturing,* vol. 81, pp. 121-128, 2// 2016.

[146] X. Dong, H. Li, M. Chen, Y. Wang, and Q. Yu, "Plasma treatment of dentin surfaces for improving self-etching adhesive/dentin interface bonding," *Clinical Plasma Medicine,* vol. 3, no. 1, pp. 10-16, 6// 2015.

[147] V. A. Lifton, G. Lifton, and S. Simon, "Options for additive rapid prototyping methods (3D printing) in MEMS technology," *Rapid Prototyping Journal,* vol. 20, no. 5, pp. 403-412, 2014.

[148] J. A. Palmer, J. D. Williams, T. Lemp, T. M. Lehecka, F. Medina, and R. B. Wicker, "Advancing three-dimensional MEMS by complimentary laser micro manufacturing," 2006, vol. 6109, pp. 61090A-61090A-8.

[149] A. Sharon, A. Bilsing, G. Lewis, and X. Zhang, "Manufacturing of 3D microstructures using novel UPSAMS process (ultra precision manufacturing of self-assembled micro systems)," in *Micro Electro Mechanical Systems, 2003. MEMS-03 Kyoto. IEEE The Sixteenth Annual International Conference on*, 2003, pp. 542-545.

[150] R. Leach, "Chapter 1 - Introduction to Metrology for Advanced Manufacturing and Micro- and Nanotechnology," in *Fundamental Principles of Engineering Nanometrology (Second Edition)*Oxford: William Andrew Publishing, pp. 1-6, 2014.

[151] M. J. Jackson *et al.*, "Fundamentals of Machining," in *Machining with Nanomaterials*, J. M. Jackson and S. J. Morrell, Eds. Cham: Springer International Publishing, pp. 1-35, 2015.

[152] H. Reinecke, C. Müller, and S. M. Karazi, "Micromachining," in *Reference Module in Materials Science and Materials Engineering*: Elsevier, 2016.

[153] M. J. Jackson *et al.*, "Micromachining from a Materials Perspective," in *Machining with Nanomaterials*, J. M. Jackson and S. J. Morrell, Eds. Cham: Springer International Publishing, pp. 77-127, 2015.

[154] E. Graham, M. Mehrpouya, R. Nagamune, and S. S. Park, "Robust prediction of chatter stability in micro milling comparing edge theorem and LMI," *CIRP Journal of Manufacturing Science and Technology,* vol. 7, no. 1, pp. 29-39, // 2014.

[155] X. Lu, Z. Jia, X. Wang, G. Li, and Z. Ren, "Three-dimensional dynamic cutting forces prediction model during micro-milling nickel-based superalloy," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 81, no. 9, pp. 2067-2086, 2015.

[156] E. Kuram and B. Ozcelik, "Micro Milling," in *Modern Mechanical Engineering: Research, Development and Education*, P. J. Davim, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 325-365, 2014.

[157] G. Kiswanto, D. L. Zariatin, and T. J. Ko, "The effect of spindle speed, feed-rate and machining time to the surface roughness and burr formation of Aluminum Alloy 1100 in micro-milling operation," *Journal of Manufacturing Processes,* vol. 16, no. 4, pp. 435-450, 10// 2014.

[158] J. Cheng *et al.*, "Experimental study on a novel minimization method of top burr formation in micro-end milling of Ti-6Al-4V," *The International Journal of Advanced Manufacturing Technology,* journal article pp. 1-21, 2016.

[159] W. Pei *et al.*, "Influence of Abrasive Particle Movement in Micro USM," *Procedia CIRP,* vol. 6, pp. 551-555, // 2013.

[160] Z. Yu, C. Ma, C. An, J. Li, and D. Guo, "Prediction of tool wear in micro USM," *CIRP Annals - Manufacturing Technology,* vol. 61, no. 1, pp. 227-230, // 2012.

[161] J. Romano, L. Ladani, J. Razmi, and M. Sadowski, "Temperature distribution and melt geometry in laser and electron-beam melting processes – A comparison among common materials," *Additive Manufacturing,* vol. 8, pp. 1-11, 10// 2015.

[162] P. Penchev, S. Dimov, and D. Bhaduri, "Experimental investigation of 3D scanheads for laser micro-processing," *Optics & Laser Technology,* vol. 81, pp. 55-59, 7// 2016.

[163] R. Menon and P. Y. Nabhiraj, "High speed micro-fabrication using inductively coupled plasma ion source based focused ion beam system," *Vacuum,* vol. 111, pp. 166-169, 1// 2015.

[164] Y. S. Liang and J. A. Shih, "Electrical Discharge Machining," in *Analysis of Machining and Machine Tools*Boston, MA: Springer US, pp. 167-179, 2016.

[165] X. Fu, Q. Zhang, L. Gao, Q. Liu, K. Wang, and Y.-W. Zhang, "A novel micro-EDM—piezoelectric self-adaptive micro-EDM," *The International Journal of Advanced Manufacturing Technology,* journal article pp. 1-8, 2015.

[166] S. Hinderer, S. L. Layland, and K. Schenke-Layland, "ECM and ECM-like materials — Biomaterials for applications in regenerative medicine and cancer therapy," *Advanced Drug Delivery Reviews,* 2015.

[167] A. Spieser and A. Ivanov, "Design of a pulse power supply unit for micro-ECM," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 78, no. 1, pp. 537-547, 2014.

[168] M. Angelozzi *et al.*, "Composite ECM–alginate microfibers produced by microfluidics as scaffolds with biomineralization potential," *Materials Science and Engineering: C,* vol. 56, pp. 141-153, 11/1/ 2015.

[169] S. S. Joshi and D. Marla, "11.15 - Electrochemical Micromachining A2 - Yilbas, Saleem HashmiGilmar Ferreira BatalhaChester J. Van TyneBekir," in *Comprehensive Materials Processing*Oxford: Elsevier, pp. 373-403, 2014.

[170] "Jejunum," in *Merriam-Webster's dictionary*, 11th ed. Springfield, MA: Merriam-Webster, 2003.

[171] C. H. Kung and A. S. lvberg, "Activity modeling and behavior modeling," presented at the Proc. of the IFIP WG 8.1 working conference on Information systems design methodologies: improving the practice, Noordwijkerhout, Netherlands, 1986.

[172] R. Quiza, O. López-Armas, and J. P. Davim, "Introduction," in *Hybrid Modeling and Optimization of Manufacturing: Combining Artificial Intelligence and Finite Element Method*Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1-11, 2012.

[173] S. N. Ahmed, "8 - Signal processing," in *Physics and Engineering of Radiation Detection (Second Edition)*: Elsevier, pp. 477-540, 2015.

[174] I. F. Apolinário and P. S. R. Diniz, "Chapter 1 - Introduction to Signal Processing Theory," in *Academic Press Library in Signal Processing*, vol. Volume 1, J. A. K. S. R. C. Paulo S.R. Diniz and T. Sergios, Eds.: Elsevier, pp. 3-28, 2014.

[175] Z. Zhang and J. C. Moore, "Chapter 3 - Filter Design," in *Mathematical and Physical Fundamentals of Climate Change*Boston: Elsevier, pp. 79-109, 2015.

[176] U. Meyer-Baese, "Infinite Impulse Response (IIR) Digital Filters," in *Digital Signal Processing with Field Programmable Gate Arrays*Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 225-304, 2014.

[177] O. A. Akanbi, I. S. Amiri, and E. Fazeldehkordi, "Chapter 4 - Feature Extraction," in *A Machine-Learning Approach to Phishing Detection and Defense*Boston: Syngress, pp. 45-54, 2015.

[178] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, "Chapter 4 - Processing in the feature and model domains," in *Robust Automatic Speech Recognition*Oxford: Academic Press, pp. 65-106, 2016.

[179] R. Quiza and J. P. Davim, "Computational modeling of machining systems," in *Intelligent Machining: Modeling and Optimization ofthe Machining Processes and Systems*, T. Özel and J. P. Davim, Eds. London, UK: ISTE Publishers, pp. 173–213, 2009.

[180] G. D. Pelegrina, L. T. Duarte, and C. Jutten, "Blind source separation and feature extraction in concurrent control charts pattern recognition: Novel analyses and a comparison of different methods," *Computers & Industrial Engineering*, vol. 92, pp. 105-114, 2// 2016.

[181] E.-J. Wang, C.-Y. Lin, and T.-S. Su, "Electricity monitoring system with fuzzy multi-objective linear programming integrated in carbon footprint labeling system for manufacturing decision making," *Journal of Cleaner Production*, vol. 112, Part 5, pp. 3935-3951, 1/20/ 2016.

[182] A. Singh, M. K. Dutta, M. ParthaSarathi, V. Uher, and R. Burget, "Image processing based automatic diagnosis of glaucoma using wavelet features of segmented optic disc from fundus image," *Computer Methods and Programs in Biomedicine*, vol. 124, pp. 108-120, 2// 2016.

[183] D. W. Kim, Y. S. Lee, M. S. Park, and C. N. Chu, "Tool life improvement by peck drilling and thrust force monitoring during deep-micro-hole drilling of steel," *International Journal of Machine Tools and Manufacture*, vol. 49, no. 3–4, pp. 246-255, 3// 2009.

[184] Z.-D. Xu, F.-H. Xu, and X. Chen, "Vibration suppression on a platform by using vibration isolation and mitigation devices," *Nonlinear Dynamics,* journal article vol. 83, no. 3, pp. 1341-1353, 2015.

[185] Q. Ren, M. Balazinski, L. Baron, K. Jemielniak, R. Botez, and S. Achiche, "Type-2 fuzzy tool condition monitoring system based on acoustic emission in micromilling," *Information Sciences,* vol. 255, pp. 121-134, 1/10/ 2014.

[186] Z. Liao and D. A. Axinte, "On monitoring chip formation, penetration depth and cutting malfunctions in bone micro-drilling via acoustic emission," *Journal of Materials Processing Technology,* vol. 229, pp. 82-93, 3// 2016.

[187] Z. Shen, C. Y. Tan, K. Yao, L. Zhang, and Y. F. Chen, "A miniaturized wireless accelerometer with micromachined piezoelectric sensing element," *Sensors and Actuators A: Physical,* vol. 241, pp. 113-119, 4/15/ 2016.

[188] "Wavelet Transform," in *Encyclopedia of Biometrics*, S. Z. Li and A. Jain, Eds. Boston, MA: Springer US, pp. 1407-1408, 2009.

[189] K. K. Shukla and A. K. Tiwari, "PVM Implementation of DWT-Based Image Denoising," in *Efficient Algorithms for Discrete Wavelet Transform: With Applications to Denoising and Fuzzy Inference Systems*London: Springer London, pp. 51-59, 2013.

[190] M. S. Reis, P. M. Saraiva, and B. R. Bakshi, "2.03 - Denoising and Signal-to-Noise Ratio Enhancement: Wavelet Transform and Fourier Transform A2 - Walczak, Steven D. BrownRomá TaulerBeata," in *Comprehensive Chemometrics*Oxford: Elsevier, pp. 25-55, 2009.

[191] N. E. Huang and S. S. P. Shen, *Hilbert-Huang Transform and Its Applications,* 2nd ed. Singapure: World Scientific Publishing Co Pte Ltd, 2014.

[192] K. Zhu and B. Vogel-Heuser, "Sparse representation and its applications in micro-milling condition monitoring: noise separation and tool condition monitoring," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 70, no. 1, pp. 185-199, 2013.

[193] G. Beruvides, R. Quiza, R. del Toro, and R. E. Haber, "Sensoring systems and signal analysis to monitor tool wear in microdrilling operations on a sintered tungsten–copper composite material," *Sensors and Actuators A: Physical,* vol. 199, no. 0, pp. 165-175, 9/1/ 2013.

[194] F. J. Alonso and D. R. Salgado, "Analysis of the structure of vibration signals for tool wear detection," *Mechanical Systems and Signal Processing,* vol. 22, no. 3, pp. 735-748, 4// 2008.

[195] A. P. Markopoulos and D. E. Manolakos, "Modeling of Micromachining," in *Modern Mechanical Engineering: Research, Development and Education*, P. J. Davim, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 285-323, 2014.

[196] A. P. Markopoulos, "Cutting Mechanics and Analytical Modeling," in *Finite Element Method in Machining Processes*London: Springer London, pp. 11-27, 2013.

[197] W. Bai, R. Sun, Y. Gao, and J. Leopold, "Analysis and modeling of force in orthogonal elliptical vibration cutting," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 83, no. 5, pp. 1025-1036, 2015.

[198] K. S. Woon and M. Rahman, "The effect of tool edge radius on the chip formation behavior of tool-based micromachining," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 50, no. 9, pp. 961-977, 2010.

[199] L. Zhanqiang, S. Zhenyu, and W. Yi, "Definition and determination of the minimum uncut chip thickness of microcutting," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 69, no. 5, pp. 1219-1232, 2013.

[200] Y. Karpat and T. Özel, "Predictive Analytical and Thermal Modeling of Orthogonal Cutting Process—Part I: Predictions of Tool Forces, Stresses, and Temperature Distributions," *Journal of Manufacturing Science and Engineering,* vol. 128, no. 2, pp. 435-444, 2005.

[201] J. Wu and Z. Liu, "Modeling of flow stress in orthogonal micro-cutting process based on strain gradient plasticity theory," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 46, no. 1, pp. 143-149, 2009.

[202] E. Demir, "A method to include plastic anisotropy to orthogonal micromachining of fcc single crystals," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 43, no. 5, pp. 474-481, 2008.

[203] Z. Jian-hua, W. Li-ying, T. Fu-qiang, Z. Yan, and W. Zhi, "Modeling study on surface roughness of ultrasonic-assisted micro end grinding of silica glass," *The International Journal of Advanced Manufacturing Technology,* journal article pp. 1-12, 2015.

[204] R. Haj Mohammad Jafar, J. K. Spelt, and M. Papini, "Surface roughness and erosion rate of abrasive jet micro-machined channels: Experiments and analytical model," *Wear,* vol. 303, no. 1–2, pp. 138-145, 6/15/ 2013.

[205] M. Arif, M. Rahman, and W. Yoke San, "Analytical modeling of ductile-regime machining of tungsten carbide by endmilling," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 55, no. 1, pp. 53-64, 2010.

[206] K. B. Mustapha and Z. W. Zhong, "A hybrid analytical model for the transverse vibration response of a micro-end mill," *Mechanical Systems and Signal Processing,* vol. 34, no. 1–2, pp. 321-339, 1// 2013.

[207] S. A. Tajalli, M. R. Movahhedy, and J. Akbari, "Chatter instability analysis of spinning micro-end mill with process damping effect via semi-discretization approach," *Acta Mechanica,* journal article vol. 225, no. 3, pp. 715-734, 2013.

[208] Y. Shi, F. Mahr, U. von Wagner, and E. Uhlmann, "Gyroscopic and mode interaction effects on micro-end mill dynamics and chatter stability," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 65, no. 5, pp. 895-907, 2012.

[209] A. D. Rock, R. Zhang, D. Wilkinson, and U. S. C. F. L. H. Division, *Velocity Variations in Cross-hole Sonic Logging Surveys: Causes and Impacts in Drilled Shafts*. Central Federal Lands Highway Division, 2008.

[210] J. Fish and T. Belytschko, "A First Course in Finite Elements," in *A First Course in Finite Elements*: John Wiley & Sons, Ltd, pp. 320-320, 2007.

[211] C. H. Lauro, L. C. Brandão, S. L. M. Ribeiro Filho, R. A. F. Valente, and J. P. Davim, "Finite Element Method in Machining Processes: A Review," in *Modern Manufacturing Engineering,* P. J. Davim, Ed. Cham: Springer International Publishing, pp. 65-97, 2015.

[212] T. Özel and E. Zeren, "Finite element modeling the influence of edge roundness on the stress and temperature fields induced by high-speed machining," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 35, no. 3, pp. 255-267, 2006.

[213] A. P. Markopoulos, "Finite Element Modeling," in *Finite Element Method in Machining Processes*London: Springer London, pp. 29-57, 2013.

[214] A. P. Markopoulos, "Application of FEM in Metal Cutting," in *Finite Element Method in Machining Processes*London: Springer London, pp. 59-69, 2013.

[215] M. S. Shunmugam, "Machining Challenges: Macro to Micro Cutting," *Journal of The Institution of Engineers (India): Series C,* journal article pp. 1-19, 2015.

[216] K. S. Woon, M. Rahman, K. S. Neo, and K. Liu, "The effect of tool edge radius on the contact phenomenon of tool-based micromachining," *International Journal of Machine Tools and Manufacture,* vol. 48, no. 12–13, pp. 1395-1407, 10// 2008.

[217] C. H. Lauro, S. L. M. Ribeiro Filho, D. Baldo, S. A. A. d. G. Cerqueira, and L. C. Brandão, "Optimization of micro milling of hardened steel with different grain sizes using multi-objective evolutionary algorithm," *Measurement,* vol. 85, pp. 88-99, 5// 2016.

[218] M. Abouridouane, F. Klocke, and D. Lung, "Microstructure-based 3D Finite Element Model for Micro Drilling Carbon Steels," *Procedia CIRP,* vol. 8, pp. 94-99, // 2013.

[219] T. Thepsonthi and T. Özel, "3-D finite element process simulation of micro-end milling Ti-6Al-4V titanium alloy: Experimental validations on chip flow and tool wear," *Journal of Materials Processing Technology,* vol. 221, pp. 128-145, 7// 2015.

[220] S. M. Afazov, R. Ronaldo, D. Londsdale, D. Zdebski, and S. M. Ratchev, "Analysis of micro-drilling of glassy ceramic Macor nozzles for scanning droplet systems," *Journal of Materials Processing Technology,* vol. 213, no. 2, pp. 221-228, 2// 2013.

[221] H. B. Wu and S. J. Zhang, "3D FEM simulation of milling process for titanium alloy Ti6Al4V," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 71, no. 5, pp. 1319-1326, 2014.

[222] D. D. Cui, K. Mylvaganam, L. C. Zhang, and W. D. Liu, "Some critical issues for a reliable molecular dynamics simulation of nano-machining," *Computational Materials Science,* vol. 90, pp. 23-31, 7// 2014.

[223] S. Goel, X. Luo, A. Agrawal, and R. L. Reuben, "Diamond machining of silicon: A review of advances in molecular dynamics simulation," *International Journal of Machine Tools and Manufacture,* vol. 88, pp. 131-164, 1// 2015.

[224] C.-J. Kim, R. Mayor, and J. Ni, "Molecular dynamics simulations of plastic material deformation in machining with a round cutting edge," *International Journal of Precision Engineering and Manufacturing,* journal article vol. 13, no. 8, pp. 1303-1309, 2012.

[225] M. J. Jackson, G. M. Robinson, M. D. Whitfield, W. Ahmed, and J. S. Morrell, "Chapter 7 - Micro- and nanomachining," in *Emerging Nanotechnologies for Manufacturing (Second Edition)*Boston: William Andrew Publishing, pp. 202-229, 2015.

[226] I. D. Marinescu, W. B. Rowe, B. Dimitrov, and H. Ohmori, "7 - Molecular dynamics for nano-contact simulation," in *Tribology of Abrasive Machining Processes (Second Edition)*Oxford: William Andrew Publishing, pp. 185-212, 2013.

[227] E. W. Grafarend, *Linear and Nonlinear Models: Fixed Effects, Random Effects, and Mixed Models*. Walter de Gruyter, 2006.

[228] A. Harfield and M. Beynon, "Empirical Modeling for constructionist learning in a Thai secondary school mathematics class," in *Ninth International Conference on eLearning for Knowledge-Based Society, 13-14 December 2012, Thailand*, 2012.

[229] P. Wilson and H. A. Mantooth, "Chapter 11 - Statistical and Stochastic Modeling," in *Model-Based Engineering for Complex Electronic Systems*Oxford: Newnes, pp. 369-400, 2013.

[230] W. Sha, "Comment on "Prediction of the flow stress of 0.4C–1.9Cr–1.5Mn–1.0Ni–0.2Mo steel during hot deformation" by R.H. Wu et al. [J. Mater. Process. Technol. 116 (2001) 211]," *Journal of Materials Processing Technology,* vol. 171, no. 2, pp. 283-284, 1/20/ 2006.

[231] G. Beruvides, R. Quiza, R. Toro, F. Castaño, and R. E. Haber, "Correlation of the holes quality with the force signals in a microdrilling process of a sintered tungsten-copper alloy," *International Journal of Precision Engineering and Manufacturing,* journal article vol. 15, no. 9, pp. 1801-1808, 2014.

[232] N. Shetty, M. A. Herbert, R. Shetty, D. S. Shetty, and G. S. Vijay, "Soft computing techniques during drilling of bi-directional carbon fiber reinforced composite," *Applied Soft Computing,* vol. 41, pp. 466-478, 4// 2016.

[233] J. S. Nam, D. H. Kim, H. Chung, and S. W. Lee, "Optimization of environmentally benign micro-drilling process with nanofluid minimum quantity lubrication using response surface methodology and genetic algorithm," *Journal of Cleaner Production,* vol. 102, pp. 428-436, 9/1/ 2015.

[234] B. Bhandari *et al.*, "Development of a micro-drilling burr-control chart for PCB drilling," *Precision Engineering,* vol. 38, no. 1, pp. 221-229, 1// 2014.

[235] G. Kibria, B. Doloi, and B. Bhattacharyya, "Predictive model and process parameters optimization of Nd:YAG laser micro-turning of ceramics," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 65, no. 1, pp. 213-229, 2012.

[236] I. E. Saklakoglu and S. Kasman, "Investigation of micro-milling process parameters for surface roughness and milling depth," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 54, no. 5, pp. 567-578, 2010.

[237] R. Quiza, G. Beruvides, and J. P. Davim, "Modeling and Optimization of Mechanical Systems and Processes," in *Modern Mechanical Engineering: Research, Development and Education*, P. J. Davim, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 169-198, 2014.

[238] N. Siddique and H. Adeli, "Introduction to Computational Intelligence," in *Computational Intelligence*: John Wiley & Sons Ltd, pp. 1-17, 2013.

[239] E. Kussul, T. Baidyk, and D. C. Wunsch, "Introduction," in *Neural Networks and Micromechanics*Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1-5, 2010.

[240] R. E. Haber, R. Haber, A. Alique, and S. Ros, "Application of knowledge-based systems for supervision and control of machining processes," *Handbook of software engineering and knowledge engineering,* vol. 2, pp. 673-710, 2002.

[241] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics,* journal article vol. 5, no. 4, pp. 115-133, 1943.

[242] E. Kussul, T. Baidyk, and D. C. Wunsch, "Classical Neural Networks," in *Neural Networks and Micromechanics*Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 7-25, 2010.

[243] H. Yu, "Network Complexity Analysis of Multilayer Feedforward Artificial Neural Networks," in *Applications of Neural Networks in High Assurance Systems*, J. Schumann and Y. Liu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 41-55, 2010.

[244] P. Tino, L. Benuskova, and A. Sperduti, "Artificial Neural Network Models," in *Springer Handbook of Computational Intelligence*, J. Kacprzyk and W. Pedrycz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 455-471, 2015.

[245] D. E. Tamir, N. D. Rishe, and A. Kandel, "Complex Fuzzy Sets and Complex Fuzzy Logic an Overview of Theory and Applications," in *Fifty Years of Fuzzy Logic and its Applications*, E. D. Tamir, D. N. Rishe, and A. Kandel, Eds. Cham: Springer International Publishing, pp. 661-681, 2015.

[246] R. l. M. del Toro, M. C. Schmittdiel, R. E. Haber-Guerra, and R. Haber-Haber, "System Identification of the High Performance Drilling Process for Network-Based Control," no. 48027, pp. 827-834, 2007.

[247] R. E. Haber and J. R. Alique, "Fuzzy Logic-Based Torque Control System for Milling Process Optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 37, no. 5, pp. 941-950, 2007.

[248] D. Martin, R. del Toro, R. Haber, and J. Dorronsoro, "Optimal tuning of a networked linear controller using a multi-objective genetic algorithm and its application to one complex electromechanical process," *International Journal of Innovative Computing, Information and Control,* vol. 5, no. 10, pp. 3405-3414, 2009.

[249] S. Dalecky and F. V. Zboril, "An Approach to ANFIS Performance," in *Mendel 2015: Recent Advances in Soft Computing*, R. Matoušek, Ed. Cham: Springer International Publishing, pp. 195-206, 2015.

[250] K.-L. Du and M. N. S. Swamy, "Probabilistic and Bayesian Networks," in *Neural Networks and Statistical Learning*London: Springer London, pp. 563-619, 2014.

[251] L. E. Sucar, "Hidden Markov Models," in *Probabilistic Graphical Models: Principles and Applications*London: Springer London, pp. 63-82, 2015.

[252] W.-H. Hsieh, M.-C. Lu, and S.-J. Chiou, "Application of backpropagation neural network for spindle vibration-based tool wear monitoring in micro-milling," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 61, no. 1, pp. 53-61, 2011.

[253] S. Mandal, A. Kumar, and Nagahanumaiah, "Assessment of micro turning machine stiffness response and material characteristics by fuzzy rule based pattern matching of

cutting force plots," *Journal of Manufacturing Systems,* vol. 32, no. 1, pp. 228-237, 1// 2013.

[254] G. Beruvides, R. Quiza, M. Rivas, F. Castaño, and R. E. Haber, "Online detection of run out in microdrilling of tungsten and titanium alloys," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 74, no. 9, pp. 1567-1575, 2014.

[255] G. Beruvides, F. Castaño, R. Quiza, and R. E. Haber, "Surface roughness modeling and optimization of tungsten–copper alloys in micro-milling processes," *Measurement,* vol. 86, pp. 246-252, 5// 2016.

[256] D. M. D'Addona, A. M. M. S. Ullah, and D. Matarazzo, "Tool-wear prediction and pattern-recognition using artificial neural network and DNA-based computing," *Journal of Intelligent Manufacturing,* journal article pp. 1-17, 2015.

[257] K. Zhu, Y. S. Wong, and G. S. Hong, "Multi-category micro-milling tool wear monitoring with continuous hidden Markov models," *Mechanical Systems and Signal Processing,* vol. 23, no. 2, pp. 547-560, 2// 2009.

[258] L. Fu, S.-F. Ling, and C.-H. Tseng, "On-line breakage monitoring of small drills with input impedance of driving motor," *Mechanical Systems and Signal Processing,* vol. 21, no. 1, pp. 457-465, 1// 2007.

[259] Q. Ren, M. Balazinski, K. Jemielniak, L. Baron, and S. Achiche, "Experimental and fuzzy modeling analysis on dynamic cutting force in micro milling," *Soft Computing,* journal article vol. 17, no. 9, pp. 1687-1697, 2013.

[260] G. Beruvides, R. Quiza, M. Rivas, F. Castano, and R. E. Haber, "A fuzzy-genetic system to predict the cutting force in microdrilling processes," in *Industrial Electronics Society, IECON 2014 - 40th Annual Conference of the IEEE,* 2014, pp. 34-37.

[261] S. N. Bhavsar, S. Aravindan, and P. V. Rao, "Investigating material removal rate and surface roughness using multi-objective optimization for focused ion beam (FIB) micro-milling of cemented carbide," *Precision Engineering,* vol. 40, pp. 131-138, 4// 2015.

[262] R. Coppel, J. V. Abellan-Nebot, H. R. Siller, C. A. Rodriguez, and F. Guedea, "Adaptive control optimization in micro-milling of hardened steels—evaluation of optimization approaches," *The International Journal of Advanced Manufacturing Technology,* journal article pp. 1-20, 2015.

[263] G. Beruvides, R. Quiza, R. M. Del Toro, and R. E. Haber, "Extracción de rasgos de las señales para la monitorización indirecta de la herramienta en el microtaladrado," *Dyna,* vol. 88, pp. 405-413, 2013.

[264] A. Alique, R. E. Haber, R. H. Haber, S. Ros, and C. Gonzalez, "A neural network-based model for the prediction of cutting force in milling process. A progress study on a real case," in *Proceedings of the 2000 IEEE International Symposium on Intelligent Control. Held jointly with the 8th IEEE Mediterranean Conference on Control and Automation (Cat. No.00CH37147),* 2000, pp. 121-125.

[265] R. E. Haber, J. R. Alique, A. Alique, and R. H. Haber, "Controlling a complex electromechanical process on the basis of a neurofuzzy approach," *Future Generation Computer Systems,* vol. 21, no. 7, pp. 1083-1095, 7// 2005.

[266] A. G. Martin and R. E. H. Guerra, "Internal Model Control Based on a Neurofuzzy System for Network Applications. A Case Study on the High-Performance Drilling Process," *IEEE Transactions on Automation Science and Engineering,* vol. 6, no. 2, pp. 367-372, 2009.

[267] A. A. Krimpenis, N. A. Fountas, I. Ntalianis, and N. M. Vaxevanidis, "CNC micromilling properties and optimization using genetic algorithms," *The International Journal of Advanced Manufacturing Technology,* journal article vol. 70, no. 1, pp. 157-171, 2014.

[268] H. Watanabe, H. Tsuzaka, and M. Masuda, "Microdrilling for printed circuit boards (PCBs)—Influence of radial run-out of microdrills on hole quality," *Precision Engineering,* vol. 32, no. 4, pp. 329-335, 10// 2008.

[269] L. A. Kudla, "Deformations and Strength of Miniature Drills," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* vol. 220, no. 3, pp. 389-396, March 1, 2006 2006.

[270] D. Yang, W. Sui, and D. Zhang, "International Conference on Applied Physics and Industrial Engineering 2012Four Methods for Roundness Evaluation," *Physics Procedia,* vol. 24, pp. 2159-2164, 2012/01/01 2012.

[271] A. Sarkheyli, A. M. Zain, and S. Sharif, "A multi-performance prediction model based on ANFIS and new modified-GA for machining processes," *Journal of Intelligent Manufacturing,* journal article vol. 26, no. 4, pp. 703-716, 2015.

[272] G. Beruvides, F. Castaño, R. E. Haber, R. Quiza, and M. Rivas, "Intelligent Models for Predicting the Thrust Force and Perpendicular Vibrations in Microdrilling Processes," in *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*, 2014, pp. 506-511.

[273] F. Castaño, R. E. Haber, R. M. d. Toro, and G. Beruvides, "Application of hybrid incremental modeling for predicting surface roughness in micromachining processes," in *Computational Intelligence for Engineering Solutions (CIES), 2014 IEEE Symposium on*, 2014, pp. 54-59.

[274] G. Beruvides, R. Quiza, and R. E. Haber, "Multi-objective optimization based on an improved cross-entropy method. A case study of a micro-scale manufacturing process," *Information Sciences,* vol. 334–335, pp. 161-173, 3/20/ 2016.

[275] A. Costa, O. D. Jones, and D. Kroese, "Convergence properties of the cross-entropy method for discrete optimization," *Operations Research Letters,* vol. 35, no. 5, pp. 573-580, 9// 2007.

[276] D. P. Kroese, R. Y. Rubinstein, and T. Taimre, "Application of the cross-entropy method to clustering and vector quantization," *Journal of Global Optimization,* Article vol. 37, no. 1, pp. 137-157, 2007.

[277] R. Y. Rubinstein and D. P. Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.

[278] GAMHE-Group, "Multi-objective Optimization Cross Entropy (MOCE+)," ed, 2015.

[279] D. A. Van Veldhuizen, "Scalable multi-objective optimization test problems," Ph.D., Air Force Institute of Technology, Wright-Patterson AFB, 1999.

[280] A. J. Nebro, F. Luna, E. Alba, B. Dorronsoro, J. J. Durillo, and A. Beham, "AbYSS: Adapting Scatter Search to Multiobjective Optimization," *Evolutionary Computation, IEEE Transactions on,* vol. 12, no. 4, pp. 439-457, 2008.

[281] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," 2001.

[282] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on,* vol. 6, no. 2, pp. 182-197, 2002.

[283] K. Li, S. Kwong, R. Wang, K.-S. Tang, and K.-F. Man, "Learning paradigm based on jumping genes: A general framework for enhancing exploration in evolutionary multiobjective optimization," *Information Sciences,* vol. 226, no. 0, pp. 1-22, 3/20/ 2013.

[284] Y. Huo, Y. Zhuang, J. Gu, and S. Ni, "Elite-guided multi-objective artificial bee colony algorithm," *Applied Soft Computing Journal,* Article vol. 32, pp. 199-210, 2015.

[285] X. Dai, X. Yuan, and Z. Zhang, "A self-adaptive multi-objective harmony search algorithm based on harmony memory variance," *Applied Soft Computing Journal,* Article vol. 35, pp. 541-557, 2015, Art. no. 3021.

[286] M. Ramteke, N. Ghune, and V. Trivedi, "Simulated binary jumping gene: A step towards enhancing the performance of real-coded genetic algorithm," *Information Sciences,* Article vol. 325, pp. 429-454, 2015.

[287] Z. Qingfu and L. Hui, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *Evolutionary Computation, IEEE Transactions on,* vol. 11, no. 6, pp. 712-731, 2007.

[288] Z. Qingfu, Z. Aimin, and J. Yaochu, "RM-MEDA: A Regularity Model-Based Multiobjective Estimation of Distribution Algorithm," *Evolutionary Computation, IEEE Transactions on,* vol. 12, no. 1, pp. 41-63, 2008.

[289] D. M. D'Addona and R. Teti, "Genetic algorithm-based optimization of cutting parameters in turning processes," in *Procedia CIRP*, 2013, vol. 7, pp. 323-328.

[290] R. Quiza Sardiñas, M. Rivas Santana, and E. Alfonso Brindis, "Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes," *Engineering Applications of Artificial Intelligence,* vol. 19, no. 2, pp. 127-133, 2006.

[291] P. M. Dixit and U. S. Dixit, *Modeling of Metal Forming and Machining Processes: by Finite Element and Soft Computing Methods (Engineering Materials and Processes) (Engineering Materials and Processes)*. Springer Publishing Company, Incorporated, p. 590, 2008.

[292] D. I. Lalwani, N. K. Mehta, and P. K. Jain, "Experimental investigations of cutting parameters influence on cutting forces and surface roughness in finish hard turning of MDN250 steel," (in English), *Journal of Materials Processing Technology,* vol. 206, no. 1-3, pp. 167-179, 2008.

[293] C. R. Dandekar and Y. C. Shin, "Modeling of machining of composite materials: A review," (in English), *International Journal of Machine Tools and Manufacture,* vol. 57, pp. 102-121, 2012.

[294] B. Sieben, T. Wagner, and D. Biermann, "Empirical modeling of hard turning of AISI 6150 steel using design and analysis of computer experiments," (in English), *Production Engineering,* vol. 4, no. 2, pp. 115-125, 2010.

[295] S. Velchev, I. Kolev, K. Ivanov, and S. Gechevski, "Empirical models for specific energy consumption and optimization of cutting parameters for minimizing energy consumption during turning," (in English), *Journal of Cleaner Production,* vol. 80, pp. 139-149, 2014.

[296] A. M. Zain, H. Haron, and S. Sharif, "Optimization of process parameters in the abrasive waterjet machining using integrated SA-GA," (in English), *Applied Soft Computing Journal,* vol. 11, no. 8, pp. 5350-5359, 2011.

[297] N. Baskar, P. Asokan, G. Prabhaharan, and R. Saravanan, "Optimization of machining parameters for milling operations using non-conventional methods," (in English), *International Journal of Advanced Manufacturing Technology,* vol. 25, no. 11-12, pp. 1078-1088, 2005.

[298] N. Yusup, A. M. Zain, and S. Z. M. Hashim, "Overview of PSO for optimizing process parameters of machining," in *2012 International Workshop on Information and Electronics Engineering, IWIEE 2012*, Harbin, 2012, vol. 29, pp. 914-923.

[299] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag New York, Inc., 2006.

[300] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," (in English), *Reliability Engineering and System Safety,* vol. 91, no. 9, pp. 992-1007, 2006.

[301] R. Sanz, C. Hernández, A. Hernando, J. Gómez, and J. Bermejo, "Grounding robot autonomy in emotion and self-awareness," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* vol. 5744 LNCS, ed, pp. 23-43, 2009.

[302] H. Imamizu and M. Kawato, "Brain mechanisms for predictive control by switching internal models: Implications for higher-order cognitive functions," *Psychological Research,* vol. 73, no. 4, pp. 527-544, 2009.

[303] J. Peterburs and J. E. Desmond, "The role of the human cerebellum in performance monitoring," *Current Opinion in Neurobiology,* vol. 40, pp. 38-44, Oct 2016.

[304] T. Ishikawa, S. Tomatsu, J. Izawa, and S. Kakei, "The cerebro-cerebellum: Could it be loci of forward models?," *Neuroscience Research,* vol. 104, pp. 72-79, Mar 2016.

[305] C. Szepesvári, "Algorithms for Reinforcement Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning,* vol. 4, no. 1, pp. 1-103, 2010/01/01 2010.

[306] H. Boubertakh, M. Tadjine, P.-Y. Glorennec, and S. Labiod, "Tuning fuzzy PD and PI controllers using reinforcement learning," *ISA Transactions,* vol. 49, no. 4, pp. 543-551, 10// 2010.

[307] C. Juanes, "Diseño e implementación de estrategias self-x en una arquitectura de control cognitivo artificial," Grado en ingeniería informática y en matemática, Escuela Politécnica Superior, Universidad Autónoma de Madrid, Madrid, España, 2014.

[308] (2017). *jFuzzyLogic Fuzzy Control Language*. Available: http://jfuzzylogic.sourceforge.net/html/manual.html

[309] (2016). *COGNETCON packages function repository*. Available: http://gamhe.eu/downloads/?route=.%2FCognetcon

[310] K. Park, Y. Han, T. h. Hur, and Y.-K. Lee, "Correlated subgraph search for multiple query graphs in graph streams," presented at the Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication, Bali, Indonesia, 2015. Available: http://delivery.acm.org/10.1145/2710000/2701234/a49-park.pdf?ip=150.244.109.166&id=2701234&acc=ACTIVE%20SERVICE&key=E16C5911FC136D5E%2EE16C5911FC136D5E%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=626563090&CFTOKEN=37393155&__acm__=1465285647_7bf35a9c3de3b8793cc45156c10f4c0b

[311] D. C. Schmidt, D. L. Levine, and S. Mungee, "Quality of Services in Distributed SystemsThe design of the TAO real-time object request broker," *Computer Communications*, vol. 21, no. 4, pp. 294-324, 1998/04/10 1998.

[312] M. Henning, "Choosing Middleware: Why Performance and Scalability do (and do not) Matter," 2009.

[313] J. F. Koning, C. J. M. Heemskerk, P. Schoen, D. Smedinga, A. H. Boode, and D. T. Hamilton, "Evaluating ITER remote handling middleware concepts," *Fusion Engineering and Design*, vol. 88, no. 9–10, pp. 2146-2150, 10// 2013.

[314] (2016, 11/08). *Unified Modeling Language*. Available: http://www.uml.org/

[315] (2016, 11/08). *Real Time Specification for Java*. Available: http://www.rtsj.org/

[316] (2016, 11/08). *S.W.a.I. Generator*. Available: http://www.swig.org/index.php

[317] (2016, 11/08). *Raspberry Pi 2 model B specifications,* . Available: http://docs-europe.electrocomponents.com/webdocs/1392/0900766b8139232d.pdf

[318] G. Beruvides, F. Castaño, R. E. Haber, R. Quiza, and M. Rivas, "Artificial intelligence-based modeling and optimization of microdrilling processes," in *Computational Intelligence for Engineering Solutions (CIES), 2014 IEEE Symposium on*, 2014, pp. 49-53.

[319] CONMICRO, "Control cognitivo artificial en procesos de micromecanizado mecánico. Método y aplicación," Ministerio de Economía y Competitividad (MINECO), DPI2012-355042012-2015.

[320] AM4G, "Advanced Manufacturing 4th Generation," Centro para el Desarrollo Tecnológico Industrial (CDTI), Programa Cien 20152015 - 2017.

ANNEX I. GLOSSARY OF TERMS

Acronymics

AAE	Average of absolute error
AAR	Cognitive Architecture
AbYSS	Archive-based hybrid scatter search
ACT	Adaptive Control of Thought
ACT-R	Adaptive Control of Thought-Rational
AI	Artificial Intelligence
AMARSi	Adaptive Modular Architecture for Rich Motor Skills
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial neural network
ANOVA	ANalysis Of VAriance
ART	Adaptive Resonance Theory
BDI	Belief-desire-intention
C	General-purpose, imperative computer programming language, supporting structured programming, lexical variable scope and recursion.
C++	General-purpose programming language, object-oriented, generic programming features and low-level memory manipulation
CE	Cross-Entropy method
CEFAS	Research Centre of Advanced and Sustenible Manufacturing, University of Matanzas, Matanzas, Cuba
CLARION	Connectionist Learning with Adaptive Rule Induction On-line
CNC	Computer numeric control
CPS	Cyber-physical Systems

CORBA	Common Object Request Broker Architecture
CONMICRO	Artificial Cognitive Control for Micromechanical Machining
CV	Convergence
CVD	Chemical Vapor Deposition
DDE	Dynamic Data Exchange
DEXMART	DEXterous and autonomous dual-arm/hand robotic manipulation with sMART sensory-motor skills: A bridge from natural to artificial cognition
DLM	Digital Lifecycle Manufacturing group, Warwick Manufacturing Group, Warwick, United Kingdom
EAs	Evolutionary Algorithms
EDAs	Estimation-of-Distribution Algorithms
EHU/UPV	University of the Basque Country
EDM	Electrical Discharge Machining
EMD	Empirical Mode Decomposition
EMOABC	Elite-guided multi-objective artificial bee colony
EPAM	Elementary Perceiver and Memorizer
EPIC	Explicitly Parallel Instruction Computing
FCL	Fuzzy Control Language
FEM	Finite Element Method
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FPI	Research Staff Training Grant
FR	Functional requirements
GAMHE	Group of advanced Automation of Machines, Highly complex processes and Environments, Centre for Automation and Robotic, Madrid, Spain
GD	Generational distance
HANDLE	Developmental pathway towards autonomy and dexterity in robot in-hand manipulation
HHT	Hilbert-Huang Transform
HR	Hyperarea ratio
HUMANOBS	Humanoids that Learn Socio-Communicative Skills by Observation
IBM	International Business Machines Corp
ICARUS	A cognitive architecture for physical agents
IceGrid	Suite of frameworks that provide object-oriented load balancing, failover, object-discovery and registry services
ICT	Information and communications technology

IDE	Integrated development environment
IIR	Infinite Impulse Response
IK4	IK4 Research Alliance
IMC	Internal mode control
IPK	Institut für Produktionsanlagen und Konstruktionstechnik, Fraunhofer, Berlin, Germany
IoT	Internet of Things
ISE	Integral of square error
JC	empirical Johnson-Cook model
JNI	Java Native Interface
JVM	Java Virtual Machine
KURT	Kurtosis
LIDA	Learning Intelligent Distribution Agent
LIGA	Lithography Electroplating and Moulding
MACE	Multi-objective cross-entropy
MACE-gD	Multi-objective cross-entropy using generalized decomposition
MAX	Maximum
MD	Molecular Dynamic
ME	Maximum Pareto front error
MEAN	Mean value
MEMS	Micro Electrical-Mechanical Systems
ML	Machine learning
MLR	Multiple Linear Regression
micro-ECM	Micro Electro Chemical Machining
micro-PCM	Micro Photo Chemical Micromachining
MIN	Minimum
MLP	Multilayer Perceptron
MOEA/D	Multi-objective evolutionary algorithms based on decomposition
MOCE	Cross entropy multi-objective optimization algorithm
MOCE+	Cross entropy multi-objective optimization algorithm modified
MOGA	Multi-objective genetic algorithm
MOP	Multi-objective optimization test functions
MSCM	Modified Shared Circuits Model
MSE	Mean square error
NFR	Non-functional requirements
NPGA	Niched Pareto genetic algorithm
NSGAII	Non-dominated sorting genetic algorithm II

NSGAII+JGBL	Non-dominated sorting genetic algorithm II + learning paradigm based on jumping genes
NIFTi	Natural human-robot cooperation in dynamic environments
non-MEMS	non-Micro Electrical-Mechanical Systems
NOPTILUS	autoNomous, self-Learning, OPTImal and compLete Underwater Systems
Ovt	Overshoot
PASAR	Prediction, Anticipation, Sensation, Attention and Response
PDF	Probability density function
PSO	Particle Swarm Optimization Algorithm
PVD	Physical Vapor Deposition
RBFN	Radial Basis Function Networks
RF-MEMS	Radio Frequency Micro Electrical-Mechanical Systems
RL	Reinforcement learning
RMS	Root Mean Square
RM-MEDA	Regularity model-based estimation of distribution algorithms
RMI	Java Remote Method Invocation
RNSGA-II-SBJG	Real-coded NSGA-II with simulated binary jumping gene operators
ROBOCAST	ROBOt and sensors integration as guidance for enhanced Computer Assisted Surgery and Therapy
RT-CORBA	Real-time Common Object Request Broker Architecture
RTSJ	Real Time Specification for Java
RVE	Representative Volume Element
SAMOHS	Self-adaptive multi-objective harmony search
SARSA	State-Action-Reward-State-Action
SASE	Cognitive Architecture
SCM	Shared Circuit Model
SKEW	Skewness
SMOCE	Simple Multi-Objective Cross Entropy method
SP	Spacing
SPEA2	Strength Pareto evolutionary algorithm
SOAR	Cognitive Architecture
SOM	Self-organized Maps
SOP	Self-observation principle
STD	Standard Deviation
STFT	Short-time Fourier transform

SWIG	Open-source software tool used to connect computer programs or libraries written in C or C++ with scripting languages such as: Python, C#, Java, JavaScript, and Octave.
TD	Difference learning
TiAlN	Titanium Aluminum Nitrate coated
UML	Unified Modeling Language
USM	Ultrasonic Micromachining
WFG	Set of multi-objective optimization test functions
WT	Wavelet Transform
ZeroC	Focus-deliver best-of-breed tools to help the Network Software
ZDT	Set of multi-objective optimization test functions

Equations

a	Action (Q-learning)
$a_1 \ldots a_4$	Coefficients of force differential model
Ap	Axial cutting deep
a_t	Action taken in time
A	Set of actions in a Markov Decision Problem
Ap	Amplitude
b	Coefficient obtained by minimizing the sum of the square
$b_1 \ldots b_3$	Coefficients of force differential model
$[b_i^{\min}, b_i^{\max}]$	Minimum and maximum values of the *i*-th objective in the elite population
$bias$	Linear combination which is called bias or threshold
c_h	Heat capacity

$[c_{k-1}, c_k], k = 1 \ldots r + 2$	Bounds of the *k*-th histogram interval
$\left[\underline{c}_{i,k}, \overline{c}_{i,k}\right]$	Lower and upper bounds for each interval
d(t)	Disturbance or noise
$d_i\ , i = 1...Z$	Pareto ranking of the *i*-th element
$d_i^*, i = 1...Z^*$	Pareto ranking of the *i*-th elite solution
D	Diameter
$\underline{D}$	Own set of forward models
e	Output action error taking as input in the modules
E	Young’s modulus
E_{pop}	Elite population
E_i	Energy component of a signal
E_υ	Event
$\%f_{rate}$	Override feed rate
$f(\bullet): \mathbb{R}^n \rightarrow \mathbb{R}^m$	Optimization objectives function
fs	Sampling frequency
f_{rate}	Feed rate
f_{tooth}	Feed rate per tooth
fw	Work frequency
∇f	Gradient of the function
Fx	Force in x-axis
Fy	Force in y-axis
Fz	Force in z-axis or thrust force
F_{z}^{al}	Allowable thrust force
F_{ref}	Force values preset to control the process
F_{process}	Force values measured in process
F_{est}	z-component of the force estimated in the mirroring
$g(\bullet): \mathbb{R}^n \rightarrow \mathbb{R}^p$	Constraints function
Gc	Generalization capabilities
h	A priori appropriate probability density function
$h_i(\bullet)$	Equality constraints
h_d	drilling depth
H	Finite-horizon
HRB	Hardness
i, j, k	Iteration variables
I_{min}	Minimum area moment of inertia of the drill cross-section
I_υ	Collection of functions

J	Performance index
J_{max}	To actions to maximize the expected return
Js	Performance index switch
kc	Thermal conductivity
K	Set of model/models parameters
KE	Error gains
KDE	Change in error gains
K_{out}	Gain of the output of the controller
K_{for}	Gain of the input of the forward controller
K_{for_out}	Gain of the output of the forward controller
K_{inv}	Gain of the input of the inverse controller
K_{inv_out}	Gain of the output of the inverse controller
$\left[K_i^{\min}, K_i^{\max}\right]$	own bounds given by the model's parameters
$l_j \in \mathbb{R}, j = 1 \ldots n$	Lower bound of the j-th variable
L	Length of the drill flute
$m \in \mathbb{N}^+$	Number of optimization objectives
m_K	Possible values of each *K* parameter
$\underline{M}$	Models
Mz	Moment in z-axis
$n \in \mathbb{N}^+$	Number of decision variable
n_{peak}	Number of peaks
n_{rpm}	Spindle rotation speed
$N \in \mathbb{N}^+$	Epochs number
N_{hole}	Number of elaborated holes
N_{reg}	Number of input–output pairs in the dataset
$p \in \mathbb{N}^+$	Number of constraints
$p_{control}$	Control mechanism sampling time
$p_{learning}$	Learning sampling time
P	Probabilities matrix
$\underline{P}$	Process inputs
P_x	Probability density function
q_{cen}	Considered angle from the provisionally center
q_{reg}	Number of parameters in a regression model
$Q(s,a)$	Quality of a state-action combination in a Markov Decision Problem
r	Number of inner histogram intervals
$\ddot{r}_p$	Plane perpendicular to the drilling axis
$radius$	Circle radius for holes quality error determination

$radius_{max}$	Maximum possible circle radius for holes quality error
$radius_{min}$	Minimum possible circle radius for holes quality error
R	Reward received after performing action
R^2	Correlation coefficients
Ra	Surface roughness
s	State (Q-learning)
s_t	State in time
s'	Next state
$step$	Drill step
S	Set of states in a Markov Decision Problem
Sc	Evaluations count
$Sc_{\max} \in \mathbb{N}^+$	Maximum evaluation number
t	Time
T	Transition probabilities matrix
$u_j \in \mathbb{R}, j = 1 \ldots n$	Upper bound of the *j*-th variable
$\mathbf{v}^*$	No-dominate vector
$V^*(s')$	Optimal value function in a reinforcement learning algorithm
Vc	Cutting speed
Vx	Vibration in x-axis
Vy	Vibration in y-axis
Vz	Vibration in z-axis
$W(x)$	Likelihood ratio
x	Vector input variables
x_{coord}	Position in x-axis
$x_{ij}, i = 1...Z, j = 1 \ldots n$	Value of the *j*-th variable for the *i*-th solution
$x^*_{ij}, i = 1...Z^*, j = 1 \ldots n$	Value of the *j*-th variable for the *i*-th elite solution
X	Matrix input variables
y	Vector output variables
y_{coord}	Position in y-axis
$y_{ij}, i = 1...Z, j = 1 \ldots n$	Value of the *j*-th objective for the *i*-th solution
$y^*_{ij}, i = 1...Z^*, j = 1 \ldots n$	Value of the *j*-th objective for the *i*-th elite solution
$\mathrm{z}_{\mathrm{coord}}$	Position in z-axis
$Z \in \mathbb{N}^+$	Population size
$Z^* \in \mathbb{N}^+$	Elite population size
$Z^*_{\max} \in \mathbb{N}^+$	Maximum elite population size
α	Learning rate

β_t	Dynamic smoothing
γ	Discount factor
$\gamma_j \in \mathbb{R},\ j = 1 \ldots n$	Weight of the *j*-th constraint
Γ	Kullback-Leibler distance or cross-entropy
δ_{learn}	Learning correction factor
δm	Material elongation
$\delta_i, i = 1 \ldots Z^*$	Distance between the *i*-th and (*i* + 1)-th elite elements
ε	Error in the single loop controller
$\Delta\varepsilon$	Variation of error in the single loop controller
$\varepsilon_j,\ j = 1 \ldots n$	Change in the standard deviation for the *j*-th variable
$\varepsilon_{\max} \in \mathbb{R}^+$	Convergence limit
$\varepsilon_\mathrm{quality}$	Hole quality error
$\zeta_k, k = 1 \ldots r + 2$	Number of children for the *k*-th histogram interval
η	Neurons number
η_f	Security factor
θ	Transfer function in MLP
$\kappa \in \mathbb{R}^+$	Elite threshold decreasing factor
$\lambda_0 \in \mathbb{R}^+$	Initial elite threshold
$\lambda_t, t = 1 \ldots N$	Elite threshold at the *t*-th epoch
μ_b	Poisson's coefficient
μ_d	Decrease factor
$\mu_j,\ j = 1 \ldots n$	Mean value of the *j*-th variable
$\nu_i(\bullet)$	Frequency component of a signal
$[(\xi,\upsilon)_1,\ldots,(\xi,\upsilon)_E]$	Elite solutions clustered by using the histogram of the objective functions
$\xi_{\min}, \xi_{\max}$	Minimum and maximum ordinal position for children
ϕ	Circle angles
$\phi_k, k = 1 \ldots r + 2$	Frequency of the *k*-th histogram interval
$\phi(t)$	Performance index associated with the action that is taken
π_p	Optimal policy
ρ	Density function
ρm	Material mass density
$\sigma_j,\ j = 1 \ldots n$	Standard deviation of the *j*-th variable
σ_Y	Yield tensile

τ_0	Unit machining time
$\varphi \in [0,1]$	Frequency inversion likelihood
$\psi \in [0,1]$	Smoothing factor
ω	Artificial neural network weights
$\bar{\varpi}$	Third-order frequency poles model of the drilling process
Ω	Decision variable
$\hat{\Omega}$	Importance sampling
$\mathcal{N}$	Normal random distribution
$\mathcal{U}$	Uniform random distribution

ANNEX II. PREVIOUS PROJECT REVIEW

Acronym	Techniques	Contributions	Applications
ChiRoPing	- Embodied active sonar perception systems - Model bat’s coordination of its acoustic, behavioural and morphological choices	- Engineering versatile and robust systems able to respond sensible to challenges not precisely specified in their design	- Navigation
CHRIS	- Exploration of engineering principles for safe movement and dexterity - Language, communication and decisional action planning	- Safe human robot interaction - Integration of cognition in Co-operative manipulation of real world objects	- Service robotics
DEXMART	- Decision between different manipulation options	- Allows a dual-arm robot to grasp and manipulate objects used by human beings	- Service robotics

Acronym	Techniques	Contributions	Applications
	- Acquisition knowledge by learning new action sequences - Designing new hand components and sensors	- Safe human robot interaction - Integration of cognition in Co-operative manipulation of real world objects	
CoFRIEND	- Identification of objects and events - Feedback and multi-data fusion - Heterogeneous sensor network	- Prototype system for the representation and recognition of human activity and behaviour	- Knowledge representation
CogX	- Identifying gaps in its own understanding of the environment and then plans how to fill those gaps to deal with novelty and uncertainty in task execution	- Creates a theory of how a cognitive system can model its own knowledge - Models the environment of a cognitive system, its own understanding of the environment and how it changes under action - Extends knowledge so as to perform future tasks more efficiently	- Task execution - Knowledge modeling
DIPLECS	- Bootstrapping and learning - Defining hierarchical perception-action cycles - Using scenario of a driver assistance system, it continuously improves its capabilities by observing the human driver, the car data, and the environment	- Designs an Artificial Cognitive System architecture that learns and adapts in dynamic and interactive real-world scenarios	- Navigation - Control

Acronym	Techniques	Contributions	Applications
LIREC	- Studying human-pet interactions	- Establishes a multi-faceted (memory, emotions, cognition, communication, learning, etc.) theory of artificial long-term companions	- Safe robotics - Interactions between robots
ROBOCAST	- Learning and interactive plan updating capabilities - Fuzzy representation - Context-based interpretation of surgeon commands	- Aids surgeons in keyhole neurosurgery - A interface allows surgeons to receive maximum feedback data with minimum extra effort on their side	- Surgery
ROSSI	- Sensorimotor and neural/computational mechanism	- Flexibly manipulate and use objects in the environment - Novel approaches to the grounding of robotic conceptualization and language	- Manipulator robots - Communications between agents
SPARK II	- Hierarchical architecture - Parallel sensory-motor pathways - Insect brain inspired	- Implements reflex-driven basic behaviours - Self-organizing complex dynamics	- Control
BRICS	- Studying to harmonize interfaces, communications and data exchange between agents - Developing a software repository of best practice robotics algorithms	- Researches shortening the development cycles for new robot systems and applications	- Telecommunication - Automotive industry - Embedded systems industry
Co3 AUVs	- 3D perception and mapping - Online data process	- Coordination and cooperative control of multiple Autonomous Underwater Vehicles	- Navigation - Multi-agent coordination

Acronym	Techniques	Contributions	Applications
		- Robustness with respect to failures and environmental changes	
ECCEROBOT	- Anthropo-mimetic robots for human-like action and interaction in the world - Motion capture - Causal analysis - Classical control theory - Internal models - Sensory-motor strategies	- Exploits human-like characteristics to produce some human-like cognitive features	- Control in robots
EUROPA	- Probabilistic scene interpretation - Modeling of the environment	- Robustly and reliable addressing the autonomous navigation problem in complex and populated environments - Reasons based on the verbal and natural interaction with users	- Navigation - Commercial applications of service robots
FILOSE	- Detecting hydrodynamic patterns in the surrounding environment	- Develops technologies in underwater robotics navigation and understanding fish biology - Understands how fish do and robots could sense the underwater environment, adaptability and reliability.	- Underwater humanitarian - Anti-terrorist activities - Surveillance of harbours - Coast security - Entertainment - Edutainment and fishery
HANDLE	- Integrating disciplines such as neuroscience, developmental psychology,	- Works in requirement for robots to carry out accurate and intelligent tasks on behalf of and in collaboration with people	- Robot grippers - Service robotics

Acronym	Techniques	Contributions	Applications
	cognitive science, robotics, multimodal perception and machine learning - Learning and predicting behaviours from imitation and motor babbling observing human manipulation gestures		
HUMANOBS	- Model-driven architectures - Integrated cognitive control - Distributed real-time systems	- Automatic learning of multi-dimensionally constrained problems - Automatic control of human-like communications skills and in large-scale perception-action architectures - Cognitive architectural for learning human interaction modeling	- Interpersonal communication systems in virtual environments
HUMOUR	- Combining behavioural studies on motor learning and its neural correlates with design, implementation, and validation of robot agents - Agents learns using information of human sensor motor systems	- Develops robot strategies to facilitate the acquisition of motor skills	- Services robotics - Robot rehabilitation - Helping professionals (arts, sports, medicine…)
I'M Clever	- Abstraction of sensory information - Study of mechanism underlying intrinsic motivations - Hierarchical recursive architectures which permit cumulative learning	- Designs robot controllers that learns cumulatively new skills and reuse them for accomplishing multiple, complex, and externally-assigned tasks - High versatility in solving task	- Autonomous learning systems and robots

Acronym	Techniques	Contributions	Applications
ROSETTA	- Sensor-based task execution skills - Management of knowledge repository	- Robot controller technology for dual-arm industrial robots that works together with humans - A knowledge repository enriched by interactions with humans and other machines	- Service robotics - Assembly of consumer devices
ALIZ-E	- Studying how long-term experience can be acquired to ground actions and interactions across time - Studying how a system can deal robustly with inevitable differences in quality in perceiving - Understanding how a system can adapt its interaction based on the way user behaviour changes	- Mobile robots that can adapts of a possibly non-continuous succession of interactions	- Human-robot interactions (robotics services) - Evaluation of interactive robots
AMARSi	- Compliant mechanics, pervasive learning and dynamical-systems based control architectures - Morphological computing - Study of principles of reservoir computing - Control architectures based on dynamical (neural) systems	- Achieves biological richness in robotic motor skills	- Motor control

Acronym	Techniques	Contributions	Applications
FIRST-MM	- Specification language in robot programming - Learning by instruction - Statistical relational learning	- Technology for flexible autonomous mobile manipulation robots - Novel robot programming environment that allows non-expert users to specify complex manipulation tasks in real-world environments	- Service robotics
IURO	- Environment perception, communication, navigation - Information retrieval from humans - Knowledge representation and assessment	- Information retrieval from humans into robot control architectures to complement their perception and action control capabilities - Identification of knowledge gaps arising from dynamically hanging situations and missing information from humans	- Commercial service robots
NIFTi	- Cognitive control model - Interconnecting contents across modules - Using prediction models to anticipate how adapt acting and communication to align with the human - Learning off- and online: reinforcement learning and statistical (relational) learning	- Investigates how natural behaviour in human-robot cooperation can arise - Aims at operationalizing natural cooperation by balancing operational and cooperation demands in a cognitive architecture	- Urban search and rescue

Acronym	Techniques	Contributions	Applications
RoboEarth	- Object recognition and localization - Control strategies by linking perception and action - Learning - World-wide web-style database	- Allows robots to share any reusable knowledge independently of their hardware and configuration - Modular design of robotic system	- Multi-agent interactions
CoCoRo	- Locally and globally acting self-organizing mechanisms - Cognition-generating algorithms to mimic each other's behaviour and to learn from each other - principles of swarm-level cognition	- Creates a swarm of interacting, cognitive, autonomous robots to improve of collective performance	- Multi-agent systems
COMPLACS	- Bandit problems - Markov Decision Processes (MDPs) - Partially Observable MDPs (POMDPs) - Continuous stochastic control - Multi-agent systems	- A unified framework in machine learning to intelligent systems that can address a wide variety of control problems of many different types - A toolkit that provides methods for the automatic construction of representations and capabilities	- Development of intelligent systems
CORBYS	- High-level cognitive control modules - A semantically-driven self-awareness module, - A cognitive framework for anticipation	- A cognitive robot control architecture to cope with highly dynamic environments as humans are demanding, curious and often act unpredictably	- Control architectures

Acronym	Techniques	Contributions	Applications
eSMCs	- Biologically-inspired information theoretic principles - Definition of object concepts and action plans - Goal-oriented behaviour - Investigating learning and adaptivity in artificial systems - Sensorimotor interactions - Object recognition - Action planning	- Sensorimotor contingencies: law-like relations between actions and associated changes in sensory input - Object concepts and action plans and that their mastery can lead to goal-oriented behaviour	- Controllers for autonomous robots
IntellAct	- Parsing scenes into spatiotemporal graphs and semantic Event Chains - Probabilistic models of objects and their manipulation - Probabilistic rule learning - Dynamic motion primitives for trainable - Descriptions of robotic motor behaviour	- Understands and exploits the meaning of manipulations in terms of objects, actions and their consequences for reproducing human actions with machines - Analysis of low-level observation data for semantic content (Learning) and the synthesis of concrete behaviour (Execution)	- Interaction human-robot
NeuralDynamics	- Detection and selection of scene representation and sequence operation	- Links cognition to sensory and motor surfaces developing low-level mechanisms	- Study of cognition
NOPTILUS	- Cooperative & cognitive-based communications	- An effective fully-autonomous multi-AUV concept/system to overcome human	- Navigation

Acronym	Techniques	Contributions	Applications
	- Gaussian Process-based estimation as well - Perceptual sensory-motor - Learning motion control - Learning/cognitive based situation understanding and motion strategies	shortcomings, by replacing human-operated operations by a fully autonomous one	
RUBICON	- Robotics & Multi-agent systems - Novelty detection - Dynamic planning - Statistical and computational neuroscience methods - Robot/wireless sensor network middleware	- Self-learning robotic ecology consisting of a network of sensors, effectors and mobile robot devices, supporting one another's learning, fulfilling tasks more effectively and efficiently - Self-adaptation to environment changes	- Ambient assisted living - Security - Multi-agent system interactions
XPERIENCE	- Exploits prior experience via generative inner models	- Automating introspective, predictive, and interactive understanding of actions and dynamic situations based on structural bootstrapping, based on inferring new actions and knowledge from their locations and use in the process	- Autonomous robotics applications

ANNEX III. CLASS DIAGRAMS

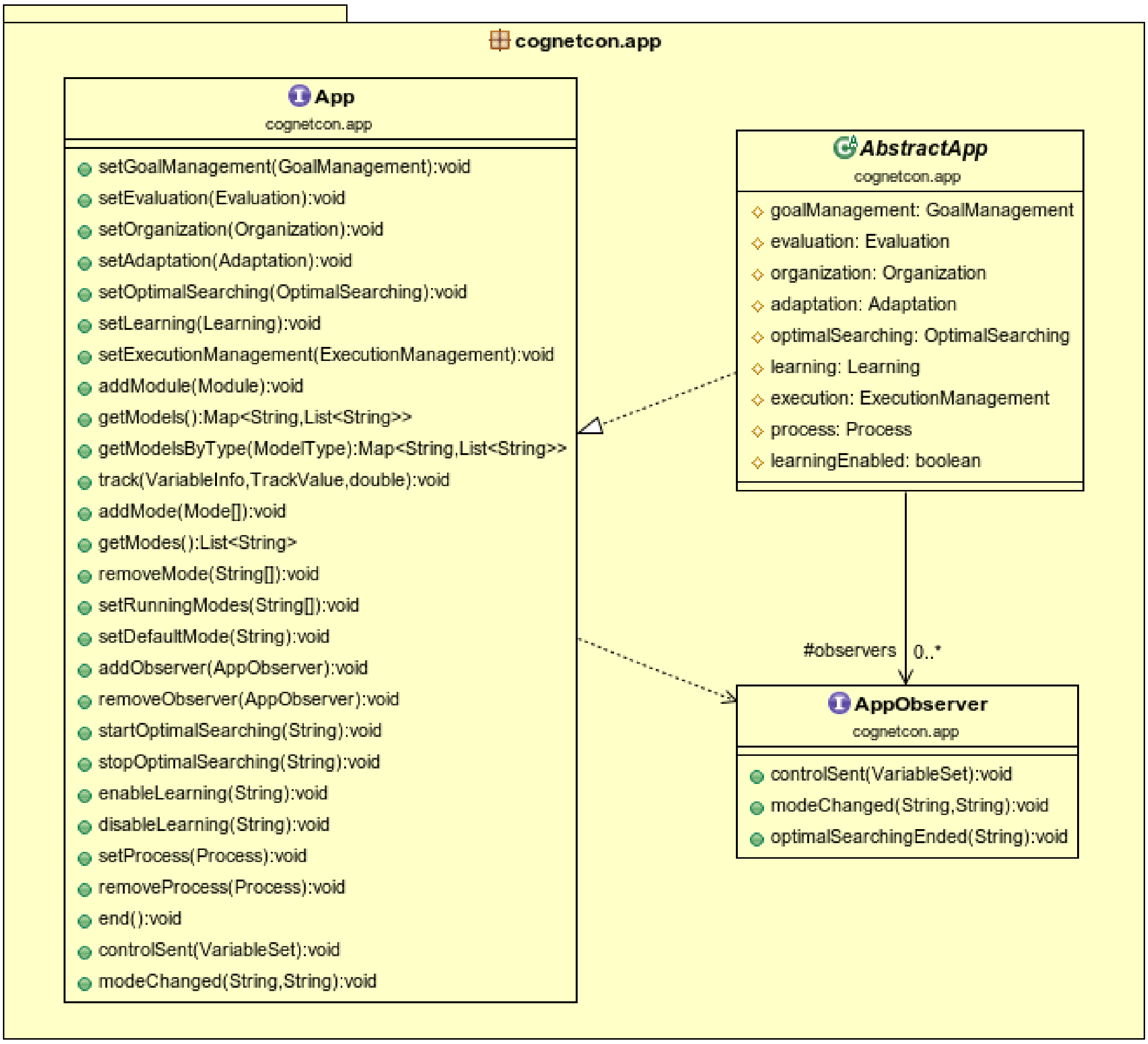

cognetcon.app
App
cognetcon.app
setGoalManagement(GoalManagement):void
setEvaluation(Evaluation):void
setOrganization(Organization):void
setAdaptation(Adaptation):void
setOptimalSearching(OptimalSearching):void
setLearning(Learning):void
setExecutionManagement(ExecutionManagement):void
addModule(Module):void
getModels():Map<String,List<String>>
getModelsByType(ModelType):Map<String,List<String>>
track(VariableInfo,TrackValue,double):void
addMode(Mode[]):void
getModes():List<String>
removeMode(String[]):void
setRunningModes(String[]):void
setDefaultMode(String):void
addObserver(AppObserver):void
removeObserver(AppObserver):void
startOptimalSearching(String):void
stopOptimalSearching(String):void
enableLearning(String):void
disableLearning(String):void
setProcess(Process):void
removeProcess(Process):void
end():void
controlSent(VariableSet):void
modeChanged(String,String):void
AbstractApp
cognetcon.app
goalManagement: GoalManagement
evaluation: Evaluation
organization: Organization
adaptation: Adaptation
optimalSearching: OptimalSearching
learning: Learning
execution: ExecutionManagement
process: Process
learningEnabled: boolean
#observers
0..*
AppObserver
cognetcon.app
controlSent(VariableSet):void
modeChanged(String,String):void
optimalSearchingEnded(String):void

Figure III.1 Class diagram of app package

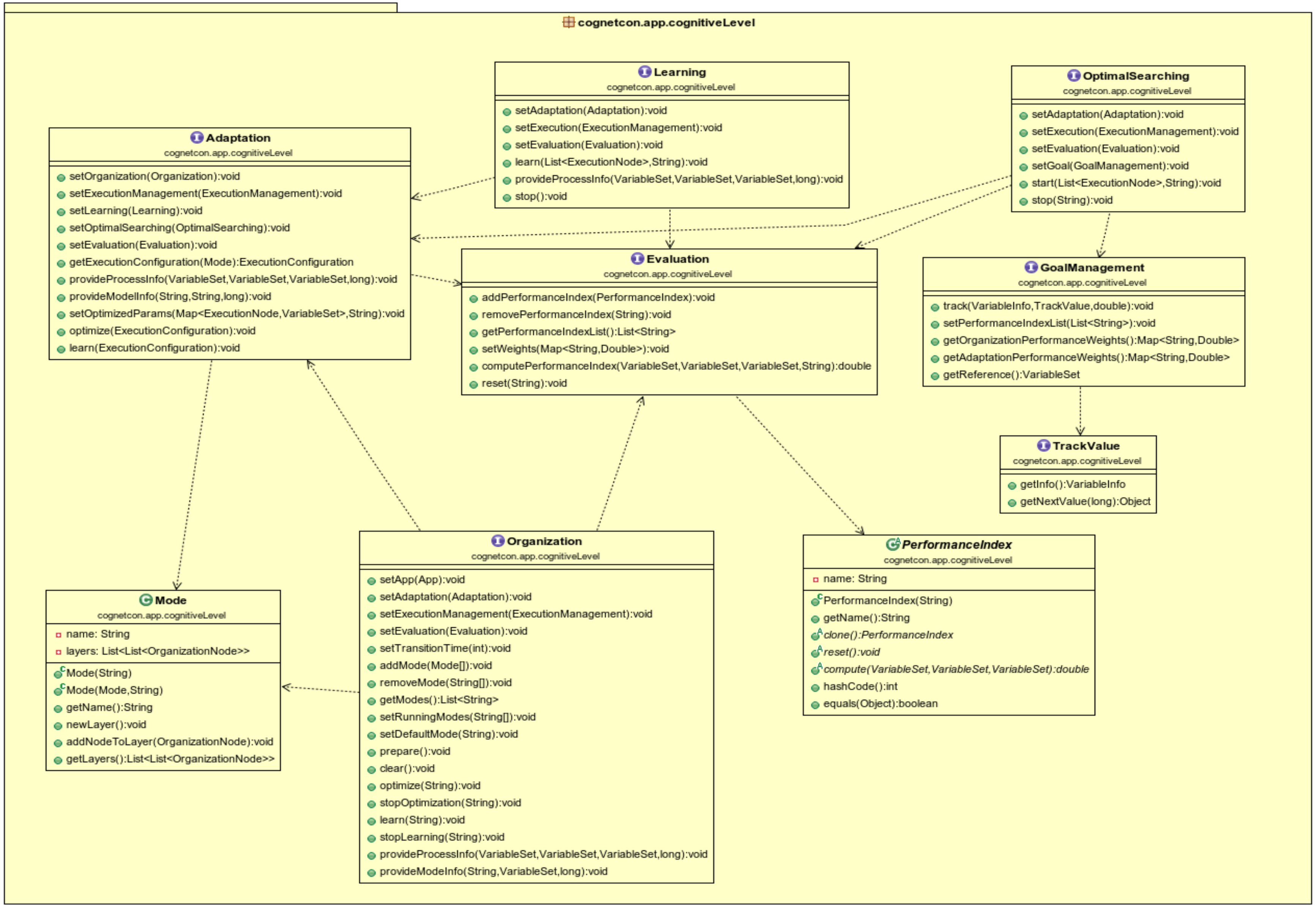

cognetcon.app.cognitiveLevel
Learning
cognetcon.app.cognitiveLevel
setAdaptation(Adaptation):void
setExecution(ExecutionManagement):void
setEvaluation(Evaluation):void
learn(List<ExecutionNode>,String):void
provideProcessInfo(VariableSet,VariableSet,VariableSet,long):void
stop():void
OptimalSearching
cognetcon.app.cognitiveLevel
setAdaptation(Adaptation):void
setExecution(ExecutionManagement):void
setEvaluation(Evaluation):void
setGoal(GoalManagement):void
start(List<ExecutionNode>,String):void
stop(String):void
Adaptation
cognetcon.app.cognitiveLevel
setOrganization(Organization):void
setExecutionManagement(ExecutionManagement):void
setLearning(Learning):void
setOptimalSearching(OptimalSearching):void
setEvaluation(Evaluation):void
getExecutionConfiguration(Mode):ExecutionConfiguration
provideProcessInfo(VariableSet,VariableSet,VariableSet,long):void
provideModelInfo(String,String,long):void
setOptimizedParams(Map<ExecutionNode,VariableSet>,String):void
optimize(ExecutionConfiguration):void
learn(ExecutionConfiguration):void
Evaluation
cognetcon.app.cognitiveLevel
addPerformanceIndex(PerformanceIndex):void
removePerformanceIndex(String):void
getPerformanceIndexList():List<String>
setWeights(Map<String,Double>):void
computePerformanceIndex(VariableSet,VariableSet,VariableSet,String):double
reset(String):void
GoalManagement
cognetcon.app.cognitiveLevel
track(VariableInfo,TrackValue,double):void
setPerformanceIndexList(List<String>):void
getOrganizationPerformanceWeights():Map<String,Double>
getAdaptationPerformanceWeights():Map<String,Double>
getReference():VariableSet
TrackValue
cognetcon.app.cognitiveLevel
getInfo():VariableInfo
getNextValue(long):Object
Organization
cognetcon.app.cognitiveLevel
setApp(App):void
setAdaptation(Adaptation):void
setExecutionManagement(ExecutionManagement):void
setEvaluation(Evaluation):void
setTransitionTime(int):void
addMode(Mode[]):void
removeMode(String[]):void
getModes():List<String>
setRunningModes(String[]):void
setDefaultMode(String):void
prepare():void
clear():void
optimize(String):void
stopOptimization(String):void
learn(String):void
stopLearning(String):void
provideProcessInfo(VariableSet,VariableSet,VariableSet,long):void
provideModelInfo(String,VariableSet,long):void
PerformanceIndex
cognetcon.app.cognitiveLevel
name: String
PerformanceIndex(String)
getName():String
clone():PerformanceIndex
reset():void
compute(VariableSet,VariableSet,VariableSet):double
hashCode():int
equals(Object):boolean
Mode
cognetcon.app.cognitiveLevel
name: String
layers: List<List<OrganizationNode>>
Mode(String)
Mode(Mode,String)
getName():String
newLayer():void
addNodeToLayer(OrganizationNode):void
getLayers():List<List<OrganizationNode>>

Figure III.2 Class diagram of cognitive level package

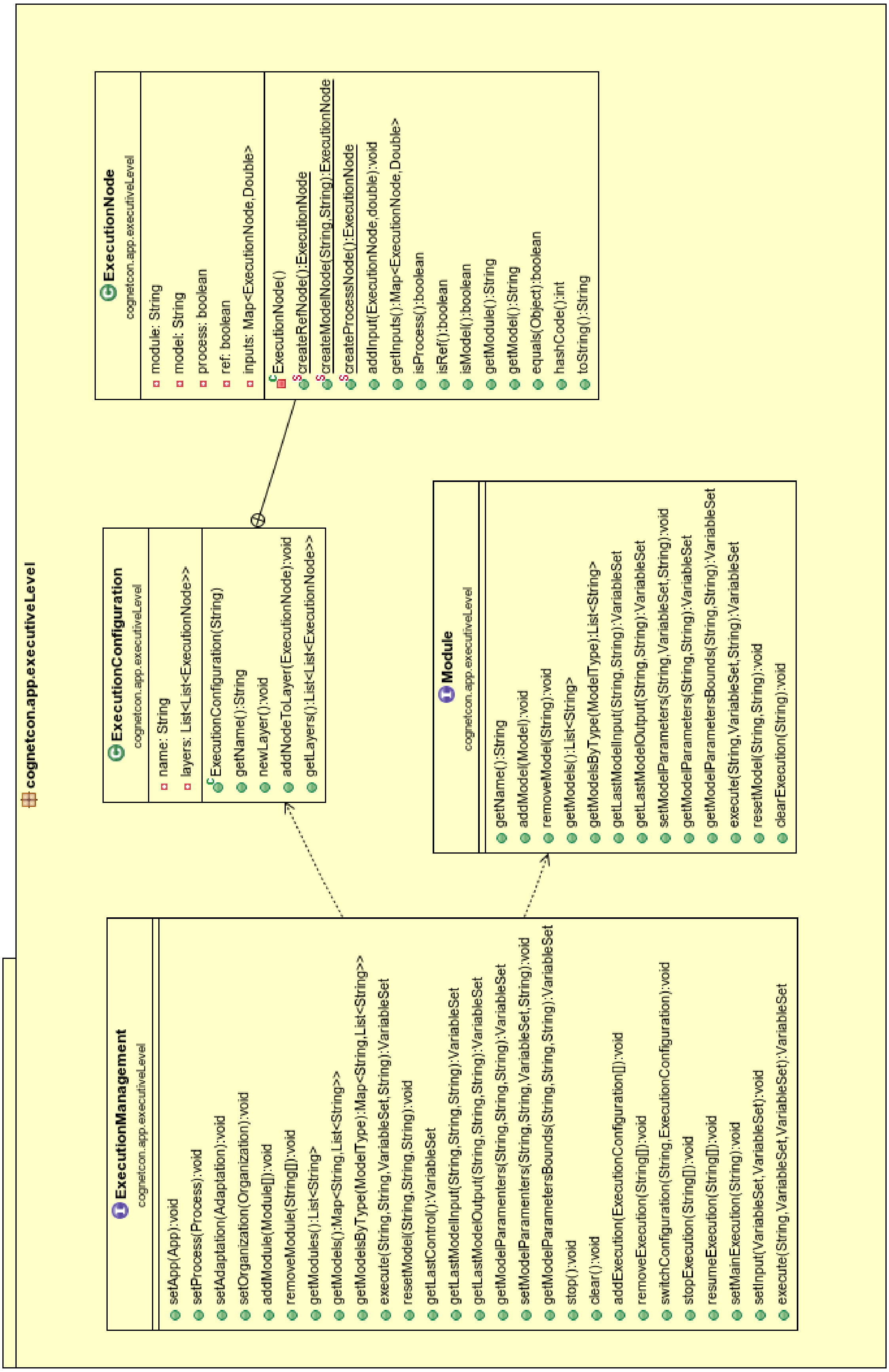

cognetcon.app.executiveLevel
ExecutionNode
cognetcon.app.executiveLevel
module: String
model: String
process: boolean
ref: boolean
inputs: Map<ExecutionNode,Double>
ExecutionNode()
createRefNode():ExecutionNode
createModelNode(String,String):ExecutionNode
createProcessNode():ExecutionNode
addInput(ExecutionNode,double):void
getInputs():Map<ExecutionNode,Double>
isProcess():boolean
isRef():boolean
isModel():boolean
getModule():String
getModel():String
equals(Object):boolean
hashCode():int
toString():String
ExecutionConfiguration
cognetcon.app.executiveLevel
name: String
layers: List<List<ExecutionNode>>
ExecutionConfiguration(String)
getName():String
newLayer():void
addNodeToLayer(ExecutionNode):void
getLayers():List<List<ExecutionNode>>
Module
cognetcon.app.executiveLevel
getName():String
addModel(Model):void
removeModel(String):void
getModels():List<String>
getModelsByType(ModelType):List<String>
getLastModelInput(String,String):VariableSet
getLastModelOutput(String,String):VariableSet
setModelParameters(String,VariableSet,String):void
getModelParameters(String,String):VariableSet
getModelParametersBounds(String,String):VariableSet
execute(String,VariableSet,String):VariableSet
resetModel(String,String):void
clearExecution(String):void
ExecutionManagement
cognetcon.app.executiveLevel
setApp(App):void
setProcess(Process):void
setAdaptation(Adaptation):void
setOrganization(Organization):void
addModule(Module):void
removeModule(String[]):void
getModules():List<String>
getModels():Map<String,List<String>>
getModelsByType(ModelType):Map<String,List<String>>
execute(String,String,VariableSet,String):VariableSet
resetModel(String,String,String):void
getLastControl():VariableSet
getLastModelInput(String,String,String):VariableSet
getLastModelOutput(String,String,String):VariableSet
getModelParamenters(String,String,String,String):VariableSet
setModelParamenters(String,String,VariableSet,String):void
getModelParametersBounds(String,String,String,String):VariableSet
stop():void
clear():void
addExecution(ExecutionConfiguration[]):void
removeExecution(String[]):void
switchConfiguration(String,ExecutionConfiguration):void
stopExecution(String[]):void
resumeExecution(String[]):void
setMainExecution(String):void
setInput(VariableSet,VariableSet,VariableSet):void
execute(String,VariableSet,VariableSet):VariableSet

Figure III.3 Class diagram of executive level package

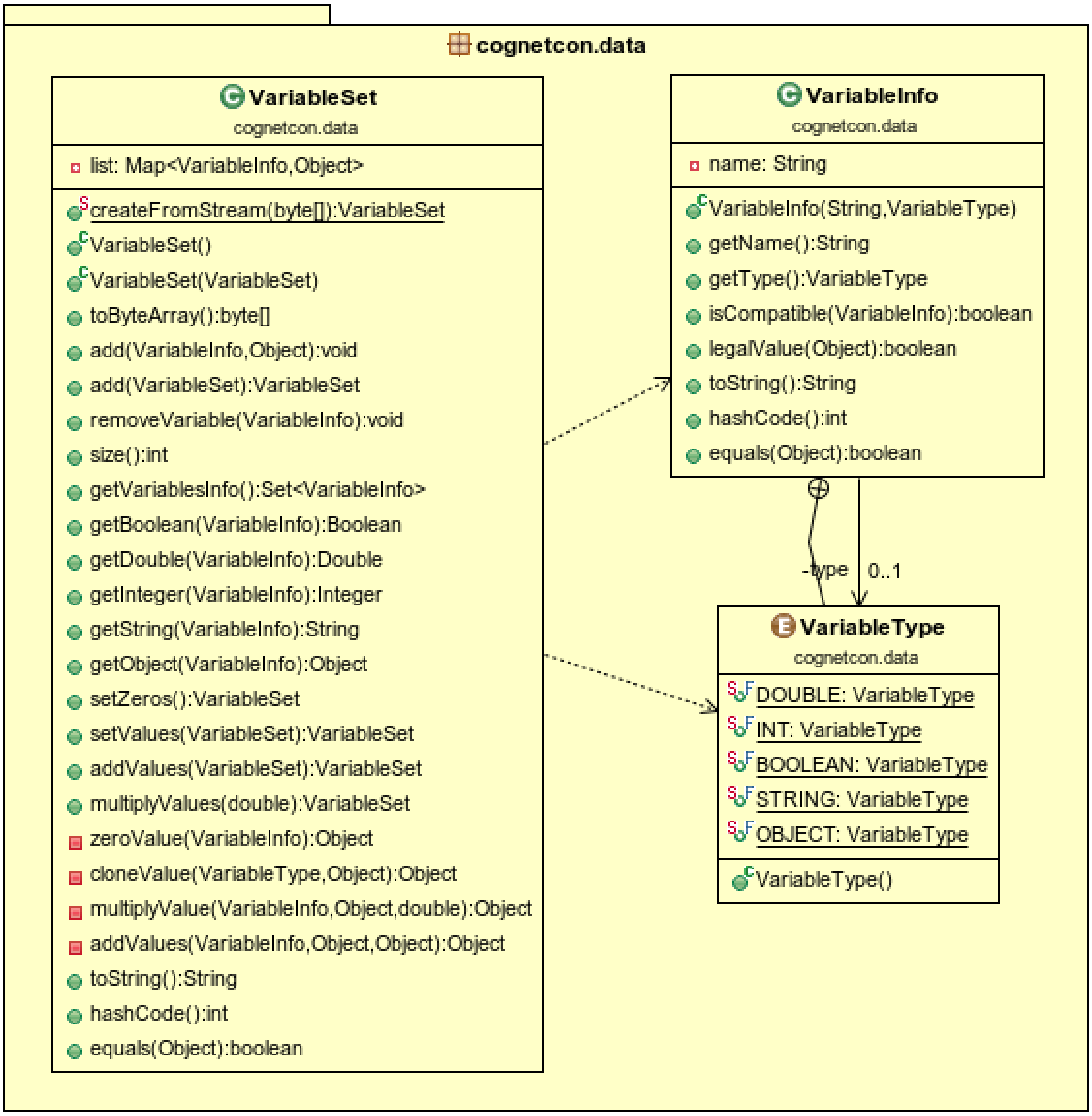

cognetcon.data
VariableSet
cognetcon.data
list: Map<VariableInfo,Object>
createFromStream(byte[]):VariableSet
VariableSet()
VariableSet(VariableSet)
toByteArray():byte[]
add(VariableInfo,Object):void
add(VariableSet):VariableSet
removeVariable(VariableInfo):void
size():int
getVariablesInfo():Set<VariableInfo>
getBoolean(VariableInfo):Boolean
getDouble(VariableInfo):Double
getInteger(VariableInfo):Integer
getString(VariableInfo):String
getObject(VariableInfo):Object
setZeros():VariableSet
setValues(VariableSet):VariableSet
addValues(VariableSet):VariableSet
multiplyValues(double):VariableSet
zeroValue(VariableInfo):Object
cloneValue(VariableType,Object):Object
multiplyValue(VariableInfo,Object,double):Object
addValues(VariableInfo,Object,Object):Object
toString():String
hashCode():int
equals(Object):boolean
VariableInfo
cognetcon.data
name: String
VariableInfo(String,VariableType)
getName():String
getType():VariableType
isCompatible(VariableInfo):boolean
legalValue(Object):boolean
toString():String
hashCode():int
equals(Object):boolean
-type
0..1
VariableType
cognetcon.data
DOUBLE: VariableType
INT: VariableType
BOOLEAN: VariableType
STRING: VariableType
OBJECT: VariableType
VariableType()

Figure III.4 Class diagram of data package

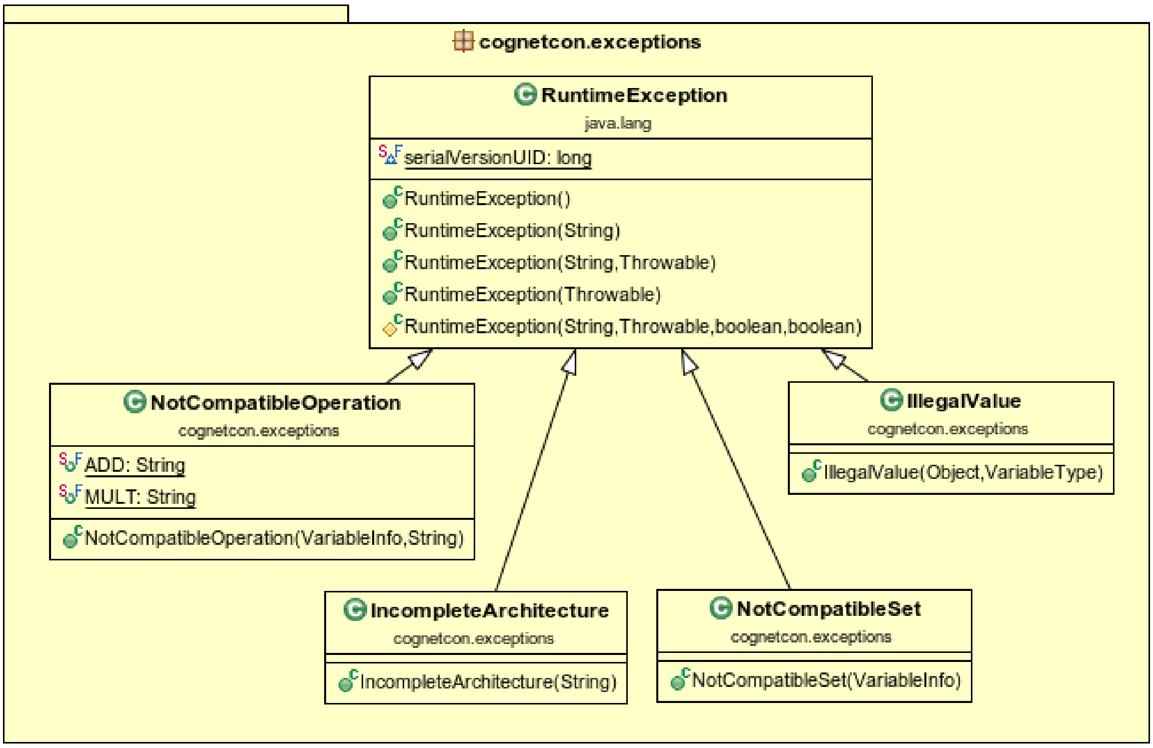


Figure III.5 Class diagram of exceptions package

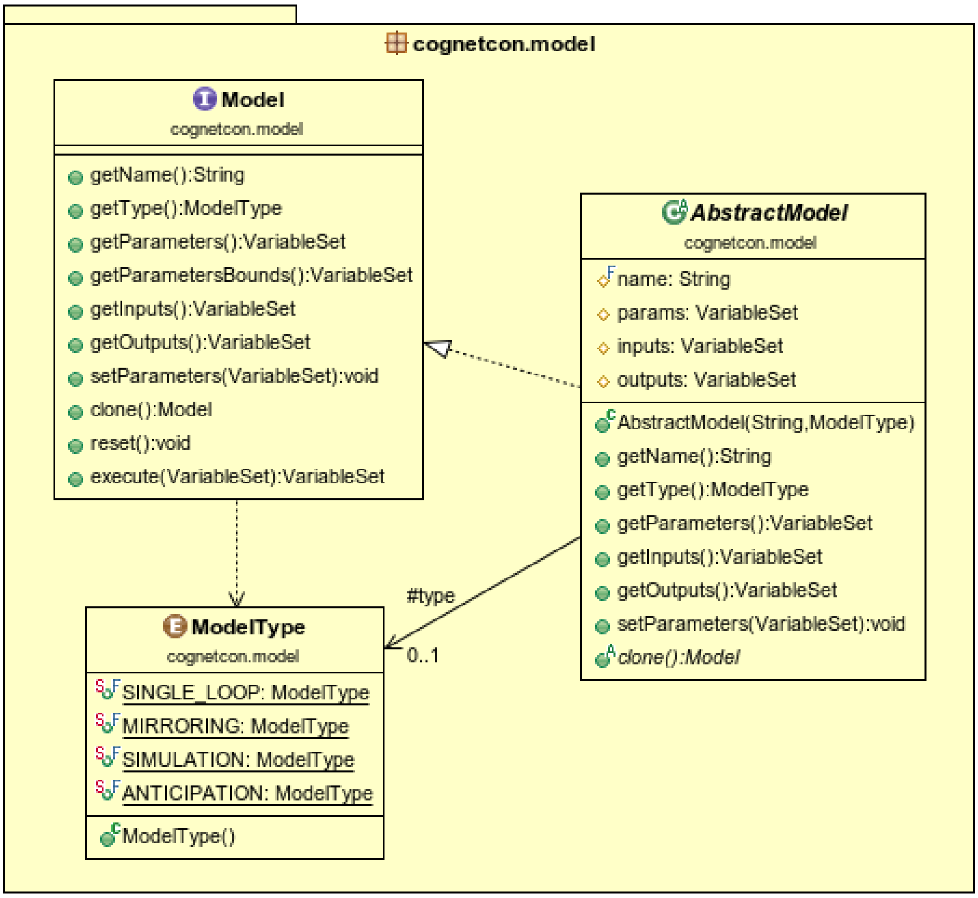


Figure III.6 Class diagram of model package

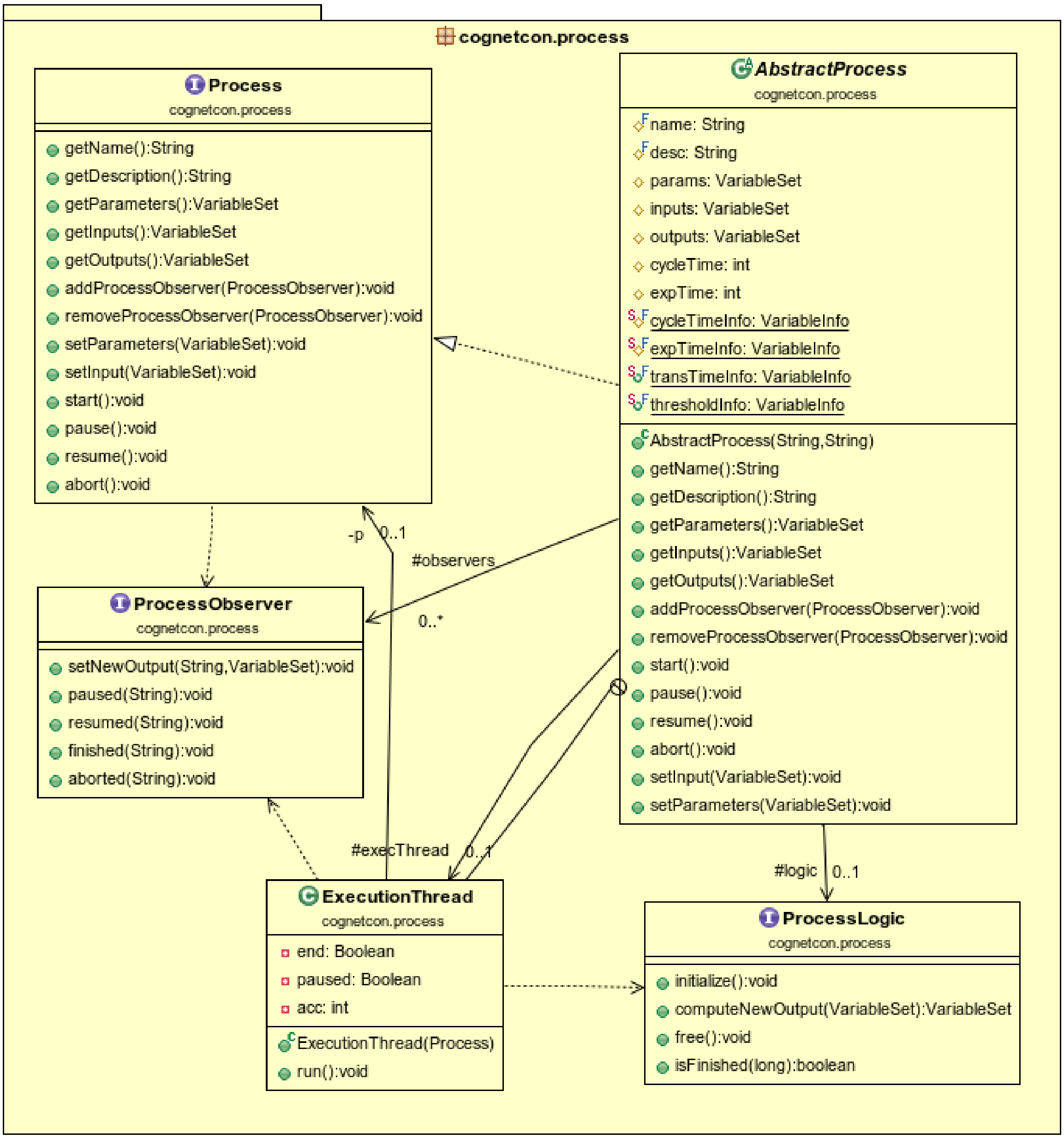


Figure III.7 Class diagram of process package

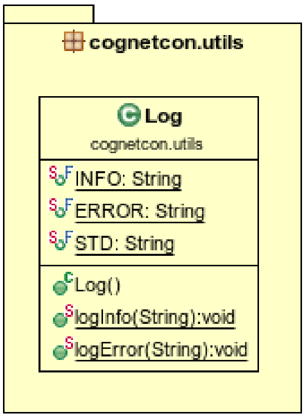
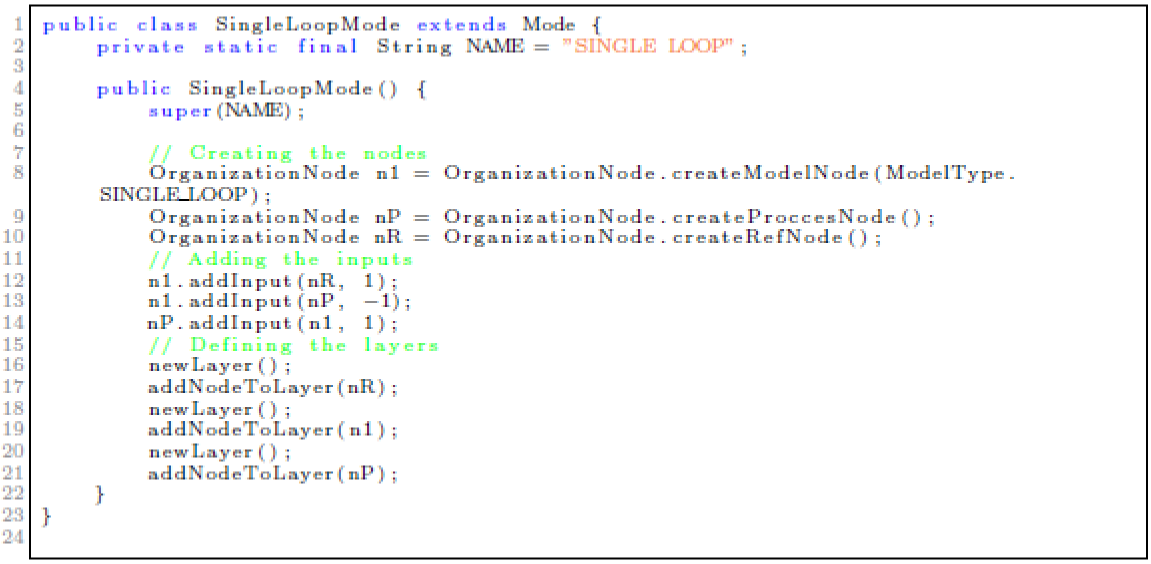


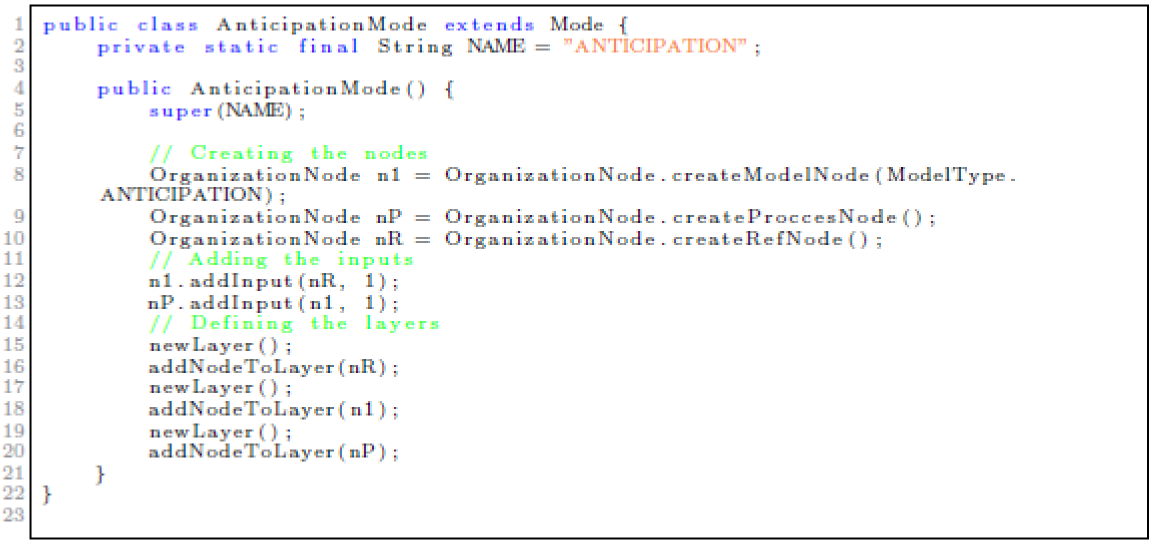
Figure III.8 Class diagram of utils package

ANNEX IV. CODE LIST



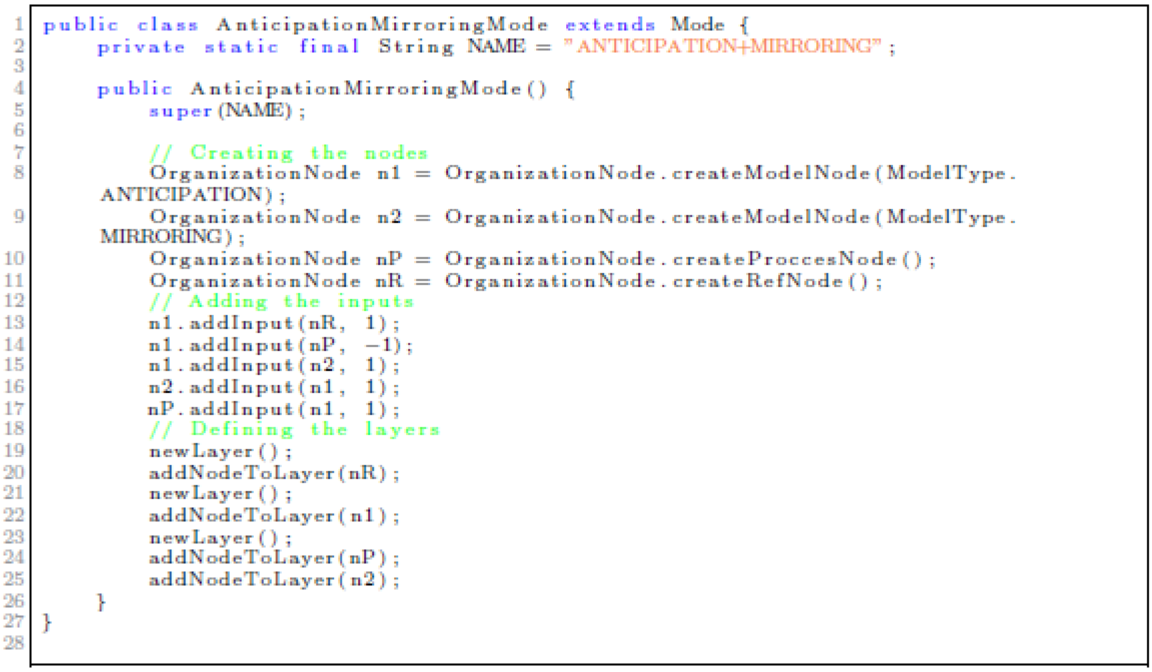
```java
public class SingleLoopMode extends Mode {
    private static final String NAME = "SINGLE LOOP";

    public SingleLoopMode() {
        super(NAME);

        // Creating the nodes
        OrganizationNode n1 = OrganizationNode.createModelNode(ModelType.
    SINGLE_LOOP);
        OrganizationNode nP = OrganizationNode.createProccesNode();
        OrganizationNode nR = OrganizationNode.createRefNode();
        // Adding the inputs
        n1.addInput(nR, 1);
        n1.addInput(nP, -1);
        nP.addInput(n1, 1);
        // Defining the layers
        newLayer();
        addNodeToLayer(nR);
        newLayer();
        addNodeToLayer(n1);
        newLayer();
        addNodeToLayer(nP);
    }
}
```

Figure IV.1 Single loop mode implementation



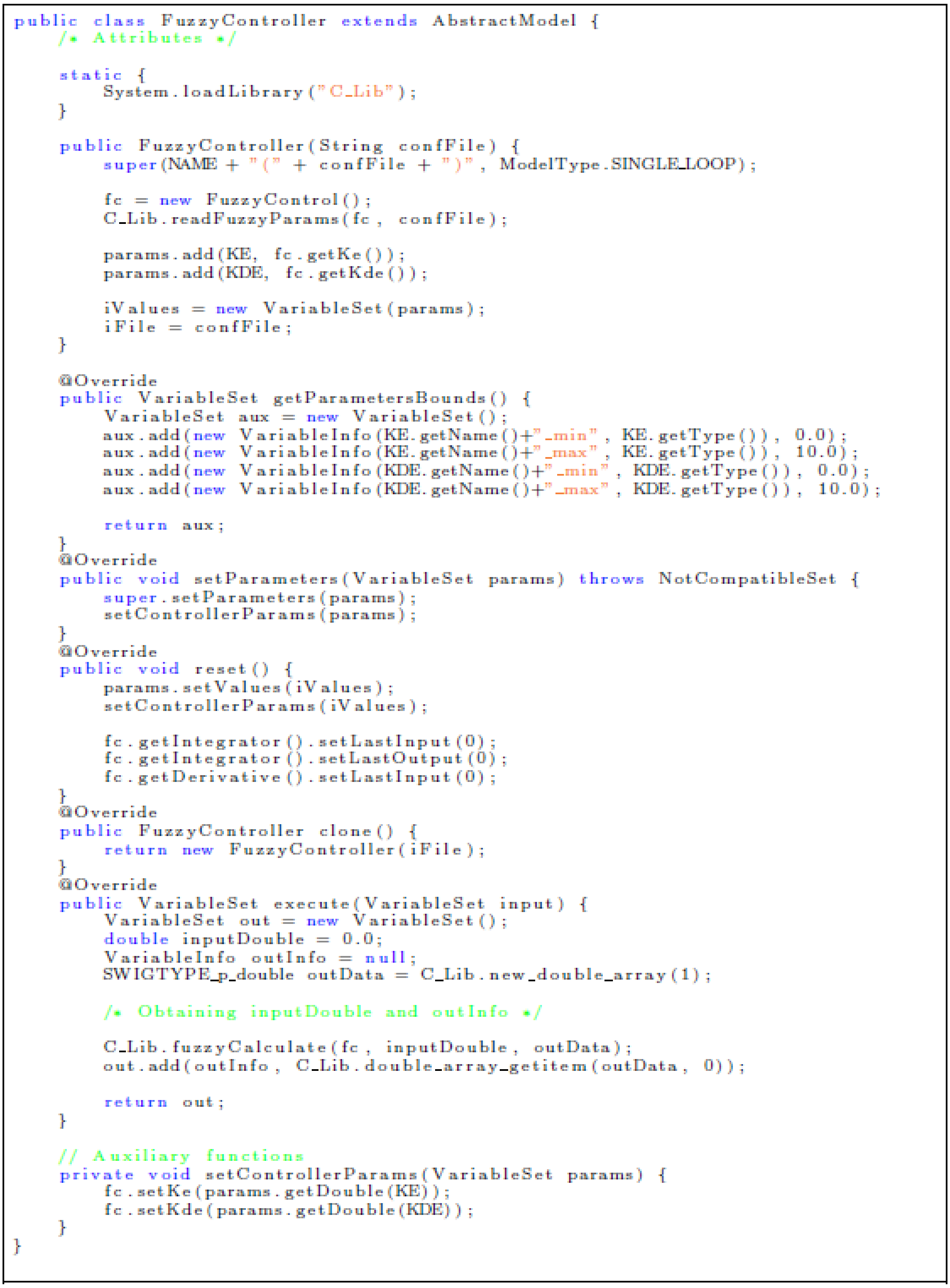
```java
public class AnticipationMode extends Mode {
    private static final String NAME = "ANTICIPATION";

    public AnticipationMode() {
        super(NAME);

        // Creating the nodes
        OrganizationNode n1 = OrganizationNode.createModelNode(ModelType.
    ANTICIPATION);
        OrganizationNode nP = OrganizationNode.createProccesNode();
        OrganizationNode nR = OrganizationNode.createRefNode();
        // Adding the inputs
        n1.addInput(nR, 1);
        nP.addInput(n1, 1);
        // Defining the layers
        newLayer();
        addNodeToLayer(nR);
        newLayer();
        addNodeToLayer(n1);
        newLayer();
        addNodeToLayer(nP);
    }
}
```

Figure IV.2 Anticipation mode implementation



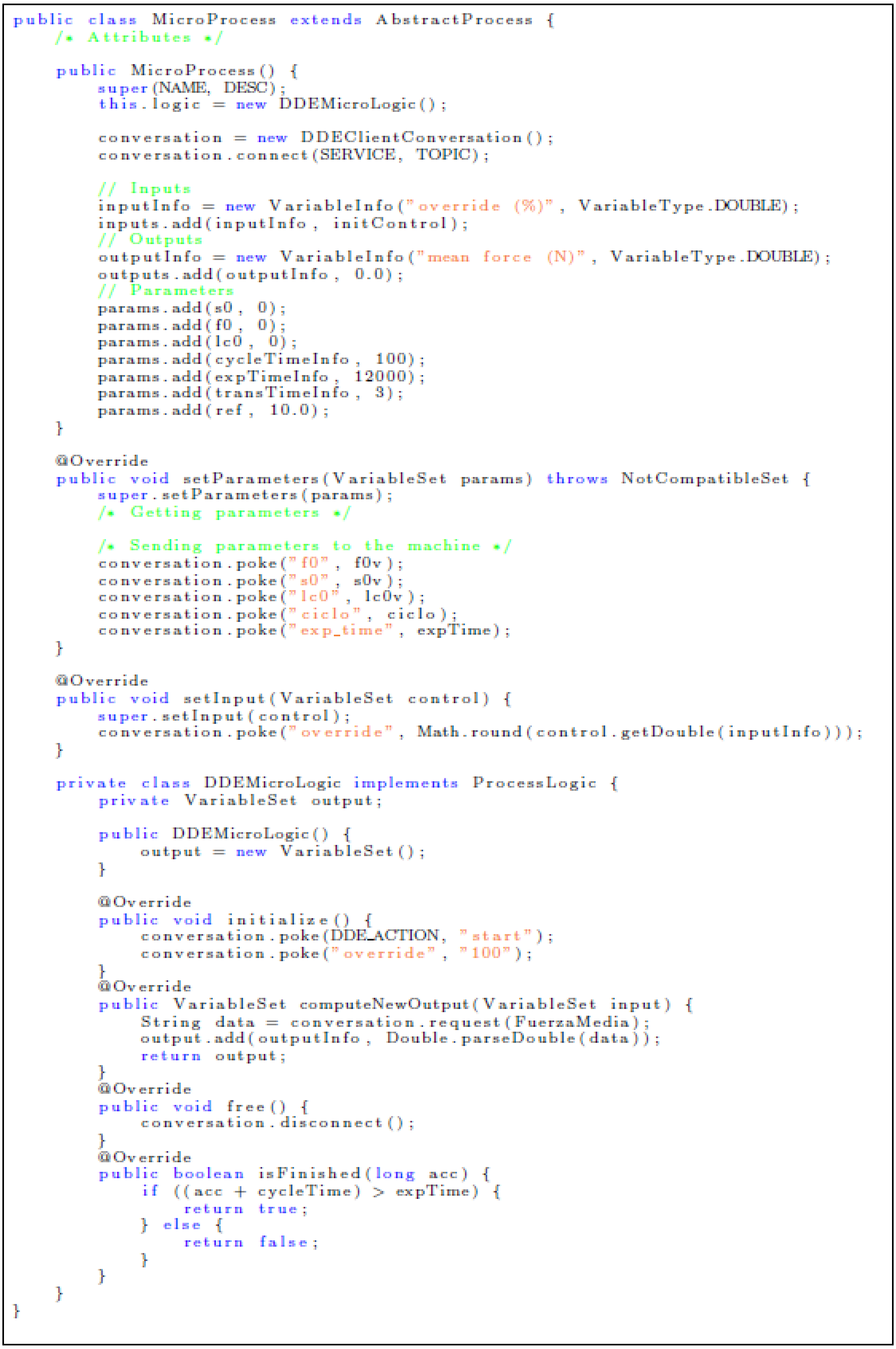
```java
public class AnticipationMirroringMode extends Mode {
    private static final String NAME = "ANTICIPATION+MIRRORING";

    public AnticipationMirroringMode() {
        super(NAME);

        // Creating the nodes
        OrganizationNode n1 = OrganizationNode.createModelNode(ModelType.
    ANTICIPATION);
        OrganizationNode n2 = OrganizationNode.createModelNode(ModelType.
    MIRRORING);
        OrganizationNode nP = OrganizationNode.createProccesNode();
        OrganizationNode nR = OrganizationNode.createRefNode();
        // Adding the inputs
        n1.addInput(nR, 1);
        n1.addInput(nP, -1);
        n1.addInput(n2, 1);
        n2.addInput(n1, 1);
        nP.addInput(n1, 1);
        // Defining the layers
        newLayer();
        addNodeToLayer(nR);
        newLayer();
        addNodeToLayer(n1);
        newLayer();
        addNodeToLayer(nP);
        addNodeToLayer(n2);
    }
}
```

Figure IV.3 Internal control (Anticipation+Mirroring) mode implementation



```
public class FuzzyController extends AbstractModel {
    /* Attributes */

    static {
        System.loadLibrary("C_Lib");
    }

    public FuzzyController(String confFile) {
        super(NAME + "(" + confFile + ")", ModelType.SINGLE_LOOP);

        fc = new FuzzyControl();
        C_Lib.readFuzzyParams(fc, confFile);

        params.add(KE, fc.getKe());
        params.add(KDE, fc.getKde());

        iValues = new VariableSet(params);
        iFile = confFile;
    }

    @Override
    public VariableSet getParametersBounds() {
        VariableSet aux = new VariableSet();
        aux.add(new VariableInfo(KE.getName()+"_min", KE.getType(), 0.0));
        aux.add(new VariableInfo(KE.getName()+"_max", KE.getType(), 10.0));
        aux.add(new VariableInfo(KDE.getName()+"_min", KDE.getType(), 0.0));
        aux.add(new VariableInfo(KDE.getName()+"_max", KDE.getType(), 10.0));

        return aux;
    }
    @Override
    public void setParameters(VariableSet params) throws NotCompatibleSet {
        super.setParameters(params);
        setControllerParams(params);
    }
    @Override
    public void reset() {
        params.setValues(iValues);
        setControllerParams(iValues);

        fc.getIntegrator().setLastInput(0);
        fc.getIntegrator().setLastOutput(0);
        fc.getDerivative().setLastInput(0);
    }
    @Override
    public FuzzyController clone() {
        return new FuzzyController(iFile);
    }
    @Override
    public VariableSet execute(VariableSet input) {
        VariableSet out = new VariableSet();
        double inputDouble = 0.0;
        VariableInfo outInfo = null;
        SWIGTYPE_p_double outData = C_Lib.new_double_array(1);

        /* Obtaining inputDouble and outInfo */

        C_Lib.fuzzyCalculate(fc, inputDouble, outData);
        out.add(outInfo, C_Lib.double_array_getitem(outData, 0));

        return out;
    }

    // Auxiliary functions
    private void setControllerParams(VariableSet params) {
        fc.setKe(params.getDouble(KE));
        fc.setKde(params.getDouble(KDE));
    }
}
```

Figure IV.4 Fuzzy controller implementation



```java
public class MicroProcess extends AbstractProcess {
    /* Attributes */

    public MicroProcess() {
        super(NAME, DESC);
        this.logic = new DDEMicroLogic();

        conversation = new DDEClientConversation();
        conversation.connect(SERVICE, TOPIC);

        // Inputs
        inputInfo = new VariableInfo("override (%)", VariableType.DOUBLE);
        inputs.add(inputInfo, initControl);
        // Outputs
        outputInfo = new VariableInfo("mean force (N)", VariableType.DOUBLE);
        outputs.add(outputInfo, 0.0);
        // Parameters
        params.add(s0, 0);
        params.add(f0, 0);
        params.add(lc0, 0);
        params.add(cycleTimeInfo, 100);
        params.add(expTimeInfo, 12000);
        params.add(transTimeInfo, 3);
        params.add(ref, 10.0);
    }

    @Override
    public void setParameters(VariableSet params) throws NotCompatibleSet {
        super.setParameters(params);
        /* Getting parameters */

        /* Sending parameters to the machine */
        conversation.poke("f0", f0v);
        conversation.poke("s0", s0v);
        conversation.poke("lc0", lc0v);
        conversation.poke("ciclo", ciclo);
        conversation.poke("exp_time", expTime);
    }

    @Override
    public void setInput(VariableSet control) {
        super.setInput(control);
        conversation.poke("override", Math.round(control.getDouble(inputInfo)));
    }

    private class DDEMicroLogic implements ProcessLogic {
        private VariableSet output;

        public DDEMicroLogic() {
            output = new VariableSet();
        }

        @Override
        public void initialize() {
            conversation.poke(DDE_ACTION, "start");
            conversation.poke("override", "100");
        }
        @Override
        public VariableSet computeNewOutput(VariableSet input) {
            String data = conversation.request(FuerzaMedia);
            output.add(outputInfo, Double.parseDouble(data));
            return output;
        }
        @Override
        public void free() {
            conversation.disconnect();
        }
        @Override
        public boolean isFinished(long acc) {
            if ((acc + cycleTime) > expTime) {
                return true;
            } else {
                return false;
            }
        }
    }
}
```

Figure IV.5 Microprocess implementation