

Article

# Classifying Intelligence in Machines: A Taxonomy of Intelligent Control <sup>†</sup>

Callum Wilson <sup>\*</sup>, Francesco Marchetti , Marilena Di Carlo , Annalisa Riccardi  and Edmondo Minisci 

Intelligent Computational Engineering Laboratory, Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow G1 1XJ, UK; francesco.marchetti@strath.ac.uk (F.M.); dicarlo.marilena@gmail.com (M.D.C.); annalisa.riccardi@strath.ac.uk (A.R.); edmondo.minisci@strath.ac.uk (E.M.)

\* Correspondence: callum.j.wilson@strath.ac.uk

† This paper is an extended version of our paper published in Wilson, C.; Marchetti, F.; Di Carlo, M.; Riccardi, A.; Minisci, E. Intelligent Control: A Taxonomy. In Proceedings of the 8th International Conference on Systems and Control (ICSC 2019), Marrakech, Morocco, 23–25 October 2019.

Received: 30 June 2020; Accepted: 18 August 2020; Published: 21 August 2020



**Abstract:** The quest to create machines that can solve problems as humans do leads us to intelligent control. This field encompasses control systems that can adapt to changes and learn to improve their actions—traits typically associated with human intelligence. In this work we seek to determine how intelligent these classes of control systems are by quantifying their level of adaptability and learning. First we describe the stages of development towards intelligent control and present a definition based on literature. Based on the key elements of this definition, we propose a novel taxonomy of intelligent control methods, which assesses the extent to which they handle uncertainties in three areas: the environment, the controller, and the goals. This taxonomy is applicable to a variety of robotic and other autonomous systems, which we demonstrate through several examples of intelligent control methods and their classifications. Looking at the spread of classifications based on this taxonomy can help researchers identify where control systems can be made more intelligent.

**Keywords:** intelligent control; taxonomy; robotics; neural network control; fuzzy logic control; evolutionary algorithms; artificial intelligence

---

## 1. Introduction

Over the past century, more advanced methods have become necessary to handle an increase in the complexity of control problems. We now require control systems which can operate in very challenging environments with limited knowledge. This motivated the use of AI techniques in control to incorporate human reasoning. The combination of AI with theories from automatic control and operations research is referred to as “Intelligent Control” [1].

IC has received a great deal of attention in many control applications since the term was first coined by Fu [2]. Its applications are wide ranging and especially useful in robotics and other autonomous machines that seek to minimise human intervention. Due to its wide use, a “terminology war” ensued where there were several competing definitions for different concepts in IC—especially “adaptive” and “learning” control [3]. Following this era, there are now concrete definitions for the various concepts relating to IC. The definition of IC we present here is based on the work of Saridis and Antsaklis, which both give clear definitions [1,4]. Under this definition there are still many different types of control systems that can be classed as IC with varying levels of complexity. The goal of this work is to quantify the similarities and differences between these control systems by defining levels of intelligence

that an IC system can possess. This allows comparisons to be made between different methods and applications and highlights gaps where further research is needed to develop IC techniques.

When considering the intelligence of machines, it is clear that “intelligence” cannot be a binary label but rather a scale from non-intelligent to highly-intelligent. As pointed out in [3], it is hard to strictly define a threshold for something to be considered intelligent, as for some definitions even a thermometer could be classed as intelligent. Obviously, in this extreme example a thermometer possesses a very low level of intelligence—if any. The problem then becomes quantifying how intelligent a controller is, but this measurement will certainly be subjective.

From the initial development of control theory up to the present day, one key concept in control has been uncertainty. This is where some aspect of a system is unknown due to lack of knowledge or stochastic behaviour. As control systems developed over the past century they have expanded their ability to autonomously deal with uncertainties to a point where they require human-like behaviours such as reasoning and learning. This can be considered intelligence [5]. A control problem where everything about the system is known can be very simple to solve using long known and understood methods. Accounting for uncertainties makes this more difficult. Here we suggest that the level of intelligence of a control system is related to the level of uncertainty at design time: more intelligent controllers cope with greater levels of uncertainty.

In this work, we propose a novel taxonomy of IC, which classifies the level of intelligence of a controller. A contribution of this work is the multi-dimensional view of IC, which considers different aspects of these systems that can possess intelligence. There are other works that attempt to classify levels of IC such as that of Krishnakumar [6]. This has four levels of intelligence that focus on a controller’s ability to self-improve and does not consider how it deals with uncertainties. Furthermore, each level in their classification is additive to the previous. Here we show that, in practice, control systems can have different levels of classification in each of the dimensions of IC, which are independent of each other and therefore not additive. An aerospace industry survey from the American Institute of Aeronautics and Astronautics also defines six “stages of intelligent reasoning” for spacecraft [7]. These classifications are specific to spacecraft operations. Here we demonstrate through examples that the taxonomy can be applied to any field that uses IC.

The remainder of this paper is organised as follows. Section 2 describes the development of control systems towards intelligent control with a particular focus on the ability to handle uncertainties. Based on literature we provide a definition of IC in section 3 and describe the dimensions of IC. From these dimensions we propose a taxonomy of IC in section 4. Finally we demonstrate the application of the taxonomy in section 5 with some relevant examples and show the current spread of methods and their classifications.

## 2. Path to Intelligent Control

Control of dynamical systems has been of fundamental importance in engineering for centuries and remains a widely studied field. One of the earliest examples of a control system is Watt’s flyball governor from the late 18th century [8]. This is designed to relieve steam pressure in an engine by means of rotating masses. It can also be considered to be one of the first examples of feedback control, where the controller measures a quantity in the system and uses this to dictate its actions. The governor also demonstrates the importance of mathematical models for designing and operating control systems, which was important in refining the design and explaining unexpected behaviour observed in operation. Modern control theory originated more recently in the early 20th century and has developed rapidly since. Here we describe this progression and how it leads towards the modern definition of intelligent control.

At its most fundamental, a controller is designed to make a system behave in a desired manner. To do this a controller takes actions on the system through potentially several types of actuator. The simplest type of control system does not observe measurements from the system and decides what actions to take based only on the desired system state. This is called open-loop control (Figure 1).

Designing an effective open-loop control system in practice requires a precise model of the system being controlled. A simple controller can then be the inverse model of the system, which takes desired states as input and returns the necessary actions to achieve that state. These types of controllers that do not observe the system are not intelligent and are generally incapable of coping with any uncertainties.

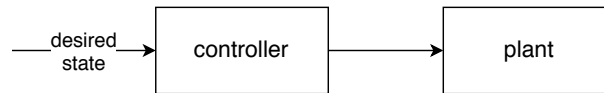


Figure 1. Open loop control scheme.

In most real systems, we expect their mathematical models to be imperfect. For this reason, most controllers use measurements of the system’s state to dictate their control actions. This is feedback control, also known as closed loop control (Figure 2). The earliest literature discussing feedback control comes from Nyquist in 1932 [9]—also referred to as ‘regeneration’ in his work. Feedback is the foundation of most modern control systems. The analysis of these closed-loop systems yields very important concepts for control systems such as stability and tracking error. Having feedback from the system removes the requirement for a precise model; however, to design an effective feedback controller without online adaptation still requires knowledge of the system dynamics.

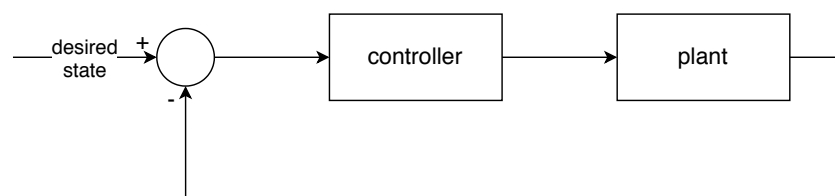


Figure 2. Feedback control scheme.

Within the realms of feedback control there are a multitude of methods and architectures for designing controllers. In engineering this naturally leads to the question of how to create the “best” control system. The concept of creating a controller with the quantifiably best performance is encapsulated in the field of optimal control. Techniques from optimal control solve the problem of maximising or minimising a measurable characteristic of a dynamical system. This field has its origins from Bellman whose pioneering work in dynamic programming defined criteria for optimal systems [10]. Theories from optimal control are still very influential in the design of modern controllers.

The control systems we have considered until now only deal with uncertainties, which is due to our lack of knowledge of the system. When we consider systems that behave in a non-deterministic manner, we require further advanced methods to deal with these types of uncertainty. This falls into the class of control systems known as stochastic control (Figure 3). In this case, uncertainties about the system are modelled as probability densities and cannot be specified as exact values [1].

At this point in the development of control systems, lack of system knowledge becomes a more significant issue. The control schemes described previously require knowledge of the system dynamics and any uncertainties must be statistically quantifiable. Beyond these systems that can be modelled, there are classes of systems where the dynamics may change over time, which degrades the controller’s performance. Furthermore, for certain systems the dynamics may be entirely unknown or incompletely known and so the methodologies discussed so far cannot be used. These are referred to as Self-Organising Control (SOC) systems, simply defined as any system with features “beyond stochastic control systems” [1].

Adaptive control methods handle changing environments by adjusting the control scheme online. This means the controller maintains a favourable performance even as the environment varies. These methods broadly fall under two categories: direct and indirect. Indirect adaptive control schemes do not alter the controller directly, but instead adapt other components that affect the control

scheme, such as a system model. Direct adaptive control schemes adapt the controller parameters themselves instead. These approaches can also be referred to as Parameter-Adaptive (indirect) or Performance-Adaptive (direct) SOC. Online adaptation is not only desirable in changing environments. Where the system being controlled is deterministic and “static”, adaptive control can also be used to tune the controller performance online. Most often, however, adaptive control is used where the environment behaviour is broadly understood at design time, but subject to significant uncertainties, for example, in parameters of the mathematical model.

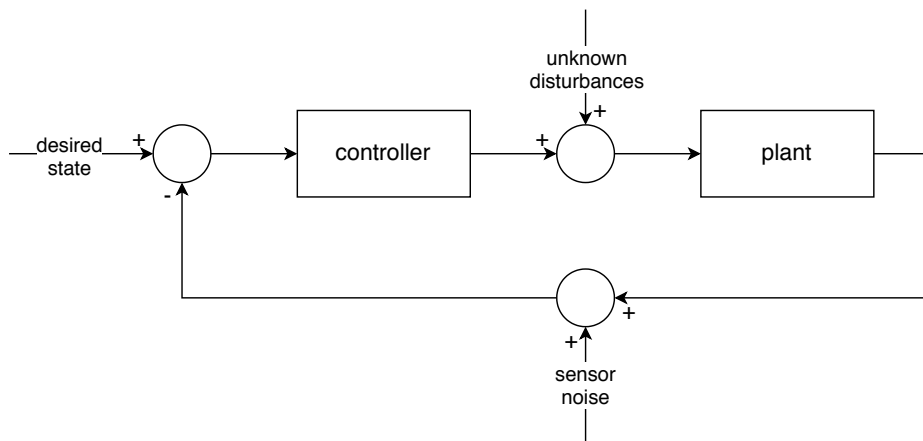


Figure 3. Stochastic control scheme.

Learning control is the final step towards intelligent control methods. This can be seen as a more specific form of Self-Organising Control (SOC) where the controller retains information pertaining to the system’s operation and uses this knowledge to alter its control scheme (Figure 4). This is where control systems begin to incorporate planning, where future actions can be selected by the system in advance based on its knowledge. At this level of development we also begin to see the more behavioural approach towards control where we aim to mimic human problem solving. As learning is a fundamental aspect of human intelligence, so the ability to learn about a system demonstrates a degree of intelligence in that control system. Learning can occur offline such that the control system is trained before operation, or online where knowledge is accumulated during its operation. This distinction is important when it comes to intelligent control systems, as will be discussed.

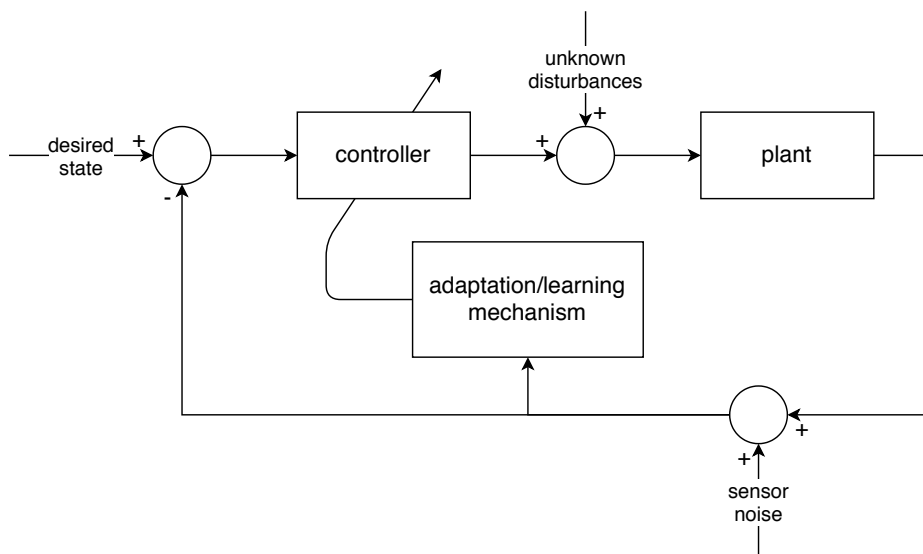


Figure 4. Adaptive or learning control scheme.

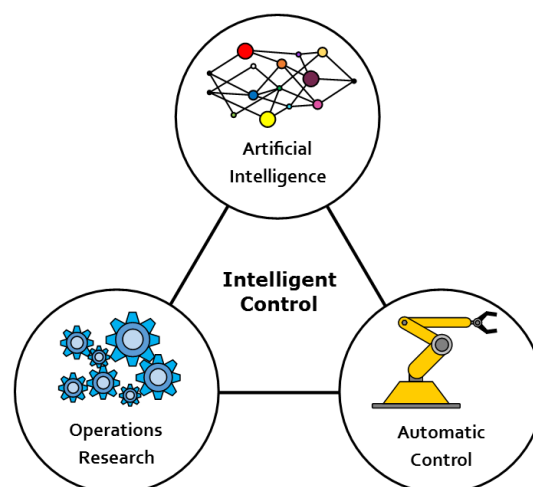
### 3. Defining Intelligent Control

As with many new concepts, the term “Intelligent Control” very quickly became widely used and often abused by many scholars both from the control community and wider fields. This made it difficult to create a suitable definition for IC since it was so commonly used to describe disparate concepts. As a result, in 1993 the IEEE Control Systems Society designated a task force to research and define “Intelligent Control” [4]. In their report they gave the following defining characteristics of IC systems:

“An intelligent control system is designed so that it can autonomously achieve a high level goal, while its components, control goals, plant models and control laws are not completely defined, either because they were not known at the design time or because they changed unexpectedly.”

This importantly shows that IC systems deal not only with system uncertainties, but also cases where the controller does not have specifically defined goals or structures. Saridis gives a more general definition of IC as an interaction between three fields: Artificial Intelligence, Operations Research, and Automatic Control Systems (Figure 5) [1]. This builds on the definition given by Fu, who originally described IC systems as the “intersection of artificial intelligence and automatic control” [2].

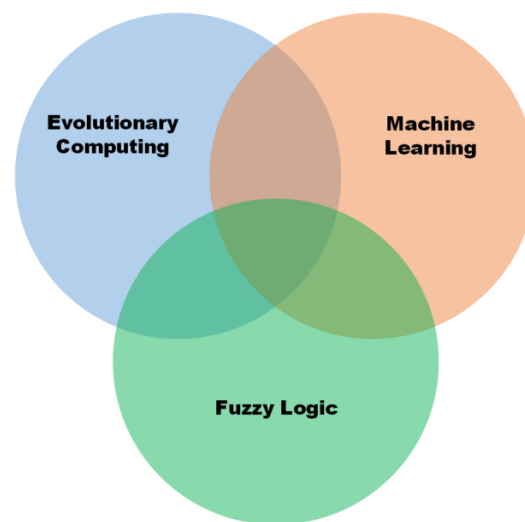
Considering the definitions of conventional control methodologies presented previously, direct or indirect adaptive control systems and learning control systems can be considered intelligent where they incorporate AI techniques. It is important to note, however, that not all adaptive control systems are intelligent since it is possible to derive adaptive systems using analytical formulations, therefore missing the AI component. Similarly, a controller is not necessarily intelligent if it is derived using AI techniques as it must still show adaptivity or learning online to be classed as intelligent. Both these points represent the most common misunderstandings in what is, and is not, classed as IC. For example, a controller may use AI to define its control scheme offline using a system model and then, when in operation, not update its control scheme further. Such a controller is not classed as IC since it does not adjust to substantial environmental uncertainties.



**Figure 5.** Intelligent control is the interaction of the fields of artificial intelligence, operations research, and automatic control.

### 3.1. Methods for Intelligent Control

While the methods within the three fields of automatic control, artificial intelligence, and operations research are broad ranging, there are a few notable techniques from AI that are most commonly used for IC. These are Machine Learning (ML), Evolutionary Computing (EC), and Fuzzy Logic (FL). Other intelligent techniques do appear in control, however these are the most frequently seen in the literature. Furthermore, there are significant synergies between these methods, which means they are often combined in control systems, as shown schematically in Figure 6.



**Figure 6.** Synergies of artificial intelligence techniques used for intelligent control.

Below we briefly describe how each method listed above can be incorporated into intelligent control schemes. This is not an exhaustive list but demonstrates the most common architectures of intelligent controllers.

#### 3.1.1. Machine Learning

Since these techniques are by definition used to learn, they are well suited to being used in IC. Of all the techniques classed as Machine Learning (ML), Neural Networks (NNs) and their closely related counterparts Support Vector Machines (SVMs) are by far the most common. While traditionally used in classification problems, these architectures can also be used as non-linear function approximators. This means they can be used to approximate uncertainties, model entire systems, or directly incorporated as controllers [11].

#### 3.1.2. Evolutionary Computing

This broad field of AI encompasses Evolutionary Algorithms (EAs), Genetic Algorithms (GAs), and Genetic Programming (GP). While these have subtle differences in approach, they are all based on biological evolution and develop solutions by “evolving” populations of potential solutions, which are evaluated based on a fitness function. They are often very computationally expensive, which can be prohibitive for online learning, however there are still several applications where Evolutionary Computing (EC) is used for online parameter adjustment in IC [12].

#### 3.1.3. Fuzzy Logic

The name of this field reflects its main characteristic of dealing with partial truth to develop reasoning. One of the advantages of Fuzzy Logic (FL) is its ability to incorporate expert knowledge when designing a controller and thus the interpretability of the derived control scheme. FL control

schemes are also commonly used in a non-intelligent manner where its control law is not updated online, but it can also be used intelligently by adjusting its parameters while operating [13].

### 3.1.4. Hybrid Methods

Due to the different strengths and weaknesses of each of the methods listed previously, they are often combined to exploit the advantages of each method. For example, a Neural Network (NN) can be used to approximate membership functions in a Fuzzy Logic (FL) controller, or a Neural Network (NN) can have its structure updated using Evolutionary Computing (EC). Any IC methods which employ multiple AI techniques are here referred to as “hybrid” methods.

## 3.2. Dimensions of Intelligent Control

Since IC deals primarily with substantial uncertainties, it is sensible to define the level of intelligence of a controller in terms of the level of uncertainty in its task. In the task force definition of IC, there are three clear dimensions where uncertainty can be present: the environment (represented by plant models and the environment models in which the plant has to operate), the controller laws and components, and the control goals. In more abstract terms, this is *what* is being controlled, *how* it is being controlled, and *why* it is being controlled.

### 3.2.1. Environment

We consider knowledge of the environment to be the ability to express a model of the environment mathematically, where under the term environment we consider both the plant model and the environment model in which it has to operate. The design of a control system conventionally requires such a model and the level of knowledge of the model affects the level of intelligence necessary in the controller. Equation (1) shows the general form of a non-linear system being controlled:

$$\dot{x} = f(x, u) \quad (1a)$$

$$y = h(x) \quad (1b)$$

where  $y$  is the system output,  $u$  is the system input,  $x$  is the system's state variables, and the functions  $f$  and  $h$  are mappings (linear or non-linear) from their inputs to appropriately dimensioned vectors. In the following equations, for simplicity we will only consider expressions for  $\dot{x}$ . The environment model may also contain some parameters,  $A = \{a_1, a_2, \dots, a_{na}\}$ , which vary with time. In this case the deterministic mapping from current state and control action to system output in Equation (1) no longer applies and now becomes time dependent. This is shown in Equation (2)

$$\dot{x} = f(x, u, A(t)) \quad (2)$$

Thus far we have assumed the function  $f$  to be known to a precision that allows reasonable tracking accuracy between the model and real environment. This is not possible when the environment's dynamics are poorly understood. We indicate this in Equation (3) with the function  $\hat{f}$  representing an uncertain mapping.

$$\dot{x} = \hat{f}(x, u, A(t)) \quad (3)$$

### 3.2.2. Controller

Similarly to the environment, a controller can be mathematically modelled with varying levels of knowledge about its components. More intelligent controllers are more flexible and have less precise knowledge of their control laws at design time. A general feedback controller is described as follows:

$$u = g(e) \quad (4)$$

where  $e = y_d - y$  is the error between the desired system output,  $y_d$  and actual system output. This represents a controller with fixed parameters that are selected at design time. A general adaptive controller has control parameters  $K = \{k_1, k_2, \dots, k_{nk}\}$ , which can vary with any number of observations. Such a controller is described in Equation (5).

$$u = g(e, K(\cdot)) \quad (5)$$

The error  $e$  between desired and true system output can also be subject to significant uncertainties relating to the behaviour of sensors and actuators. Even if the environment itself is stationary and deterministic, there may be errors, for example, in the measurements or unmodelled actuator dynamics. This case is described as shown:

$$u = g(\hat{e}, K(\cdot)) \quad (6)$$

where  $\hat{e} = y_d - \hat{y}$  is the measured error given the measured and uncertain state variables  $\hat{y}$ . The controller must then cope with these uncertain measurements. More sophisticated controllers display more significant variations in their structure than just the control parameters. In this case, there may be several different control laws to select based on observations, or new control laws may be derived online. A general form of such a controller is given here:

$$u = \begin{cases} g_1(\hat{e}, K_1(\cdot)) \\ g_2(\hat{e}, K_2(\cdot)) \\ \vdots \\ g_{ng}(\hat{e}, K_{nk}(\cdot)) \end{cases} \quad (7)$$

### 3.2.3. Goals

Compared to the previous two dimensions, goals are more abstract in general and less rigorously mathematically defined. The level of knowledge of goals can then be thought of as how well it could be expressed mathematically, as well as the level of awareness of goals in the controller. In most cases a controller's goal is defined as fulfilling some stability criterion or maintaining some performance measure across its operating range. In this case, the goal is entirely defined at design time and the controller has no awareness of this goal.

Another approach to defining control goals is to have some cost function that gives the controller an indication of its performance in a task. The controller then seeks to minimise this cost function with its control policy. In doing so the controller now has some awareness of its goals and creates ways to achieve them instead of following prescribed routines to achieve a predetermined level of performance.

Beyond control systems with defined goals or cost functions, the goals become more abstract and defined in high level language rather than mathematically. In some cases, specific short-term goals may change over time as determined by the controller's internal planning. This is done with respect to some global goal, which remains constant. In cases where a global goal cannot be defined mathematically and the controller can only be given high level goals, this requires an intelligent system to deduce how to act appropriately and achieve such goals.

## 4. Taxonomy

As discussed in Section 3, IC methods are used where there is a substantial lack of knowledge at design time. This lack of knowledge comes under three main categories: the environment, the controller, and the goals. Within each of these categories, any controller, including conventional ones, can have a varying degree of knowledge at design time. Here we present a classification scheme for IC methods, which is based on the level of knowledge present in the control system at design time. In each case the highest level of uncertainty, level 4, is the hypothetical maximum uncertainty.



#### 4.1. Environment Knowledge

1. *Complete and precise environment model:*

If the environment is precisely known (where Equation (1) captures all dynamics), an open loop controller could be used, thus requiring no degree of intelligence. In reality there are often aspects of the system that are not perfectly modelled or subject to uncertainties. This then requires a more sophisticated controller.

2. *Complete environment model subject to minor variations:*

Any real system can only be modelled to a certain degree of precision. At this level we consider environments which have bounded uncertainties that are small enough such that simple feedback controllers can be used with little or no need for adaptation. These controllers are not necessarily intelligent, since they only require low levels of adaptation for dealing with slight uncertainties and do not learn online. There are still some examples of intelligent controllers within this category.

3. *Environment subject to change during operation:*

At this level the environment has time-varying parameters that describe its behaviour (Equation (2)). Now a higher degree of intelligence is required, since substantial changes in the environment cannot always be predicted or may be too complex to model. At this level of uncertainty, some conventional adaptive control methods can still perform sufficiently as well as intelligent ones.

4. *Underlying physics of environment not well defined:*

Denoted here as an uncertain mapping from states and actions to future states as in Equation (3). This is an uncommon scenario for Earth applications, however it is a fundamental problem for many space applications, such as Mars entry vehicles. Here some information about the environment is known, but there are still substantial knowledge gaps requiring an intelligent controller.

5. *No knowledge of environment:*

Where no model exists for the environment and the control designer cannot incorporate any environmental knowledge into the controller, this requires an intelligent control system to safely explore its environment.

#### 4.2. Controller Knowledge

1. *Stationary, globally stable controller:*

Most feedback controllers have guarantees of stability and maintain a certain level of performance under given assumptions. In simple cases, these assumptions allow the control system to perform well with a fixed set of parameters without any need for adaptation (Equation (4)).

2. *Varying controller parameters:*

There are many examples of intelligent and non-intelligent applications that vary some control parameters online (Equation (5)). This accounts for a lack of knowledge in the controller parameters, where fixed parameters at design time are insufficient to cover the entire operating range of the system.

3. *Unknown sensor/actuator behaviour:*

This comes under the broad category of fault tolerant control, which itself has many dimensions. Here we consider fault tolerance to represent a level of uncertainty in the controller, where measurements may be erroneous and actions may not create the predicted effect (Equation (6)). Some fault tolerant systems use simple thresholds for indicating faults that are

specified at design time, but since these are known this does not fall under this category. Here we are instead referring to a control system that must deal with unknown faults.

4. *Varying controller configurations:*

At higher levels of intelligence, a controller can alter its own control structure online (Equation (7)). This is commonly done offline using techniques such as evolutionary computation to define the controller structure. An intelligent controller requires online adaptation and therefore an efficient means of adjusting its configuration while operating.

5. *No known controller structure:*

The controller itself designs the control system from scratch using, for example, mathematical operations, control blocks, and intelligent architectures. An intelligent controller must be able to do this online, but perhaps with a rudimentary initial controller to give a stable starting point.

#### 4.3. Goal Knowledge

1. *Goals entirely predetermined by designer:*

Most control systems, including intelligent ones, have a clearly defined goal that entirely shapes the control system design. In this case the control system is not 'aware' of its goals and is therefore unable to update its goals or improve its performance with respect to the current goals. An example of such systems are those where the tracking error between a reference state and the current state must be reduced to zero.

2. *Goal specified implicitly, for example, as a reward function:*

Many optimal control problems come under this category, since the aim of the controller is often to minimise or maximise a defined cost function when the means of optimising this function are not specified. The high level goal of the controller is then to derive a control policy which achieves optimal control with respect to this cost function. This is also the case where the controller is punished for detrimental actions and must find a control policy which avoids such actions. These examples fit well into the framework of reinforcement learning control, where an agent learns by interacting with the environment and observing its state and a reward.

3. *Specific goals subject to change during operation with a globally defined goal:*

In a dynamic environment, the definition of specific goals depends on contingent events and observations. Moreover, if the allocation of goals is performed on ground, such as in a space mission, the robot/spacecraft will have to wait for new instructions every time a new, unforeseen event occurs or a new set of scientific data is available. This requires an intelligent goal planner to elaborate new specific goals based on changes in the environment.

4. *One or several abstract goals with no clear cost function:*

There are cases where the goals cannot be easily defined mathematically and so the controller requires an understanding of high level goals. For example, a controller's goal might be "capture images of scientifically interesting events" or "explore this region and collect data". The controller must be able to decide what events are scientifically interesting or which data are worth collecting.

5. *No knowledge of goals:*

The controller has to deduce what actions to take when, to begin with, it has no knowledge or indication of what actions are favourable.

### 5. Classification of Relevant Examples

Using the taxonomy presented here, we now give some examples of intelligent controllers and their classification. The specific examples detailed here are used to illustrate the applicability of the

taxonomy to a range of methods with varying levels of intelligence. The following notation is used in the classification below:

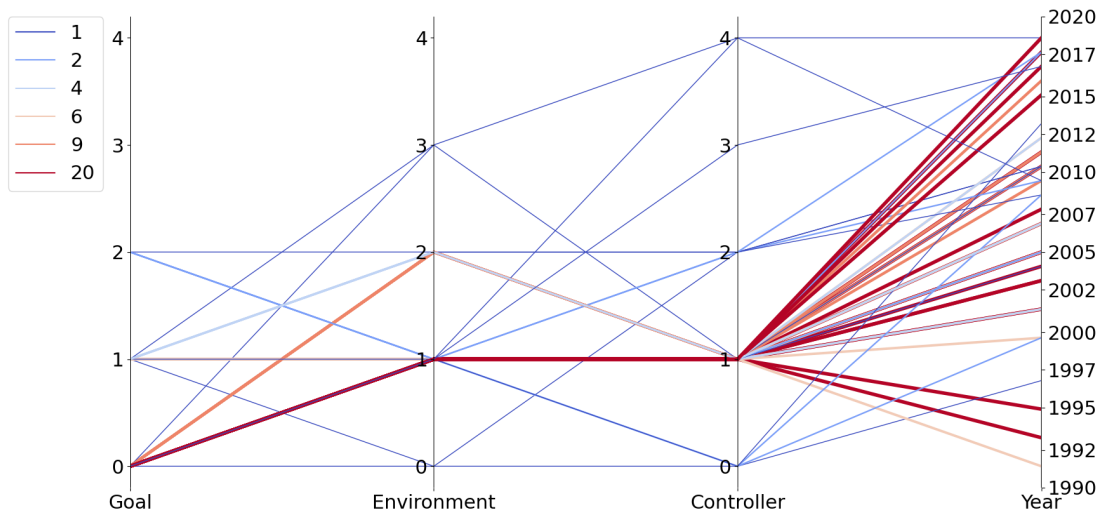
- G: Goal Knowledge
- E: Environmental Knowledge
- C: Controller Knowledge

Table 1 shows the classification of IC methods presented in numerous papers with references, also grouped by the AI technique employed. For clarity, only levels of classification observed in the reviewed applications are listed in the table. Figure 7 presents these classifications in a parallel coordinate plot. As with Table 1, the applications are divided into different levels of intelligence and the colours and line thicknesses indicate the number of applications found in the respective level of intelligence. We add an additional dimension of publication year to show the spread of dates over which these are published. This shows there are no examples with a goal classification greater than two. The most common classification is G-0, E-1, C-1 (20 applications) by a significant margin—the second most populated intelligence level is G-0, E-2, C-1 (9 applications). The years of publication of the reviewed works are well spread from 1990 to 2020.

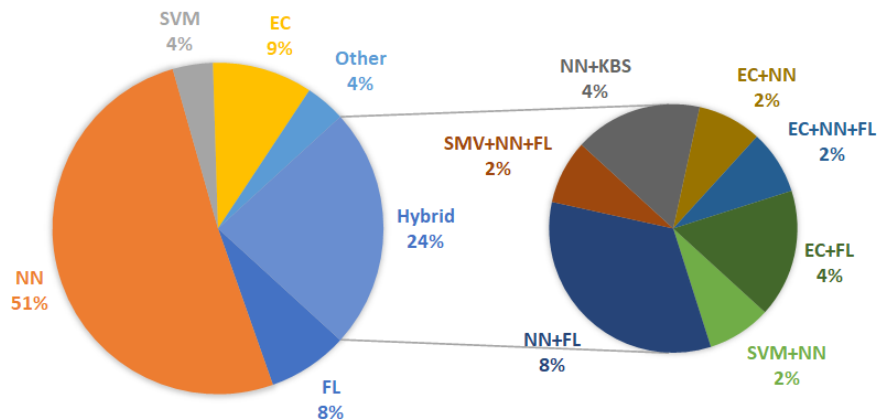
As discussed in Section 3.1, different AI techniques can be used for IC applications, but most of them come under Machine Learning (ML) (especially Neural Network (NN) and Support Vector Machine (SVM)), Fuzzy Logic (FL), Evolutionary Computing (EC) or hybrid methods. We quantify the distribution of these methods in Figure 8 from which we see that the majority of analysed applications make use of Neural Networks (NNs) (51%). These controllers also cover a broad range of intelligence levels. Evolutionary Computing (EC) and Fuzzy Logic (FL) are employed on a similar number of applications but with different levels of goal knowledge. In general, Evolutionary Computing (EC) is employed for symbolic regression—as in the case of Genetic Programming (GP)—or optimisation applications, especially coupled with other AI techniques, while Fuzzy Logic (FL) appears more often in higher intelligence applications (e.g., G-2). In particular, FL is often employed where human-like reasoning is desired. Hybrid methods are also widely used (24%) and among them the most common is the combination of Neural Network (NN) and FL.

**Table 1.** Artificial intelligence techniques used for intelligent control applications.

	G0							G1						G2		
	E0	E1			E2		E3	E0	E1		E2		E3	E1		E2
	C2	C1	C2	C3	C1	C2	C1	C0	C1	C4	C1	C2	C4	C0	C1	C1
FL		[14,15]									[16]				[17]	
NN		[18–30]		[31]	[32–36]	[37]	[38]		[39]		[40–42]	[43]				
SVM		[44,45]														
EC								[46]	[47,48]	[49]			[50]			
Other														[51]	[52]	
Hybrid Methods	[53]	[54,55]	[56]		[57–60]				[61,62]					[63]		[64]



**Figure 7.** Parallel coordinate plot of the observed levels of intelligence. The colour scale and the different line thickness refers to the number of applications observed in the considered intelligence level.



**Figure 8.** AI methods used for intelligent control (IC).

In the following we give a short description of an application for each intelligence level observed. These applications are from various engineering domains related to robotics, which are all systems that achieve their goals without human intervention.

- *G-0, E-0, C-2:*

In the work of Kankar et al. [53], Neural Network (NN) and Support Vector Machine (SVM) are compared on the task of predicting ball bearing failures. Both techniques prove to be useful for this application. While the presented system is not a complete controller itself, it is a fault detection system that can be integrated in a controller for a rotating machine.

- *G-0, E-1, C-1:*

Ichikawa and Sawa give an early example of Neural Networks (NNs) being used as direct controllers [55]. In their paper they combine a direct NN controller with genetic model reference adaptive control, which trains the NN based on a model of the ideal plant dynamics. This system is designed to deal with changing environment dynamics and continually updates its network to optimise performance.

- *G-0, E-1, C-2:*

A common technology for intelligent control and particularly Fault Detection, Isolation, and Recovery (FDIR) is Adaptive Network-based Fuzzy Inference System (ANFIS), which was developed by Jang [65]. An example of such an application of Adaptive Network-based Fuzzy Inference System (ANFIS) is presented by Wang et al. [66]. Their system comprises an adaptive backstepping sliding mode controller augmented with an Adaptive Network-based Fuzzy Inference System (ANFIS) Fault Detection, Isolation, and Recovery (FDIR) system that controls a robotic airship. The ANFIS observer predicts the environment state at each time step. If these values disagree with those from the sensors, then a sensor fault is declared and the ANFIS output is used as input to the controller. The level of Goal knowledge is 0 since the goal of the control system is to minimise a tracking error following a predetermined trajectory.
- *G-0, E-1, C-3:*

The Neural Network (NN) controller proposed by Wu et al. [31] has a unique feature which makes its classification C-3. The controller can change the network topology and its parameters online based on the output of a learning algorithm. Such a change in the topology requires a trade-off between maintaining sufficient computational speed for online usage and the required precision in its output values.
- *G-0, E-2, C-1:*

One of the most popular IC methods is the neuro-fuzzy controller, which combines the adaptability of a Neural Network (NN) with the human-like reasoning of fuzzy controllers [59]. In this example, the authors apply a neuro-fuzzy model reference adaptive control scheme to an electric drive system. They show the controller is robust to changes in the environment parameters and adapts quickly to suppress vibrations and improve tracking accuracy.
- *G-0, E-2, C-2:*

Another example of Fault Detection, Isolation, and Recovery (FDIR) incorporated into control systems is presented in [37]. Here a fault tolerant control scheme based on a backstepping controller integrated with a Neural Network (NN) is used to recognise unknown faults, with online adaptation of the NN weights. The overall system uses two networks to approximate unknown system faults and compensate for their effect. NN weights are updated online using a modified back-propagation algorithm.
- *G-0, E-3, C-1:*

Such an uncertain environment as a Mars entry vehicle benefits from having an intelligent control system [38]. In this paper the authors develop a Neural Network (NN) based sliding-mode variable structure controller. This controller has a fast loop, which is a conventional PID controller, and a slow loop, which contains the adaptive NN element. The goal is completely defined by the user through the definition of a nominal entry trajectory.
- *G-1, E-0, C-0:*

In this control system of modest intelligence—classified as G-1—a Genetic Algorithm (GA) is used to optimise the temperature for ethanol fermentation online [46]. This process is not an online adaptation of the controller parameters, but instead the optimal fermentation temperature is obtained online in a manner similar to optimal control approaches. What makes this system intelligent, in contrast to classical optimal control approaches, is that the optimisation is performed online according to the plant states.
- *G-1, E-1, C-1:*

As discussed in Section 3, it can be advantageous for IC methods to combine different AI techniques to exploit their benefits. Handelman et al. create such a system which comprises

a Knowledge Based System (KBS) for devising learning strategies and a Neural Network (NN) controller, which learns the desired actions and performs these consistently in real-time [61]. This is designed to mimic human learning, which combines rule based initial learning and fine tuning by repetitive learning. The environment and controller considered here have low levels of uncertainty, and the control goals are only implicitly defined.

- *G-1, E-1, C-4:*

Despite being a well known technique for symbolic regression applications, Genetic Programming (GP) is still not widely used in IC. An example of the use of Genetic Programming (GP) for control purposes is presented in [49], where it is used to derive a control law for a mobile robot moving in an environment with both known and unknown obstacles. The use of GP to create a control law gives this system a controller classification of C-4 since it derives the control law only using predefined mathematical functions without any prior knowledge of the controller structure. The environment has slight uncertainty from the unknown obstacles.

- *G-1, E-2, C-1:*

Kawana and Yasunobu present an intelligent controller capable of dealing with a failure in the actuators [16]. Since the introduced failure is known and defined by the user, this example is not classified as G-2, but does constitute a major change in the environment. The peculiarity of this controller is its ability to generate a model of the environment through online learning. This model is then used to update the fuzzy control rules.

- *G-1, E-2, C-2:*

Talebi et al. give another example of Fault Detection, Isolation, and Recovery (FDIR) enhanced with IC [43]. Here two recurrent Neural Networks (NNs) are employed to detect and isolate faults—one for sensor faults and another for actuator faults. These NNs also compensate for these faults directly without the need for an additional subsystem for fault isolation.

- *G-1, E-3, C-4:*

An approach similar to [49], which also uses Genetic Programming (GP), is proposed by Marchetti et al. [50]. Here GP generates a control law online and the controller is tested on different failure scenarios. In addition to these, they consider the case of an unknown environment model at design time.

- *G-2, E-1, C-0:*

In the work of Ceriotti et al. [51], they design a controller capable of modifying the goal of a planetary rover during its mission. The controller fuses navigation data with pieces of scientific data from different sources to yield a single value of “interest” for each point on the map. This map evolves during the mission depending on observed data. The fusion of data from different sensors is realised using the Dezert-Smarandache Theory (DSmT) of plausible and paradoxical reasoning, which can overcome the limitations of both fuzzy logic and evidence theory. In particular, paradoxical reasoning is able to provide a solution even in the case of conflicting information.

- *G-2, E-1, C-1:*

The Autonomous Sciencecraft Experiment onboard NASA’s Earth Observing One is one of the most advanced satellite IC systems [52]. As with many intelligent control systems, this system has a hierarchical structure. In this case the highest level in the control hierarchy is the CASPER planner, which uses information from the onboard science to plan its activities. This is fed to the spacecraft command language, which then carries out the plan using lower level actions. This level can also adapt to environmental changes and make control adjustments as necessary. Below this level is conventional software, which simply carries out control actions as instructed by higher levels. While this system does not operate in a substantially varying environment,

it alters its controller parameters online and contains highly autonomous decision making and goal updating.

- *G-2, E-2, C-1:*

WISDOM is a control system for rovers, which is capable of high level planning and adaptive control [64]. Again this control system has a hierarchical structure with three layers. The top layer is responsible for generating plans, which are fed to the adaptive controller at a lower level. This adaptive system deals with immediate changes in the environment and gives instructions to the lowest level in the hierarchy, which is connected directly to the actuators. This system adapts to changing or uncertain environments and has varying parameters. The goals are also evolved over time in the system's planner.

With the examples listed above we show the applicability of the taxonomy to a range of fields and applications involving IC. To demonstrate the consistency of the taxonomy, here we show how different applications using different AI techniques can be grouped in the same classification according to the task they perform. A few examples are listed below:

- *Activity planning—G2:* All reviewed applications involving planning and reasoning were classified as G2. Although they use a variety of AI techniques, in all these cases the controller needs to choose its desired states and how to achieve them. For example, in [52] the control system uses information from the onboard science subsystem to plan its activities. In [63] a reasoning strategy based on forward chaining is adopted to find optimal concentrations of chemicals for an electrolytic process.
- *Robotic navigation and manipulation—E2:* Robotic systems that operate in the presence of unknown obstacles or environment characteristics are classified as E2. Under this category fall control systems that deal with parametric uncertainties in the dynamic models of the plant as in [35,36] and robotic exploration in an uncertain environment [64].
- *Adaptive intelligent control—C1:* Where the system is not subject to unknown faults, all control systems that adapt their parameters online are classified as C1. This can be achieved with various AI techniques, but the common aspect within this category is some adaptation mechanism that updates the control law parameters. By comparing two different applications like [17] and [29], it can be seen that despite the different AI techniques they employ (in this case Fuzzy Logic (FL) and Neural Network (NN)) and their differing overall goal, they are both classified as C1 since they both adjust control law parameters online.
- *Fault Detection, Isolation, and Recovery (FDIR)—C2:* This encompasses control systems that can deal with failures in its sensors or actuators. An often used technique in this category is Adaptive Network-based Fuzzy Inference System (ANFIS) as in [56] for fault detection and diagnosis of an industrial steam turbine and in [66] where a controller is designed to reliably track the trajectory of a robotic airship in the presence of sensor faults.

## 6. Conclusions

It is clear that the use of intelligent control covers many classes of systems that require various levels of intelligence. Simpler environments can benefit from low levels of intelligence to tune their performance online, whereas more complex and uncertain environments require much higher levels of intelligence to operate effectively. This level of intelligence can be related to the extent to which the control system deals with uncertainties.

Here we have proposed a taxonomy of IC that classifies control systems based on their level of knowledge in three dimensions: goals, environment, and controller. From the applications studied, it seems most IC systems focus on the environment and controller dimensions, which tend to have higher levels of intelligence than the goal dimension. Some more recent references show development towards higher levels of intelligence in each dimension, however there is still work to be done developing the level of intelligence with respect to goal knowledge.

We expect future intelligent controllers will continue to find useful applications with lower levels of intelligence. However, as we require more autonomous machines, we will also see more applications at higher levels of intelligence. Future work will likely investigate how to create controllers able to deal with much higher level of uncertainty for goals, which is needed, for example, in developing controllers for machines that can explore new and unknown environments. The recent developments in AI and Explainable AI will also see a growth in applications in intelligent control, where human operators can have access to autonomous goal decisions and reasoning process explanations. This will be a fundamental step toward the building of trustworthy systems.

**Author Contributions:** Conceptualization, C.W., F.M., M.D.C., A.R. and E.M.; methodology, C.W., F.M., M.D.C., A.R. and E.M.; formal analysis, C.W., F.M., M.D.C.; data curation, C.W., F.M. and M.D.C.; writing—original draft preparation, C.W. and F.M.; writing—review and editing, M.D.C., A.R. and E.M.; visualization, C.W. and F.M.; supervision, A.R. and E.M.; project administration, A.R.; funding acquisition, A.R. and E.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the European Space Agency, (Contract Number: 4000124916/18/NL/CRS/hh).

**Acknowledgments:** The authors would like to thank Irene Huertas from the European Space Agency for insightful discussions on Intelligent Control over the duration of the research contract.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ANFIS	Adaptive Network-based Fuzzy Inference System
CASPER	Continuous Activity Scheduling Planning Execution and Replanning
CMAC	Cerebellar Model Articulation Controller
DSmT	Dezert-Smarandache Theory
EA	Evolutionary Algorithm
EC	Evolutionary Computing
FDIR	Fault Detection, Isolation, and Recovery
FL	Fuzzy Logic
GA	Genetic Algorithm
GGAC	General Genetic Adaptive Control
GMRAC	Genetic Model Reference Adaptive Control
GP	Genetic Programming
IC	Intelligent Control
KBS	Knowledge Based System
ML	Machine Learning
NN	Neural Network
SOC	Self-Organising Control
SVM	Support Vector Machine

## References

1. Saridis, G.N. Toward the Realization of Intelligent Controls. *Proc. IEEE* **1979**, *67*, 1115–1133. [[CrossRef](#)]
2. Fu, K. Learning control systems and intelligent control systems: An intersection of artificial intelligence and automatic control. *IEEE Trans. Autom. Control* **1971**, *16*, 70–72. [[CrossRef](#)]
3. Antsaklis, P.J. Intelligent Learning Control. *IEEE Control Syst.* **1995**, *15*, 5–7. [[CrossRef](#)]
4. Antsaklis, P.J. Defining Intelligent Control. Report to the Task Force on Intelligent Control. *IEEE Control Syst. Soc.* **1993**, *58*, 4–5.
5. Linkens, D. Learning systems in intelligent control: An appraisal of fuzzy, neural and genetic algorithm control applications. *Control Theory Appl.* **1996**, *143*, 367–386. [[CrossRef](#)]



6. Krishnakumar, K.; Kulkarni, N. Inverse Adaptive Neuro-Control for the control of a turbofan engine. In Proceedings of the AIAA conference on Guidance, Navigation and Control, Portland, OR, USA, 9–11 August 1999.
7. Lavalley, D.B.; Olsen, C.; Jacobsohn, J.; Reilly, J. Intelligent Control For Spacecraft Autonomy—An Industry Survey. In Proceedings of the AIAA Space Forum (Space 2006), San Jose, CA, USA, 19–21 September 2006.
8. Bennett, S. A Brief History of Automatic Control. *IEEE Control Syst.* **1996**, *16*, 17–25. [[CrossRef](#)]
9. Nyquist, H. Regeneration Theory. *Bell Syst. Tech. J.* **1932**, *11*, 126–147. [[CrossRef](#)]
10. Bellman, R. The Theory of Dynamic Programming. *Bull. Am. Math. Soc.* **1954**, *60*, 503–515. [[CrossRef](#)]
11. Hunt, K.J.; Sbarbaro, D.; Zbikowski, R.; Gawthrop, P.J. Neural networks for control systems—A survey. *Automatica* **1992**, *28*, 1083–1112. [[CrossRef](#)]
12. Fleming, P.; Purshouse, R. Evolutionary algorithms in control systems engineering: A survey. *Control Eng. Pract.* **2002**, *10*, 1223–1241. [[CrossRef](#)]
13. Passino, K.M.; Yurkovich, S. *Fuzzy Control*; Addison-Wesley: Menlo Park, CA, USA, 1998; Volume 42.
14. Guan, P.; Liu, X.J.; Liu, J.Z. Adaptive fuzzy sliding mode control for flexible satellite. *Eng. Appl. Artif. Intell.* **2005**, *18*, 451–459. [[CrossRef](#)]
15. Elkilany, B.G.; Abouelsoud, A.A.; Fathelbab, A.M.; Ishii, H. Potential field method parameters tuning using fuzzy inference system for adaptive formation control of multi-mobile robots. *Robotics* **2020**, *9*, 10. [[CrossRef](#)]
16. Kawana, E.; Yasunobu, S. An intelligent control system using object model by real-time learning. In Proceedings of the SICE Annual Conference, Takamatsu, Japan, 17–20 September 2007; pp. 2792–2797.
17. Yu, Z. Research on intelligent fuzzy control algorithm for moving path of handling robot. In Proceedings of the 2019 International Conference on Robots and Intelligent System, ICRIS 2019, Chengdu, China, 23–25 February 2019; pp. 50–54.
18. Gu, Y.; Zhao, W.; Wu, Z. Online adaptive least squares support vector machine and its application in utility boiler combustion optimization systems. *J. Process. Control* **2011**, *21*, 1040–1048. [[CrossRef](#)]
19. Lee, T.; Kim, Y. Nonlinear Adaptive Flight Control Using Backstepping and Neural Networks Controller. *J. Guid. Control Dyn.* **2001**, *24*, 675–682. [[CrossRef](#)]
20. Brinker, J.S.; Wise, K.A. Flight Testing of Reconfigurable Control Law on the X-36 Tailless Aircraft. *J. Guid. Control Dyn.* **2001**, *24*, 903–909. [[CrossRef](#)]
21. Johnson, E.N.; Kannan, S.K. Adaptive Trajectory Control for Autonomous Helicopters. *J. Guid. Control Dyn.* **2005**, *28*, 524–538. [[CrossRef](#)]
22. Williams-Hayes, P. *Flight Test Implementation of a Second Generation Intelligent Flight Control System*; Technical Report November 2005; NASA Dryden Flight Research Center: Edwards, CA, USA, 2005.
23. Krishnakumar, K. Adaptive Neuro-Control for Spacecraft Attitude Control. In Proceedings of the 1994 IEEE International Conference on Control and Applications, Glasgow, UK, 24–26 August 1994.
24. Sabahi, K.; Nekoui, M.A.; Teshnehlab, M.; Aliyari, M.; Mansouri, M. Load frequency control in interconnected power system using modified dynamic neural networks. In Proceedings of the 2007 Mediterranean Conference on Control and Automation, Athens, Greece, 27–29 June 2007.
25. Becerikli, Y.; Konar, A.F.; Samad, T. Intelligent optimal control with dynamic neural networks. *Neural Netw.* **2003**, *16*, 251–259. [[CrossRef](#)]
26. Kuljaca, O.; Swamy, N.; Lewis, F.L.; Kwan, C.M. Design and implementation of industrial neural network controller using backstepping. *IEEE Trans. Ind. Electron.* **2003**, *50*, 193–201. [[CrossRef](#)]
27. San, P.P.; Ren, B.; Ge, S.S.; Lee, T.H.; Liu, J.K. Adaptive neural network control of hard disk drives with hysteresis friction nonlinearity. *IEEE Trans. Control Syst. Technol.* **2011**, *19*, 351–358. [[CrossRef](#)]
28. Yen, V.T.; Nan, W.Y.; Van Cuong, P. Robust Adaptive Sliding Mode Neural Networks Control for Industrial Robot Manipulators. *Int. J. Control Autom. Syst.* **2019**, *17*, 783–792. [[CrossRef](#)]
29. Hamid, M.; Jamil, M.; Butt, S.I. Intelligent control of industrial robotic three degree of freedom crane using Artificial Neural Network. In Proceedings of the 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference (ITNEC 2016), Chongqing, China, 20–22 May 2016; pp. 113–117.
30. Ligutan, D.D.; Abad, A.C.; Dadios, E.P. Adaptive robotic arm control using artificial neural network. In Proceedings of the 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM 2018), Baguio City, Philippines, 29 November–2 December 2018.

31. Wu, Q.; Lin, C.M.; Fang, W.; Chao, F.; Yang, L.; Shang, C.; Zhou, C. Self-organizing brain emotional learning controller network for intelligent control system of mobile robots. *IEEE Access* **2018**, *6*, 59096–59108. [[CrossRef](#)]
32. Dai, S.L.; Wang, C.; Luo, F. Identification and learning control of ocean surface ship using neural networks. *IEEE Trans. Ind. Inform.* **2012**, *8*, 801–810. [[CrossRef](#)]
33. Nicol, C.; MacNab, C.J.; Ramirez-Serrano, A. Robust adaptive control of a quadrotor helicopter. *Mechatronics* **2011**, *21*, 927–938. [[CrossRef](#)]
34. How, B.V.E.; Ge, S.S.; Choo, Y.S. Dynamic load positioning for subsea installation via adaptive neural control. *IEEE J. Ocean. Eng.* **2010**, *35*, 366–375. [[CrossRef](#)]
35. He, W.; Chen, Y.; Yin, Z. Adaptive Neural Network Control of an Uncertain Robot with Full-State Constraints. *IEEE Trans. Cybern.* **2016**, *46*, 620–629. [[CrossRef](#)] [[PubMed](#)]
36. Klecker, S.; Hichri, B.; Plapper, P. Neuro-inspired reward-based tracking control for robotic manipulators with unknown dynamics. In Proceedings of the 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE 2017), Shanghai, China, 29–31 December 2018; pp. 21–25.
37. Xu, Y.; Jiang, B.; Tao, G.; Gao, Z. Fault tolerant control for a class of nonlinear systems with application to near space vehicle. *Circuits Syst. Signal Process.* **2011**, *30*, 655–672. [[CrossRef](#)]
38. Li, S.; Peng, Y.M. Neural network-based sliding mode variable structure control for Mars entry. *Proc. Inst. Mech. Eng. Part J. Aerosp. Eng.* **2011**, *226*, 1373–1386. [[CrossRef](#)]
39. Yang, C.; Li, Z.; Li, J. Trajectory planning and optimized adaptive control for a class of wheeled inverted pendulum vehicle models. *IEEE Trans. Cybern.* **2013**, *43*, 24–36. [[CrossRef](#)]
40. Johnson, E.; Calise, A.; Corban, J.E. Reusable launch vehicle adaptive guidance and control using neural networks. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, QC, Canada, 6–9 August 2001.
41. Qazi, M.U.D.; Linshu, H.; Elhajian, T. Rapid Trajectory Optimization Using Computational Intelligence for Guidance and Conceptual Design of Multistage Space Launch Vehicles. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, CA, USA, 15–18 August 2005; pp. 1–18.
42. Wen, C.M.; Cheng, M.Y. Development of a recurrent fuzzy CMAC with adjustable input space quantization and self-tuning learning rate for control of a dual-axis piezoelectric actuated micromotion stage. *IEEE Trans. Ind. Electron.* **2013**, *60*, 5105–5115. [[CrossRef](#)]
43. Talebi, H.A.; Khorasani, K.; Tafazoli, S. A recurrent neural-network-based sensor and actuator fault detection and isolation for nonlinear systems with application to the satellite’s attitude control subsystem. *IEEE Trans. Neural Netw.* **2009**, *20*, 45–60. [[CrossRef](#)]
44. Zhang, X.; Xu, D.; Liu, Y. Intelligent control for large-scale variable speed variable pitch wind turbines. *J. Control Theory Appl.* **2004**, *2*, 305–311. [[CrossRef](#)]
45. Wong, P.K.; Xu, Q.; Vong, C.M.; Wong, H.C. Rate-dependent hysteresis modeling and control of a piezostage using online support vector machine and relevance vector machine. *IEEE Trans. Ind. Electron.* **2012**, *59*, 1988–2001. [[CrossRef](#)]
46. Moriyama, H.; Shimizu, K. On-line optimisation of culture temperature for ethanol fermentation using a genetic algorithm. *J. Chem. Technol. Biotechnol.* **1996**, *66*, 217–222. [[CrossRef](#)]
47. Caponio, A.; Cascella, G.L.; Neri, F.; Salvatore, N.; Sumner, M. A fast adaptive memetic algorithm for online and offline control design of PMSM drives. *IEEE Trans. Syst. Man Cybern. Part Cybern.* **2007**, *37*, 28–41. [[CrossRef](#)] [[PubMed](#)]
48. Ponce, H.; Souza, P.V.C. Intelligent control navigation emerging on multiple mobile robots applying social wound treatment. In Proceedings of the 2019 IEEE 33rd International Parallel and Distributed Processing Symposium Workshops (IPDPSW 2019), Rio de Janeiro, Brazil, 20–24 May 2019; pp. 559–564.
49. Chiang, C.H. A genetic programming based rule generation approach for intelligent control systems. In Proceedings of the 2010 International Symposium on Computer, Communication, Control and Automation (3CA), Tainan, Taiwan, 5–7 May 2010; Volume 1, pp. 104–107.
50. Marchetti, F.; Minisci, E.; Riccardi, A. Towards Intelligent Control via Genetic Programming. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020.
51. Ceriotti, M.; Vasile, M.; Giardini, G.; Massari, M. An approach to model interest for planetary rover through Dezert-Smarandache Theory. *J. Aerosp. Comput. Inf. Commun.* **2009**, *6*, 92–108. [[CrossRef](#)]

52. Chien, S.; Sherwood, R.; Tran, D.; Cichy, B.; Rabideau, G.; Castano, R.; Davis, A.; Mandl, D.; Frye, S.; Trout, B.; et al. Using Autonomy Flight Software to Improve Science Return on Earth Observing One. *J. Aerosp. Comput. Inf. Commun.* **2005**, *2*, 196–216. [[CrossRef](#)]
53. Kankar, P.K.; Sharma, S.C.; Harsha, S.P. Fault diagnosis of ball bearings using machine learning methods. *Expert Syst. Appl.* **2011**, *38*, 1876–1886. [[CrossRef](#)]
54. Ahn, K.K.; Kha, N.B. Modeling and control of shape memory alloy actuators using Preisach model, genetic algorithm and fuzzy logic. *Mechatronics* **2008**, *18*, 141–152. [[CrossRef](#)]
55. Ichikawa, Y.; Sawa, T. Neural Network Application for Direct Feedback Controllers. *IEEE Trans. Neural Netw.* **1992**, *3*, 224–231. [[CrossRef](#)]
56. Salahshoor, K.; Kordestani, M.; Khoshro, M.S. Fault detection and diagnosis of an industrial steam turbine using fusion of SVM (support vector machine) and ANFIS (adaptive neuro-fuzzy inference system) classifiers. *Energy* **2010**, *35*, 5472–5482. [[CrossRef](#)]
57. Gabbar, H.A.; Sharaf, A.; Othman, A.M.; Eldessouky, A.S.; Abdelsalam, A.A. Intelligent control systems and applications on smart grids. In *New Approaches in Intelligent Control. Intelligent Systems Reference Library*; Nakamatsu, K., Kountchev, R., Eds.; Springer: Cham, Switzerland, 2016; Volume 107. [[CrossRef](#)]
58. Al-isawi, M.M.A.; Sasiadek, J.Z. Guidance and Control of Autonomous, Flexible Wing UAV with Advanced Vision System. In Proceedings of the 2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR), Międzyzdroje, Poland, 27–30 August 2018; pp. 441–448.
59. Orłowska-Kowalska, T.; Szabat, K. Control of the drive system with stiff and elastic couplings using adaptive neuro-fuzzy approach. *IEEE Trans. Ind. Electron.* **2007**, *54*, 228–240. [[CrossRef](#)]
60. Kaitwanidvilai, S.; Parnichkun, M. Force control in a pneumatic system using hybrid adaptive neuro-fuzzy model reference control. *Mechatronics* **2005**, *15*, 23–41. [[CrossRef](#)]
61. Handelman, D.A.; Lane, S.H.; Gelfand, J.J. Integrating Neural Networks and Knowledge-Based Systems for Intelligent Robotic Control. *IEEE Control Syst. Mag.* **1990**, *10*, 77–87. [[CrossRef](#)]
62. Lennon, W.K.; Passino, K.M. Intelligent control for brake systems. *IEEE Trans. Control Syst. Technol.* **1999**, *7*, 188–202. [[CrossRef](#)]
63. Wu, M.; Nakano, M.; She, J.H. An expert control strategy using neural networks for the electrolytic process in zinc hydrometallurgy. In Proceedings of the 1999 IEEE International Conference on Control Applications, Kohala Coast, HI, USA, 27–30 August 1999; Volume 16, pp. 135–143.
64. Vasile, M.; Massari, M.; Giardini, G. *Wisdom—An Advanced Intelligent, Fault-Tolerant System for Autonomy in Risky Environments*; Technical Report, ESA ITI Contract 18693/04/NL/MV; ESA ESTEC: Noordwijk, The Netherlands, 2004
65. Jang, J.S.R. ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685. [[CrossRef](#)]
66. Wang, Y.; Zhou, W.; Luo, J.; Yan, H.; Pu, H.; Peng, Y. Reliable Intelligent Path Following Control for a Robotic Airship Against Sensor Faults. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 2572–2582. [[CrossRef](#)]

