UNIVERSIDAD NACIONAL DE COLOMBIA

Design of an emotion-based controller for dynamical systems

José Danilo Rairán Antolines

Universidad Nacional de Colombia

Facultad de ingeniería, Departamento de Ingeniería de Sistemas e Industrial

Bogotá D.C., Colombia

2014

Design of an emotion-based controller for dynamical systems

José Danilo Rairán Antolines

This thesis has been presented because is a requirement to obtain the title:

Doctor en ingeniería - Sistemas y computación

Advisor:

Ph.D. Luis Fernando Niño Vásquez

Research Topic:

Intelligent Systems

Research Group:

Laboratorio de Investigación en Sistemas Inteligentes (LISI)

Universidad Nacional de Colombia

Facultad de ingeniería, Departamento de Ingeniería de Sistemas e Industrial

Bogotá D.C., Colombia

2014

Acknowledgments

The work during the seven years that were necessary to finish this dissertation has been continuous and exhausting. I hope that all the work done becomes evident by the quality of the results presented herein, which by the way were possible by the collaboration of many people.

First, I want to thank Dr. Luis Fernando Niño for selfless, timely, invaluable, and unceasing help, especially in the most critical moments, such as accepting me as student at the end of the first year, and making the initial contacts for my internship in Memphis TN. In addition, I am especially grateful for his advice about when and how to publish partial results, as well as guidance in the writing of this final document.

In addition, it is important to recognize and thank Dr. Alberto Delgado. His advice and conversations during the first year helped me define the main hypothesis of the PhD, that emulated emotions play an important role in the control of dynamical systems.

I also thank Dr. Max Garzón for the opportunity to work at his lab for two and a half years, resulting in the work documented in Chapter Two and Sections 3.2 and 3.3.

I also thank Steve Strain for the proofreading of this thesis. His dedication certainly improved the quality of this document.

Abstract

This thesis studies the effect of emulating human emotions in control systems. The introduction of a new control strategy applies well known facts from neuroscience and psychology about decision-making processes in humans, and combines them with computer science and artificial intelligence tools into control system design.

The approach to the concept of emotion specified in this thesis requires a dynamical system to be controlled, a second system is employed as a reference model, and a mechanism for predicting the future dynamics for both systems. The comparison between these predictions triggers an emotion in the controller, and each emotion plus its intensity serves to define the appropriate input to the plant. Thus, the controller avoids negative emotions such as anger or fear, while seeking for positive emotions such as calm or satisfaction. This scheme causes the plant to approach the dynamic of the reference model.

Given the importance of prediction in the emotion-based controller, this thesis proposed three algorithms to anticipate the value of a function. The first algorithm approximates the rate of change of any function given samples, the second reconstructs a periodic signal, and the third approximates the period of a periodic target function. In several tests, including linear and nonlinear systems, the proposed controller outperforms a classical controller for all considered cases. One of the main contributions of this research pertains to the reduction of the negative effects of nonlinearities in a plant by defining a linear reference model and leading the system to approach it.

The application of emulated emotions in control systems opens a wide range of new solutions for dynamical systems, which may include aspects such as fault tolerance, motion planning, and control of complex systems.

Keywords: anticipation, approximation algorithms, control systems, decision-making process, emotion-based control, emulated emotions, reference model.

Resumen

En esta tesis se estudia el efecto de la emulación de emociones humanas sobre el control de sistemas dinámicos. La aplicación de la estrategia de control nueva utiliza hechos bien conocidos con respecto a neurociencia y psicología acerca del proceso de toma de decisiones humanas, y combina esto con resultados en ciencias de la computación y de inteligencia artificial en el control de sistemas dinámicos.

El concepto de emociones especificado en esta tesis requiere un sistema dinámico a ser controlado y un segundo sistema utilizado como modelo de referencia, además requiere un mecanismo de predicción del futuro de la dinámica de los dos sistemas. La comparación entre las dos predicciones hace que se dispare una emoción en el controlador, y cada emoción más su propia intensidad sirve para definir el valor de entrada para la planta en cada instante. Así, el controlador evita emociones negativas, tales como ira y temor, mientras que busca experimentar emociones positivas, tales como calma o satisfacción. Este esquema logra que la planta se aproxime al modelo de referencia.

Dada la importancia de la predicción en el controlador basado en emociones, en esta tesis se proponen tres algoritmos para anticipar el valor de una función. El primer algoritmo aproxima la razón de cambio de una función dadas muestras, el segundo algoritmo reconstruye una señal periódica, y el tercero aproxima el periodo de una función periódica. Se vio, por medio de muchas pruebas, incluyendo sistemas lineales y no lineales, que el controlador propuesto supera el rendimiento de un controlador clásico, para todos los casos. Una de las contribuciones más importantes de esta investigación tiene que ver con la reducción del efecto negativo de las no linealidades en una planta, mediante la definición de modelo de referencia lineal.

La aplicación de emociones emuladas en el control de sistemas dinámicos abre posibilidades para soluciones nuevas en sistemas dinámicos, las cuales pueden incluir aspectos como la tolerancia a fallos, la planeación de movimiento, y el control de sistemas complejos.

Palabras clave: anticipación, algoritmos de aproximación, sistemas de control, toma de decisiones, control basado en emociones, emociones emuladas, modelo de referencia.

Contents

List of figures

List of tables

List of acronyms

ASHRAE	-	American Society of Heating, Refrigerating and
Air	-	Conditioning Engineers
BELBIC	-	Brain Emotional Learning Based Intelligent Controller
db	-	Decibels
DC	-	Direct Current
DFT	-	Discrete Fourier Transform
DSP	-	Digital Signal Processor
EDDIE	-	Emotion-Display with Dynamic Intuitive Expressions
FFT	-	Fast Fourier Transform
FIS	-	Fuzzy Inference System
FPGAs	-	Field-Programmable Gate Array
HVAC	-	Heating, Ventilation, and Air Conditioning
ITAE	-	Integral Time Absolute Error
MSE	-	Mean Squared Error
NN	-	Artificial Neural Network
P	-	Proportional
PCI	-	Peripheral Component Interconnect
PCs	-	Personal Computers
PI	-	Proportional-Integral
PID	-	Proportional-Integral-Derivative
PWM	-	Pulse-Wide Modulation
RM	-	Reference Model
RMSE	-	Root Mean Squared Error
S	-	Complex variable, Laplace transformation, Laplace plane
UAVs	-	Unmanned Aerial vehicles
µc	-	Microprocessor

Introduction

A classical control strategy aims to change the dynamic of a plant in such a way that the plant's output approaches a set point, regardless of the fluctuations in the environment surrounding the system. Thus, the controller compares the plant's output with the set point and compensates any difference, by means of a mathematical formula. Therefore, tuning a controller consists of defining the parameters (also known as gains) in this formula. Once the design stage ends, these gains remain constant, because the plant is assumed to be linear. This strong assumption allows the use of traditional mathematical tools to model systems and design controllers, such as Laplace and Z transforms, but real systems are hardly ever linear. Thus, the assumption of linearity may be the main disadvantage of this tuning process for practical applications. The nonlinear behavior of a plant may diminish the performance of the control system for dynamics out of the assumed linear zone, causing poor performance or even instability.

The negative effects of control system nonlinearities have been addressed by different control strategies, such as nonlinear control, robust control, adaptive control, and intelligent control [31], [72]. Each solution may assure stability in a wide region, dynamically change the gains of the controller, or emulate cognitive capabilities in humans. The solution proposed in this thesis lies in the latter category, given the demonstrated ability of humans to handle changes outside and inside themselves. This thesis studies the emulation of human emotions, as well as their effects in the decision-making process, akin to the emulation of the brain's learning capacity by neural networks. The definition of the new controller strategy in this thesis applies cognitive science to classical control systems. This effort brings well known facts from neuroscience and psychology about how humans make decisions, and combines them with tools from computer science and artificial intelligence.

Every emotion in the proposed control algorithm bases its definition on two pillars. On the one hand, a set point value that becomes a dynamical system, which is called *reference dynamic*. Thus, the control strategy leads the plant to approach another dynamical system instead of following a single value. This concept finds its inspiration in the relationship between two systems in nature, such as brain and body, husband and wife, planet and satellite, and many others. An application in this research uses a linear reference dynamic. As a result, the control system decreases the negative effects of nonlinearities in the plant. On the other hand, an emotion appears from *the comparison between two anticipated scenarios*: a predicted state of the plant in contrast with a predicted state of the reference dynamic. Matching these two scenarios (for desirable predicted scenarios) results in emotions such as calmness or satisfaction, while a disagreement may trigger anger, fear, or other negative emotions. These two pillars establish the base of a metaphor in which an agent (the controller) travels a landscape in order to reach the predicted location of the reference dynamic. Thus, the agent makes decisions seeking for experiencing positive emotions while avoiding negative emotions.

The use of anticipation as a bridge between cognitive concepts and classical control theory may be the most important result in this thesis. Anticipation allows the definition of emotional states for dynamical systems. The experimental results demonstrate the convenience of implementing emulated emotion in control systems to influence the decision-making process in a controller. Also important, the emotion-based controller allows the inclusion of any control algorithm, akin to the logical and emotional aspects of the rational process in humans. The emotion-based controller outperformed a classical controller for all the experiments in this thesis.

Since anticipation plays a central role in the proposed control strategy this thesis introduces three algorithms to predict future states of a function. The application of the first algorithm supposes that the system will always follow its current trend. Thus, the prediction can be reduced to the approximation of the rate of change and the definition of a time window. This algorithm restricts its application to data coming from uniform sampling. A second algorithm predicts future states of a periodical target function by reconstructing a single period of the signal. Consequently, the algorithm approximates the period value in an initial stage, and then reconstructs the signal in a second and final stage. The period approximation uses discrete data and computes the period by minimizing the mean squared error between a sinusoidal function and the target function. In the second stage, a perceptron neural network approximates the Fourier coefficients, which are represented by the weights of the network. The third algorithm focuses on the period approximation. It compares two sections of the target function to approximate the period, which reduces the amount of data required to make the approximation relative to the second algorithm.

This document is organized as follows. Chapter One presents general concepts of classical control and emphasizes the negative effect of nonlinearities. In addition, this chapter describes five of the most common solutions to reduce undesired nonlinear effects. In contrast to the classical control architecture, Chapter Two proposes a new control framework that eliminates the use of a set-point and stresses the relationship between two dynamical systems as a way to generate new controllers based on reference models. Once the new control framework has been established, Chapter Three presents the key concept of anticipation in the emulation of emotions. This concept was studied from a mathematical standpoint to solve two particular problems. The first corresponds to approximating the rate of change of a sampled variable, and second is reconstructing a periodic signal as a method to predict future behaviors. The new framework plus the concept of anticipation serves as a basis to define the proposed control strategy, in Chapter Four. This chapter also includes a review of some of the theories of emotions from psychology and neuroscience that justify the Circumplex Model of Affect as the selected model of emotions. Several applications of the proposed controller in dynamical systems validate the proposed architecture. The final chapter presents the conclusions in three sections. The first section presents general conclusions, the second focuses on results, and third lists future work.

The application of emulating emotions to enhance control systems goes beyond the problems solved in this thesis, which include systems modeled with fourth order differential equations of lower order. As a result, the metaphor of the coupling of two dynamical systems could be applied to problems different in nature, such as fault tolerance algorithms, human computer interfaces, or the control of biological systems, among others.

1. Control systems – classical point of view

This chapter establishes a foundation for a discussion of the control of dynamical systems, including the definition of control and how a control system works. Given the amount of data available in literature regarding classical control, this chapter focuses exclusively on the concepts most relevant to this thesis, such as identification processes, control design, and performance evaluation.

Classical Proportional-Integral-Derivative (PID) control is by far the most common control algorithm in industry. This type of controller has proven simple and applicable in virtually all kinds of industrial processes. However, nonlinearities in the plant affect the controller performance. Therefore, the end of this chapter presents five solutions to the negative effects of the nonlinearities. These effects are stressed in this chapter, since the final result of this thesis, an emotion-based control algorithm serves as a solution. The proposed controller causes the plant to approach a linear dynamic, resulting in a decrease of the negative effects of the nonlinearities.

1.1 Design and analysis of a classical controller

A PID controller design attempts to compute the value of three gains (*Kp*, *Ki*, and *Kd*) in order to alter the future state of a plant so as to minimize the error. The error $e(t)$ corresponds to the difference between sensor variable $b(t)$ and a reference signal $r(t)$, as shown in Fig. 7-1. In other words, the error represents the difference between the current state and a desired state.

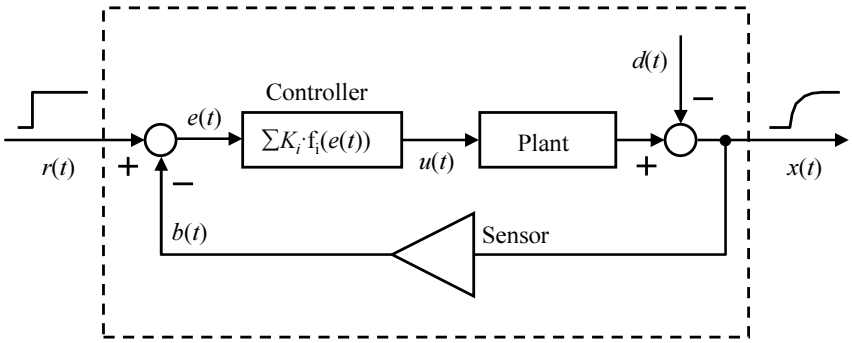


Fig. 7-1 Classical approach to control.

The gains *Kp*, *Ki*, and *Kd* in Eq. 1.1 indicate the influence of the contributions from the error's present, past, and future behavior, respectively. The integral component corresponds to the history, and the derivative represents the tendency, which can be associated with an anticipated value of error. The tuning process finds the values for the gains that match a set of response criteria for the plant, such as settling time, overshoot, and steady state error. As a result, and given that the plant is

assumed to be linear and invariant in time, the gains do not change their values once the tuning process finishes.

$$u(t) = Kp \cdot e(t) + Ki\int e(t)dt + Kd\,\frac{de(t)}{dt} \qquad (1.1)$$

It is therefore not surprising that in practice, problems reducible to linear equations elicit the attention of a control designer almost exclusively. Theoretically, only plants that permit such reduction are amenable to rigorous analyses, and therefore problems of significant practical or theoretical importance may escape consideration or lack adequate solutions. The problems become even more difficult when external conditions such as noise $d(t)$, environmental conditions that affect the plant, or the effect of human actors involved in the plant's management are taken into account, as happens increasingly in intelligent control.

The speed control of an electric motor illustrates the design process in this chapter. Specifically, the controller regulates the dynamic of a direct current (DC) motor. Controller designers use this type of motor because its model can be written as a second order linear differential equation. Thus, the simplicity of the motor dynamic makes it ideal to try control algorithms. The control algorithm can be implemented in Matlab, in a Digital Signal Processor (DSP), a Field-Programmable Gate Array (FPGA), or a microprocessor (μc), among other technologies, as shown in [74], [1], [25].

The experimental setup in Fig. 1-2 serves as an instantiation of the general layout in Fig. . This control system has two parts: software and hardware. The connection between these two parts may be a data acquisition card when the actuator (the power amplifier in this case) required voltage as input. The hardware part generates voltage through a sensor, whereas the output of the control algorithm feeds the actuator using voltage as well. In addition, a filter smoothes the feedback signal. Thus, this filter indirectly smoothes the control law and the behavior of the plant.

The tuning process for the controller generally has four steps, as follows: 1) sensor characterization 2) system identification; 3) control tuning, using any technique, such as root locus; and 4) performance evaluation. An explanation of these steps follows.

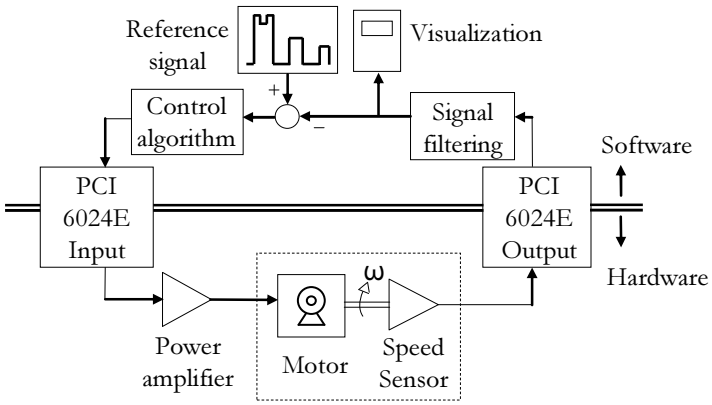


Fig. 1-2 Experimental setup for a classical control system.

Sensor characterization

Sensor characterization requires recording the corresponding outputs and inputs of a set of sensors. This set should uniformly cover the whole rank of the sensor, given that the relationship is linear. If the input data lie on the x-axis of a two dimensional plot, and the output data lie on the y-axis,

then characterizing the sensor is equivalent to finding the slope of the plot. Thus, the model of a sensor usually corresponds to a gain. The transformation of any physical variable by a sensor requires time; however, the characterization process does not take into account this delay. This approximation results from the plant's constant time exceeding the sensor delay by a factor of 10 or more; from this perspective, the sensor may be seen as making an instantaneous transformation of the plant's output. Finally, a common practice in control is to plug the sensor's output into a low-pass filter. This filter smoothes any signal with frequencies higher than a threshold called the cutoff frequency.

Plant identification

Identification corresponds to the process of modeling a plant from input-output data. In other words, an identification process finds a mathematical model that matches the experimental data. Therefore, the data set should represent as much of the dynamic as possible. For instance, the input may be a step function, changing randomly in magnitude and time length, as shown in Fig. 1-3. This figure presents the plant's input in part a), the plant's output in b), and the model's output in c). Classical control systems use a transfer function $H(s)$ as a model for the plant. One example of this type of model is defined in Eq. 1.2. The numerator of the model describes the gain of the system. Thus, a system with this $H(s)$ model increases 201 for every unitary increment in the input. The denominator corresponds to the constant times of the plant, and describes the transient behavior of the system. For example, the slower constant time for this example is 374 ms, and because a transient lasts approximately five constant times, the transient lasts 1.8 s. In general, a transfer function $H(s)$ indicates how a system transforms an input signal into its corresponding output.

$$H(s) = \frac{201}{(0.374s + 1)(0.072s + 1)} \tag{1.2}$$

Controller tuning

The tuning process sets the values for the gains of a PID controller or its combinations, such as Proportional (P) and Proportional-Integral (PI); first by simulation and then experimentally in the lab. In linear systems, the root locus method may be one of the most common methods for tuning controllers. This method attempts to meet several goals at the same time, for instance to reduce the stabilization time, to maintain the gain of the plant, and/or to minimize the overshoot in the response [50]. This multi-goal requirement is met by trial and error, locating poles and zeros of the system in a closed loop.

The root locus method guarantees that the poles of the whole system follow a certain deterministic path as result of changing the values for the gains of the controller. These changes imply also changes in the time response of the plant, due to the relation between the time behavior and pole placements. Thus, the method requires finding the set of gains that, for a particular motion of the poles, fulfills the temporal criteria for the response.

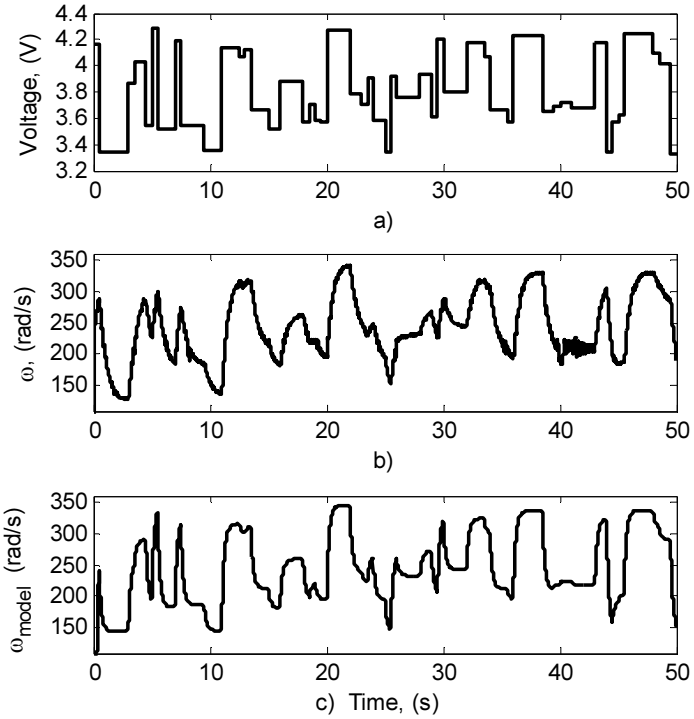


Fig. 1-3 Input-output data for the identification process.

The integral component of the classical controller eliminates the steady state error, whereas the derivative component reduces the settling time, as shown in [22]. The plot in Fig. 1-4 shows the pole positions, by ■ symbols, when a PI controller governs the dynamics of the system. The set of gains for this controller gives good performance during the simulation stage, but unfortunately, the experimentation in the lab may not be as good. Therefore, the control process usually retunes the gains at the end of the controller design.

Performance evaluation

Stability serves as a criterion for evaluating the performance of a controller. This evaluation requires observing the pole locations for a closed loop system. A stable system has every pole at the left side of the S plane (for instance the case shown in Fig. 1-4), whereas one or more poles at the right implies unstability. Stability guaranties that the output of the plant, at any bounded reference signal, will also be bounded.

Other criteria for the evaluation of the controller performance in addition to stability may be the measurement of bandwidth and the study of noise rejection. The bandwidth measurement defines the system's response to a pure sinusoidal input, when its frequency changes, as shown in Fig. 1-5, a). The example in this figure shows a cutoff frequency for the open loop system of 2 rad/s, whereas it reaches 10 rad/s in the closed loop system. This increase in the operating frequency effects as much as a fivefold enhancement in task performance over the open loop alternative.

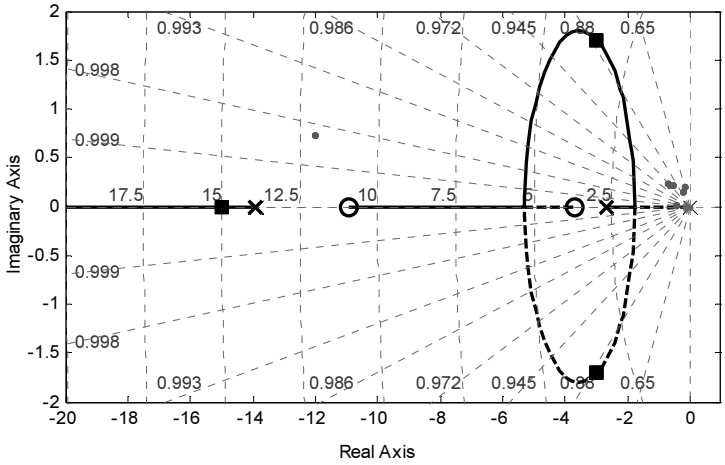


Fig. 1-4 Root locus for the PI controller in the S plane.

In general, the control tuning process does not include a noise rejection study. This study belongs to the final stage in the control design, the controller performance evaluation. The noise rejection study serves to evaluate the ability of the controller to compensate for the presence of an unknown and undesirable signal. For example, Fig. 1-5, b) shows noise in the sensor lasting 0.3 s, while its amplitude is at most 10% of the reference signal. As a result, the controller needs less than 0.2 s to return to the steady state behavior.

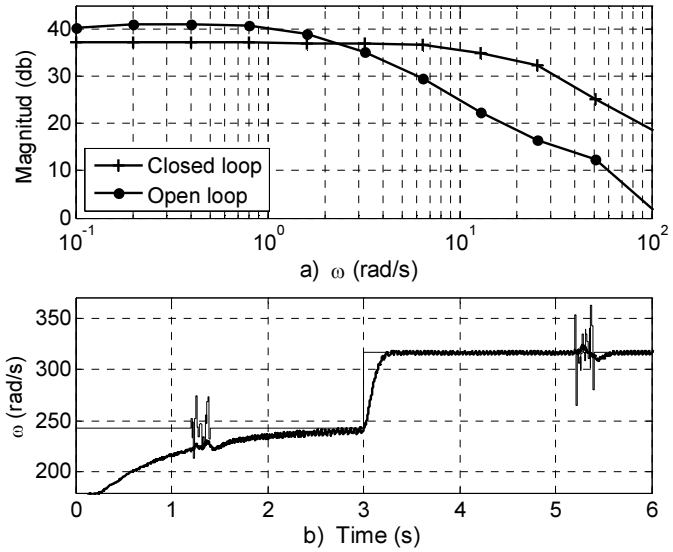


Fig. 1-5 Performance evaluation for a classical controller.

1.2 Negative effects of nonlinearities and their solutions

Nonlinearity corresponds to any relation between input and output on a system that does not fulfill the mathematical principles of superposition and proportionality [59]. There may be an infinite number of nonlinearities in dynamical systems. However, some of them, such as dead-zone, consistently appear in industrial applications. For example, the input voltage of an electronic transistor must pass a threshold value in order to output current. This effect can also be seen in many devices such as hydraulic cylinders or electrical motors. In the case of the motor, it only starts to spin when the input voltage exceeds a minimum voltage value different from zero.

The most common kind of nonlinearity on industrial applications may be the saturation. If a system has saturated, then even when the input changes, the output will remain still. For instance a hydraulic cylinder has a bounded range of motion, so the dynamic for the position of the cylinder has a saturation equivalent to the limit of motion. Even if the input indicates that the cylinder should continue its path, a physical limit prevents it. The classical electronic transistor serves again as an example of saturation. The output of these devices never exceeds a current value define by the relation between the voltage source and the load value regardless of the input voltage.

The third kind of nonlinearity is hysteresis. The behavior of a system with this nonlinearity depends on its past. As a result, the output value may differ regarding the increasing or decreasing nature of input and output. A magnetic coil serves as a classical example of the hysteresis effect, because the relation between the current and the magnetic force in a ferromagnetic coil expresses this nonlinearity. Thus, even for a null value of current through the coil, there may be a remanent magnetic field in the core. In addition, the value and sign of this remanent field depend on the last maximum current value.

Nonlinearities are not problems in themselves; the actual problem resides in the assumption made by control engineers concerning the linearity of the whole system. As a result of this strong assumption the gains in the controller remain fixed while the control system stays active. Therefore, the performance of the controller changes as a result of changes in the plant. In general, the performance decreases, because the tuning process attempts to locate the dynamic around a desirable state. Thus, any other dynamic will be worse, including the extreme case of unstabilty. The effect of the nonlinearities may be also impossible to overcome without changing the plant; such as in the case of saturation.

Classical PID control guarantees solutions for linear systems, but considering a plant to be linear is just an idealization of its behavior. Therefore, the first, and possibly simplest, strategy to reduce the negative effects of nonlinearities could be just to avoid them, or in other words, to work only in zones that appear to be linear. As a result there may be a guarantee of stability for the specified linear range. However, nothing can be said about the rest of the dynamic. One example of this solution may be the control of an electrical motor around the nominal voltage while working with nominal load. Unfortunately, the linear behavior cannot be extrapolated to zones outside the nominal conditions, such as transients, overloads, or failures in the system.

A second solution to avoid the negative effects of nonlinearities divides the entire dynamic of a system into sections that can be considered linear. As a result, every quasi-linear zone requires an individual tuning process. Thus, the system must have a method to identify the current working zone, and then appropriately set the predefined and fixed controller gains. Some dynamical systems may permit the use of an algorithm to fit data in such a way that the transitions between zones become smooth. Nevertheless, the application of this solution requires knowledge of the entire dynamic as well as an adequate division into quasi-linear zones, which may be a very hard problem.

A third solution bases its results on the heuristic that comes from tuning a classical controller instead of the dynamic linear zoning already explained. The heuristic process observes the relation between the value of each controller gain and a parameter of the control system. For instance, increasing a gain may consistently shrink the error in a system. Thus, an adaptive law will define a mathematical function to relate the error and the gain value. The number of relations between controller gains and system parameters comes from the experience in dealing with the control system. As a consequence, the results depend on the ability of the designer to come up with an appropriate set of relations. This may be the biggest pitfall of this approach.

The fourth solution, fuzzy logic control, entirely depends on the experience of a so called domain expert. Instead of formulating mathematical functions that associate controller gains and system parameters, as was done in the previous solution, this controller expresses the knowledge in a set of logical and linguistic rules. These rules typically define the relation between the error of the control system $e(t)$ and the actuating signal $u(t)$. In addition, a fuzzy logic controller can also use the relation between the rate of change and the integral of the error as inputs, thus resembling a nonlinear PID controller. Fuzzy logic control seems to be a natural way to control, because it expresses knowledge in a linguistic fashion as humans might do. Nevertheless, most of the times the system should be already working in closed loop in order to acquire experience. Moreover, in many cases, fuzzy control slightly surpasses the performance of the previous two solutions.

The fifth solution, the use of neurocontrollers, may be one of the most advanced methods to control dynamical systems, because these controllers can learn and control simultaneously. This capability makes them adaptable, and thereby useful for dealing with nonlinearities. On the contrary, the four previous solutions to deal with nonlinearities remain fixed once they have been tuned. In other words, they control only once the learning stage has finished. The learning process in a neurocontroller consists in minimizing the mean squared error, where the error corresponds to the difference between a reference and the actual output of the plant. One of the aspects that make the neuro-architecture adequate is the ability to dynamically change the rate of learning, as well as the possibility of learning both online and offline. However, this control strategy has some disadvantages. In general, there are not criteria to choose between architectures and learning algorithms in association with the application; there is neither a method to define the number of neurons per layer nor the number of layers, according to the problem; the definition of the learning rate is another pitfall, because this definition lies on the ability of an expert; finally, not every industrial process allows a controller to be learning while operating either for a short frame of time.

2. A new dynamical approach to control theory

The first chapter described classical control as well as some ways to deal with nonlinearities, because it was shown that nonlinearities are one of the hardest problems to be solved in the control of dynamical systems. Now, the second chapter follows the same path and shows that there must be a radical change on the classical control paradigm in order to endow systems with more powerful ways to cope with nonlinearities and also to solve more complex problems.

This chapter starts with a set of definitions regarding a new control framework, where the dynamic of the so called reference model enhances the behavior of a plant. These definitions are explained using a thermal room as an example. The traditional solution of the thermal room problem states that energy must be added to compensate for the possible deviation from the current temperature to a fixed ideal temperature. In contrast, the last section of this chapter shows how the solution of the thermal room problem could be improved when the reference includes the perception of comfort of a human being into the system. As a result, the reference model that replaces the traditional concept of set point corresponds to a dynamical system called Comfort Zone. This new control scheme allows savings of about 80% compare to a traditional control system.

2.1 Basic definitions and control framework

The critical issue in control systems is the recursive effect of control actuators on a plant and the feedback loop they create. The resolution of these problems is usually simplified by specifying a desired behavior in relatively simple terms (e.g., “Bring the temperature of the room to 22°C”), when the actual desirable reference behavior may be harder to describe, but perhaps easier or more efficient to effect. For example, “Bring the temperature of a room to 22°C, but do not bother to cool the other rooms where no one is present during a certain time interval”.

A second important issue in the control of dynamical systems is the problem of stability and robustness in the behavior of the plant under unknown or unexpected disturbances from the environment or the plant itself (e.g., the temperature is set regardless of the cycles in temperature due to variations in the weather, a change of season, etc.) A third issue is the identification problem, i.e., how to find an appropriate model of a plant to capture the control problem in accurate, objective and feasible ways. For instance, what will the weather be like tonight and tomorrow?

In order to address the problems already mentioned, the formulation of a control problem must allow for the developing of much more sophisticated conditions in the plants and the set points for the controls. The definitions below will hardly appear new, as they have been indirectly pointed out in both early and recent reviews (e.g., [48]), but the difference here will be that a) they will not be used as mere tools, but as the main actors in formulating and solving control problems; and b) they will affect the efficiency and effectiveness of the resulting solutions to greater extent.

Dynamical Systems

As mentioned above, dynamical systems have played an important role in control theory as tools for analysis and solution. Here it is proposed that they can play a much larger and significant role.

Definition. 1. An *open dynamical system* ***Xs*** consists of four components ***Xs*** = <T, ***X***, ***U***, $\boldsymbol{\Phi}_t$ in T>, where $\boldsymbol{\Phi}_t$ is a function that describes the evolution of the states of the system $x \in \boldsymbol{X}$, given an input $u \in \boldsymbol{U}$, over a period of time $t \in T$. The time coordinate is either discrete ($T = \boldsymbol{Z}$, natural numbers) or continuous ($T = \boldsymbol{R}+$, the positive real numbers), whereas $\boldsymbol{\Phi}$ is subject to the following constraints:

Consistency: $\boldsymbol{\Phi}(0,x_0,u) = x_0$, where x_0 is called the *initial condition*.

Determinism: $\boldsymbol{\Phi}(t_2,\boldsymbol{\Phi}(t_1,x_{0,}\, u_1),u_2) = \boldsymbol{\Phi}(t_1 + t_2,x_{0,}\, u_1+u_2)$, for t_1, t_2, $t_1 + t_2 \in T$.

Important qualitative properties of a dynamical system include the trajectories of arbitrary initial conditions, their fixed and stable points, the basins of attraction, and their ensemble as the corresponding phase space. (For more details about dynamical systems, see [11] [7].)

Plants

Definition 2. A *plant* ***Ys*** is an open deterministic dynamical system, usually characterized by its phase space, including any stable points (attractors), basins of attraction, cyclic or transient behaviors.

In general, just some indication of the qualitative behavior of a plant is known in advance. For instance, just the nature of trajectories, cycles, and long-term behavior for a system may be known.

Control Systems

The following concept represents a departure from the traditional definitions of control theory, but will allow for several enhancements, including a better description of set points as well as a more explicit and more robust handling of uncertain input and of the so-called noise and disturbances in traditional control.

Definition 3. A dynamical system ***Ys*** *approximates* a dynamical system ***Xs*** if there exists a *sensor* mapping s: $\boldsymbol{X} \rightarrow \boldsymbol{Y}$ that maps every whole basin of attraction of ***X*** into a single basin of attraction of ***Y***. Specifically, if $\{A_1, A_2, A_3, \ldots\}$ are the attractors of ***X*** and $\{B_1, B_2, B_3, \ldots\}$ are the basins of attraction of ***Y***, the following condition holds for each i, $s(A_i) \subseteq B_j$ for some j. In particular, every attractor in the dynamical system ***X*** is entirely mapped to a corresponding attractor in the dynamical system ***Y***. No requirement is implied for the transient states in ***X*** or for every state in B_j being assigned to a state in A_i, as is shown in Fig. 2-1. The approximation must reflect the behaviors of the given system by exhibiting qualitatively similar behaviors in their own.

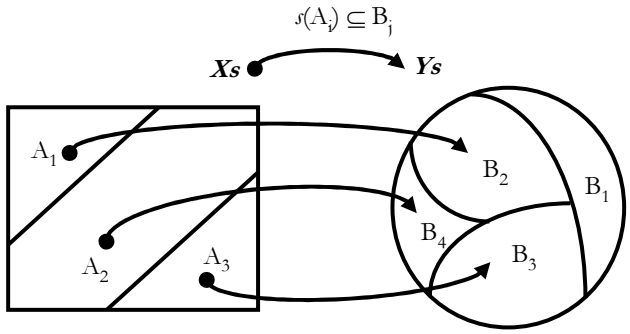


Fig. 2-1 The concept of dynamical system approximation.

Examples. It is well known that the multilayer perceptron and recurrent neural networks can approximate a variety of dynamical systems by the so-called universal approximation over a window of time [28].

Definition 4. A *control system* <***Xs***, ***Ys***, *s, c*> consists of a pair of dynamical systems ***Xs*** (the plant) and ***Ys*** (the reference behavior) and two mappings *s*: ***Xs*** → ***Ys*** and *c*: ***Ys*** → ***Xs*** so that the system ***Ys*** approximates the plant ***Xs*** via the map s, see Fig. 2-2. The system ***Ys*** is called the *reference dynamics* and the triple <*s*,***Ys***,*c*> is called the (dynamic) *controller*.

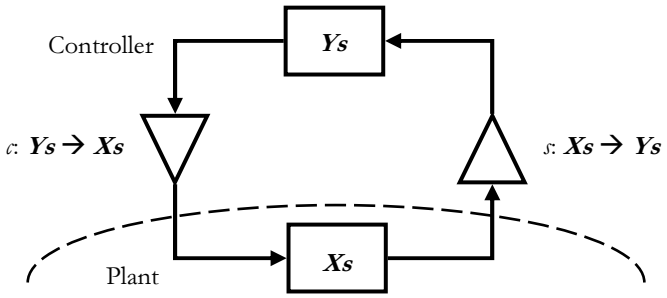


Fig. 2-2 New control framework for dynamical control system.

The plant is modeled as a dynamical system and the desired dynamic is given by a reference dynamical system synchronized with the plant. The dynamical system of the controller ***Ys*** specifies the desirable long-term behavior of the plant over time, regardless of the initial conditions, and independent of the exact trajectory of the plant. Note, however, that no constraints are placed on the transient behavior of the plant, which could even be mapped by *s* to a single attractor in ***Ys***. Examples of *s* mappings are thermometers and tachometers mapping whole basins of attraction to a single fixed global attractor. Examples of *c* mappings are mappings implemented by controls that drive actuators in a traditional control. Solving the control problem thus means finding maps *s* and *c* and their implementation, so that the plant is driven to a behavior qualitatively similar to ***Ys*** according to these mappings.

The new framework in this chapter, as shown in Fig. 2-2, could allows the study of control problems that are not even expressible using the traditional framework of control. For instance in dancing each dancer is a dynamical system, and they are joined by their senses in a stylistic motion. The coupling of these dynamical systems allows the overall system to express dynamics that are quite impossible to see if just one of the systems (dancers) forms the whole control system. In this example, each dancer is, at the same time, plant and reference dynamic, in such a way that both look for matching their dynamics in gracious ways to create a performance. It is clear that the traditional concept of reference in this example is changed by a complex dynamic.

As another example of the use of this new framework, consider an implementation of a conversational system, which follows from the area of Human Computer Interaction. In this problem an avatar assists people and exchanges information in such a way that people get what they want if possible. The person can be seen as a reference dynamic, and the avatar and all the computation behind it is the plant. The plant should match the dynamic of the reference dynamic, and this is done through the running of continuous iterations by a conversation using the control loop. The mappings are implemented through the senses of the person and a set of screen, speaker and microphone of the avatar.

One more example could be the study of the relation between brain and body, which in general is not a case of study in traditional control. Once again, there are two strongly related dynamical systems. Each can express a dynamic in isolation, but when they are combined they form an enriched coupled system. As in the previous example, the traditional concept of reference is extrapolated and enriched to a set of concepts such as increasing comfort, or improving changes of success.

The versatility of the new framework proposed in this section could be exemplified by the description of another application: the control of a biological system such as a whole forest. The reference dynamic could be a reforestation plan, which may include a program for an ideal landscape and also the definition of certain ideal biodiversity, among others. The plant system is the actual forest including trees, animals, source of water, the weather, etc. Reforestation plan and the plant influence each other while they simultaneously express their own dynamics, looking for a point of a correspondence between the two systems. The mappings s and c of the new framework could be extremely complex for the reforestation problem. For instance, s may include various sensor modalities including properties of the soil, satellite imagery, and in general any kind of geographic information system. On the other hand, the reference dynamic acts over the plant by planting, landscaping, or anything that a forestry policy could specify to move the forest toward its ideal state.

2.2 Definition of a control system using the new framework

The example in this section adequately defines the components of a control system using the new framework that was described in the previous section. The solution of the control system will be given in the next section. The plant of the control system is a thermally isolated room where the internal temperature depends only on the internal exchange of heat. This thermal room has four divisions and every division belongs to a person. Each person has a different ideal temperature for their sub-room. The sub-rooms have the same area equipped with equivalent heaters and coolers. Finally, there is at most one person inside the room at a time, else the room is empty.

Plant (Xs)

In order to understand the dynamic of the plant we can imagine that there are only two sub-rooms and also that the sub-rooms have the same size. Then all initial conditions will approach a thermal equilibrium at the mean value. In other words, every temperature is a fixed point attractor and their basins of attraction consist of all the pairs of temperatures with the same mean value. For instance initial conditions of 20° and 20° will remain at 20°, and 15° and 25° will eventually reach 20°, as will 10° and 30°, and so on. Thus, all the fixed points are placed in a line with unitary slope as shown in Fig. 2-3.

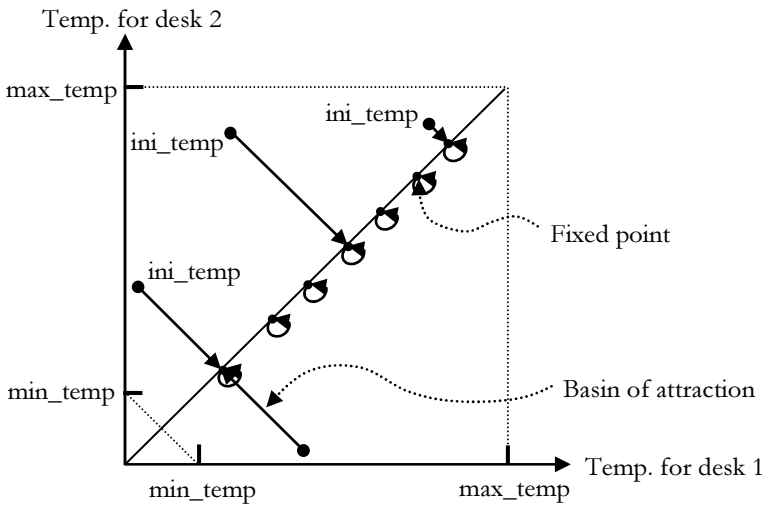


Fig. 2-3 Fixed points defining a unitary slope.

The case with four sub-rooms has the similar behavior but its dimension is four instead of two.

Reference behavior (*Ys*)

The reference behavior ***Ys*** is a hypercube with five axes, one for every sub-room temperature and another for the empty case. All initial conditions, for instance 1 and 2 in Fig. 2-4, must be attracted to state 3, where 3 is a fixed point defined by the ideal temperature of one of the sub-rooms or the temperature for the empty case. Then state 3 is a fixed point for ***Ys***, so the reference behavior must have five fixed points.

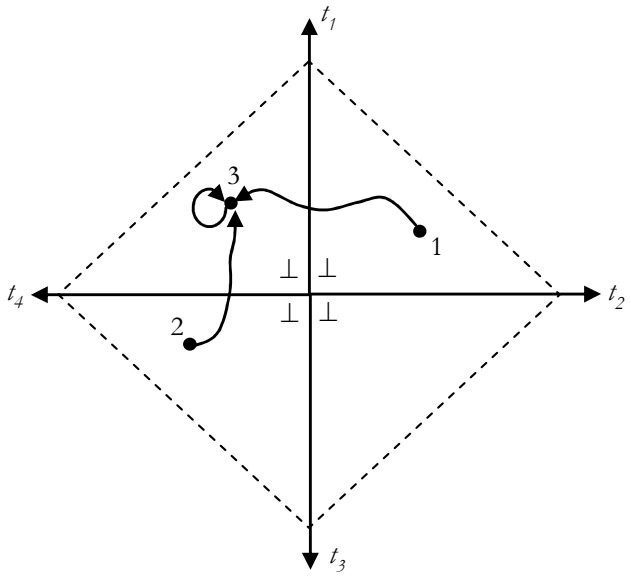


Fig. 2-4 Reference behavior for the thermal room.

State 1 in Fig. 2-4 represents a possible set of initial conditions, such as (18°,20°,22°,25°). Another possible initial condition vector in Fig. 2-4 could be (20°,25°,12°,17°), or any state with the format (t_1, t_2, t_3, t_4). If the ideal temperature for the sub-room under study is 20°, the state 3 should be (20°,20°,20°,20°) as shown in Fig. 2-5.

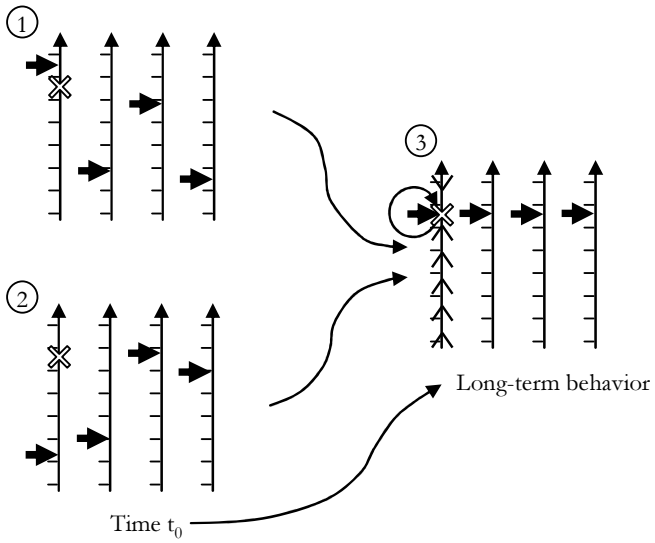


Fig. 2-5 Reference model and stable conditions.

***s* mapping**

Every sub-room has a temperature sensor that reflects its average temperature (t_i). All these four values are the output of the mapping. Each temperature sensor guaranties that all basins of attractions for sub-rooms are mapped in the reference behavior ***Ys***.

***c* mapping**

The *c* mapping computes the difference between current temperature t_i and each temperature T'_i for every sub-room, and based on these temperatures, turns on or off a heater h_i or cooler c_i, as is shown in Eq. 2.1. The sub-room temperature increases due to the hot air coming from a heater, and decreases as a result of the cold air from a cooler. Also, the cold air's minimum temperature is -10°C, and the hot air's maximum is 40°C. The temperature for the empty case is reached using all heaters or coolers at the same time.

$$c_i = \begin{cases} h_i = on \quad if \quad \left(t_i - t_i'\right) < 0 \\ c_i = on \quad if \quad \left(t_i - t_i'\right) > 0 \\ h_i, c_i = off \quad otherwise \end{cases} \tag{2.1}$$

All the components of the new framework for a control system were defined in this section: ***Xs***, ***Ys***, *s*, and *c*. Plant ***Xs*** is the same dynamical system used in traditional control, although the next section will show how more complex dynamics can be considered when using the new framework. The reference dynamic ***Ys*** is another dynamical system, which takes at least four attractors into consideration. Finally, the two mappings *s* and *c* are useful to close the control loop by means of the relation created between the outputs of one system and the inputs of the other system.

2.3 Design of a control system using the new framework

In this section, it is shown how the proposed framework can be used for modeling and resolving control problems in more effective ways. The HVAC (Heating, Ventilation, and Air Conditioning) problem is a well known problem of providing adequate comfort to humans in a variety of habitable spaces such as housing units, and even outer space stations. This is a problem of growing global importance because of global warming, as well as more demanding considerations in terms of energy minimization, the internal environmental conditions at higher latitudes, and the increasing design pressures for high levels of comfort. The traditional thermostat control pumps hotter or colder air into the plant based on readings from temperature sensors, independently of the environmental conditions or the specific characteristics of the inhabitants, other than a desirable temperature setting that is applied uniformly across the plant. Given the relative low efficiency of such controllers, as will be seen below, this example illustrates advantages of the solutions obtained by using a methodology based on the new framework already introduced.

Three control problems for thermal room plants are used to evaluate and contrast the traditional and the dynamic approaches. *Problem A* is the traditional HVAC problem of maintaining the temperatures of a thermal room at a desired value, in principle regardless of the outside conditions. The plant dynamics is here an average household in the US with an average heated area of 229 m^2 (2,464 sq. ft.) corresponding to 801 m^3 of air volume for an average height of 3.6 m (12 ft), on average [29]. The reference dynamics is a dynamical system that sets desirable temperatures for a basic weekly cycle, for example, as specified by a programmable thermostat, namely a set temperature for daylight hours, a different temperature for night hours, and perhaps a different schedule for weekends during a weekly cycle in the course of a year. The solution A to this control problem A is implemented by the traditional thermostat, as described above.

Using the dynamical approach described above, the plant for *problem B* can be considered the dynamical system of a room embedded in an external environment which is a physical dynamical system (the local weather). The trajectories of this system exhibit local cycles (day and night), as well as longer term seasonal variations, to mention only a few. The time scale will be constrained to one year for the purpose of the evaluation below. The reference dynamics is a dynamical system that sets desirable temperatures for the same basic weekly cycle as in problem A, but with the difference that only a sub-volume of the house will be maintained on that temperature schedule, say for example, about 200 m^3 (7,070 cu. ft.) This area will be moved around the house (from kitchen/den to dining room, to bedroom, following the human around the house), while the remaining areas are left uncontrolled.

In *problem C*, a more sophisticated plant includes a human as part of the plant dynamics, now (weather + human). It then becomes clear that the reference dynamics should be far more complex because what is really important for the human is not the temperature, but rather their degree of comfort. Therefore, a better reference dynamics now could involve not only the temperature around the inhabitants, but also the (absolute) humidity and air speed, three of the most important factors impacting physical human comfort [5]. The temperature settings about a certain volume (200 m^3) and about the human follow a dynamics similar to that in Problem B. Absolute humidity is measured in grams of water contained in a kilogram of air, i.e., g/kg and can be changed using electrical appliances such as humidifiers or dehumidifiers. Air speed is changed using fans. Therefore, in order to specify the reference dynamics, it is necessary to identify the appropriate range and combination of these three variables to provide a) acceptable (dis)comfort and b) ideal comfort for the average human.

The ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning Engineers) [73] has determined such conditions for (dis)comfort. These variables are physically independent. For example, in the scale of absolute humidity, three values are important [66]. The first is the maximum and is the equivalent of 12 grams of water per kilogram of air; above that value the moisture in the air makes the skin transpire on the average human (a PMV-Predicted Mean Value of -0.5, corresponding to the prediction of a 10% vote of dissatisfaction). The other two boundaries in terms of moisture are given in terms relative to this value; values between 20% and 60% indicate the percentage of moisture in the air compared to what may be before condensation. In terms of air speed, air moving at 1 m/s is perceived as a reduction of 3°C in temperature [19]. The acceptable comfort zone can thus be represented by a polyhedron shown in Fig. 2-6, while the inscribed polyhedron shows what could be likewise considered the ideal comfort zone [73].

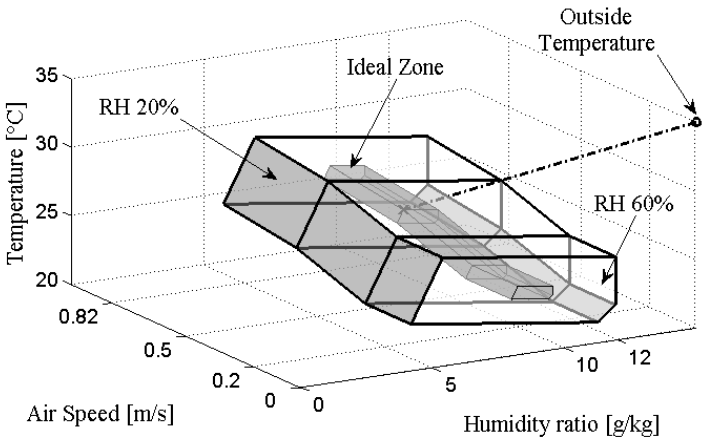


Fig. 2-6 Polyhedra for comfort zone and ideal comfort zone.

The reference dynamics is described next. A random given ambient condition x as follows *(Temp,Hum,AirS)* inside the comfort zone is a stable fixed point and needs no change. Outside the comfort zone, a state x should follow a trajectory that will take the plant to the closest point in the comfort zone region by changing the temperature, humidity and air speed proportionally to the difference along the three variables. The dynamic control used in the evaluation below is described in Table 2-1.

Table 2-1 Controller dynamics in the solution to problem C.

1.	Read the environmental conditions *(Temp, Hum, AirS)* from the sensors. If they are within the ideal comfort zone, turn the appropriate actuator off until the reading is out of the ideal comfort zone. Else go to step 2.
2.	Find the closest point on the ideal comfort zone's polyhedron.
3.	Add energy to move the variables *(Temp, Hum, AirS)* closer along every axis, for a Δt = 60 s.
4.	Go back to step 1

2.3.1 Dynamical solutions for problem A

In order to evaluate the traditional solutions, the energy expenditure in actuating the control was selected as a measure of quality. In order to approximate this energy, nine representative cities in

the USA were selected as a sample to estimate such energy with their average seasonal temperatures, as indicated in Fig. 2-7 [76].

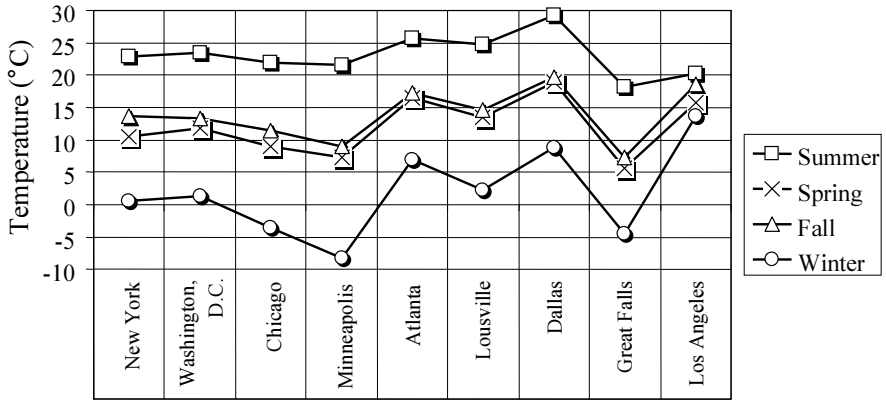


Fig. 2-7 Average seasonal temperatures in nine US cities.

The energies were computed by a simulation on Matlab of a 3D thermal room modeled as a cube and surrounded by walls insulated from an environment kept at a typical constant ambient temperature and humidity for each city in each season. The heat losses through the walls and roof were computed based on a combined thermal conductivity of 0.025 W/m K, typical house walls, doors and windows (made of wood, insulating fiber, and glass) and the difference between outside and inside temperatures. The energy to heat the air was calculated as the amount of heat loss/gain (Winter/Summer) through the walls, doors, windows and ceiling/roof. The dynamics of the plant was modeled by heat conduction and convection using the heat equation. The equation has four terms,

$$\rho C \frac{\partial T(t)}{\partial t} - \nabla\left(k \nabla T(t)\right) = Q + h(Text(t) - T(t)) \qquad (2.2)$$

The first term is a function of the material heat capacity C (J/kg), and the density ρ (kg/m^3). According to this component, when some amount of heat energy Q is applied to the material, the temperature $T(t)$ follows a transient. For linear ρ and C this yields a temperature increasing or decreasing according to an exponential decay characteristics of systems of order one [41]. The second term in Eq. 2.2 describes how the temperature is distributed in space, forming a gradient, depending on the conductivity of the material k (W/mK). This term corresponds to the distribution of temperature by conduction. The third term is the energy required for the material Q to change its temperature. The last term represents the convection heat distribution, which depends on the heat transfer coefficient h in W/m^{2}K. $Text(t)$ is the temperature outside the material. It is common to express the energy in watts, instead of Joules, so that the first term is multiplied by the volume of the material, the second by the area of heat transference between materials, and the last term by the area where convection occurs.

The classical use of the heat equation in engineering considers the change in temperature by adding or removing heat energy as needed. A more detailed knowledge, as required by the dynamic approach to control, requires understanding that the addition of electrical power is not the only way to change the temperature distribution. The heat equation has two terms by which changes can be made. The conduction term is the major component of heat losses in the walls, so that instead of adding power, insulation materials could be used with lower conduction coefficients. On the other hand, it is possible to use forced convection, through the air movement to unify the temperature.

The results of the simulation for solution A are shown in Table 2-2 for the nine cities.

Table 2-2 Evaluation of efficiency for problem A.

		Cooling [kWh]	Heating [kWh]	Total [kWh]
New England	New York	2,827	32,939	35,766
Middle Atlantic	Washington, D.C.	2,989	31,431	34,420
East North Central	Chicago	2,402	40,170	42,572
West North Central	Minneapolis	2,268	47,758	50,026
South Atlantic	Atlanta	4,644	19,276	23,920
East South Central	Louisville	3,578	27,496	31,074
West South Central	Dallas	9,421	15,509	24,930
West Mountain	Great Falls	870	48,808	49,678
West Pacific	Los Angeles	2,786	15,753	18,539
	Totals	31,785	279,140	310,925

Evaluation of thermal room efficiency of the traditional control solution to problem A as yearly annual household energy expenditures in nine US cities is described in Fig. 2-7. The figures were obtained by simulation on Matlab using a model based on the heat equation and typical statistical data about real homes in each city.

In order to validate these results, statistics were obtained for the energy expenditures of an average household in each of the nine cities [77], as shown in Table 2-3. They show a very good agreement with Table 2-2, within 1% error.

Table 2-3 Validation of the simulation for the control solution A.

		Cooling [kWh]	Heating [kWh]	Total [kWh]
New England	New York	740	43,380	44,120
Middle Atlantic	Washington, D.C.	1,506	41,439	42,945
East North Central	Chicago	1,663	42,090	43,753
West North Central	Minneapolis	2,130	36,606	38,736
South Atlantic	Atlanta	3,909	26,545	30,454
East South Central	Louisville	3,708	28,673	32,381
West South Central	Dallas	5,162	19,513	24,675
West Mountain	Great Falls	4,144	29,727	33,871
West Pacific	Los Angeles	1,242	18,953	20,195
	Totals	24,204	286,926	311,130

Validation of the simulation for the control solution A as an estimate of yearly annual household energy expenditures in nine US cities described in Fig. 2-7. The figures were obtained by extrapolation of statistical averages of weather records [76].

2.3.2 Dynamical solutions for problems B and C

A control for problem B is implemented by the traditional means, although it will require more sophisticated electronics, as described above. Analogous simulations were done for the previous

solution and the results are shown in Table 2-4. A control for problem C can likewise be implemented by the traditional means with extant technology, although it will require even more sophisticated electronics. Analogous simulations were done for the previous solution and the results are shown in Table 2-5. The air is heated or cooled proportionally to its volume, density, and heat capacity. For the discussion of solution C below, the humidifier and dehumidifier were assumed to vary linearly and proportional to ΔHum. It was estimated that it takes about 30 minutes and 500 W on average to add/remove humidity from this plant. Likewise, the fan is assumed to be on only when necessary. The estimate is that the fan takes less than 5 minutes and a nominal power consumption of 740 W to fix the necessary air speed in the plant when the control is on. The dynamic control will go off until the sensors indicate the plant has left the ideal comfort zone.

Table 2-4 Evaluation of efficiency for problem B.

		Cooling [kWh]	Heating [kWh]	Total [kWh]
New England	New York	716	8,336	9,052
Middle Atlantic	Washington, D.C.	757	7,955	8,712
East North Central	Chicago	608	10,166	10,774
West North Central	Minneapolis	574	12,085	12,659
South Atlantic	Atlanta	1,176	4,879	6,055
East South Central	Louisville	906	6,959	7,865
West South Central	Dallas	2,386	3,926	6,312
West Mountain	Great Falls	220	12,351	12,571
West Pacific	Los Angeles	706	3,988	4,694
	Totals	8,049	70,645	78,694

The evaluation of thermal room efficiency for the traditional control solution to problem B is shown in Table 2-4, as yearly annual household energy expenditures in nine US cities.

Table 2-5 Evaluation of efficiency for problem C.

		Cooling [kWh]	Heating [kWh]	Total [kWh]
New England	New York	680	5,931	6,611
Middle Atlantic	Washington, D.C.	701	5,661	6,362
East North Central	Chicago	854	7,065	7,919
West North Central	Minneapolis	1,022	8,326	9,348
South Atlantic	Atlanta	1,282	2,982	4,264
East South Central	Louisville	1,794	3,956	5,750
West South Central	Dallas	3,594	2,548	6,142
West Mountain	Great Falls	0	9,471	9,471
West Pacific	Los Angeles	1	3,331	3,332
	Totals	9,928	49,271	59,199

The evaluation of thermal room efficiency for the traditional control solution to problem C is shown in Table 2-5, as yearly annual household energy expenditures in nine US cities. The three problems A-C address increasingly complex scenarios. Their solutions can be compared on the basis of their energy utilization, as calculated in Fig. 2-8.

The estimate of energy expenditure by the simulation for the solution to problem B is comparably with that of four of the cities, and is statistically close to the average for the entire country. Therefore the estimates provided by the simulations are reasonably accurate within a small margin of error (under 1%). The fact that in some cities like Dallas there would be little difference between solution B and C while the difference would be large in cities such as Atlanta and Minneapolis/StPaul shows the importance of that humidity as a factor to be controlled. Earlier work has already pointed to a similar conclusion through different means in [47], and even played a role in the invention of the air conditioner (originally designed just to reduce relative humidity to about 55%). Using this standard, solution B would afford savings of about 74% on the average for the entire country compared to solution A, whereas solution C would afford an average savings of about 25% energy compared to the current solution B, or about 80% compared to solution A, for the entire country. The dynamics solution thus appears very competitive, in view of current problems faced by the green house solution to the problem [4], [18].

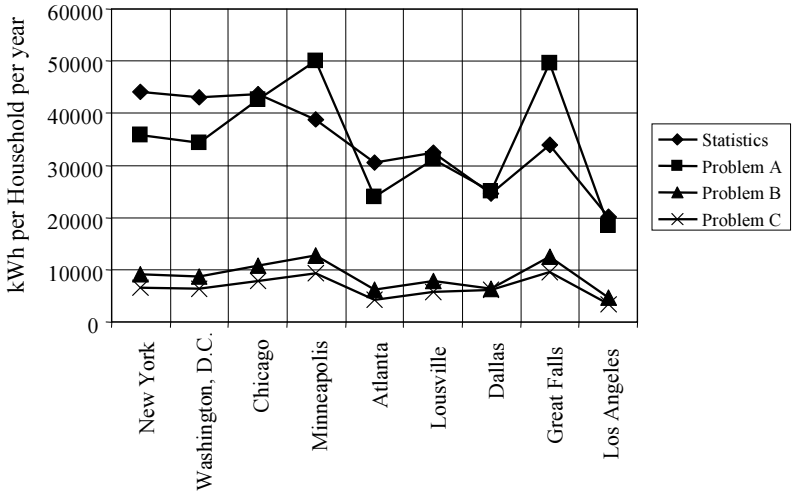


Fig. 2-8 Comparison of energy efficiency for solutions A, B, and C.

On the average, solution B provides an energy savings of about 74% over solution A, and solution C further offers 25% improvement over solution B, for an overall improvement of 80% over solution A. A comparison of the simulation results in solution A with statistics of energy consumption in the nine cities shows agreement within 1% error.

The implementation of this type of dynamic control for HVAC would therefore be desirable as an integrated system. The simulations show that the shortest trajectory (dashed line in Fig. 2-6) to the comfort zone in solution C is not a straight line. There does not seem to be enough knowledge of this dynamics to find the optimal trajectory, which would provide a more efficient solution and thus appears to be an important open problem for an optimal controller.

Final remarks

A traditional controller affects the dynamic of a plant in such a way that the output of the plant follows a reference value, whereas this reference comes from outside the control system. The concept of reference in the proposed new framework evolves to turn this simple reference into an entire dynamical system now part of the control system. As a result, the reference dynamic influences the plant, while the plant influences the reference. The new reference should identify the plant. Thus, both systems have similar nature and their relations may produce useful dynamics.

The definition of the new control framework may permit control applications that are unthinkable nowadays. For instance, one of such applications may be the use of the framework to design a conversational system, so this problem, which is a typical application in Human Computer Interfaces, could be studied using the point of view of dynamical systems. Another application could be the study of the close relationship between dancers during a performance as a control problem, which would allow coming up with new ideas about human relationships, motion planning, and also would open the door to a set interesting new research questions about dynamical systems. Finally, another example could be the control of a biological system, which is a significantly complex problem regardless of the size of the system. On the other hand, the definition of the reference dynamic in section three proposes a measure of comfort as the goal of a control system. This concept can be easily generalized to any kind of dynamical system, and could be used to replace the traditional concept of reference (also called set point), so instead of looking to follow a single value, as in traditional control, the coupling of the systems could seek to reach or increase its level of comfort.

3. Anticipation – a key concept to emulate emotions

Previous chapter introduced a new control framework that deals with more complex dynamics than those of a traditional controller. In principle, control should aim to join two dynamical systems: one labeled a reference and the other a plant. This idea has undergone successful testing using the well known thermal room problem. After establishing this strong base, this chapter will develop the key concept of anticipation. The next chapter expands on the importance of anticipation, which, in brief, arises as the key mechanism for generating an error signal via an emotional state, namely the difference between an anticipated future and what actually happens. A situation anticipated as positive, that matches reality, will trigger a positive emotion. On the contrary, a realized negative expectation will trigger a negative emotion. That triggered emotion could serve to update the correlation between the reference and the plant. If this correlation meets expectations, the system has not reason to change the relation between reference and plant. However, if nothing happens as expected, a strong indication exists for such a change.

The power of anticipating a state of a dynamical system will be explored by solving two problems. First, the traditional method to anticipate a signal requires the computation of the rate of change, but if the data comes from samples of the target function, the best result is just an approximation. Thus, the first section of this chapter proposes an algorithm to improve the quality of the approximation. Second, the reconstruction of a periodic signal involves computing the period, and this computation cannot be done if the ratio between period and sampling rate does not match a rational number. Therefore, this chapter proposes two algorithms to provide approximations of the period.

The first section in this chapter serves to present a new algorithm that uses the signals coming from an encoder to approximate speed. This algorithm eliminates the oscillations on the approximations that appear when using traditional algorithms, including the case of constant speed; instead, a single estimated value results from the synchronization of the encoder pulses and a series of impulses every time increment (delta). Finally, the harmonic mean between the proposed algorithm and a modification of itself has the smallest relative error possible, which demonstrates optimality.

Two neural networks reconstruct a periodic signal in the second section of this chapter. The first network approximates exclusively the period, and the second estimates the coefficients of the Fourier expansion. Thus, the reconstruction strategy consists of minimizing the mean squared error, by a modified backpropagation algorithm, over a single neuron with a sine transfer function. In addition, a first modification of the algorithm requires less data, rendering the algorithm suitable for real-time implementations.

Two algorithms provide estimates of the period for a periodic signal in the third section of this chapter. Those algorithms guarantee to be within a maximum error of the true period in the order

of the sampling rate. In addition, experimental evidence demonstrates the effectiveness of the proposed algorithms, even for noisy or quasiperiodic data. The results thus extend the range of applicability of the FFT and Sampling Theorem for signal reconstruction and illustrate the pertinence of these algorithms to solve other problems, such as power systems.

3.1 Anticipation by approximating the rate of change

The computation of the rate of change may be the simplest method to anticipate a future state of a system. This method, which supposes that the dynamic is linear, accurately anticipates a state for a constant rate of change, where a prediction results naturally. The anticipated value relates three parameters: the current state, the rate of change, and a delta time value. If the rate of change varies, the approximation becomes less effective as the time for the prediction gets farther from the current time.

Approximating a rate of change is important especially in the context of control, where a controller leads the dynamic of a system based on the signals given by sensors; that is, based on approximations of what actually happens. The algorithm proposed in this section looks for getting the maximum amount of information coming from an incremental encoder, thus establishing a fast and stable measure of the rate of change for the angular position of a shaft, i.e., speed. An incremental encoder is an electromechanical instrument. The mechanical part is an evenly spaced slotted disc, whereas the electric includes a light beam and a detector, as shown in Fig. 3-1. The light generates a high level (one) when the light crosses the slit and reaches the detector, and zero otherwise.

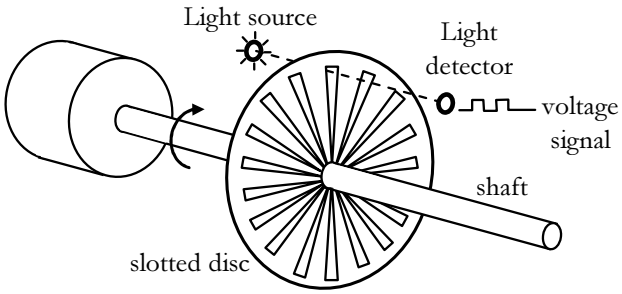


Fig. 3-1 Oscillating counting of *Nep* at constant speed.

The measurement of speed has one physical limitation: approximating a speed requires measurements at two instants in time and the result neglects one of them. In addition, the approximation of a speed has at least two sources of error, because this process requires two measurements: position and time. Since ω_m is $\Delta\theta/\Delta t$ the error on the approximation ω_m lies in the nature of those deltas (that is discrete and digital values). Digital measures possess an inevitable truncation error. Thus, $\Delta\theta$ corresponds to an integer multiple of the pulses coming from the encoder (*pul),* and *dt* is likewise an integer multiple of the clock resolution (*ts*).

The speed at which the time between two consecutive pulses from the encoder lasts exactly *dt* defines the so called *speed limit* given in revolutions per second (ω_{lim} = (1/*pul*)/*dt*). Speed values less than ω_{lim} are low, whereas those greater are high. A similar concept in [75] defines what authors call the speed resolution.

There are two classical algorithms to approximate the speed: fixed-time and fixed-space algorithms. Other algorithms generally come from these two. If an approximation uses the fixed-time approach, which means updating the counting of pulses (*Nep*) each *dt*, the algorithm generates an approximation $\omega_m = Nep\omega_{lim}$. On the other hand, the fixed-space approach updates counts of *dt's* (called *Ndt*) between every two consecutive pulses, so $\omega_m = (1/Ndt)\omega_{lim}$. Fixed-space algorithm is recommended for low speeds, whereas fixed-time algorithm is useful for high speeds. The truncation error makes ω_m to oscillate regardless the approach, for instance *Nep* in Fig. 3-2 switches between 1 and 2, even if the speed remains constant. The effect of these oscillations requires the use of low-pass filters to smooth the value of the approximation, which implies delays.

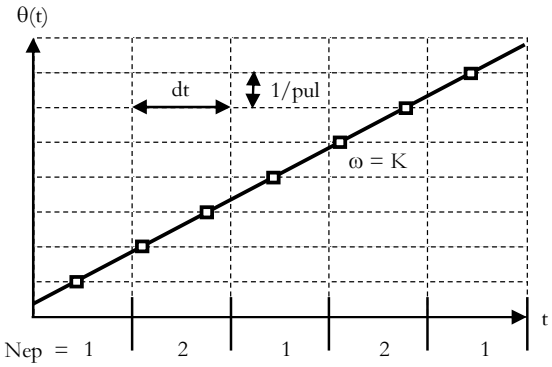


Fig. 3-2 Oscillating counting of *Nep* at constant speed.

Current solutions to avoid the effect of the truncation can be classified mainly in two groups: highly accurate measurements of time, and synchronizing position and time signals. Solutions from the first group use the power of computation of electronic circuits to measure time with high resolution, for instance microcontrollers [54], DSPs [6], and FPGAs. A more elaborate solution in [43] detects the rising edges from the encoder to make a polynomial fitting, which smoothes the approximation. The work in [36] goes so far as claim that the effect of the truncation error disappears when the resolution of the clock reaches 50 ns. The oscillations within the approximation, though small, still remain in this first solution group. On the other hand, the second group looks for synchronizing position and time measurements. Based on this synchronization concept [75] proposes an algorithm named the S method, mainly helpful for high speeds. The proposal in [67], optimal for encoders with few slits, uses synchronization for high speeds, whereas for low speeds it switches the algorithms with one another one based on its model of the system.

3.1.1 A novel algorithm to approximate speed

The new algorithm in this section eliminates oscillations in the speed approximation when they come from discrete and digital measurements, as in the case of an incremental encoder. The core of the proposal consists of synchronizing position and time pulses. Since there is no control over the time at which the encoder pulse appears, the algorithm simply starts to count *dt's* when a position pulse appears. As a result the approximation will be constant for constant speeds, without the use of state observers or models of the system as required in [67].

The guarantee of the mathematical results in this section requires the following assumptions: 1) that the encoder does not lose pulses during the counting; 2) that the measure of time is exact and in multiples of the clock time, *ts*; and 3) that the mechanical distance between pulses remains constant. These assumptions idealize the true performance of an incremental encoder, although they are necessary to assure error bounds on the approximations.

The first step in transforming the encoder signals detects the rising edge from each pulse. This detection generates a pulse train (*Iep*) as shown in Fig. 3-3. The next step counts these pulses, and as a result there is a stair type signal (*Cep*). If the counting *Cep* goes to zero, for instance because the time reaches *dt*, then a register saves the last value in the counter into a variable Nep_1. The scheme in the lower part of Fig. 3-3 illustrates an additional series of blocks. These blocks count *dt's*. A pulse *dt* results when the accumulation of *ts's* reaches the value *dt*. This event resets and starts the counting of *ts's* over. The counter block holds the *dt* counting in a variable *Cdt*. If an *Iep* pulse resets that counter, then a register holds the last value of *Cdt* in Ndt_1.

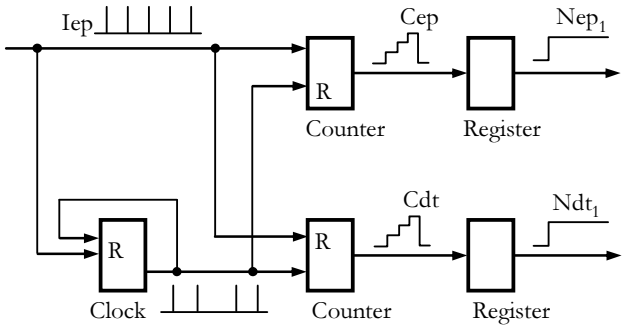


Fig. 3-3 Algorithm scheme and definition of Nep_1 and Ndt_1.

The approximation for the speed equals $\omega_{m1} = n_1\omega_{lim}$, where $n_1 = Nep_1/Ndt_1$. Counting requires that Nep_1 should be a whole number (0,1,2,…), while Ndt_1 should be a natural number. There are more pulses *Iep* in the count than pulses *dt* for speeds higher than ω_{lim}, so Ndt_1 stays fixed at one. On the contrary, there are more pulses *dt* than pulses *Iep* for speeds lower than ω_{lim}, so Nep_1 stays fixed at one. Possible values for n_1, for high speeds are 1,2,3, and so on, and for low speeds, ½, ⅓, ¼, and so on.

The Algorithm 3-1 shows the pseudo code for the new algorithm. This algorithm generates a new approximation ω_{m1} every *ts* by computing Ndt_1 and Nep_1. The core of the algorithm synchronizes the time and space pulses in line 6 where the clock goes to zero. However, this synchronization does not mean that Ndt_1 and Nep_1 update their values synchronously: Ndt_1 updates its value in line 6 (synchronized with *Iep* pulses), and Nep_1 updates its value in line 9 (synchronized with *dt* pulses).

Algorithm 3-1 Pseudo code for signals synchronization and frequency approximation.

Input: *ts*, *dt*, ω_{lim}, *Iep* impulses
Output: ω_{m1}
1 **begin**
2 $Cep \leftarrow 0$, $Cdt \leftarrow 0$, $Nep_1 \leftarrow 0$, $Ndt_1 \leftarrow 1$, clock $\leftarrow 0$
3 **repeat**
4 clock $\leftarrow$ clock + ts
5 **if** $Iep = 1$ & $Cdt \neq 0$ **then**
6 $Ndt_1 \leftarrow Cdt$, clock $\leftarrow 0$
7 **end**
8 **if** clock = dt & $Cep \neq 0$ **then**
9 $Nep_1 \leftarrow Cep$
10 **end**
11 **if** $Iep = 1$ **then**
12 $Cep \leftarrow Cep + 1$, $Cdt \leftarrow 0$
13 **end**
14 **if** *clock* = *dt* **then**
15 $Cdt \leftarrow Cdt + 1$, $Cep \leftarrow 0$, clock $\leftarrow 0$
16 **end**
17 **return**($\omega_{m1} \leftarrow \omega_{lim}(Nep_1 / Ndt_1)$)
18 **until** user stops the loop
19 **end**

Plots a) and b) in Fig. 3-4 illustrate the operation of the algorithm for low and high speeds respectively. Event *a* starts the program in both examples, and subsequent letters define other events. For instance, events *d* and *g* update Nep_1 in both examples. The approximation ω_{m1} remains constant for events later than *f* in the plot a). On the other hand, the approximation after any event later than *d* in the plot b) remains constant.

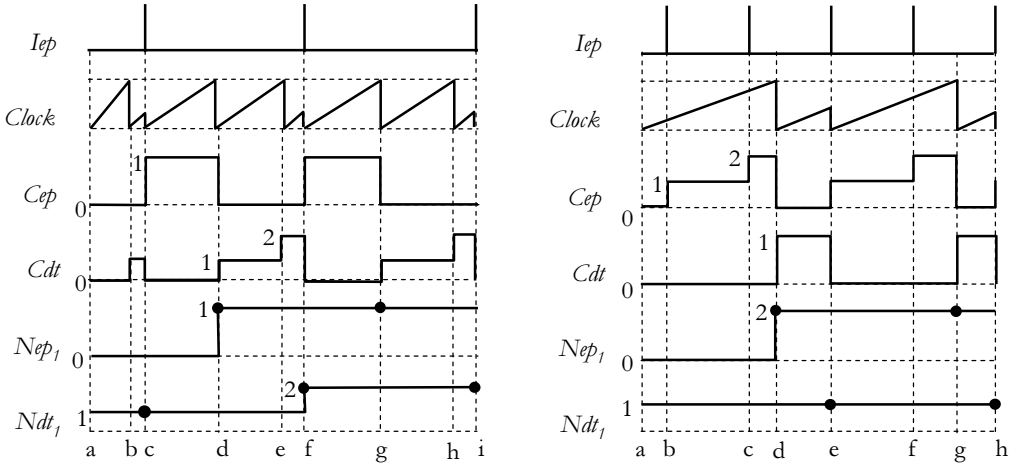


Fig. 3-4 Examples of algorithm execution. a) $\omega \approx 0.4\,\omega_{lim}$, b) $\omega \approx 1.5\,\omega_{lim}$.

Solid points in Fig. 3-4 depict the time when the algorithm updates Nep_1 or Ndt_1. The time between updates for low speeds is $dt_a = (\omega_{lim}/\omega)dt$, and for high speeds is dt_a (see Eq. 3.1). The maximum value of dt_a in Eq. 3.1 is twice *dt* as the speed approaches ω_{lim}, and the delay decreases as ω grows, with *dt* as its limit.

$$dt_a = \left(\left(1 - \frac{1}{\left\lfloor \frac{\omega}{\omega_{\lim}} \right\rfloor} \right) \cdot \frac{\omega}{\omega_{\lim}} + 2 \right) dt \quad (s) \tag{3.1}$$

The value of Nep_1 corresponds to the ceiling of the number of pulses *Iep*, as can be seen in Fig. 3-4; at the same time Ndt_1 equals the floor of the counting of *dt's*, so the approximation n_1, defined as Nep_1/Ndt_1, is always greater than the actual speed as shown in Fig. 3-5, where the X and Y axes have been scaled by ω_{lim}.

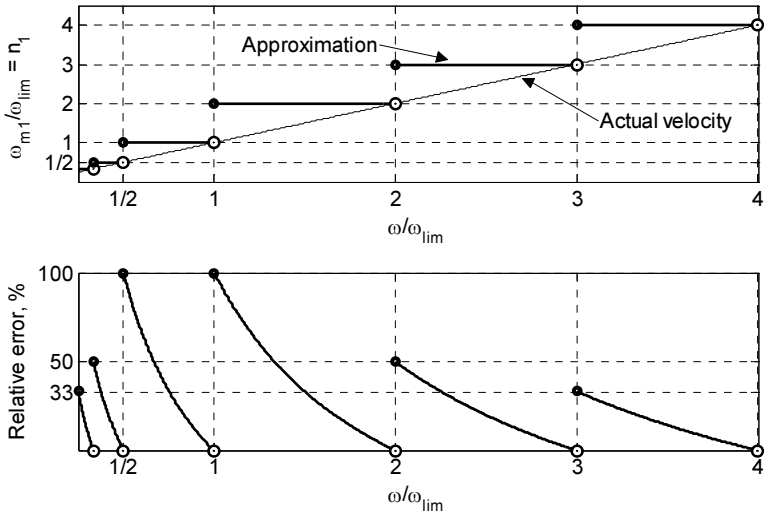


Fig. 3-5 Speed approximation, ω_{m1}, and relative error.

The first measure of quality for the speed approximation correlates relative error and speed, as shown in the lower part of Fig. 3-5. A second and final measure of quality, the maximum relative error per interval, takes into account that the value of ω actually is unknown. An interval covers the whole range of speeds with the same approximation. For instance, all speeds approximated with $n_1 = 3$ as approximation define an interval. This second measure of quality depends entirely on the output of the algorithm. Even so, Eq. 3.2 shows the definition of relative error for the second measure of quality in order to come up with an equation for this indicator.

$$er_1 = \frac{|\omega_{m1} - \omega|}{\omega} \times 100\% \tag{3.2}$$

Since $\omega_{m1} = n_1 \omega_{lim}$, then

$$er_1(\omega, n_1) = \left(n_1 \frac{\omega_{\lim}}{\omega} - 1 \right) \times 100\% \tag{3.3}$$

The relative error er_1 maximizes its value at the left of each interval when the ratio ω_{lim}/ω is maximum. For high speeds (when n_1 exceeds 1), the ratio ω_{lim}/ω corresponds to the inverse of $n_1 - 1$ as can be seen by analyzing the upper part of Fig. 3-5. For instance, if $n_1 = 3$, the maximum ratio equals $\omega_{lim}/\omega = 1/2$, so the maximum relative error reaches 50%. For low speeds the ratio corresponds to $\omega_{lim}/\omega = (1/n_1) + 1$. For instance, if $n_1 = 1/2$, then $\omega_{lim}/\omega = 3$, so the maximum

relative error reaches 50% again. The previous analysis of the ratio ω_{lim}/ω produces the expression for the maximum relative error shown in Eq. 3.4.

$$Er1_{Max}(n_1) = \begin{cases} n_1 \times 100\% & , n_1 \le 1 \\ \dfrac{1}{n_1 - 1} \times 100\% & , n_1 \ge 2 \end{cases} \tag{3.4}$$

The relative error in Eq. 3.4 matches the maximum relative error for the traditional fixed-time algorithm if $\omega \ge \omega_{lim}$ and $dt = ts$. In addition, $Er1_{Max}$ matches the maximum relative error for the traditional fixed-space algorithm, when $\omega < \omega_{lim}$ and $dt = ts$. In summary, the proposed algorithm equates its measure of quality with the relative error from traditional algorithms.

3.1.2 Optimization of the maximum relative error

The optimum for the algorithm minimizes the error Er_{Max}. This minimization comes from an observation about where the maximum error happens. According to the result in Fig. 3-5, that maximum error takes place at the left end from each interval. If the approximations shift down one place, for example from $n_1 = 3$ to a new $n_2 = 2$, or from $n_1 = ½$ to $n_2 = ⅓$, then the relative error decreases. This decrease occurs due to the maximum difference between ω and the new ω_{m2} taking place at the right of each interval instead of at the left.

The new approximation ($\omega_{m2} = n_2\omega_{lim}$) requires n_2 as given in Eq. 3.5.

$$n_2 = \begin{cases} \dfrac{1}{\dfrac{1}{n_1} + 1} & , \omega < \omega_{\lim} \\ n_1 - 1 & , \omega \ge \omega_{\lim} \end{cases} \tag{3.5}$$

Nep_1 and Ndt_1 provide an alternate means for the calculation of n_2, as shown in Eq. 3.6.

$$n_2 = \begin{cases} \dfrac{Nep_1}{Ndt_1 + 1} & , \omega < \omega_{\lim} \\ \dfrac{Nep_1 - 1}{Ndt_1} & , \omega \ge \omega_{\lim} \end{cases} \tag{3.6}$$

The relative error for the approximation ω_{m2} is shown in Eq. 3.7.

$$er_2(\omega, n_2) = \left(1 - n_2 \frac{\omega_{\lim}}{\omega}\right) \times 100\% \tag{3.7}$$

According to Eq. 3.7, and for high speeds, every integer ratio $\omega/\omega_{lim} = n_2 + 1$ maximizes the relative error. For instance if $n_2 = 2$, then $\omega_{lim}/\omega = 1/3$, so the maximum relative error reaches

33.3%. For low speeds, the integer ratio ω/ω_{lim} equals the inverse of $(1/n_2) + 1$. So, for instance, if $n_2 = ½$, the fraction $\omega_{lim}/\omega = 3$. As a result the maximum error reaches 33.3% again. The error in terms of n_2 is shown in Eq. 3.8. This equation defines a supremum and not a maximum, because the right end of each interval is open, as shown in Fig. 3-5.

$$Er2_{Sup}(n_2) = \begin{cases} n_2 \times 100\% & , n_2 < 1 \\ \dfrac{1}{n_2 + 1} \times 100\% & , n_2 \geq 1 \end{cases} \tag{3.8}$$

The computation of the approximation ω_{m1} and ω_{m2} requires the actual value of ω to classify a speed as low or high: see Eq. 3.5 and Eq. 3.6. The value of the speed in general is unknown, although there are two approaches to determine whether the speed passes the speed limit, ω_{lim}. Both approaches use an indirect measure. In the first approach, whereas low speeds have the time between consecutive *Iep's* exceeding *dt*, that time does not exceeds *dt* for high speeds. The second approach counts n_1. Whereas for high speeds, that number exceeds $n_1 \geq 2$, for low speeds it does not.

The approximation ω_{m2} reduces the maximum error up to 50% with the approximation ω_{m1}, mainly for speeds close to ω_{lim}. On the other hand, $Er1_{Max}$ and $Er2_{Sup}$ get closer and closer as the speed gets farther from ω_{lim}, or in other words, when n_1 and n_2 go either to zero or infinity.

Any approximation greater than ω_{m1} generates errors larger than $Er1_{Max}$. At the same time, any estimation under ω_{m2} results in errors over $Er2_{Sup}$. So if there is a way to reduce the maximum error per interval, the resultant approximation should fall between ω_{m1} and ω_{m2}. See, for instance the approximation $\omega_{m3} = n_3\omega_{lim}$ in Fig. 3-6. Now the work involves finding a value for n_3 that minimizes the error in the region $\omega_{m2} \leq \omega_{m3} \leq \omega_{m1}$.

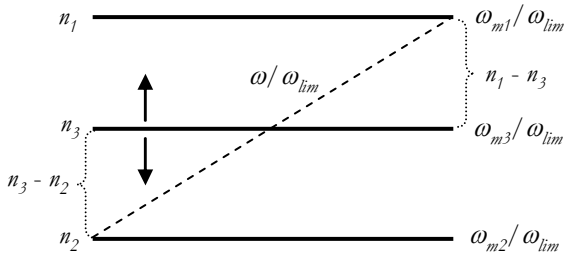


Fig. 3-6 Finding the best approximation, n_3.

The minimum relative error for ω_{m3} could be on the right or left side at each interval. If the approximation ω_{m3} decreases the relative error on the left also decreases. As a result, the relative error on the right increases. Thus, errors on the left oppose errors on the right. Then, the optimal value levels the errors on both sides, as shown by Eq. 3.9, Eq. 3.10 and Eq. 3.11.

$$er_L = \frac{n_3 - n_2}{n_2} \times 100\% = \left(n_3 / n_2 - 1 \right) \times 100\% \tag{3.9}$$

$$er_R = \frac{n_1 - n_3}{n_1} \times 100\% = \left(1 - n_3 / n_1 \right) \times 100\% \tag{3.10}$$

The error er_L reaches zero when $n_3 = n_2$ in Eq. 3.9, and according to the representation in Fig. 3-6 the maximum occurs when $n_3 = n_1$. Eq. 3.10 shows that er_R reaches zero when $n_3 = n_1$, and the maximum occurs when $n_3 = n_2$. Since er_R and er_L are linear, then the minimum from combining er_R and er_L occurs when $er_L = er_R$, as shown in Eq. 3.11. The value of n_3 in Eq. 3.11 defines the optimal approximation for ω_{m3}: the harmonic mean between ω_{m1} and ω_{m2}. No other value of n_i produces an smaller error.

$$\frac{n_3 - n_2}{n_2} = \frac{n_1 - n_3}{n_1} \quad \rightarrow \quad n_3 = \frac{2}{\frac{1}{n_1} + \frac{1}{n_2}} \tag{3.11}$$

The error $Er3_{Max}$ is defined in terms of n_3 in Eq. 3.12, and in terms of n_1 in Eq. 3.13.

$$Er3_{Max}(n_3) = \begin{cases} \dfrac{n_3}{2} \times 100\% & , n_3 < 1 \\ \dfrac{n_3 - \lfloor n_3 \rfloor}{\lfloor n_3 \rfloor} \times 100\% & , n_3 > 1 \end{cases} \tag{3.12}$$

If $n_1 = 1$ or 2 in Eq. 3.13 then $Er3_{Max}$ equals one third of the error given by Eq. 3.4. On the other hand, when $n_1 \rightarrow \infty$ or $n_1 \rightarrow 1/\infty$ then the error $Er3_{Max}$ goes below half of $Er1_{Max}$. Therefore this third approximation ω_{m3} in fact optimizes the error by setting the smallest possible bound for the relative error at each interval.

$$Er3_{Max}(n_1) = \begin{cases} \dfrac{1}{1 + \dfrac{2}{n_1}} \times 100\% & , n_1 \leq 1 \\ \dfrac{1}{2n_1 - 1} \times 100\% & , n_1 \geq 2 \end{cases} \tag{3.13}$$

The bounds for the relative error have been calculated for constant speed. Nevertheless, these results also apply for variable speeds, if the acceleration does not go beyond a given bound. The method for finding the maximum acceleration begins by supposing constant acceleration and high speeds. Under these assumptions, and according to the result in Eq. 3.1, any approximation ω_{mi} updates its value at most every $2dt$, so the maximum acceleration has an increment of speed equal to ω_{lim} in an interval of time equal to $2dt$. The same analysis can be done for speeds under ω_{lim}, as shown in Eq. 3.14.

$$a_{\max} = \begin{cases} \dfrac{\omega_{\lim}}{2dt} \dfrac{1}{Ndt(Ndt+1)}, \left(rev/s^2 \right) & \omega < \omega_{\lim} \\ \dfrac{\omega_{\lim}}{2dt}, \left(rev/s^2 \right) & \omega \geq \omega_{\lim} \end{cases} \tag{3.14}$$

The solid lines in Fig. 3-7 illustrate the approximations ω_{mi}, and the dashed line represents the actual speed. The approximation ω_{m1} is always greater than ω_{m3}, and ω_{m3} is always greater than ω_{m2}. The parameters of the simulation in Fig. 3-7 are dt = 1 ms, pul = 10,000, and ts = 0.1 µs.

These values make ω_{lim} = 0.1 rev/s, so the maximum acceleration is a_{max} = 50 rev/s^2. The ramp with positive acceleration has a constant acceleration equal to 8.33 rev/s^2, whereas the negative acceleration is -12.5 rev/s^2.

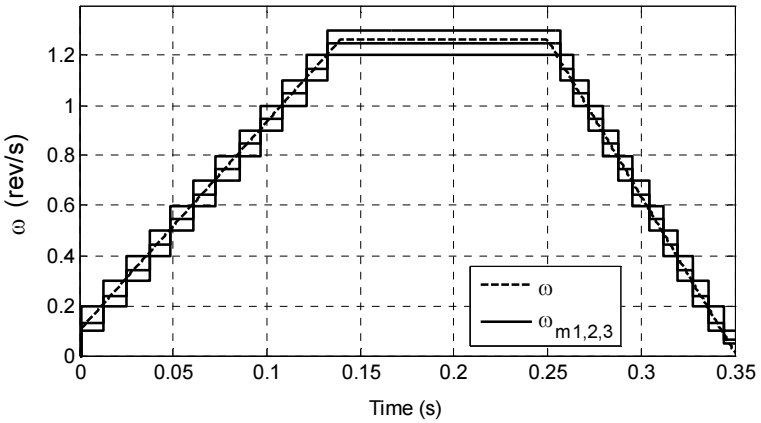


Fig. 3-7 Approximations ω_{mi} for the variable speed case.

The approximation technique that was tested in this section used an incremental encoder, so the results appear to be limited for measuring angular speeds, even though this technique can be equally applied to approximate the rate of change of any variable. There is only a single restriction on the data: the sampling rate should be constant. This restriction, along with the discrete nature of the measurements, generates oscillations on traditional algorithms of approximation, even when the rate of change remains constant; however, those oscillations disappear when using the proposed algorithm. The new algorithm could be used to estimate a derivative required in the control algorithm in the next chapter, which would show the algorithm's usefulness.

3.1.3 Experimental results

In this section, the algorithm generates approximations using real signals coming from an encoder. The scheme in Fig. 3-8 illustrates an encoder coupled to a shaft, where the system looks to approximate the shaft's angular speed. The encoder's output fluctuates between zero to five volts for low and high levels respectively. A data acquisition card captures this voltage signal, and a computer records its values in order to use them later. The encoder has 160 slots, and the DC motor has the commercial reference Minertia 6GFMED. The data acquisition card has a sampling time *ts* equal to 100 μs.

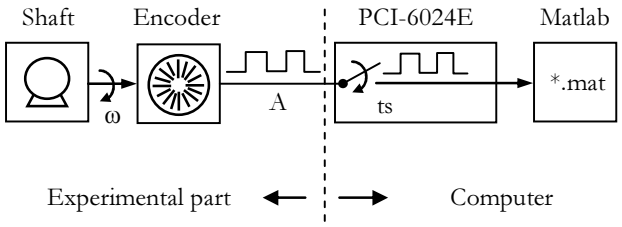


Fig. 3-8 Experimental setup and signal processing scheme.

The algorithm's verification requires the actual value of ω to compute the relative error. Instead of using a highly accurate sensor to get that value, the proposed algorithm finds approximations using the best parameters for the smallest feasible error. For instance, ideal

parameters are *pul* = 160 and *dt* = 2 s. These parameters guarantee a boundary for the relative error equal to 0.07%, when $\omega = 2$ rev/s, as shown in Eq. 3.12. This error decreases to 0.0065%, if $\omega = 24$ rev/s.

The first experiment has the next parameters *pul* = 160 and *dt* = 1 ms, so *dt* overcomes *ts* by a factor of ten. In order to capture the data, the motor runs at constant voltage. Then, a computer saves data for ten seconds starting only when the motor enters its stationary stage. The algorithm gets approximations using the recorded data as input. The same steps are repeated every 0.5 V for constant voltages from 3 through 24 V.

The solid line in Fig. 3-9 represents the bound for the relative error per interval, as indicated in Eq. 3.4, 3.8, and 3.13. On the other hand, dots represent the experimental values for relative errors: e_{r1} for ω_{m1}, e_{r2} for ω_{m2}, and e_{r3} for ω_{m3}. Results in Fig. 3-9 agree with the analysis given in the previous section. Relative errors always fall under bounds, and the approximation ω_{m3} consistently has the minimum relative error.

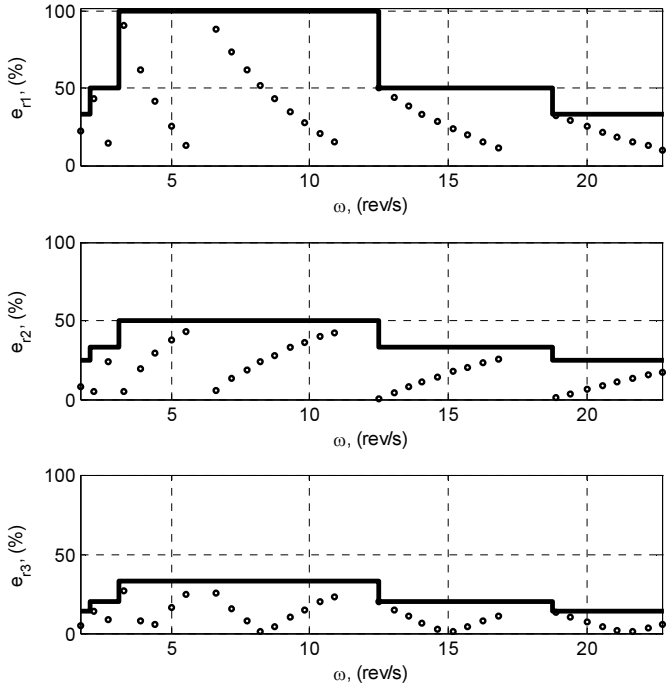


Fig. 3-9 Relative error for *pul* = 160, *dt* = 1 ms, and *ts* = 100 μs.

Other experiments use *dt* equal to 2, 5, 10, 20, 50 and 100 ms. If *dt* increases ω_{lim} decreases. Then, for instance if *dt* is greater than 5 ms, ω_{lim} goes below experimental shaft speeds. The greater *dt*, the larger n_i, and as a result the relative errors decrease, but the updating time increases. So the relation between *dt* and relative errors becomes a trade-off problem, the solution of which will depend on the application.

The relative error also decreases by making ω_{lim} higher than any experimental speed. This change on the ω_{lim} value requires the use of an encoder with a lower number of slots, or the definition of a smaller *dt*. Thus, instead of changing the encoder to run this experiment, only one of every *k* pulses of the encoder was recorded. The experiment uses *k* = 40 and *dt* = 1 ms. So ω_{lim} = 250 rev/s, and since all speeds fall under 30 rev/s, the maximum relative error falls under 10%, as was experimentally proved.

3.2 Approximation of periodic dynamical systems

The knowledge of a certain dynamic usually does not go beyond a region around an operational point. A tool that provides approximations of the entire dynamic on the phase space might offer considerable advantage for the genesis of new control strategies. This implies the capability, not only of approximating states at the current time, but of anticipating future states as well. This section and the next one explore the generation of such a tool for the simplest case, a periodic system. This problem has apparently been solved, for instance by Fourier approximations among many other tools for reconstructing signals. However, an initial exploration in this section shows that anticipating states of periodic systems proves quite challenging for systems with an unknown period.

The first approach to reconstruct a periodic dynamic uses a neural network. The weights of the network should match the Fourier coefficients, Fourier phases, biases, and frequencies. The resultant reconstruction can be used to generalize data from a periodic system. Unfortunately, the approximation degenerates with increasing time and exceeds the maximum time in the data. The final algorithm divides the problem of reconstruction in two: first, to approximate the period, and second, to approximate the Fourier coefficients.

3.2.1 Approximation by a Fourier Neural Network

A Fourier expansion is, by far, the most common mathematical tool to deal with periodical functions. This expansion rebuilds curves based on a sum of sinusoidal functions with appropriated coefficients and frequencies, as shown in Eq. 3.15.

$$f(t) \approx \frac{a_0}{2} + \sum_{m=1}^{n} \left(a_m \cos(m\omega t) + b_m \sin(m\omega t) \right) \tag{3.15}$$

The target function *f(t)* in Eq. 3.15 can be any periodic function. In addition, the coefficient $a_0/2$ represents the mean value of the approximation, whereas ω stands for the fundamental angular frequency, and m could be any natural number from 1 to n. With m = 1, $m\omega$ yields the fundamental frequency, with m = 2, the second harmonic, and so on. Determining the coefficients a_m and b_m requires the computation of integrals as shown in Eq. 3.16 and Eq. 3.17. These equations suppose absolute precision in the value for the frequency and period.

$$a_m = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos(m\omega t) dt \tag{3.16}$$

$$b_m = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin(m\omega t) dt \tag{3.17}$$

There are many ways to approximate the Fourier expansion as well as the transform. For instance, the work in [42] proves that a neural network with a unique neuron in the hidden layer can learn the Fourier transform of any function. In addition, according to the work in [32], a Hopfield neural network learns the coefficients of the Fourier expansion. A third work shows that a sinusoidal activation function improves the generalization of a neural network [21]. References [42], [21], and

[32] show that the implementation of a feedforward neural network also learns a Fourier expansion.

Based on the literature review, the proposed architecture for the application in this section has one input and one output: the time and the approximation at any given input time, respectively. The network has one hidden layer with sinusoidal activation functions, and one output with a linear activation function. Thus, the weights between the input and hidden layers should resemble the multiples of the fundamental frequency. On the other hand, the weights between the hidden layer and the output should resemble the coefficients of the Fourier expansion. The bias signal for the neurons in the hidden layer must reach zero to reproduce the sine terms of the expansion, and $\pi/2$ for the cosine terms. Finally, the bias of the output neuron approximates the average value of the function, as shown in Eq. 3.18.

$$f^*(t) = b_0 + \sum_{n=1}^{\infty} a_n \cdot \sin\left(\left(\omega_n \cdot t\right) + b_n\right) \qquad (3.18)$$

The comparison between the Fourier neural network in Eq. 3.18 and the Fourier expansion in Eq. 3.15 shows that b_0 represents $a_0/2$; also ω_n stands for $m\omega$; and b_n may be 0 or $\pi/2$; finally, a_n in Eq. 3.18 corresponds to the coefficients a_n y b_n in Eq. 3.15. Finally, the training algorithm is the modified backpropagation called Levenberg-Marquardt in Matlab.

The Fourier neural network was trained to approximate the saw tooth function in the lower part of Fig. 3-10. The data consist of two periods and a hundred data points per period, as marked by a line at 4π *s*. The upper part of the figure shows the true frequencies and coefficients as well as their approximations. The lower part of the figure illustrates the target function and the approximation $f_a(t)$. The approximation looks promising for times smaller than or equal to the validation time window (eight periods), but the approximation degenerates as the time increases.

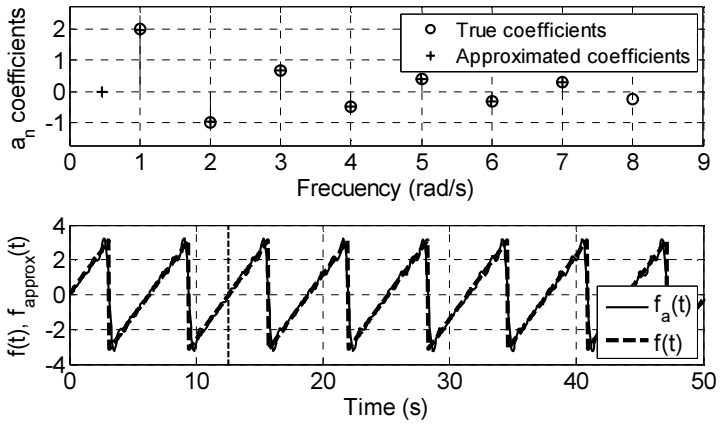


Fig. 3-10 Output of a Fourier neural network after the training process.

The error on the parameters after the learning process has two sources: errors in magnitude and errors in frequency. An analysis of these errors concludes that frequency errors have greater significance than magnitude errors. The Fourier network approximates periodic functions within the temporal range of the available data, so the network actually generalizes. The network is a good candidate for interpolation. However, if the network attempts to extrapolate to times beyond the data limit, then the error increases with the input time.

3.2.2 Approximation of the fundamental frequency

A conclusion from the previous section states that improving the approximation of the fundamental frequency ω also improves the approximation of the whole signal. Consequently, this section looks to create an algorithm to approximate ω. Since $\omega = 2\pi/T$, then the period T can be used to approximate the frequency ω. The most important challenge in approximating the period is the fact that the sampling time may not match the period. For example, a sampling time of 24 hours does not match a year, because the ratio between the earth's period of rotation and its translational period may not be a rational number. The most common adjustment of this difference corresponds to an additional day every leap-year. On the other hand, Fourier expansion and polynomial approximations neglect the irrational ratio between sampling and period.

The approximation of the period in this section uses a neural network. The training process of the network looks to minimize the mean squared error between a perfect sine function and the data. The frequency of the sine changes, whereas any other parameter, such as amplitude, phase, and DC level remains constant through the training. Unfortunately, this minimization process never comes up with the true frequency. To understand why this happens, let suppose that the data set includes exactly one period, from $t = 0$ through the period at $t = 2\pi$, as shown in Fig. 3-11, where the data set resembles a square function.

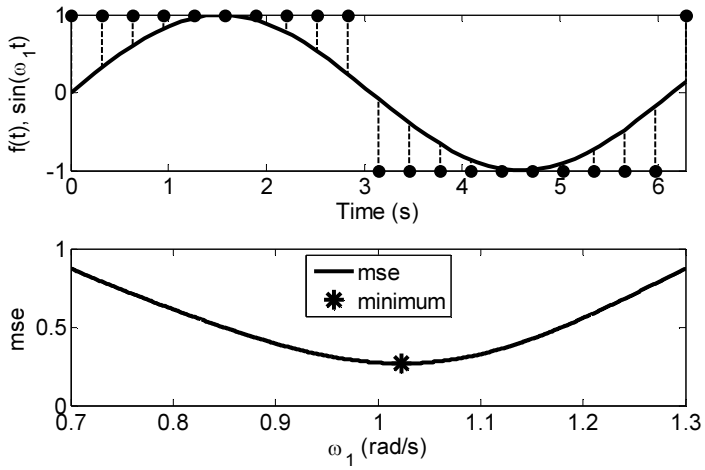


Fig. 3-11 Root mean squared error for a sequence of points approximated by a sine function.

The error *mse* differs from zero because the sine can only match the target function at two points, when $sine(\omega_1 t) = \pm 1$. Any other point increases the value of the *mse* regardless of the sampling rate. In addition, the frequency that minimizes the error differs from the true frequency, as shown by an asterisk in Fig. 3-11, where $\omega = 1$. The next experiment increases the length of the data from one period to 64 periods, shifting the approximation closer to ω, as shown in Fig. 3-12.

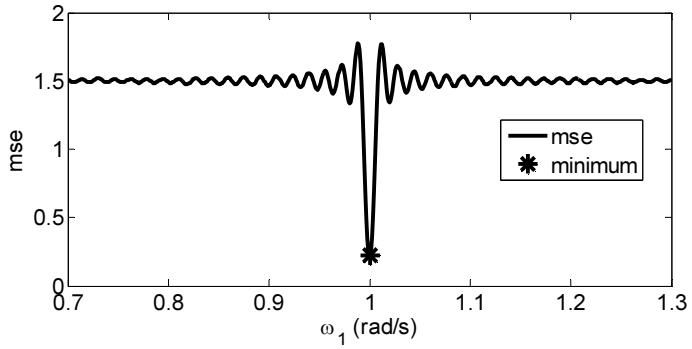


Fig. 3-12 MSE for a square function including 64 periods.

This minimization strategy cannot produce the true frequency, although increasing the data set size does reduce the distance between the true frequency and the approximation. Another feature of the approximation process relates to the ratio between the sampling time and the period. An integer multiple of the sampling time does not necessarily match the period; thus, there may be a residual that can be expressed as a percentage of the sampling time. For instance, 0.5 stands for a residue equal to half sampling time. The simulation in Fig. 3-13 means that in spite of the residue, the estimation approaches the true frequency as the data length increases.

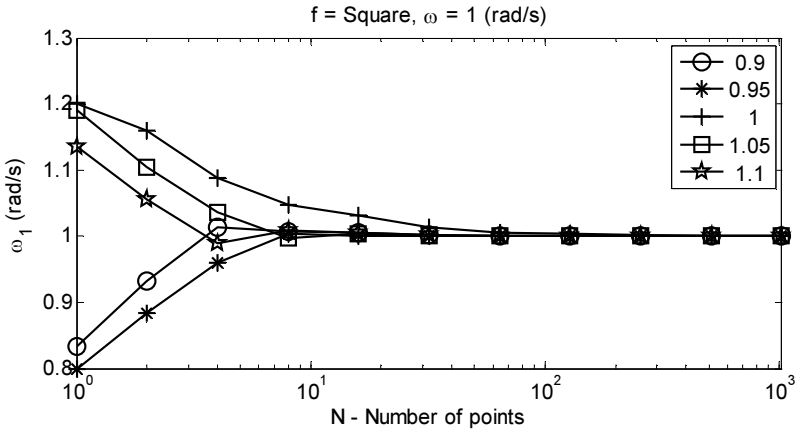


Fig. 3-13 Approximation of the frequency at different sampling rates.

In summary, two problems come from approximating the frequency by minimizing *mse*. On the one hand, the error profile depends on the target function: different shapes produce different errors. On the other hand, any initial estimation of the frequency ends at the closest local minima on the error profile, as shown in Fig. 3-14. As a result, there is no guarantee of getting the best approximation, which coincides with the global minimum. All curves in Fig. 3-14 were simulated using 15 points per period. The lower curve in part a) represents the *mse* when the data set has just 15 points (one period). Subsequent curves represent examples for 2, 4, 8, and 16 periods.

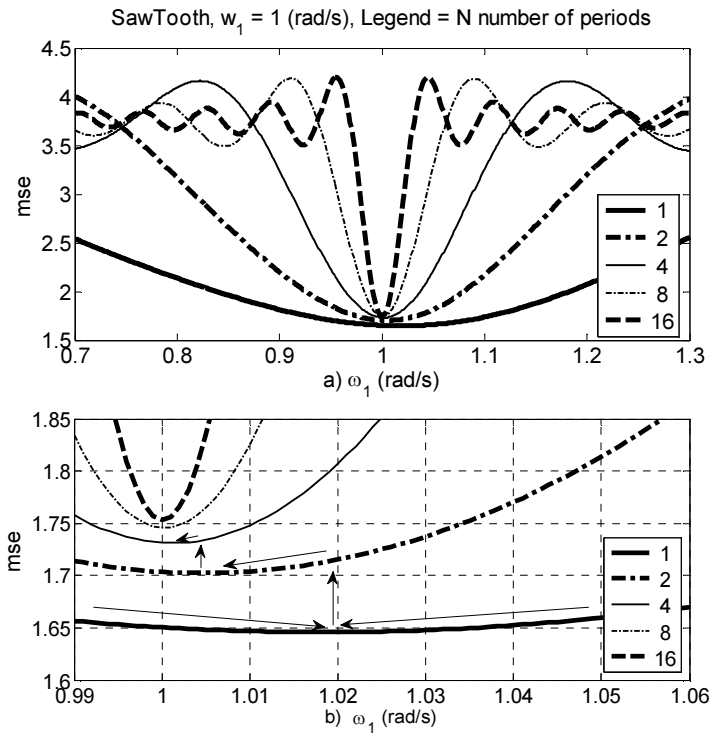


Fig. 3-14 Successive approximations of the fundamental frequency.

The number of local minima in Fig. 3-14 increases with the number of periods in the data set, as shown also in Fig. 3-12. Then, the strategy to find the global minimum starts with data equivalent to one guessed period, where the number of local minima finds its minimum. Then, any new iteration starts with an initial guess equal to the frequency that minimizes the *mse* in the previous iteration. In addition, increasing data with every new iteration can result, for instance, in doubling the size in the previous iteration, as shown in Fig. 3-14 part b).

The approximation algorithm halts when one of two circumstances happen. When $|\omega_{1(i)} - \omega_{1(i-1)}| \leq \delta$, in other words, when the difference between current and previous approximations equals a delta limit. Experimentally was seen that the indicated difference changes linearly with the error on the approximation. The second criterion is to halt when no more data is available.

3.2.3 Mathematical proof of convergence

The mean squared error between a function *f(t)* and a sinusoidal function with amplitude A_1 and frequency ω_1 in discrete time is given by Eq. 3.19.

$$mse_n = \frac{\sum_{i=1}^{n} \left(f(t_i) - B_1 \sin(\omega_1 t_i) \right)^2}{n} \tag{3.19}$$

Lemma 1:

Let *f*(*t*) be a discrete version of a periodical target function, and let $B\sin(\omega_1 t)$ be a sinusoidal approximant with variable frequency. Suppose constant sampling rate and fixed amplitude for the

sinusoidal. Then there is an $\varepsilon > 0$ such that $\varepsilon < |\omega - \omega_1|$, where ω_1 minimizes the mean squared error between the set $f(t)$ and the sinusoidal approximant.

The mean squared error in continuous time *MSE* equals an integral, as shown in Eq. 3.20.

$$MSE = \frac{1}{T}\int_0^T \left(f(t) - B_1 \sin(\omega_1 t)\right)^2 dt \tag{3.20}$$

Since mse_n tends to *MSE* when n tends to infinity, Lemma 1 can be proved using the continuous version of *MSE* instead of the discrete definition of error. In addition, a Fourier expansion replaces $f(t)$, so instead of dealing with any kind of periodic function the proof deals with an infinite series of harmonics. One advantage in using Fourier expansion relates to a property stating that every cosine coefficient in the expansion equals zero for an appropriate shifting phase of $f(t)$. This shifting simplifies the expression in Eq. 3.21, where an approximant T_1 equals $2\pi/\omega_1$.

$$MSE = \frac{1}{T_1}\int_0^{T_1} \left(\left(\sum_{j=1}^{\infty} b_j \sin(j\omega t) \right) - B_1 \sin(\omega_1 t) \right)^2 dt \tag{3.21}$$

The solution of the integral in Eq. 3.21 is done using an iterative process based on the symbolic toolbox of Matlab. The first step includes the first term of the infinite sum exclusively. The second iteration includes the first two terms, $j = 1$ and $j = 2$. This process is repeated for j and the expression is written in Eq. 3.22.

$$MSE = -\frac{1}{4}B_1^2 \frac{\sin(2\omega_1 T_1)}{\omega_1 T_1} + \sum_{j=1}^{\infty} \left(-b_j B_1 \frac{\sin((j\omega - \omega_1)T_1)}{(j\omega - \omega_1)T_1} + b_j B_1 \frac{\sin((j\omega + \omega_1)T_1)}{(j\omega + \omega_1)T_1} + \Delta_j \right) \tag{3.22}$$

The variable Δ_j in Eq. 3.22 represents the sum of the terms independent of ω_1. The result in Eq. 3.22 allows searching for the minimum *MSE* by computing the derivative $dMSE(\omega_1)/d\omega_1$, as shown in Eq. 3.23.

$$\begin{aligned} \frac{dMSE(\omega_1)}{d\omega_1} &= \frac{1}{4}B_1^2 \frac{\sin(2\omega_1 T_1)}{\omega_1^2 T_1} - \frac{1}{2}B_1^2 \frac{\cos(2\omega_1 T_1)}{\omega_1} + \ldots \\ \ldots &+ \sum_{j=1}^{\infty} \left(b_j B_1 \frac{\cos((j\omega - \omega_1)T_1)}{(j\omega - \omega_1)} - b_j B_1 \frac{\sin((j\omega - \omega_1)T_1)}{(j\omega - \omega_1)^2 T_1} + b_j B_1 \frac{\cos((j\omega + \omega_1)T_1)}{(j\omega + \omega_1)} - b_j B_1 \frac{\sin((j\omega + \omega_1)T_1)}{(j\omega + \omega_1)^2 T_1} \right) \end{aligned} \tag{3.23}$$

Since $T_1 = (2\pi)/\omega_1$, then:

$$\begin{aligned} \frac{dMSE(\omega_1)}{d\omega_1} &= -\frac{1}{2}\frac{B_1}{\omega_1} + \ldots \\ \ldots &+ \sum_{j=1}^{\infty} b_j \left(\frac{\cos((j\omega - \omega_1)T_1)}{(j\omega - \omega_1)} - \frac{\sin((j\omega - \omega_1)T_1)}{(j\omega - \omega_1)^2 T_1} + \frac{\cos((j\omega + \omega_1)T_1)}{(j\omega + \omega_1)} - \frac{\sin((j\omega + \omega_1)T_1)}{(j\omega + \omega_1)^2 T_1} \right) \end{aligned} \tag{3.24}$$

For $\omega_1 = \omega$, and using l'Hôpital's rule in Eq. 3.25 and Eq. 3.26:

$$\left.\frac{dMSE(\omega_1)}{d\omega_1}\right|_{\omega_1 = \omega} = -\frac{1}{2}\frac{B_1}{\omega_1} + \sum_{j=1}^{\infty} b_j \left(\frac{\sin((j\omega - \omega_1)T_1)}{1} + \frac{\sin((j\omega - \omega_1)T_1)}{2T_1} + \frac{-\sin((j\omega + \omega_1)T_1)}{1} + \frac{\sin((j\omega + \omega_1)T_1)}{2T_1} \right) \tag{3.25}$$

$$\left.\frac{dMSE(\omega_1)}{d\omega_1}\right|_{\omega_1=\omega} = -\frac{1}{2}\frac{B_1}{\omega_1}\Bigg|_{\omega_1=\omega} = -\frac{1}{2}\frac{B_1}{\omega} \tag{3.26}$$

Since the derivative in Eq. 3.26 differs from zero, then minimizing *MSE*, using T_1 around a period, does not result in $\omega_1 = \omega$. On the other hand, A_1 remains constant because the goal of the minimization process is not to reduce *MSE*, but to minimize the distance between ω_1 and ω.

Lemma 2:

In the limit as $T \to \infty$, the frequency $\omega_1 \to \omega$

The value of T_1 from Eq. 3.24 will be replaced by Tz to simplify the analysis, so $T_z = ZxT_1 + T_1/8$, and Z is the set of natural numbers.

$$\begin{aligned}&\frac{dMSE(\omega_1)}{d\omega_1} = \frac{1}{4}B_1^2\frac{1}{\omega_1^2 T_z} + \ldots \\ &\ldots + \sum_{j=1}^{\infty}\left(b_j B_1\frac{\cos((j\omega-\omega_1)T_z)}{(j\omega-\omega_1)} - b_j B_1\frac{\sin((j\omega-\omega_1)T_z)}{(j\omega-\omega_1)^2 T_z} + b_j B_1\frac{\cos((j\omega+\omega_1)T_z)}{(j\omega+\omega_1)} - b_j B_1\frac{\sin((j\omega+\omega_1)T_z)}{(j\omega+\omega_1)^2 T_z}\right)\end{aligned} \tag{3.27}$$

L'Hôpital's rule simplifies Eq. 3.27 to compute the limit when $\omega_1 \to \omega$, and then the limit when $T_z \to \infty$.

$$\begin{aligned}&\lim_{\omega_1\to\omega}\frac{dMSE(\omega_1)}{d\omega_1} = \ldots \\ &\lim_{\omega_1\to\omega}\left(\frac{1}{4}\frac{B_1}{\omega_1^2 T_z} + \sum_{j=1}^{\infty} b_j\left(\frac{\sin((j\omega-\omega_1)T_z)}{-1} + \frac{\sin((j\omega-\omega_1)T_z)}{2T_z} + \frac{-\sin((j\omega+\omega_1)T_z)}{1} + \frac{\sin((j\omega+\omega_1)T_z)}{2T_z}\right)\right)\end{aligned} \tag{3.28}$$

$$\lim_{T_z\to\infty}\left(\lim_{\omega_1\to\omega}\frac{dMSE(\omega_1)}{d\omega_1}\right) = \lim_{T_z\to\infty}\left(\frac{1}{4}\frac{B_1}{\omega^2 T_z}\right) = 0 \tag{3.29}$$

Two conclusions emerge from Eq. 3.29. First, the algorithm converges. The frequency that minimizes the expression gets closer and closer to the true ω when $T_z \to \infty$. And second, the approximation decreases with the increasing data size T_z.

Lemma 3:

The number of local minima of the error *MSE* around ω increases as the data size increases.

Recall Eq. 3.24, but instead of T_1 use NT_1 in Eq. 3.30 and Eq. 3.31, where N is a natural number.

$$\begin{aligned}&\frac{dMSE(\omega_1)}{d\omega_1} = -\frac{1}{2}\frac{B_1}{\omega_1} + \ldots \\ &\ldots + \sum_{j=1}^{\infty} b_j\left(\frac{\cos((j\omega-\omega_1)NT_1)}{(j\omega-\omega_1)} - \frac{\sin((j\omega-\omega_1)NT_1)}{(j\omega-\omega_1)^2 NT_1} + \frac{\cos((j\omega+\omega_1)NT_1)}{(j\omega+\omega_1)} - \frac{\sin((j\omega+\omega_1)NT_1)}{(j\omega+\omega_1)^2 NT_1}\right)\end{aligned} \tag{3.30}$$

By simulation it was seen that the first term in the sum, $j = 1$, approximates the entire sum. An intuitive way to understand this is by analyzing the denominator of every term. If $\omega \approx \omega_1$ then the

difference $\omega - \omega_1$ goes to zero, so the whole term is weighted more than any other. On the other hand, increasing the j index decreases the weight of the term in the whole sum.

$$\lim_{NT_1 \to \infty} \frac{dMSE(\omega_1)}{d\omega_1} = -\frac{1}{2}\frac{B_1}{b_1} + \frac{\omega_1 \cos((\omega - \omega_1)NT_1)}{(\omega - \omega_1)} \tag{3.31}$$

The first term on the right of Eq. 3.31 is constant, whereas the second grows as $\omega_1 \to \omega$, so Eq. 3.31 can approximate as shown in Eq. 3.32.

$$\lim_{NT_1 \to \infty} \frac{dMSE(\omega_1)}{d\omega_1} = 0 = \left(\frac{\omega_1}{\omega - \omega_1}\right)\cos\left(2\pi N \frac{\omega}{\omega_1}\right) \tag{3.32}$$

Since the frequency of the cosine term grows with N in Eq. 3.32, the number of local minima also increases. So, increasing the data size increases the likelihood of falling into a local minimum different than the global minima. This effect justifies the use of the iterative algorithm where the first iteration begins with samples of small data size, which are gradually increased in subsequent iterations.

This section concludes analyzing the effect of changing the amplitude of the sine function that approximates a periodic function. Based on the result in Eq. 3.23, the critical points can be approximated using only the first term of the sum, as shown in Eq. 3.33. This experimental proof supposes that the fundamental frequency of the Fourier expansion overcomes any other harmonic, which is the case in most of the periodic functions.

$$-\frac{1}{2}\frac{B_1}{b_1}\frac{1}{\omega_1} + \frac{\cos((\omega - \omega_1)NT_1)}{(\omega - \omega_1)} - \frac{\sin((\omega - \omega_1)NT_1)}{(\omega - \omega_1)^2 NT_1} + \frac{\cos((\omega + \omega_1)NT_1)}{(\omega + \omega_1)} - \frac{\sin((\omega + \omega_1)NT_1)}{(\omega + \omega_1)^2 NT_1} = 0 \tag{3.33}$$

If B_1 and b_1 have the same sign, ω_1 falls into a minimum; otherwise, ω_1 is a maximum.

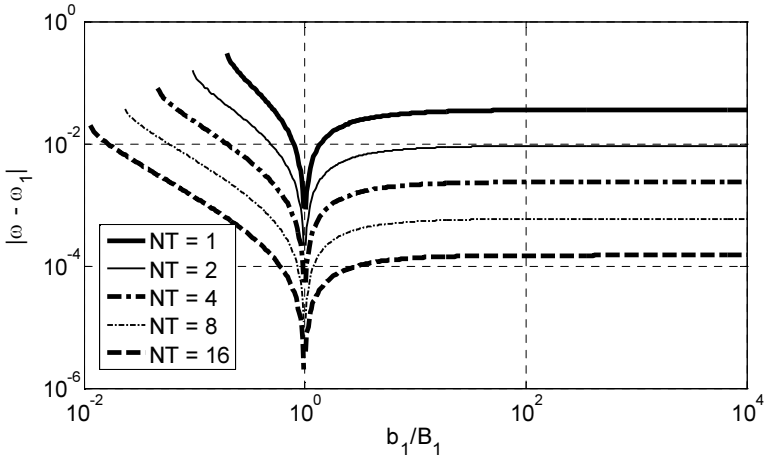


Fig. 3-15 Influence of A_1 amplitude on the approximation ω_1.

The relationship between amplitudes for approximant and fundamental harmonic influences the value of ω_1. However, if b_1 is greater than or equal to a hundred times B_1, then ω_1 does not change anymore, as shown in Fig. 3-15. On the other hand, if b_1 is a hundred times smaller than B_1 there could be non minimum. So, the rule of thumb for choosing B_1 says that B_1 should be smaller than or equal to the amplitude of the function.

3.2.4 Reconstruction using a neural network

Any result coming from the three lemmas in the previous section can be generalized for an approximant equal to the second or higher (*nth)* harmonic. If such a case occurs, the minimization of *MSE* estimates double or *n* times the fundamental frequency, respectively. This conclusion also applies for the sum of harmonics. So, instead of making approximations with one pure sine function, the approximant would have a sum of sine functions, which may resemble the shape of a target function.

The difference between using a pure sine function or several harmonics for the approximant seems subtle. However, the enlarged approximant reduces the error, as shown in Fig. 3-16. The example in that figure corresponds to a square function. The sampling rate equals four points per period, whereas the data size, *N*, covers four periods. The approximant increases in complexity, and this has a computational cost. Nonetheless, the data size requirements diminish.

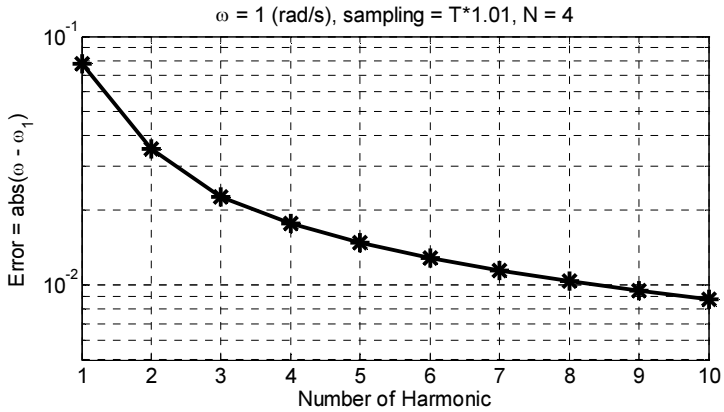


Fig. 3-16 Effect of harmonics on the approximant function.

The approximation ω_1 serves to transform the input of the neural network in Fig. 3-17. Instead of using the time, these inputs match the terms of a Fourier expansion. Thus, the network has only one job, to learn the coefficients of the Fourier expansion. By definition, the coefficients minimize the mean square error, resulting in an approximation of the Fourier expansion.

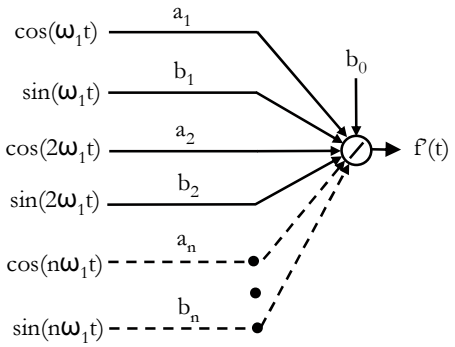


Fig. 3-17 Network architecture to learn Fourier coefficients.

Experiments show that the modified backpropagation algorithm effectively trains the network and stabilizes after just *one* epoch. In addition, increasing the sampling rate improves the approximations. Results of the estimation for a triangular function are shown in Fig. 3-18. The value of $\omega_1 = 1.0001$, whereas $\omega = 1$ (rad/s).

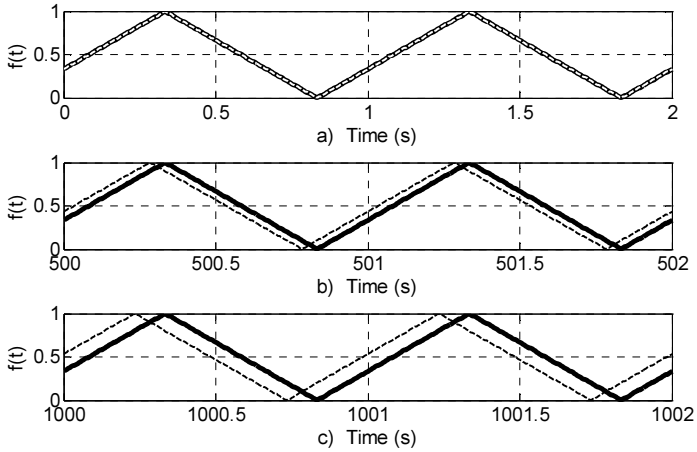


Fig. 3-18 Prediction of the value for a triangular function.

The prediction in Fig. 3-18, part a), approximates the target closely. In addition, the network can effectively anticipate the function for times more than five hundred periods ahead, as shown in part b), where the continuous line represents the target function, and the dashed line represents the anticipated value. However, the error gets visible for times on the order of a thousand periods ahead. The anticipation of smooth functions yields better results due to the Gibbs phenomenon, a well-known effect for Fourier expansions. This condition predicts problems around discontinuities.

This section concludes with three examples of approximation and anticipation. The plots in Fig. 3-19 correspond to a) Saw tooth, b) Van Der Pol, and c) Square functions. Each approximation has a hundred harmonics. The error is the difference between the target value and the anticipated value at each time.

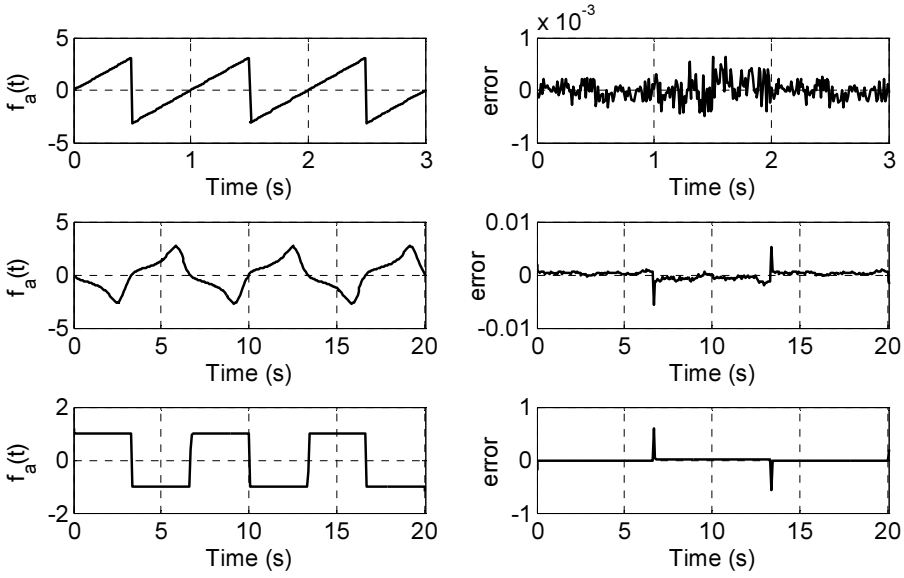


Fig. 3-19 Three examples of anticipation.

The reconstruction of periodic signals can be performed by approximating the period of the signal, which can be seen also as approximating the fundamental frequency. This frequency is estimated using a single neuron with a sine transfer function and Levenberg-Marquardt as training algorithm, as explained in [71]. The approximation leads to an inevitable error that can be minimized only increasing the data size in an iterative fashion. This section has presented an algorithm that uses

evenly spaced data samples to approximate the period of a signal through successive iterations. The first iteration requires a rough approximation of the period and sufficient data points to cover at least the guessed period; the minimization of the mean squared error then provides a first estimation for the period. The estimation at the end of each iteration is used as the initial guess for the next iteration; at the same time, the data size progressively increases every iteration, which guaranties that each iteration will provide a better approximation than the previous one.

3.3 Period estimation by minimizing the variation

Signal refers to any abstract, symbolic, or physical manifestation of information such as music, language, text, graphics, multimedia, and sensor data. A classical result in signal processing, the Sampling Theorem asserts that reconstruction of a (perhaps unknown) periodic target signal up to order n harmonic frequencies requires that the data be sampled at a rate of at least $2n$ data points per cycle. Under these conditions, the Discrete Fourier Transform (DFT) computes the period of a sampled signal, but only if a multiple of the sampling rate matches the period of the signal. Otherwise, the transformation cannot guarantee an accurate result. The reconstruction of a signal requires *a priori* knowledge of the period, or at least of its multiplicity in the samples. Such an assumption may lead to large errors when the estimation performs predictions, especially in the long term. The problem becomes even more difficult in the presence of sparse data or real-time applications, or, due to sampling errors that are unavoidable when collecting discrete observations made by sensors.

Given the availability of several methods for constructing the signal given its period and appropriate data sampled at the appropriate rate [55], the problem consists of estimating the period of the signal. Under the standard assumptions of classical algorithms such as the FFT, period estimation can be done through estimation of its inverse, the signal frequency. Frequency estimation has focused on single sinusoidal signals in [39], [58], and [69]. In [39], authors use a method called phase unwrapping, which estimates the frequency by performing a linear regression on the phase of the signal. However, this method cannot be used in real-time applications due to a high computational time. The method in [58] is a deterministic method under noiseless conditions, and bases its results on the solution of a system of linear equations using DTF for two time intervals. The method in [69] requires iterations, combining a matrix of elements converted to two unitary vectors in order to give the final frequency estimate. Other works allow several sinusoidal components. For instance, in [12] the estimate of the frequency for a power system minimizes the mean squared error between an assumed signal model and the actual signal. This work focuses on methods that can be implemented in electronic circuits, but they neither prove nor compare the performance of their algorithm with popular estimators.

In [51], authors suggest a time-varying sinusoidal representation to estimate frequency for signals such as speech and audio. Experimental results show that, although computationally intensive, the suggested algorithm outperforms FFT-based frequency approaches in nonstationary environments. Finally, a more general family of algorithms uses the DFT as a basis to estimate the frequency. In [9], authors propose a method exclusively for complex exponential functions under white noise. They use DFT in the first stage, after which the three terms with highest spectrum magnitude produce a better estimate by combining the terms. The authors claim that the estimation performs well for small/medium sampling rates, but nothing is said about high sampling rates. In [78], a noniterative method consists of two parts: a coarse estimation, given by the FFT, and a fine estimation, using least square minimization of three spectrum lines. The authors assume minimal additive effects from noise in the coarse estimates used to construct the combined estimate.

3.3.1 Period estimation

In this section, the proposed algorithms will be described in three parts. The first defines the problem precisely. The second includes the definition of all the variables and measurements of error in order to approximate the period, as well as an example of the application of the algorithm. The third part corresponds to the report of two applications of the approximation algorithm.

Problem definition

This document only concerns a restricted, but fairly general class of signals, namely those described by so-called functions of *bounded variation* over a finite interval. This class includes the usual signals describable by smooth functions, but does not include all continuous functions. On the other hand, it includes large families of discontinuous functions. The precise definition is as follows.

Definition 1. A function $f : [a,b] \rightarrow \boldsymbol{R}$ is of bounded variation, denoted $f \in BV[a,b]$, if there exists some constant M such that $\sum |f(t_i)-f(t_{i+1})| \leq M$ for every partition π: $a=t_0<t_1<\ldots< t_n=b$ of a finite interval $[a,b]$.

The function f defines a periodic function over the real numbers $\boldsymbol{R}$ (still denoted f) with $[a,b]$ as fundamental period. The problem to be solved is to find the best estimated value T' of the unknown period T, based on a discrete set of observations of the values of f at points sampled at a constant rate t_1.

Assumptions on the target signals:

A1. The function f has a well defined period T;

A2. The variation of the function f is bounded over the fundamental period;

A3. The system samples data at an exact sampling rate. The sensitivity of algorithms A and B under noise conditions will be addressed later.

Input and output of the algorithm

Input: A set D of data points of the form $D = \{(it_1, f(it_1)), i=0,1,2, \ldots, n\}$, where t_1 corresponds to the sampling rate, and f is a periodic signal. The algorithm also requires an initial estimation of the period T_0, so that $T_- < T_0 < T_+$. For instance, $T_- = 3/4T$ and $T_+ = 5/4T$.

Output: An estimate T' of T, or "*The period cannot be determined*".

Note. After a time shift, and without loss of generality, $t_0 = 0$. Thus, the length of data should be $t_n \geq \delta$, where $\delta = 8T_0/3$. This constant results from the following analysis. If $T_0 = T_-$, then the data length should be $1/T_-$ in order to include T in the search. Also, if $T_0 = T_+$, then the data length should be $1/T_+$ to include T. Thus, the upper bound is $1/T_-$, and because the algorithm will require two section of data, then $\delta = 2/T_- = 8T_0/3$. The data points $(it_1, f(it_1))$ will sometimes be written as $(t_i f_i)$ or as (it_1,f_i).

3.3.2 Two period approximation algorithms

In this section, we present pseudocode for two algorithmic solutions to the period approximation problem. Given a tentative value t for the period, the term 'section' refers to data points taken at time points in the interval $[t_0,t]$ or to data points taken at times in the interval $(t,2t]$. $\Delta_n(t)$, or just Δ_n will refer to the value $\Delta_n = (T - t_n) / t_1$, where n corresponds to the last data index such that $t_n \le t$. Note that any of the three values t_1, $t_n = nt_1$, and Δ_n can be calculated from the other two. N will denote the unique (unknown) n such that $t_N \le T < t_{N+1}$. Below, we may also refer to a time index i as the index of the data point $t_i = it_1$. With this notation in place, we now turn to the description of the algorithms for estimating T.

Algorithm *A*

This algorithm minimizes an error functional $\Delta f(t)$ given by

$$\Delta f(t) = \sum_{i=1}^{n-1} \max\left(\left|f(t_{n+i}) - f(t_i)\right|, \left|f(t_{n+i}) - f(t_{i-1})\right|\right) \qquad (3.34)$$

over the sampling partition of the interval $[0,t]$ into intervals of the same length t_1. The error $\Delta f(t)$ measures the periodicity of the data between two consecutive sections for a candidate period value t of the target function. Given a data set D, the functional $\Delta f(t)$ is piecewise constant on the intervals $[t_i,t_{i+1})$. Therefore, any approximation of T on the basis of D cannot be guaranteed to be any closer to T than half the radius of the partition $t_1/2$. The algorithm returns the value of $T' = t_N + 0.5t_1$ such that $\Delta f(t_N)$ minimizes the error over the interval $[0,t_N]$, as shown in Algorithm 3-2. The search interval of the period has two limits $[\gamma_-, \gamma_+]$, where $\gamma_- = 4T_0/5$ and $\gamma_+ = 4T_0/3$.

Algorithm 3-2 Algorithm *A*.

Input: data set (f), initial guess of the period (T_0), sampling rate (t_1)
Output: An approximate value of the period (T')
$Ni \leftarrow$ floor(γ_-/t_1) $Nf \leftarrow$ ceiling(γ_+/t_1) for $n = Ni : Nf$ compute $\Delta f(t_n)$ end for $m \leftarrow n$ that minimizes Δf over $[0, t_{Nf}]$ $T' \leftarrow (m + 0.5)t_1$

Algorithm *B*

The algorithm effects a search for a better estimate T' using the Δf value as a guide. Once the algorithm computes a measure of error $\Delta f(t_n)$ for each such t_n in the interval $[\gamma_-, 2\gamma_+]$, the algorithm eliminates candidate periods t_n for the candidates yielding $\Delta f(t_n)$ exceeding a bound V_π given by Eq. 3.35, where $1+t_n$ is the number of data points in the first section.

$$V_\pi = \sum_{i=1}^{n-1} \left|f(t_i) - f(t_{i-1})\right| \qquad (3.35)$$

If fewer than two points has a $\Delta f(t_n)$ value falling under V_π, the algorithm returns "The period cannot be determined" and halts. If there are two such points or more surviving candidates, the next step runs a multiplicity test on each of them, as shown in Algorithm 3-3, where ones is a vector with all the entries one. The basic idea behind this test asserts that multiples of the true approximation t_N should also be candidates for the period of the function; however, given the discrete nature of the data, the multiplicity allows a maximum error t_1. If there is only one candidate in $[\gamma_-, \gamma_+]$ with multiples of itself across $[\gamma_-, 2\gamma_+]$, the output of the algorithm B equals $T' = t_N + 0.5t_1$ otherwise, the algorithm returns "The period cannot be determined" and halts, as shown in Algorithm 3-4.

Algorithm 3-3 Test for multiplicity condition.

Input: index vector of period candidates (NB), last survivor time index (Nf), maximum number of data points (Np)
Output: index vector of the candidates that satisfy the multiplicity condition (M)

M = [], $i \leftarrow 1$
while $NB_i \le Nf$
$m \leftarrow 2$
if $mNB_i \le Np$
while min(|NB - $mNBi$×ones$_{\text{length}(NB)}$|) ≤ (m-1)
$m \leftarrow m + 1$
end while
else exit
if $mNB_i \ge Np - m$ then
$M \leftarrow [M\ NB_i]$,
else exit
$i \leftarrow i + 1$
end while

Algorithm 3-4 Algorithm B.

Input: data set (f), an initial guess of the period (T_0), a sampling rate t_1
Output: approximated period (T') or "The period cannot be determined"

$NB \leftarrow$ [ ]
$Ni \leftarrow$ floor(γ_-/t_1)
$Nf \leftarrow$ ceiling(γ_+/t_1)
$Np \leftarrow$ ceiling($2\gamma_+/t_1$)
for $n = Ni : Np$
compute Δf and V_π
If $\Delta f \le V_\pi$ then
$NB \leftarrow [NB\ n]$
else exit
end for
$M \leftarrow$ run multiplicity test for NB
if the length of M is one
$T' \leftarrow (M + 0.5)t_1$
else
'The period cannot be determined'
end if

Example

To illustrate the previous algorithms, consider the target signal $f(t) = sin(2t)sin(20t)$, with period $T = \pi$ and data sampled at intervals $t_1 = 0.2$ s. Taking $T_0 = 5T/4$, the evaluation of Δf in Fig. 3-20 indicates that there is only one candidate under the bound V_π that also satisfies the multiplicity property. The error for that estimation is about $7t_1$. The markers *x* or *o* shown in Fig. 3-20 illustrate the feasible candidates after running algorithm *A* and algorithm *B*.

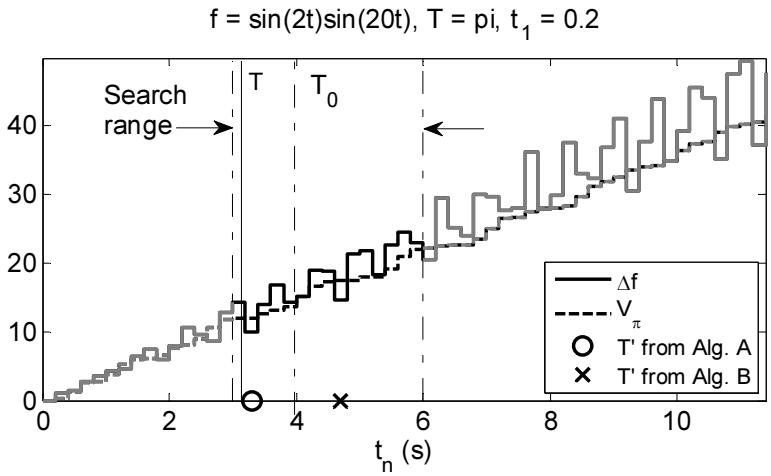


Fig. 3-20 Low sampling rate.

The corresponding situation with a finer sampling of target values ($t_1 = 0.01$ *s*) is shown in Fig. 3-21. The returned value for *T'* equals 3.145 *s*. The relative error can be computed as $|T - T'| \approx 0.001T$ (about 0.1%), which shows that actually *T'* approximates the presumably unknown period *T*. The output of algorithm *B* matches the output of algorithm *A*.

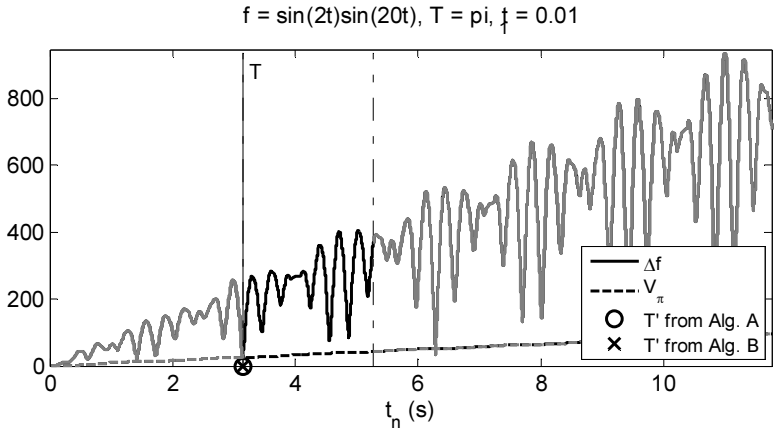
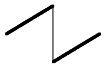
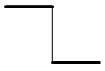


Fig. 3-21 Output of the algorithm with sufficient sampling data.

Experimental test of the approximation algorithms

Table 3-1 shows an assessment of the quality of the period approximation provided by algorithms *A* and *B*.

Table 3-1 Relative error in period approximation of some target functions.

Sine: $f = \sin\left(\frac{2\pi}{T}t\right)$		np = 60, Δ_N = 0.1 T' = 3.162855 Error = 0.3999
|sine|: $f = \left\|\sin\left(0.5\frac{\pi}{T}t\right)\right\|$		np = 100, Δ_N = 0.2 T' = 3.151093 Error = 0.3000
sine2: $f = \sin\left(\frac{2\pi}{T}t\right)\sin\left(10\frac{2\pi}{T}t\right)$		np = 200, Δ_N = 0.3 T' = 3.144745 Error = 0.1999
tsine(1/t): $f = \mathrm{mod}(t,T)\sin\left(\frac{1}{\mathrm{mod}(t,T)}\right)$		np = 500, Δ_N = 0.4 T' = 3.142222 Error = 0.1000
Chebyshev: $f = 256\bar{t}^9 - 579\bar{t}^7 + 432\bar{t}^5 + \ldots$ $\ldots - 120\bar{t}^3 + 9t; \quad \bar{t} = \left(\frac{\mathrm{mod}(t,T)}{T/2}\right) - 1$		np = 60, Δ_N = 0.5 T' = 3.141593 Error = 0.0000
Saw tooth: $f = \mathrm{mod}\left(\frac{t+T/2}{T/2},2\right) - 1$		np = 100, Δ_N = 0.6 T' = 3.138438 Error = 0.0999
Tent map: $f = \mathrm{mod}\left(\frac{t+T/2}{T/2},2\right) - 1$		np = 200, Δ_N = 0.7 T' = 3.138446 Error = 0.1999
Square: $f = -2\left(\left\lfloor \mathrm{mod}\left(\frac{t}{T/2},2\right) - 0.5 \right\rfloor\right)$		np = 500, Δ_N = 0.8 T' = 3.139707 Error = 0.3000

The period for all the experiments was set to a common value $T = \pi$ for convenience in comparing results across different functions, and the sampling rate t_1 can be computed from the number of data points per period (denoted *np*) and Δ_N in Table 3-1, as shown in Eq. 3.36. For each target function, *np* was set to 60, 100, 200 and 500; for each *np* there where eleven values for Δ_N, as shown in Fig. 3-22. The error in the estimation was defined as $|T - T'|/t_1$.

The results in Fig. 3-22 show that regardless of the shape, sampling rate, amplitude of the target function, or algorithm, the error on the approximation appears to be only a function of Δ_N. In addition to the experiments in Table 3-1 and Fig. 3-22, many other experiments were run with different phase shifts and the same conclusion held true. Table 3-1 shows data for some of the experiments, whereas Fig. 3-22 shows the error for the eight target functions in Table 3-1 and different sampling rates. The linearity on the error in Fig. 3-22 comes from the definition of the error $|T - T'|/t_1$. T comes from Eq. 3.36, and the value of T' comes from the next analysis. The estimation of the period given by algorithm *A* always meet the point on the left side of T, and because of the addition of $t_1/2$ (as can be seen on Algorithm 3-2 and Algorithm 3-4), then $T' = (np - 1)t_1 + t_1/2$. As a result, replacing T and T' on the expression of the error, the error is $|\Delta_N - ½|$.

$$t_1 = \frac{T}{np - 1 + \Delta_N} \tag{3.36}$$

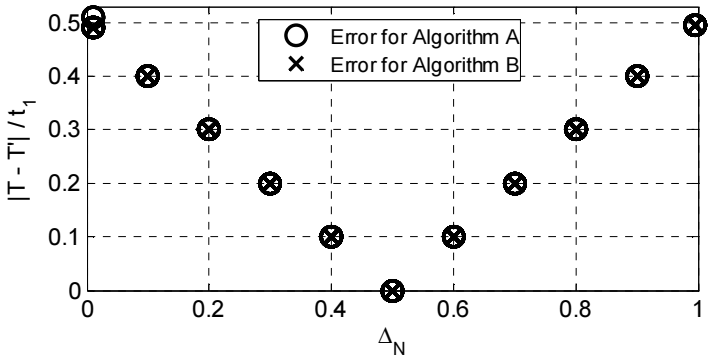


Fig. 3-22 Period approximation using algorithms *A* and *B*.

3.3.3 Two applications of the approximation algorithm

The first application of algorithm *A* estimates the frequency for a power grid, and the second one predicts the next maximum for the sunspots number. The first problem will demonstrate the robustness of algorithm *A* on noisy data with harmonic content, under real-time constraints, with estimations every 5 *ms*. The second problem concerns the prediction of the maximum of monthly observations of solar activity that exhibits period variations around a historical average of 11 years. Results for the power system application show that algorithm *A* is competitive when compared with traditional algorithms, such as Prony algorithms.

Frequency estimation in power systems

An ideal value for the frequency in North America is 60 ± 0.05 *Hz*, as defined by the North American Electric Reliability Council – NERC, which also defines over frequency or under frequency when the frequency 60 ± 1 *Hz*, and defines equipment damages for deviations of ± 4 *Hz* or more. On the other hand, the frequency of a power grid changes according to its inertia and the load on the grid that inevitably causes acceleration or deceleration of the generators. Changes in the frequency affect the reliability of the grid in various ways, including damage to generators and transformers collapse of the power grid, and overload of transmission lines.

Controlling the frequency in a power system requires a prior estimate of its current value. There are a variety of algorithms to make this estimation. Many of them use some facts about ideal sinusoidal functions, but such estimates do not always prove robust to noise, as in the case of switching-type transients. Another family of algorithms minimizes the square error between a sinusoidal function and the data, such as the one proposed in the IEEE standard 1057™-2007 for Digitizing Waveform Recorders [30]. The standard uses an iterative method to approximate the amplitude, phase, continuous component and frequency of a sinusoidal signal. Its results accurately approximate signals with low harmonic content, but the approximation degrades as the signal deviates from a perfect sine function. Other algorithms base their results on the Discrete Fourier Transform, Kalman Filters, and maximum likelihood estimates, among others.

The performance of algorithm *A* will be compared with the adaptive algorithm of Moore and the Prony method, as reported in [49]. The adaptive algorithm of Moore uses the fact that values for a sine and cosine decomposition of a periodic signal and their derivatives in continuous time can be used to compute the frequency. So the discrete approximation for each term can be used to estimate the frequency. The main source of error in this estimation corresponds to the decomposition process, which uses a FIR filters that themselves require an estimation of the

frequency. As a result, the method requires a closed loop with adaptive compensation to find the best estimate of the frequency based on the data for a quarter of period. In addition the algorithm requires pre-filtering so as to reduce the effect of harmonic content. On the other hand, the Prony method defines an estimate of the frequency by finding the parameters of a complex expression that minimizes the difference with the true sine and cosine decomposition of the data. The algorithm changes the minimization process into a linear fitting problem based on three consecutive computations of the orthogonal components of the signal. An additional pre-filtering step also requires reducing the influence of the harmonics and the noise in the data.

The evaluation of the algorithms implies running two tests: an amplitude modulation test and harmonic content test. In the modulation test, the amplitude of a pure sine function changes as indicated in Eq. 3.37, where the nominal amplitude A_0 equals 167 or 1 in pu (per unit); t_i represents the current time, and $A[t_i]$ the modulated amplitude at any given time t_i.

$$A[t_i] = \begin{cases} A_0 & t_i < 0 \\ A_0\left(1 + 0.5\sin\left(2\pi\frac{t_i}{2}\right)\right) & t_i \geq 0 \end{cases} \tag{3.37}$$

The second test assumes a sine function with amplitude A_0 and harmonic content of $0.1A_0$ for the second harmonic; $0.1A_0$ for the third harmonic; and $0.05A_0$ for the fifth harmonic. Every test was run for three frequency transients. First, a constant acceleration ramp; second, a constant ramp of deceleration; and finally, a sinusoidal transient, as shown in Eq. 3.38 through Eq. 3.40 respectively. These transients properly test any algorithm for approximating the fundamental frequency of the power system because their rate of change exceeds those of real events, as can be seen in [16].

$$f[t_i] = \begin{cases} f_0 & ,t_i < 0 \\ f_0 + t_i & ,0 \leq t_i \leq 1 \\ f_0 + 1 & ,t_i > 1 \end{cases} \tag{3.38}$$

$$f[t_i] = \begin{cases} f_0 & ,t_i < 0 \\ f_0 - t_i & ,0 \leq t_i \leq 1 \\ f_0 - 1 & ,t_i > 1 \end{cases} \tag{3.39}$$

$$f[t_i] = \begin{cases} f_0 & ,t_i < 0 \\ f_0 + \sin(2\pi t_i) & ,0 \leq t_i \leq 1 \\ f_0 & ,t_i > 1 \end{cases} \tag{3.40}$$

Results of the simulations for Adaptive and Prony algorithms can be found in [49]. The data in that paper were pre-conditioned using a second-order Butterworth low-pass filter with a cutoff frequency of 600 *Hz* and sampling frequency of 7,680 *Hz*. The estimates were compared with the actual frequency delayed 132 samples, which nearly corresponds to a fundamental period of the nominal sine function. On the other hand, the proposed algorithm in this section, with a sampling rate of 0.1 μs, produces an estimate every 5 *ms*. The reference frequency was defined as the actual frequency at the moment of the computation due to the fact the proposed algorithm does not have any associated filter delays. The noise in the signal was simulated by adding white noise of 0.1% off the function amplitude.

The real-time constraint restricts the application of the algorithm, so algorithm A was modified to obtain an appropriate algorithm A'. For instance, based on the knowledge of the signal and the accuracy obtained, the frequency can be safely searched in a relatively small interval of 55-65 *Hz*. In addition, the minimization of Δf uses the so-called zooming process. The zooming starts with a very coarse subsampling (e.g., once every 10,000 data points), then based on the minimization produced by this subset, runs another iteration with a finer subsampling (e.g., every two thousand data points), minimizing only around the candidate provided by the previous iteration. This procedure was done for every five hundred data points, every hundred, every twenty, every five, and finally for all data points. This modification reduces the computation time of algorithm A' by a factor of over 10^4.

Six experiments were run on Matlab using simulated data. These experiments provide a measure of quality for an approximation algorithm, based on precision and time elapsed to compute the approximation. The computation time was scaled by a factor of 25 for all the experiments, assuming a possible implementation of the algorithm on a chip such as a DSP. The first part of every experiment corresponds to a stationary condition at 60 Hz, and the relative error was always kept under 0.001%, due solely to the noise in the data. Algorithm A' closely tracks the (random) changes on the frequency in all simulations, as can be observed in Fig. 3-23 for: Sine + Amplitude modulation + noise + Ramp Up transient, and Fig. 3-24 for: Harmonics + noise + Sinusoidal transient.

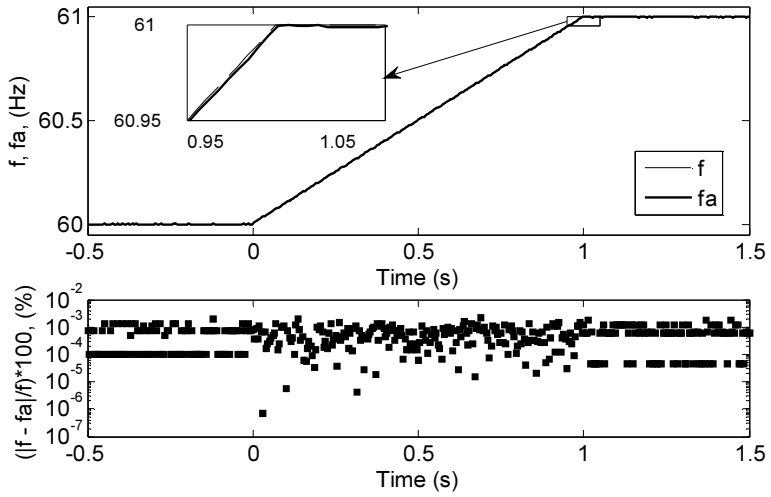


Fig. 3-23 Power grid application of algorithm A', ramp transient.

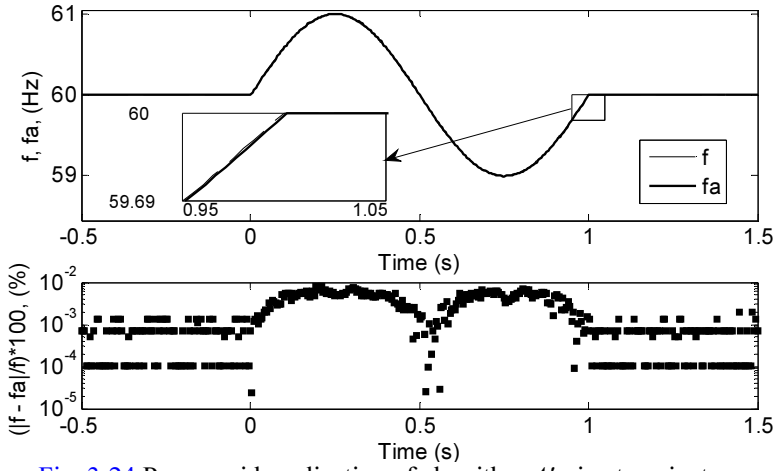


Fig. 3-24 Power grid application of algorithm *A'*, sine transient.

The comparison of three algorithms, as shown in Table 3-2, shows that algorithm *A* can track the changes in the frequency and also provides an accurate estimation of the frequency in a power grid. The quality of the estimate overcomes current algorithms, especially when the signal has harmonic content, because the minimization of Δf does not make assumptions about the signal shape, a common assumption in other algorithms. In addition, the algorithm displays robustness to noise in the data, so no filters are required, whereas current algorithms produce delays associated with the compulsory filtering process. The experiments also show that amplitude modulation imposes more restrictions for the algorithm *A* than for harmonic content.

Table 3-2 Average estimation errors for three algorithms.

Test Condition	Frequency Change	Average Estimation Error (%)		
		Adaptive Algorithm (from [49])	Prony Analysis (from [49])	Algorithm *A'*
Amplitude Modulation	Ramp up	10×10-4	5×10-4	16×10-4
	Ramp down	10×10-4	6×10-4	15×10-4
	Sinusoidal	25×10-4	22×10-4	90×10-4
Harmonics	Ramp up	43×10-4	7×10-4	4×10-4
	Ramp down	48×10-4	21×10-4	4×10-4
	Sinusoidal	34×10-4	28×10-4	20×10-4

Sunspot prediction

Magnetic poles in the sun reverse positions quasiperiodically. A quasiperiodic system, such as the sun, has an irregular periodicity caused by an unpredictable component that does not allow a precise measurement of its period. As a result of the quasiperiodicity of the sun, a number of its parameters consequently change in a quasiperiodic fashion, such as flares or the so called sunspots, which are areas with very intensive magnetic activity that appear darker compared with the surrounding area. The number of sunspots (*ssn*) has a strong correlation with the energy emanated by the sun reaching planet earth (which might cause communication problems at the maximum amplitude of that number). This number also has a relation with the weather on earth [45]. The *ssn* values can be used to predict the maximum activity of the sun because the linear relation between its value and other parameters such as 10.7 cm radio flux, total solar irradiance, and M- and X-class flares per month [27]. In addition, *ssn* has been collected for over three hundred years through daily observations.

The data set for *ssn* can be divided in Solar Cycles (SCs), defined as subsets between consecutive minima. The first one SC1 started in 1755, and the current one SC24 started officially in December 2008. The data set is considered reliable after 1900, so experiments typically use data from SC14, which started in 1901.

Algorithm *A*' will be used to predict the occurrence of the next maximum *ssn* based on monthly observations of the *ssn*. Thus, the predicted next maximum happens at the time of the current maximum plus the estimated next period. Therefore, the estimate of the amplitude for the maximum *ssn* per cycle, which is the main objective for current models for sunspots forecasts, will not be considered in this document.

The length of the next period SC24 may be estimated in four different ways as follows:

1. The average period of the known cycles since SC16;

2. The fundamental period based on the last two known cycles, using FFT, averaged with all estimations, also using FFT, since SC19;

3. The estimation of the current period, using algorithm *A'*, averaged with all the estimation, also using Δf, since SC19;

4. The estimation of the current cycle using a variation of the current algorithm, i.e, $\Delta f2$.

A second variation of algorithm *A* ($\Delta f2$) finds its motivation in the fact that the true magnetic period approaches 22 and not 11 years, as suggested by the statistically-based odd-even cycle models [33]. The data then has three sections instead of two, and the algorithm compares the first section with the third one, instead of the second one as done by algorithm *A*.

Although daily records of *ssn*, for prediction purposes the data is usually aggregated monthly and smoothed [68]. In addition, a normalization procedure made over the data eliminates influences on estimates caused by differences in maxima at each section during the computation of Δf. The normalization process moves each minimum *ssn* to a new minimum equal to zero. Then, data on the left and right of the new minimum change their values to have the same maximum, as shown in Fig. 3-25. That normalized data serves to estimate the last known period using FFT, Δf and $\Delta f2$. The four normalized cycles on the left have the same minimum and the same maximum.

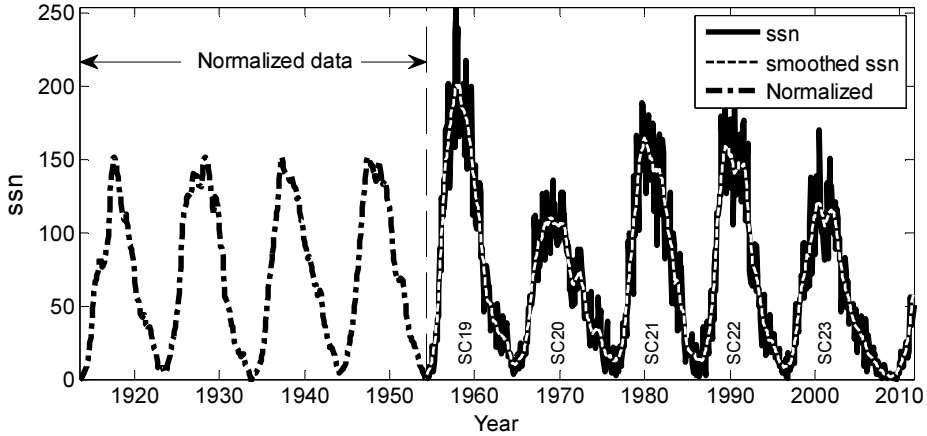


Fig. 3-25 Sunspot number.

Furthermore, the initial estimate of the period, T_0, for Δf and $\Delta f2$ equals the average of all true periods from SC2 through the last known maximum, which gives $T_0 \approx 130$ months to date. The search covers the range 3/4 through 5/4 of T_0.

The error in the prediction matches the absolute value of the difference in months between the true and the predicted period for each solar cycle, as shown in Fig. 3-26. Results in Fig. 3-26.a indicate the prediction for cycles between the maxima. The prediction for any other month equals the last known prediction. These monthly predictions enhance the influence of longer periods, over the required averaging process. Errors in Fig. 3-26.b come from time warping, with only one prediction per cycle, which causes differences in the predictions via the mean, FFT, and Δf.

The prediction using time warping does not cause large differences in the error for mean as predictor, as can be seen from the maximum error, which is close to 10 months in both cases. The maximum error using FFT reaches 16 months, and equals 10 months for Δf. The largest error corresponds to 18 months by $\Delta f2$, although errors for SC22 and SC23 using $\Delta f2$ fall under 6 months. Thus Δf or $\Delta f2$ provides a competitive prediction of the length of the next cycle (error under 3 months) based on pseudoperiodic data. Unfortunately, the decreasing trend of the error in Fig. 3-25 cannot be sustained; for example, if the maximum for the SC24 were in December 2012, then the prediction error would be as large as 20 months in the best case scenario, given by $\Delta f2$. The extreme length of SC23 causes all these rather substantial errors.

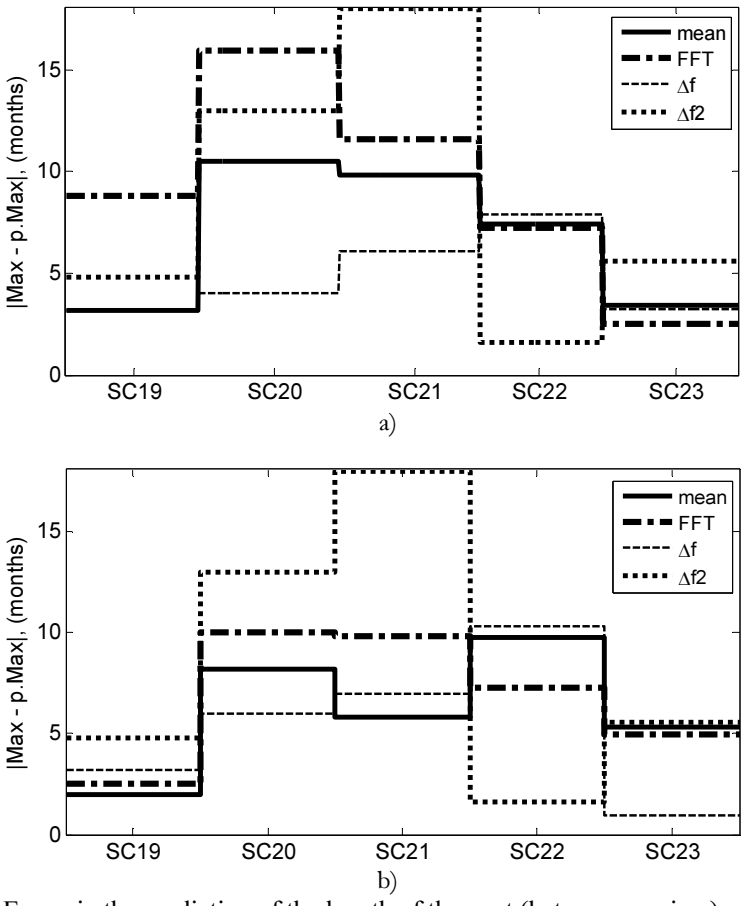


Fig. 3-26 Errors in the prediction of the length of the next (between maxima) sunspot cycle.

In summary, two algorithms for period estimation were presented, as well as two applications to power grid frequency estimation and sunspot period estimation. This estimation is *not* limited to sinusoidal functions, but can be used over a wider family of (possibly discontinuous, but mostly differentiable) signals. Such estimates enable the use of many algorithms for signal reconstruction once the period is known. The error in the estimation for noiseless and periodic signals does not depend on the shape of the signal, its amplitude, or the phase shift, but only on the sampling rate and data. The algorithms have very low computational requirements (quadratic $O(n^2)$ in the number of data samples n), which makes it feasible to run applications that require *real-time* results, such as frequency estimation. Algorithm A' tracks the frequency changes and the errors to within less than 2×10^{-3} %, which represents a twofold improvement over the state of the art for the case of harmonic content.

There are no apparent advantages to using algorithm A or its modification to predict the length of a solar cycle. The predictions are not significantly better than the ones provided by the simple mean of previous known cycles or a more elaborated and accepted estimation given by the Discrete Fourier Transform, because the data does not meet the assumptions of the approximation algorithm. However, this application shows that the algorithms produce competitive answers even in cases where the target signal is not truly periodic, or for noisy data. Given the fact that periodic signals occur abundantly in nature, this algorithm may find several other applications.

Some problems of interest remain. For example, it appears that the zoom-in technique used in algorithm A' could be refined to produce better results in the power grid application. A second problem of interest is to explore a variation of algorithm A by comparing more than one pair of sections (as was done with sun-spots) to improve the accuracy or efficiency of the algorithm.

4. Control using emulated emotions

This chapter combines concepts from previous chapters into a single emotion-based control architecture. The main difference between the emotion-based control architecture and a classical controller regards the decision-making process. Thus, an emotional controller bases its decisions on emulated human emotions, whereas a classical controller uses a mathematical law that can fairly be seen as the logic process carried out in the brain. The use of emulated emotions upgrades the capabilities of a control system and allows designers to consider solving complex problems beyond the traditional electromechanical field, such as biology, robotics, and games, among others.

This chapter has four sections. Section One defines human emotions and presents some computational models of emotions, which can simulate the decision-making process in an agent working as a controller. Section One concludes by choosing the Circumplex Model of Affect as the model to measure the emotional states coming from a control system. Due to the extensive literature about emotions, this section focuses its analysis on models and definitions that lead us to the Circumplex Model of Affect.

Section Two details the emotion-based control architecture. The definition of the controller starts with a review of the phase plane as a mathematical tool to analyze dynamical systems. Then, any change in the dynamic may be seen as the trajectory of an agent across a landscape in the phase plane. This trajectory is a powerful analogy for an emotion-based controller. In addition, the concept of reference model in Chapter Two forms an idealization of the landscape inside the mind of the agent. The difference between the plant state and the idealization in the agent generates an emotional state, and that emotional state helps in defining the actuating signal at each instant in time.

Section Three consists of a number of simulations attesting to the performance of the emotion-based controller. These experiments begin with linear systems and finish with a nonlinear system. In addition, each experiment compares the performance of a traditional PID controller with the emotion-based controller. The results show some of the advantages coming from using the proposed controller instead of a classical controller. Section Four, the final one, presents a refined version of the emotion-based control architecture. The main advantage of this upgrade architecture is an experimental assurance of stability, given by an exponential decrease in the difference between the reference dynamic and the plant dynamic for the examples in this thesis.

4.1 Computational models of emotions

The etymology of the English word emotion, according to the *Oxford English Dictionary*, derives from the Latin root meaning “of action”. Its French origin means “to move out”. The definition states that emotion is “any agitation or disturbance of mind, feeling, passion; any vehement or excited mental state.” That definition serves to introduce one of the major problems related with

emotion studies: the definition of the term itself. Nearly every author uses his or her own definition. One such definitions by Franklin E. Payne states that emotion "is the momentary (acute) and ongoing (chronic, continuous) disturbance within the mind (soul, spirit) caused by the discrepancy between perceived reality and one's desires." [52] This definition was chosen among many others because it may be used to emulate emotions in agents, given the simplicity that may allow applications in real-time.

A further definition for emotion, taken from neuroscience and quite useful for the present purpose states that emotions enhance the decision-making process. Decision making often occurs in the face of uncertainty about whether one's choices will lead to benefit or harm. Emotions aid the decision making process by means of bodily states that appear during the deliberation of future consequences, coloring the behaviors as being advantageous or disadvantageous [15]. Emotions in humans assist the reasoning process and also facilitate the processing of knowledge. In addition, emotions serve as a regulatory system, help in defining behaviors, and facilitate communications and social interaction. Emotions also pertain to attention, memory, learning, perception, creativity, and adaptation, among others. In summary, emotions are indispensable to every aspect of human life.

A definition of emotion for artificial agents does not differ greatly from the above mentioned definitions for humans. For instance, in [60], the author asserts that emotions can be usefully defined as states elicited by rewards and punishments. According to [8], emotions serve as processes designed to achieve, maintain, or change the organism's relationship with some aspect of its environment. In addition, emotions in agents are important for a number of purposes, including action selection, adaptation, self model, communication, learning, memory control, sensory integration, alarm mechanisms, strategic processing, attentional focus, motivation, and goal management
[65]. On the other hand, Scherer [63] states that an emotional state for an agent is a particular instantiation of operating parameters for the complete set of capabilities (physical or cognitive) it possesses, a viewpoint of significant interest for agent and robotic applications.

Once a set of definitions of emotions for humans and agents has been given what follows is a review of some of the most elaborated theories of emotion. The first theory, proposed by Joseph E. LeDoux, includes the so called circuit of fear emotion in [34]. LeDoux suggests that affective phenomena fall into three categories: *evaluation, expression,* and *experience*. Emotional evaluation and experience are studied as separate processes with distinct underlying brain substrates. As a further consequence, evaluative processes may be seen as pre-conscious mechanisms. Emotions, for LeDoux, in brief consist of a conscious experience as well as physiological and neurological reactions and voluntary and involuntary behaviors.

A second theory is called Damasio's somatic marker theory [13]. Damasio differentiates between subjective feelings and the emotion itself. Emotions can be seen as the combination of a mental evaluative process, simple or complex, with dispositional responses to that process resulting in an emotional body state. The prefrontal cortex has a central role in this theory. It receives inputs from both sensory cortices and limbic structures. The prefrontal cortex establishes contingencies between somatic, visceral feedback and a given situation or stimulus configuration. This relation among brain structures forms a somatic marker that can regulate behaviors according to previous experiences. Damasio sees emotions as "bioregulatory reactions that aim at promoting, directly or indirectly, the sort of physiological states that secure not just survival, but survival regulated into the range that we identify with well-being."

A third theory is proposed by Jean-Marc Fellous. He differentiates emotional expressions for communication from emotions as a mechanism for organizing behaviors, such as action selection, attention, and learning. Arbid and Fellous use four levels to explain emotions in humans: reactive, routine, reflective-deliberative, and reflective-meta-management [2]. In addition, they state that an animal comes with a set of basic drives that provide the motivation for behavior. Most of the neural circuitry underlying these drives involves specific nuclei of the hypothalamus. The hypothalamus sends information to the amygdala and the orbitofrontal cortex. The amygdala can influence cortical areas via feedback from proprioceptive, visceral or hormonal signals, via projections to arousal networks, and through interaction with the medial prefrontal cortex. The prefrontal cortex, in turn, sends distinct projections back to several regions of amygdala.

A not exhaustive list of theories of emotion from psychology can be seen in Table 4-1. They, as well as the previous three theories, may be used to generate computational models of emotions. Each theory has its main author, publication year, a reference, and an explanation of the basic idea.

Table 4-1 Other theories of emotion.

Author	Main idea	Reference
Aaron Sloman, CogAff Architecture schema, 1990’	Emotions involve complex processes produced by the interactions between the motives and beliefs of an agent. Motives help the agent to decide upon the course of action. Beliefs are real or imagined perceptions that a motive will be satisfied. Different actions may result depending upon the strength of the belief.	“Beyond Shallow Models of Emotion”, 2001, pp 177–198
Ekman, Basic emotions theory, 1980’	An emotion consists of a set of sensations experienced briefly about something that matters to the person experiencing it. Emotions happen without the experience being chosen consciously; a subconscious appraisal process triggers the emotion.	“Facial Action Coding System (FACS)” Consulting, Psychologists Press, 1979
Ortony, Clore, Cullins. OCC model, 1988	Emotions are valenced reactions to events, agents or objects. These events, agents or objects are appraised according to an individual’s goals, standards and attitudes. OCC was developed with the express aim to implement it in a computer.	The Cognitive Structure of Emotions. Cambridge, University Press, 1988
Dörner, et. al., Psi Theory of Human Action Control, 2000’	Dörner has developed a computer model that integrates cognitive, motivational, and emotional processes. Within the framework of PSI he developed the model “EmoRegul” which concerns emotional processes. The central mechanisms of emotion regulation in PSI are the motivators for certainty and competence.	The Mathematics of Emotions. 2003
Klaus Scherer, CPM The component Process Model, 1990’	Five functionally defined subsystems are involved in emotional processes. An information-processing subsystem evaluates the stimulus through perception, memory, prediction and evaluation of available information.	Emotion Serves to Decouple Stimulus and Response. Scherer, 1994, p. 127.
Frijda, “Action tendency” Theory, 1990’	Emotions evolved in order to improve the adaptation of the organism in its environment by providing it with fast harm avoidance. Frijda develops a definition of the “emotion process” that begins with stimulus, transitions through various phases of appraisal and finally results in an action tendency.	The Laws of Emotion. American Psychologist, 1988
Aaron Ben-Ze’ev, Intentionality and feelings, 2000’	Emotions result from evaluation of changes in the environment in order to reach an ideal state. Change detection comes from the comparisons between an alternative situation and a reference, so emotions require imagination. The way to adjust values is by social comparisons.	“The Subtlety of Emotions” MIT, 2000

Emotions in control systems

The importance of emotions in everyday life is readily apparent, but trying to implement them in machines may seem hard or even unethical. Scientists from all around the world have seen that this is not only possible, but important for a machine with the ability to plan, or to face unknown environments. This is the case with certain robots, artifacts, virtual agents, and specially stressed control algorithms; they should be adapted to support the uncertainty of specific conditions of the controlled process or faults inside the system.

Implementing emotions in machines gives scientists the possibility of testing emotional theories, improving the interaction between human and machines, studding things that may be consider unethical, and addressing the question regarding the necessity of real or pseudo-emotions in machines [17]. The implementation of emotions in machines must begin by observing the agent's environment for clues to calibrate the quantity, valence, arousal, and complexity of each emotion. For instance *anger* may serve to stop a current situation; *disgust* to correct something that does not contribute to need satisfaction; *fear* defends the agent from threats; *happiness* rewards when a goal is achieved and reinforces a behavior; *surprise* triggers exploration in the presence of a novel object, and *sadness* signifies that a need has not been satisfied, as stated in [3], [37], and [44].

Researchers in cognitive science and related areas have proposed several computational models of emotions; these may be used to demonstrate or refute theories about emotions. For example, one of the most complete models, based on neuroscience, studies the interaction between attention and memory [17]. Other models represent the cognitive process of decision making, especially applied to robotics or virtual agents [24], [26]. Two of the latter have been used to control dynamical systems. The first model applies neural nets to the implementation of a control law [79]. The second computational model was not originally designed for control; its authors sought to reproduce the emotional learning occurring in the amygdala, and the relation of the amygdala with the orbitofrontal cortex [46]. The equations of the model represent the connections among brain structures. The work of each component is reduced to comparisons and to the four basic arithmetic operations. These features of the model make it a practical algorithm for real-time applications.

The computational model of emotional learning in the amygdala, as reported in [46] comprises the amygdala, orbitofrontal cortex, sensorial cortex, thalamus and a rewarding signal. This last component does not have a specific correlate in human anatomy. The amygdala reacts to stimuli such as pain or certain odors, and can establish new associations; in other words, it learns from stimuli previously considered neutral according to the associative learning paradigm. In the model, the output of the amygdala never decreases its value. Therefore, once it learns it never forgets. Thus, the orbitofrontal cortex inhibits the emotional reactions if they are not adequate. The sensory cortex acquires and directs the output stimuli, and the closely associated thalamus simulates a fast system of signal classification.

The second computational model of emotions has inspired a control strategy called BELBIC, which stands for Brain Emotional Learning Based Intelligent Controller [35]. The adaptation capacity of that controller was verified through a variety of applications, such as a heat exchanger, a power system, a spatial vehicle, a synchronous motor, a washing machine, and a variable reluctance motor, among others. In particular three applications will be analyzed: the Duffing and Van der Pol oscillators, and a balancing system, as reported in [40]. Comparisons of control performance for these three systems using BELBIC are shown in Table 4-2, where the systems have output y_p, required output y_r, position x, balancing angle θc, and finally error e, sensorial input

S, reward signal *Rew*, and constants α_a and α_0 to define the learning speed of the model. A designer of a BELBIC controller should adjust, by trial and error, from six through eight parameters to setup the controller, as shown in the table. Reward signal *Rew* may be seen as the output of a classical controller, whereas *S* signal joints desired outputs and actual outputs.

Table 4-2 Adjustable parameters of a BELBIC controller.

	Van der Pol	Duffing	Balancing
S	$[2y_p \quad 3y_r]^T$	$[2y_p \quad 5y_r]^T$	$[10x \quad 1000e \quad 500\theta_c]^T$
Rew	$650e + 50\int e$	$100e + 20\dot{e}$	$1000(4e + 2\int e + 5\dot{e})$
α_a	1×10^{-6}	1×10^{-10}	1×10^{-5}
α_o	2×10^{-3}	3×10^{-2}	7×10^{-6}

Once the tuning process for the BELBIC controller in Table 4-2 is completed, the controller adapts its behavior to changes in the reference signal and added noise, but only if those changes and noise were available during the tuning process. Thus, the resulting behavior of the controller to unknown conditions results unpredictable. Another pitfall regards the control signal magnitude. The control signal peaks at 700 N for the first system, 200 N for the second and 900 V for the third one. These values cannot be obtained using conventional laboratories. Thus, the simulations are only idealizations. These past difficulties indicate that there is still work to do in order to implement a practical controller based on human emotions.

Selection of a computational model of emotion

This section exposits the selection of a computational model of emotions in order to use it as the core of the emotion-based control architecture in the next section. A feasible classification of models of emotions distinguishes between discrete and continuous computational models. Discrete models create sets of basic or principal emotions. For instance, authors in [10] and

[65] define exclusively anger and fear to define searching algorithms and also to generate autonomy in control systems. Discrete models have inspired several engineering applications, and some of its theorists include Robert Plutchik, Paul Ekman, and Nico Frijda [64]. Their models have eight, six, and six basic emotions, respectively.

As opposed to discrete models, there are mainly two continuous models of emotions: one comes from the appraisal theory, and the other comes from the dimensional theory of affection. Psychologists Richard Lazarus and Craig Smith are two of the main contributors to the dimensional theory. They seek to define emotions as the result of evaluating a situation. The definition of emotion according to appraisal theory takes into account the suddenness of a situation, the importance of the goal, the control that can be gained, and the available energy, among other aspects. Some proposals use from five to sixteen variables [37]. Lately, a lot of work related to the discrete theory has been focused on finding support for this theory from neuroscience, and also from dynamical systems theory [62].

The second point of view regarding the continuous theories of emotions is known as the dimensional theory. This is precisely the theory selected to emulate emotions in the next section. The dimensional theory was proposed by Russell in 1980, who measured emotional states in human beings by means of two variables, valence and arousal. Valence indicates the emotional

experience from positive states such as happiness, to the negative ones such as anger. Arousal ranges from inactive states such as sleepiness, to very active ones such as excitement [61].

When Russell measured the emotional states, he represented them on a Cartesian coordinate system with the valence on the horizontal axis and the arousal on the vertical axis. Therefore, emotions occupied a circular region, giving the model its name, the *Circumflex Model of Affect*. For instance Happiness in Fig. 4-1 is the combination of a high positive valence and medium arousal level; sadness is high negative in valence and low in arousal. Although this theory was proposed by a psychologist, in the last years, it has found support from neuroscience [56].

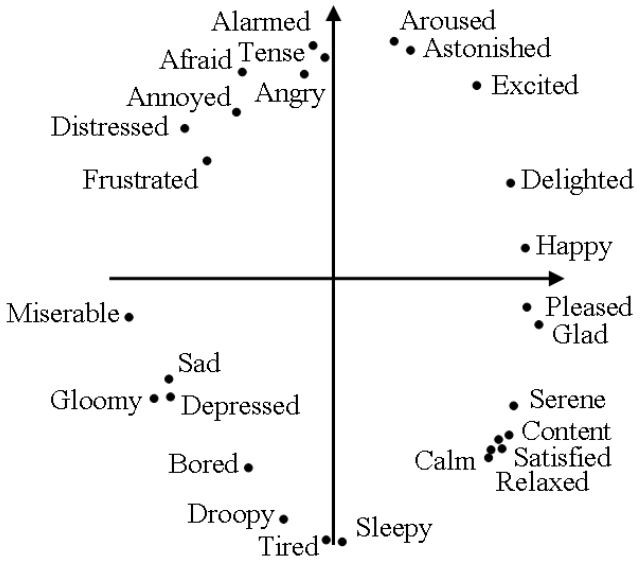


Fig. 4-1 Circumflex model of emotions. Russell 1980.

Based on the Circumflex Model of Affect, many contributions have been made to medicine, psychology, language analysis, and music, among other fields. Recent works include the measurement of emotional states in order to change screen colors on a mobile phone [14], and the measurement of emotional states in a player to alter the characteristics of a video game [53]. Another work, in robotics, is the design and construction of EDDIE (an Emotion-Display with Dynamic Intuitive Expressions). This robot regulates the action of servomotors, which control the movement of ears, eyes and mouth in the generation of emotional expressions [70]. Based on the analysis of this application, the authors concluded that the dimensional theory can be used in building a control strategy based on emotions.

4.2 Proposed emotion-based control architecture

The plants to be controlled in this section will be described by linear and nonlinear differential equations in a single state variable. However, farther analysis can generalize the analysis to systems with multiple inputs and outputs. Thus, one way to analyze the dynamic of such a plant is the phase plane [20], which requires the description of the system by means of two equations in two variables. The first variable corresponds to the output of the system itself, for instance a position, a force, a concentration or any other measurable physical parameter. The second variable may be the rate of change for this output. An example of a nonlinear plant is written in Eq. 4.1 and Eq. 4.2. This plant has output x_1, rate of change x_2, and input u.

$$\frac{d^2x_1(t)}{dt^2}+\frac{dx_1(t)}{dt}+x_1(t)\left(x_1^2(t)-0{,}2\right)=u(t) \tag{4.1}$$

$$\begin{cases}\dfrac{dx_1(t)}{dt}=x_2(t)\\ \dfrac{dx_2(t)}{dt}=-x_1(t)(x_1^2(t)-0.2)-x_2(t)+u(t)\end{cases} \tag{4.2}$$

The phase plane may be seen as a vector field, as shown in Fig. 4-2. It associates a vector to every pair (x_1, x_2) in Eq. 4.2. The horizontal component of each vector is x_1 and the vertical component is x_2. Each vector indicates the direction of the system's motion, so ($x_1(t+\Delta t)$, $x_2(t+\Delta t)$) can be predicted by current values of $x_1(t)$ and $x_2(t)$ in addition to the direction of the arrows. On the other hand, the controller seeks to change the direction of the system through an actuating signal u so as to reach a desired state at each instant in time. Left unperturbed, the dynamics of the system will take the plant to an attractor, or else will exhibit divergent behavior depending on the nature of the system. Different conditions lead the system to different attractors or global states. Fig. 4-2 shows an example of the dynamic of the plant in Eq. 4.2 where $u = 0$, which leads the system to one of two different attractors after 30 s. The phase plane was chosen to define the new control algorithm because the motion of the system following the direction given by the arrows may be seen as the motion of an agent over a landscape.

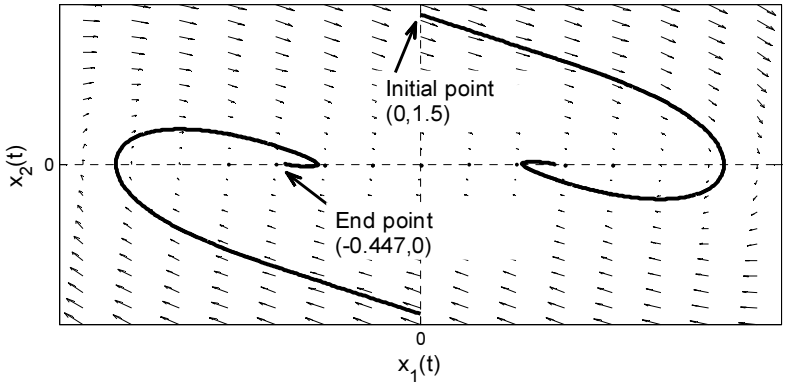


Fig. 4-2 Motion of states of a nonlinear system in the phase plane.

A traditional control algorithm makes decisions in order to transform x_1 into the reference value $r(t)$, as shown in Chapter One, through the external intervention of the input u. A controller therefore has only three possible decisions: increase, decrease or maintain the current u value. The algorithm's difficulty increases by virtue of the fact that it is necessary to determine when, how much and how long it will take to the system to reach the reference r.

The vector field of a phase plane facilitates the definition of emotions for a dynamical system (DS). The motion of a system downstream a vector field may be seen as the motion of an agent over a landscape. Also important, the decisions made by the agent (the controller) change the landscape instantly. This idea about instant changes on a landscape does not have a counter part in nature, but it facilitates the definition of emotions. So, the difference between a current landscape and an ideal one serves as input data to determine an emotional state. For instance, if a new landscape resulting from a decision moves an agent toward its goal, this event would actually generate a positive emotional state in the agent. The use of the phase plane to define the emotional states of an agent forms the basis for one of this document's significant novel ideas.

The reference model

The reference model (RM for short) is a dynamical system used to define an ideal set of dynamics for a plant. An ideal control system will ensure that the plant closely follows the RM. Therefore any deviation of the plant from the model should be addressed by the emotional controller in such a way that the distance between the dynamic of the plant and that of the RM tends to zero. The controller has a single tool to lead the entire dynamic: the prediction of the dynamic for the plant and the model. This anticipation establishes emotional states which in turn serve to make decisions about how to lead the plant.

The dynamic of the RM may be seen as the knowledge of an agent (the controller) about the plant, which may be translated into desires about future states of the system, akin to expectations in humans. That knowledge can be used to define ideal scenarios to contrast them with current and future states of the plant. The agent, which is the brain of the system, responds emotionally to any difference between ideal and anticipated scenarios for the plant. Then, as in humans, every emotional state influences the actions that will be performed in order to reduce the difference between anticipated and actual scenarios. For instance, a match between anticipated and actual scenarios triggers calmness, which may be interpreted as the success of previous actions on the system. Thus, the same actions can be performed again to continue getting the same good results. On the contrary, a strong difference between scenarios (for example if an employee receives a pink slip while he expects a promotion) should instantly evoke the implementation of a radically new control plan.

The reference r represents an ideal state for RM and system, whereas the RM defines the entire dynamic for the transition from an initial state to the ideal state dictated by r. In cognitive terms r for a human being may be to get a promotion, to get married, or to take a vacation, whereas the RM corresponds to the whole set of actions, including when, where, and how to reach that ideal state r. A difference between an agent in this chapter and its human counterpart is that r is superimposed externally to the control system. In a human, however, r may emanate from a higher cognitive level within the same person, as the Perceptual Control Theory explains [57].

In terms of the phase plane concept, the dynamic defined by the RM can be seen as a landscape. This landscape dictates the motion for agent and system, following the path indicated by the arrows on the phase plane. The controller is an agent moving down the landscape with the same coordinates as the plant. Therefore, the controller can be seen as the brain of an agent, whereas the plant is the body of that agent. The agent's trajectory always finishes in an attractor. The simplest type of an attractor, a point attractor, corresponds to the lowest point in the landscape. Thus, there must be at least one attractor on the phase plane representing the dynamic of the RM located at the reference r.

The changes in a DS input define a breakpoint in the landscape analogy, since each input value defines a new landscape. If r remains constant, then an adequate definition of the RM sets up a dynamic with a global attractor placed at $(r,0)$ on the phase plane. On the other hand, a variable r means that the attractor, and the whole dynamic of the RM, changes as time passes. The landscape analogy increases its complexity when it becomes apparent that the control system has two dynamical systems, each with its own, possibly different, input. Therefore, there are actually two landscapes: one that shapes the inside world of the agent (the RM) and another one that forms the outside world, which is the dynamic of the plant.

If r remains constant, the work of the agent consists of defining an input u for the plant in such a way that the location of the agent (controller and plant, in other words brain plus body) matches the

location of the RM regardless of the difference in landscapes (dynamics). Thus, the brain of the agent makes decisions, emotional decisions in this case, in order to change the plant's associated landscape appropriately, changing the input u so as to draw the current state (location) of the plant to the location of the RM.

The reference in Fig. 4-3 is a constant, $r = 1$, represented by an x symbol. The upper row in that figure shows the dynamic of the RM, and the lower row corresponds to the plant. Every input u generates a repelling dynamic in the plant, signifying that the system is unstable. The attractor of the RM was properly placed to match the reference after 20 s, so that the initial state (0,0) turns into (r,0), as shown by the o symbol in Fig. 4-3. Each column in the figure represents an instant in time. In addition, each column shows a different actuating signal u. The final location of the agent allows visualization of the unstable equilibrium point at which the plant matches the RM.

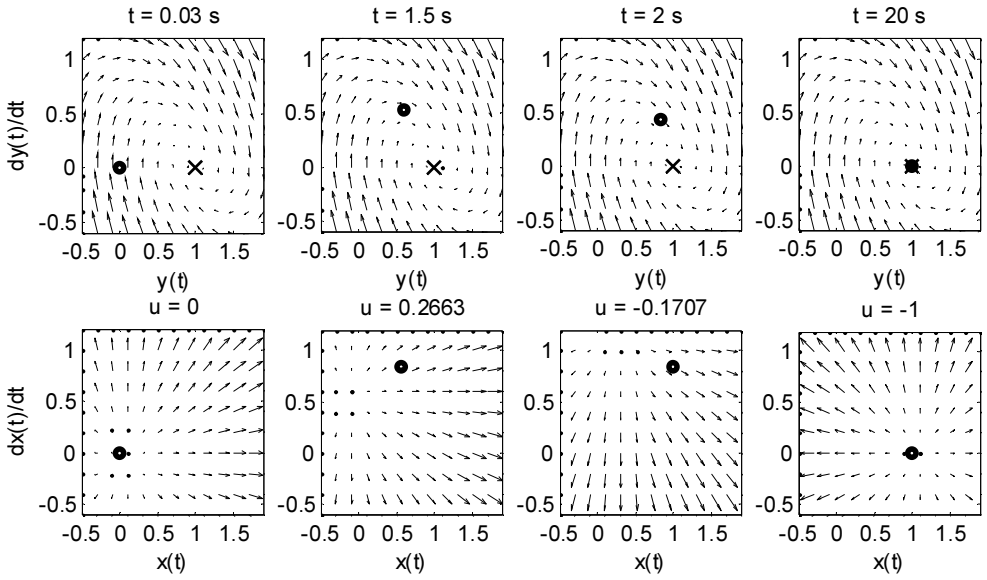


Fig. 4-3 Inner model (upper) and outside world (lower) for the agent.

The example in Fig. 4-3 shows an unstable system following the RM. In this example, a classical PI controller uses the distance between x and the instant location of the RM to feed the plant. The RM may be a linear dynamical system, such as $H(s) = 1/(s^2 + s + 1)$, and the plant is $G(s) = 1/(s - 1)$. In addition, the traditional PI controller has gains $Kp = 2$, and $Ki = 2$.

A reference r can be smoothed using a low pass filter in order to avoid instantaneous changes, which may be translated into instabilities on the whole DS. Such a filter may be considered as a RM. There are at least two models that fulfill traditional constraints for DS (such as rise time and overshoot), namely Bessel and ITAE (for Integral Time Absolute Error) [38]. The first has zero overshoot, and the second optimizes energy consumption. Once the model has been chosen using the table in [38], the work concludes by defining the settling time for the RM.

Dynamical systems in Fig. 4-4 are the same ones simulated in Fig. 4-3. The initial condition for both system equals (0,0), and the simulation lasts 6 s. The two vector fields are omitted intentionally in the figure to improve clarity. The RM knows the location of the target value, namely the reference r, whereas the system pursues the RM.

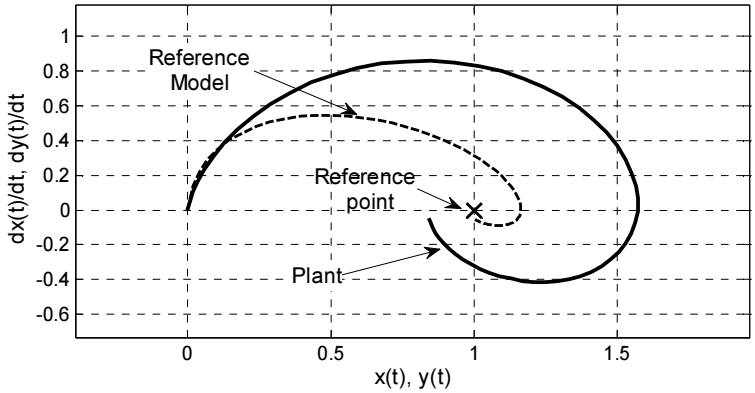


Fig. 4-4 Plant dynamic going behind the RM.

Once the continuous nature of the process is defined, for instance by means of Eq. 4.1 and Eq. 4.2, it makes sense to choose a geometric approach to measure the distance from the plant to the RM. Fig. 4-5 depicts a snapshot of the dynamic at a certain time *t*. Ideally, the plant should advance towards the RM along the line of sight between RM and the current state of the system. The coordinates (*x*, *dx*/*dt*) describes the location of the system in Fig. 4-5, whereas RM represents the ideal output, with coordinates (*y*, *dy*/*dt*), for the whole system at every instant *t*.

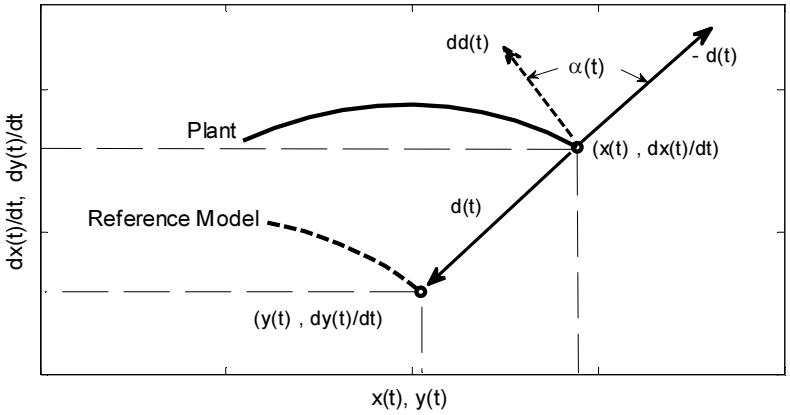


Fig. 4-5 Definition of the α angle.

The vector *d* is defined in Eq. 4.3. This vector corresponds to the difference between the position vector of RM and the system behavior in the phase plane. The ideal magnitude for *d* equals zero, and the way to assess how much it should increase or decrease is by computing a second vector *dd*, given by Eq. 4.4.

$$\overrightarrow{d(t)} = \left(y(t) - x(t)\right)a\vec{x} + \left(\frac{dy(t)}{dt} - \frac{dx(t)}{dt}\right)a\vec{y} \tag{4.3}$$

$$\overrightarrow{dd(t)} = \left(\frac{dy(t)}{dt} - \frac{dx(t)}{dt}\right)a\vec{x} + \left(\frac{d^2y(t)}{dt^2} - \frac{d^2x(t)}{dt^2}\right)a\vec{y} \tag{4.4}$$

Now, vectors *d* and *dd* are normalized to focus the attention on the angle between them. For instance, an ideal situation happens when *d* decreases and *d* has the direction exactly opposite to *dd*, because this means that not only does the error goes to zero but also that there will be no oscillations. Therefore, the final goal of a controller consists of making the angle between these

vectors equal to zero. On the other hand, in the worst case, this angle equals π rad. The definition of α in Eq. 4.5, by comparing the unitary vectors $-U_d$ and U_{dd}, is a mathematical trick to relate the definition of emotions given by the Circumplex Model of Emotion and the dynamic of a control system.

$$\alpha(t) = \cos^{-1}\left(-U_d(t) \bullet U_{dd}(t)\right) \tag{4.5}$$

Definition of emotional states for the emotion-based controller

Generally, complex tasks will require lower emotional arousal in order to achieve performance equivalent to that of a simple task. This fact is known as the Yerkes-Dodson law [23]. Therefore, $\alpha = 0$ can be labeled as the emotional state calmness.

Since it would be too ambitious to attempt to define at first a model with the full range of emotions felt by humans, only an adequate subset that helps to perform control tasks is considered. In particular, emotional disorders such as stress, phobias, manias, and so on are excluded. Likewise, more complicated behaviors such as experiencing several emotions at the same time, or combining them will be excluded.

One way to visualize an emotional state in a controller corresponds to imagine this state as being experienced by a person carrying out a control task. A person may experience happiness when encountering positive results after experiencing challenges; on the other hand, a person may experience fear or anger with the opposite outcome. These emotions can be characterized in terms of the emotional state α as shown in Table 4-3. This table includes the definition of the most representative emotional states; from the most positive (calm) through the most negative (anger). For simplicity, we assume that the emotional state of the controller depends exclusively on the instantaneous value of α.

Table 4-3 Emotional State Definition

Emotion	Definition
Calm	Zero α has been reached, or the angle is small, ($\alpha \sim 0$).
Satisfaction	The goal, $\alpha = 0$, has not been reached yet, but it is within a permissible angle, ($\alpha = \pi/6$).
Happiness	The distance between system and RM at least decreases. The magnitude of d will eventually be zero ($\alpha = \pi/3$).
Excitement	The distance d does not increase nor decrease, but maintains the ratio between RM and system, so that it oscillates around RM ($\alpha = \pi/2$).
Fear	The ratio between RM and the system increases, so ($\alpha = 3\pi/4$).
Anger	This is the worst case. The system moves away from RM, so it appears that the goal, $\alpha = 0$, would never be achieved, ($\alpha = \pi$).

The emulated emotion of the control system corresponds to the emotional state α. This emotional state α will be used to make decisions. Those decisions drive the system by defining the input u to the plant. Thus, the plant follows a desired behavior given by the reference model.

Proposed emotion-based control architecture

The goal of this section is to describe in detail the proposed emotional component of the controller, as shown in Fig. 4-6. The control architecture includes an additional variable called mood. Mood is

defined in psychology as the fixation of an emotion that has been felt for a period of time. For instance, if the mood is anger, it can turn gradually from negative to positive after experiencing a positive emotion for some time. A simplified way to implement a mood module in the model is by means of a low pass filter because it does not allow the emotional states to change instantly. The mood module uses the emotional state α to stabilize possible fluctuations of the system, especially the ones caused by noise or faults.

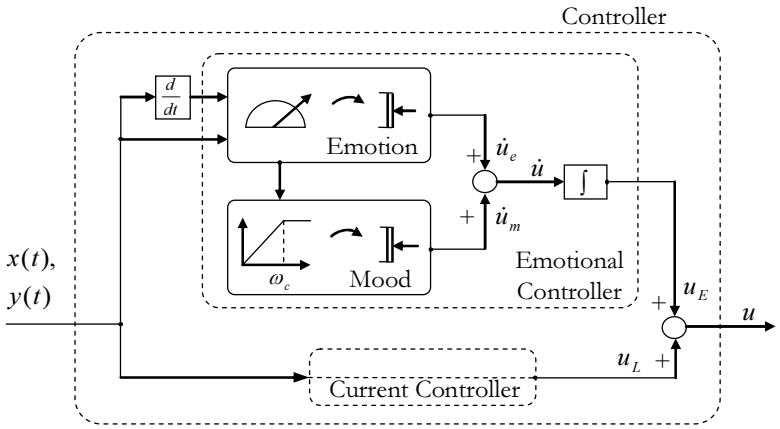


Fig. 4-6 Control algorithm architecture.

The emotional controller also includes a numerical integrator, which sums up the emulated emotion and mood outputs. The output of the integrator is u_E in Fig. 4-6. When the emotional state α matches calmness then $\dot{u}_e$ equals zero. Therefore, the integral holds u_E on the previous value, which makes sense, because the decision is to feed the plant using an actuating signal that has proved effective in leading the system to a desired dynamic. If $\dot{u}_e$ does not match zero the integral increases or decreases according to the current emotional state α. As a result the value of u changes looking to compensate any deviation from the reference model.

The last, and optional, component of the architecture is a classical controller, as shown in Fig. 4-6. This classical controller makes decisions to regulate a system based on the difference between reference y and output x, which may be associated with the logical processes in humans. The mixture of logic and emotion results in a powerful tool for defining the actuating signal by means of adaptive decisions. The relationship between the logical and the emotional components can be as simple as an additive algorithm, as shown in Fig. 4-6. However, this control can use other relationships, and can be replaced by a soft algorithm, such as a fuzzy system.

4.3 Validation of the emotion-based controller

This section has two parts. The first presents a series of simulations to determine the most representative emotional states α by tuning a traditional PID controller that acts on an unstable plant. The second part shows the result of simulations using the new emotion-based controller to control some dynamical systems. This section presents an initial design of the controller, whereas the following section describes a refinement of the same controller.

Simulation of some emotional states α

All simulations in this section have the following parameters in common: plant $H(s) = 1/(s - 1)$, reference $r = 1$, initial conditions $x(0) = 0$ and $dx(0)/dt = 0$, reference model $RM(s) = 1/(s^2 + 2s + 1)$, and

unitary feedback. The values of the traditional PID controller gains differ from one simulation to the next. Fig. 4-7 through Fig. 4-9 each has four panels. The graphs on the left show one emotion and those on the right show another one. The upper plot illustrates the behavior in time, whereas the lower illustrates the emotional state α. The radius in the lower plots corresponds to the distance between y and x, in other words, the traditional definition of error in control.

The emotional state anger, shown on the left of Fig. 4-7, begins at 3 s, at which time the error increases to reach 200 in just 15 s. The gains of the controller were $Kp = 0.2$, $Ki = 0.2$, and $Kd = 0$. The plot on the right begins at 0 s, but fear is experienced only after 2 s. The rate of change of d for fear is smaller than in the case of anger, and the highest difference between the plant and the RM barely reaches 1.25 at 4 s. Even when the radius on the polar graph changes, the emotional state α remains constant. The controller for this emotional state α has gains $Kp = 0.9$, $Ki = 0.3$, and $Kd = 0$.

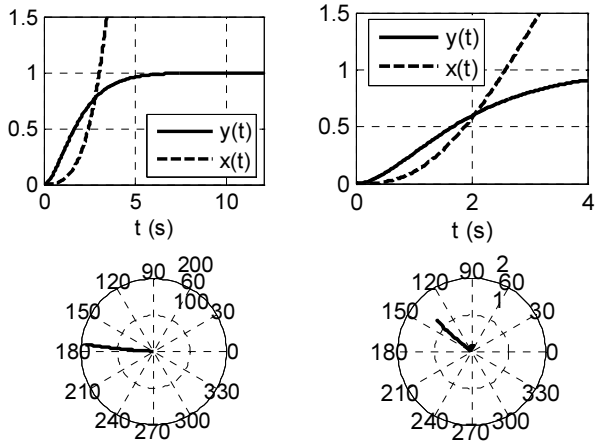


Fig. 4-7 Anger and Fear.

Excitement, on the left of Fig. 4-8, proves to be very important, since it defines the limits between stable and unstable behaviors. Excitement corresponds to the point at which the system stops diverging and begins to follow the RM. Thus, achieving stability should assure that the controller experiences emotions that are more positive than excitement for each instant in time, in other words, having emotional states α smaller than 90°. The controller on the left of Fig. 4-8 has gains $Kp = 1$, $Ki = 1$, and $Kd = 0$. Happiness, on the right, has gains $Kp = 1.5$, $Ki = 1$, and $Kd = 0$.

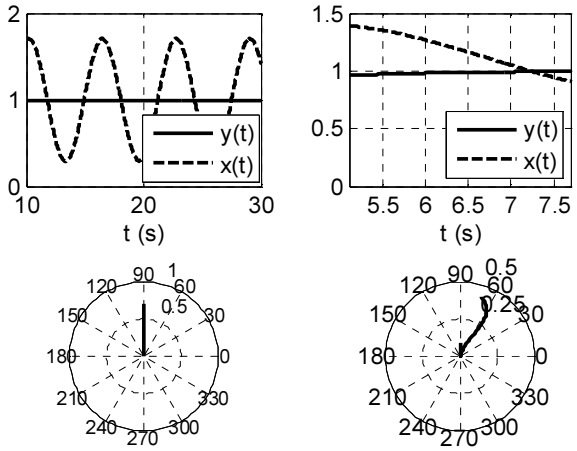


Fig. 4-8 Excitement and Happiness.

Happiness corresponds to an emotional state $\alpha = 60°$, as shown on the right of Fig. 4-8. The system experiences happiness until x equals y at approximately 7.1 s, then a negative emotion appears as x goes away from y. On the other hand, satisfaction and calm, the ideal emotional states, are shown in Fig. 4-9. During these ideal states the system leads its dynamic towards the RM. Satisfaction, on the left side of the figure, correspond to gains $Kp = 2.5$, $Ki = 1$, and $Kd = 1$, whereas calm has $Kp = 10$, $Ki = 1$, and $Kd = 1$.

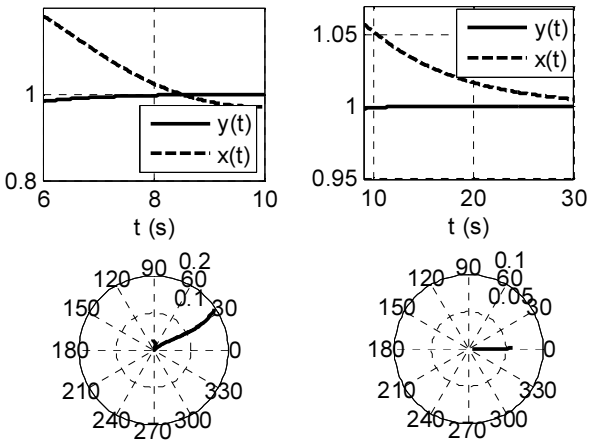


Fig. 4-9 Satisfaction and Calm.

Simulations from Fig. 4-7 through Fig. 4-9 show how the dynamic of a system can be seen as a set of emotional states α. The range from calmness to anger has been adopted, taking the order given for the Circumplex Model of Affect into account.

Implementation of the proposed controller

In this section the first version of the emotion-based controller controls four dynamical systems. The controller computes two values: the emotional state α and $\mathring{u}_e$, where $\mathring{u}_e$ can be seen as the decision made by the controller in order to keep the system within the expected behavior. The multiplication of the instantaneous emotional state α by a constant may be the simplest kind of decision. Unfortunately that multiplication only controls simple dynamical systems, because the emotional state α does not distinguish between positive and negative errors. Therefore, the multiplication should have a new term: the sign of the difference $y - x$, as was actually done in the simulations to come.

The simulation in Fig. 4-10 shows the comparison between the performance of the emotion-based controller and a traditional PID controller. The result is remarkable: the emotional controller produces a maximum error of 6×10^{-3}, whereas the PID controller shows an error of 0.4. However, this performance cannot be generalized to nonlinear systems. The control system has the following parameters: plant $H(s) = 1/(s - 1)$, reference model $RM(s) = 25/(s^2 + 10s + 25)$, gains for the PID $Kp = 7$, $Ki = 10$, and $Kd = 1$, and gain for the emotional controller $kem = 1\times10^4$. Whereas, the sampling time is 1×10^{-3} s.

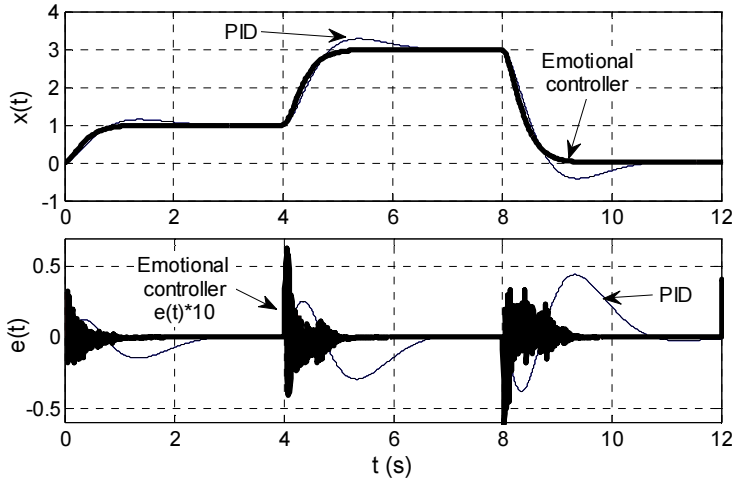


Fig. 4-10 Control of an unstable first order system.

The next example illustrates that calmness does not guarantee zero error; on the other hand, it does guarantee that the controller will seek to reduce the error, as illustrated in Fig. 4-11 from $t = 4$ *s* and onward. The control system has the following parameters: plant $H(s) = 1/(s + 1)$, reference model $RM(s) = 1/(s^2 + s + 1)$, and $kem = 5$. Whereas, the sampling time is 5×10^{-3} *s*. This simulation demonstrates that what matters during the control process is that the emotional state remains under $\alpha = 90°$ to guarantee a decreasing error.

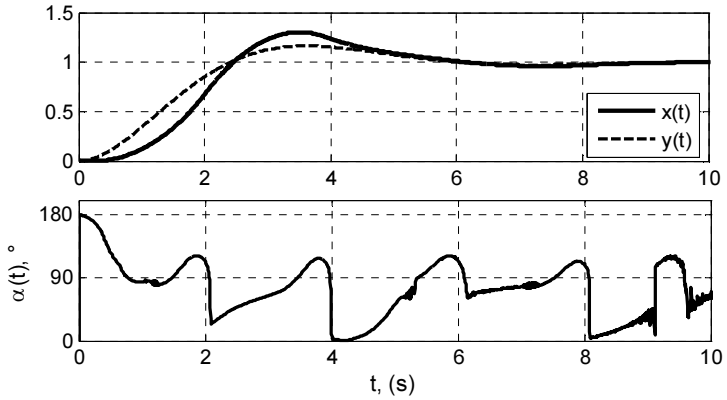


Fig. 4-11 Relation between α, y, and x during the control of a first order system.

The third example illustrates control of a second order system, as shown in Fig. 4-12. The maximum error for the emotion based controller reaches 6×10^{-3}, which is 30 times smaller than maximum PID error. The control system has the following components: plant $H(s) = 1/(s^2 + 2s + 1)$, reference model $RM(s) = 4/(s^2 + 2s + 4)$, and the gains for the PID $Kp = 5$, $Ki = 5$, and $Kd = 1$. The gain for the emotional-based controller is $kem = 1\times10^4$. Whereas, the sampling time is 1/800 s.

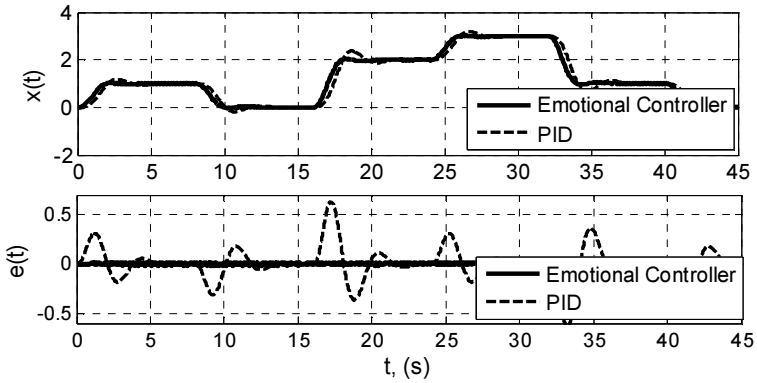


Fig. 4-12 Emotional component and control behavior for a second order system.

In the fourth example the emotion-based controller governs the dynamic of the nonlinear system given in Eq. 4.1. The reference r changes every 12 s, beginning at 1, then 0.7, 0, 0.3, and finally -0.6, as shown in the upper part of Fig. 4-13. The control system has components $RM(s) = 1/(s^2 + s + 1)$, and parameters $Kp = 5$, $Ki = 5$, $Kd = 1$, and $kem = 1\times10^4$. Whereas, the sampling time is 1×10^{-2} s.

The second plot of the Fig. 4-13 shows the performance of the classical controller, which gives a maximum error of 0.16, whereas the emotion-based controller, in the third plot, has an error of only 0.011. Thus, the emotional controller has an error 14 times smaller than the traditional PID. The third plot shows the result of combining the traditional controller and the emotional component in the control architecture. The combination sums the output of both controllers as shown in Fig. 4-6. In such a case the maximum error goes to 5.9×10^{-3}, in other words almost half of the error of the emotional component alone, and 27 times smaller than the traditional controller. This combination inherits the good performance of the emotion block and the soft behavior of the PID. Finally, the lower plot in the figure Fig. 4-13 shows a combination of the logic and emotional controller in which the output of the logic controller (PID) is weighted five times the emotion-based controller's output weight. This experiment decreases the maximum error more than two times with respect to the previous experiment. The error in the lower plot achieves a minimum, but it is still there. The plant oscillates around the reference model, even though the oscillations fall under 2.5×10^{-3}, which may be considered as zero for some applications.

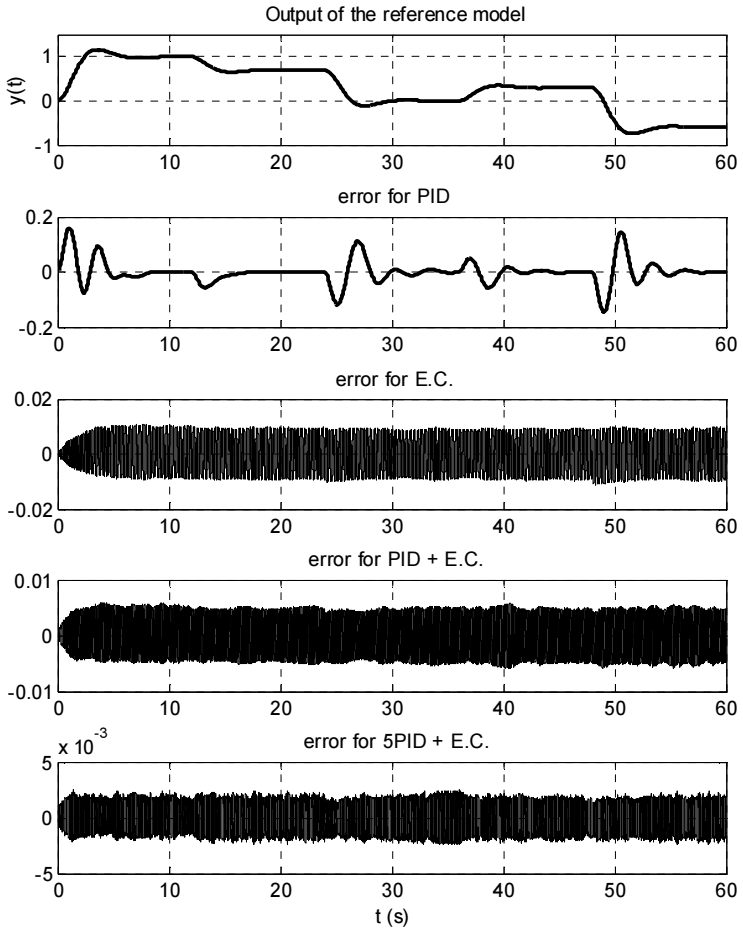


Fig. 4-13 Control of a nonlinear system.

In summary, the use of the emotion-based controller has been illustrated using two first order dynamical systems as well as a second order DS, both linear and nonlinear. The primary component of the emotional-based controller is the emotional state α, associated with the difference between the plant dynamic and the RM dynamic. This is akin to the difference between expected current scenarios triggering emotions in humans. Zero angles are associated with calmness, while the value of π is associated with anger. The strategy is consistent with the Circumplex Model of Affect.

4.4 Refined architecture for the emotion-based controller

The previous definition of the emotional state α serves to control dynamical systems, as was demonstrated by a number of examples in the preceding section. Nevertheless, the plants continued to oscillate around the reference dynamic, instead of approaching the reference asymptotically. A new definition of the emotional state α was proposed in an attempt to eliminate such oscillations within a finite time period. Again, the anticipation is the core of the new control architecture, where the difference between a current prediction for the plant and an ideal prediction defines the emotional state α (see Fig. 4-14).

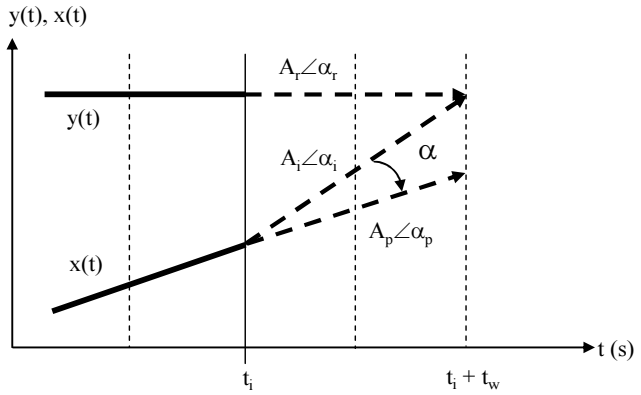


Fig. 4-14 New definition of α.

The anticipation consists of computing a future value for the reference dynamic and for the plant also, using the rate of change, m, at a current time t_i, such that $y(t_i+t_w) = y(t_i) + mt_w$. The future time, t_w, can be defined as a multiple of the sampling time. For instance, Fig. 4-14 shows that $t_w = 2dt$. This new definition of α does not require the computation of a second rate of change, which translates into asymptotic stability for the system.

A second change in the control architecture consists of replacing the input for the integral in the controller. Thus, instead of $K_{em}\alpha$, the new input equals $A_i\alpha$. The ideal magnitude, A_i in Fig. 4-14, can be seen as the intensity of an emotion. So, greater differences between the reference dynamic and the plant represent emotions with higher intensities. In another interpretation, A_i can be seen as a variable K_{em}.

A third and major change in the controller consists of adding a component to the integral, so $u = KpA_i\alpha + Ki\int A_i\alpha dt$. The proportional part, $KpA_i\alpha$, controls the instantaneous dynamic, while the old integral component, $Ki\int A_i\alpha dt$, can be seen as a measure of inertia of the system, and controls the system based on the history of the dynamic. The main disadvantage of using the integral exclusively resides in the fact that any non-zero input causes changes in the output of the integral. As a result, even a minor error may cause oscillations of the plant around the reference. These oscillations can be reduced by decreasing the sampling time and also increasing K_{em}, but the oscillations will still occur.

The whole package of new definitions will be exemplified by a simulation, as shown in Fig. 4-15. First of all, a very simple reference dynamic results from simply demanding that $y = 1$. On the other hand, the plant equals $H(s) = 100/(s^2 + 10s + 100)$. The parameters of the controller are the sampling time, $dt = 1\times10^{-3}$, the prediction window, $t_w = 2dt$, and the gains $Kp = 1\times10^4$ and $Ki = 1\times10^5$. A very important feature of the control performance regards the amplitude of the emotional state α. The emotional state α decreases with the error such that the whole system tends toward calmness. In addition, the emotional state α detects the sign of the error. Thus, the emotional state α can be positive or negative. The simulation in Fig. 4-15 also shows that $A_i\alpha$ shrinks as time pass.

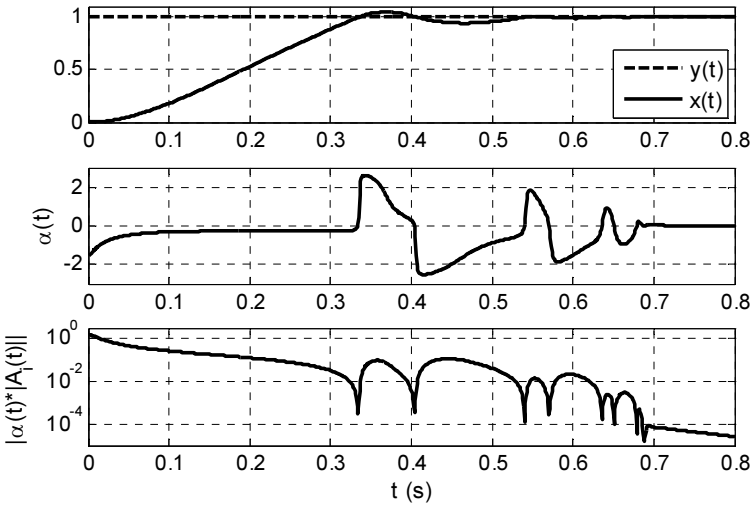


Fig. 4-15 Performance of the new control architecture.

The error, $x_r - x_p$, in Fig. 4-16 serves as a demonstration of stability. This error decreases exponentially and reaches Matlab's minimum spacing of floating point numbers slightly after three seconds.

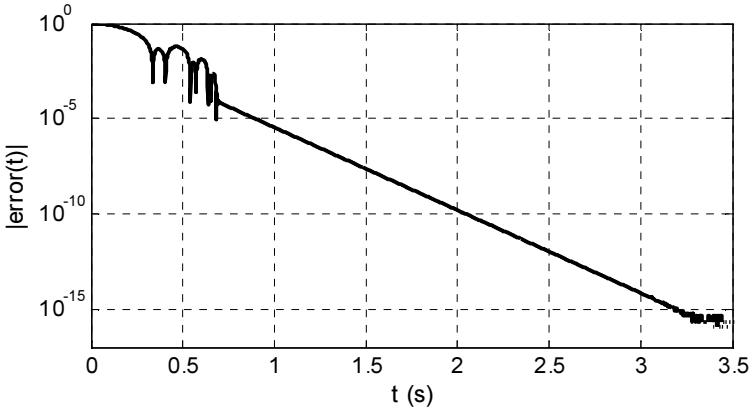


Fig. 4-16 An experimental demonstration of stability.

The modification of the control algorithm will be tested using three plants: a nonlinear plant, a frequency-controlled power system, and the plant described previously in Chapter 3, with a frequency given by an approximation algorithm.

Example 1

The nonlinear plant, given by Eq. 4.1, was already used in this chapter to explain the intuitive idea of an emotional controller. This plant has two attractors when its input equals zero. Thus, the system will naturally approach one of these attractors depending on the initial conditions. The reference dynamic, $H(s) = 1/(s^2 + s + 1)$, achieves its steady state value after 5 s, which is faster than the plant. This steady state equals one for times between 5 to 10 s. Then a transition occurs at 10 s, and the new steady state reaches zero for times higher than 15 s. The parameters of the controllers are $dt = 1$ ms, $Kp = 1\times10^4$, $Ki = 5\times10^3$, and $t_w = 2dt$.

The error caused by a traditional controller and the emotional controller differ in two ways, as shown in Fig. 4-17. One of these differences is the smoothness of the emotional controller; the

other is the value of the error itself. The ratio between these errors reaches two orders of magnitude at the end of each steady state region between 10 and 20 *s*.

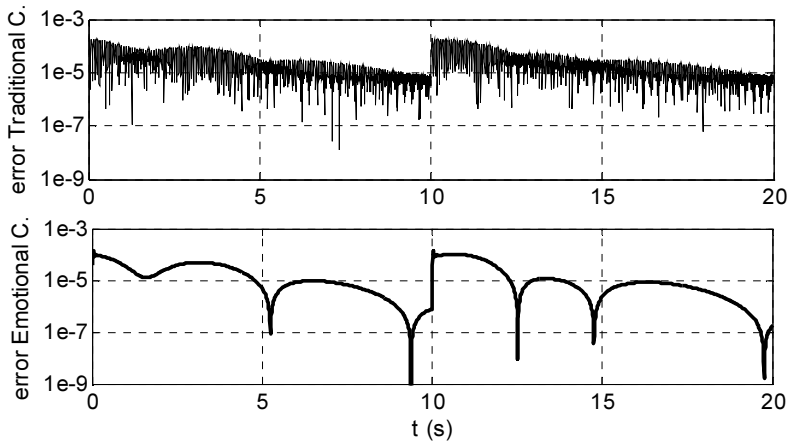


Fig. 4-17 Error for a traditional and a modified emotional controller.

The oscillations for the traditional PI controller are caused by fast oscillations in the controller's actuating signal, as shown in Fig. 4-18. This oscillatory behavior contrasts sharply with the smoothness of the emotional controller's output.

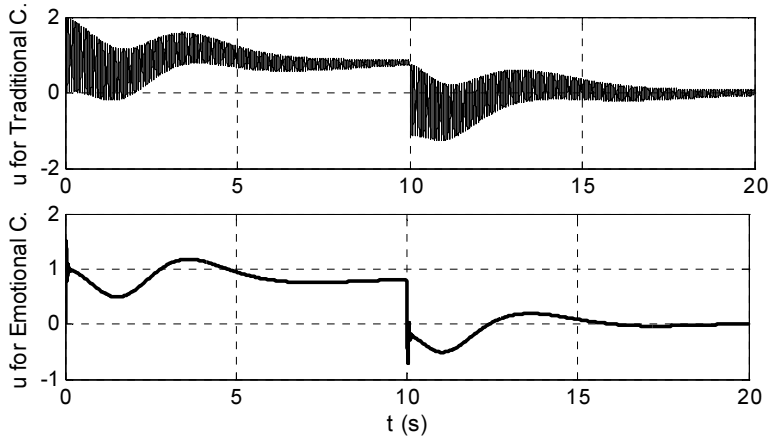


Fig. 4-18 Controllers output.

Example 2

The frequency control of a power system was addressed in Chapter Three. The goal of such an application was to demonstrate that it is possible to anticipate a periodic behavior on the basis of an approximation of the period. However, this example assumes a known frequency. Fig. 4-19 depicts the model of the power system, with values expressed in per unit (p.u) to simplify computations. The nominal power of the system is one, and extra load causes changes on the frequency. The system should compensate the extra load by changing the actuating signal *u* emanating from the controller. The frequency value is also in per unit, so the traditional 60 *Hz* are scaled into 1 *Hz*. Then, each cycle lasts one second. The constant time for each block (governor, turbine, and system inertia) represent real constant times in p.u.

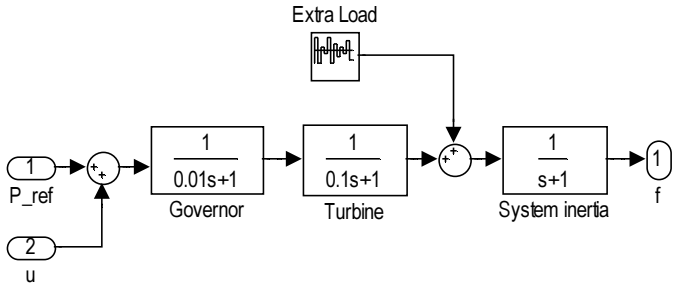


Fig. 4-19 Power system block diagram.

The parameters of both controllers are dt = 2.5 ms, Kp = 25, and Ki = 150. The prediction window t_w equals $2dt$ for the emotional controller. The experiment in Fig. 4-20 shows a non-controlled plant from 0 to 120 cycles, and a controlled plant for times higher than 120 cycles. The extra load changes every 20 cycles and its value varies randomly from -4% to 4%. This variation on the load makes the frequency to change proportionally. Once a controller starts to define the actuating signal, the frequency for the most part remains within the reference value, 1 Hz.

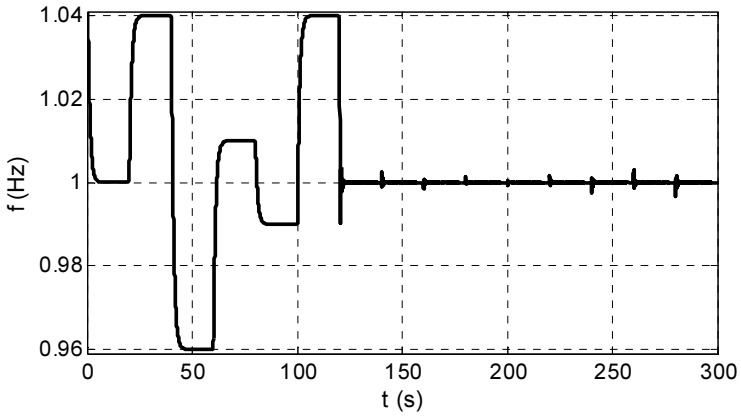


Fig. 4-20 Frequency for uncontrolled and controlled plant.

The error for the emotional controller gets smaller exponentially. As a result, after only 18 cycles the controller reaches an error smaller than 1×10^{-16}, whereas the traditional PI controller requires 20 cycles to stay at 1×10^{-12}, as shown in Fig. 4-21. Results in Fig. 4-21 serve to demonstrate that both controllers make the plant stable, but the emotional controller is definitely faster.

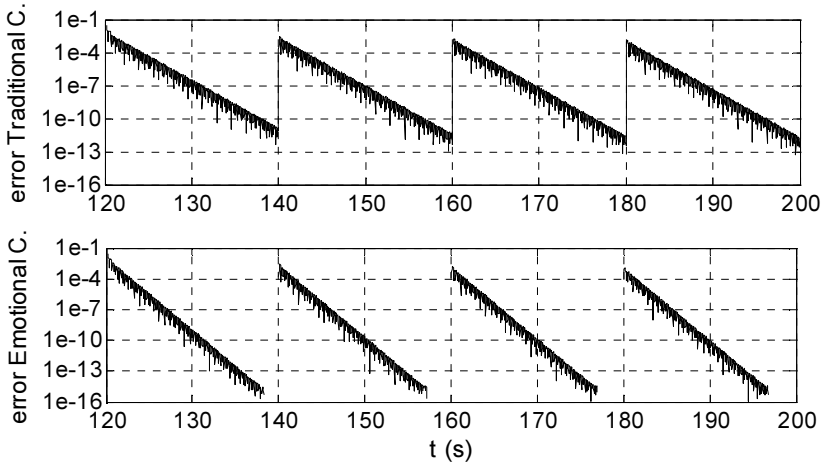


Fig. 4-21 Error for the frequency control application.

Example 3

As in the previous example, this problem implements frequency control, but approximates the frequency by means of the method described in Section 3.3, instead of using its exact value. The approximation requires sampling a voltage signal, as shown in Fig. 4-22, by means of the tapped delay.

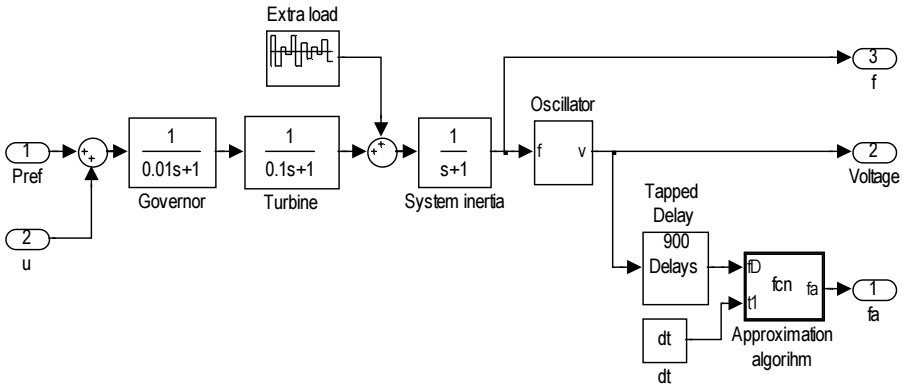


Fig. 4-22 Power system and frequency approximation blocks.

The actual value of the frequency corresponds to the output of the block labeled "system inertia" in Fig. 4-22, this value generating a sine function at the given frequency. Then, the tapped delay captures a sample every 2.5 *ms*. After that the approximation algorithm block uses those samples to produce an approximate period, which falls between 0.95 and 1.05 *s*. The algorithm has latency close to two periods, as shown in Fig. 4-23.

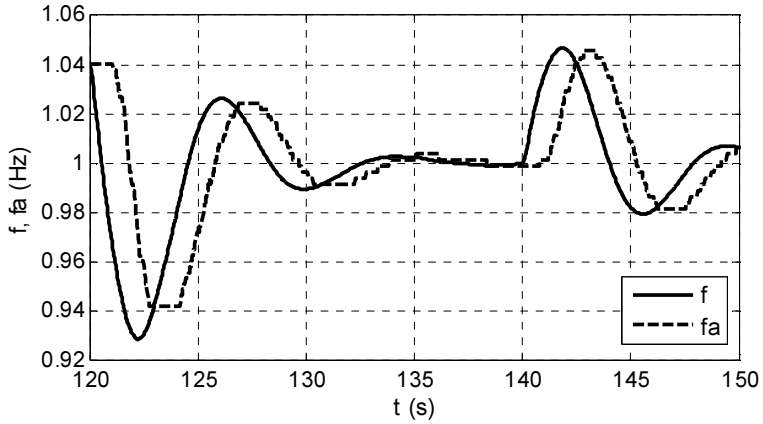


Fig. 4-23 True and approximated frequency.

These simulations demonstrate that both controllers have similar performance when the prediction window, t_w, equals $2dt$, with the gains for both controllers set to $Kp = 0.4$ and $Ki = 0.5$. In addition, and given the discretization used by the approximation algorithm, it is not possible to eliminate the error, but only to reduce it to a certain limit. Results in Fig. 4-24 are not as good as results in previous examples, but the emotional controller still performs as well as the traditional controller. Basically, having an approximation only after two cycles reduces the effectiveness of the anticipation, but at least the control system preserves stability.

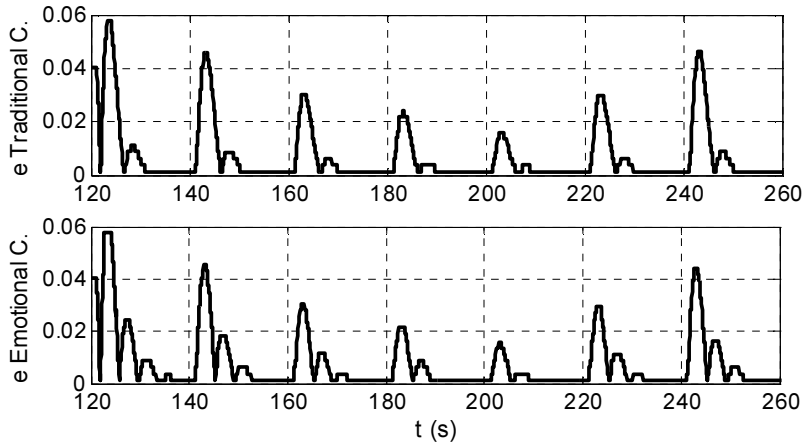


Fig. 4-24 Error for both controllers.

There are two inconveniences associated with the definition of the emotional states α in Section 4.2. On the one hand, oscillations around the reference model preserve their magnitude regardless of how long the control system works. On the other hand, the relationship between position, speed, and acceleration may cause incorrect evaluation of an emotional state. Simulations show that most of the times when the position has a smaller magnitude than speed, and the speed than acceleration, then apparently the emotional state stays around 90°, in spite of the error behavior.

The refined version of the emotional state α resolves both reported problems. First, the controller assures stability because the error always tends towards zero for the experiments in this thesis, eliminating possible oscillations. This was done by also adding a new component in the decision-making process: a proportional contribution to the error and not only integration. The solution of the second problem includes the definition of a new set of vectors to compute the emotional state α, in such a way that those vectors have the same dimensions: time on the x axis, and the magnitude of the system on the y axis.

In terms of control systems, there are only three parameters to tune during the control process. The first one, t_w, was fixed at $2dt$ for all the examples in this section. Experiments show that t_w may be set between one and several hundreds of dt. Larger prediction windows means that the algorithm looks farther ahead, which has the same effect on prediction as using delayed data. As was seen in the last example of this section, this reduces the performance of the controller. The other two parameters may be seen as Kp and Ki of a traditional PID controller. Then, different algorithms may be used to tune these gains. However, future work may look for an algorithm to determine these gains in an adaptable fashion.

4.5 Frequency and time characterization of the emotion-based controller

This section compares the performance of the proposed emotion-based controller with a traditional PI controller. This study focuses on the frequency and time behavior for both controllers under equal constraints and setups. The frequency-domain analysis computes the effect of changes in the amplitude and frequency of a sinusoidal reference on the control error and also on the controller's effort to follow a reference. A traditional measure of the error computes the difference between the reference signal and the output of the plant, once this output is transformed by a sensor. Given the proposed controller in this thesis, it is interesting to analyze another definition of error, an

emotional definition of the error. We established the new definition of error as α times A_i, according to the definitions in Fig. 4.14, Eq. 4.6 and Eq. 4.7.

$$\alpha = a\tan\left(\frac{y(t_i + t_w) - x(t_i)}{t_w}\right) - a\tan\left(\frac{x(t_i + t_w) - x(t_i)}{t_w}\right) \tag{4.6}$$

$$A_i = \sqrt{(y(t_i + t_w) - x(t_i))) + t_w^2} \tag{4.7}$$

Based on the preceding equations, the indicator of the error evolution using the emotional definition corresponds to the Integral of the Absolute value of Error (*ITAE*), defining the error as α times A_i, as shown in the Eq. 4.8.

$$ITAE\ \ emotional\ \ definition = \int_0^{20} t|\alpha A_i|dt \tag{4.8}$$

The *ITAE* index computes the error behavior for a given range of time in a single value penalizing errors that persist in time by multiplying the error by the time t. The description of the *ITAE* index for the traditional definition of the error corresponds to the expression in the Eq. 4.9.

$$ITAE\ \ traditional\ \ definition = \int_0^{20} t|y - x|dt \tag{4.9}$$

As it was already mentioned, in addition to the analysis of the error, which corresponds to determine how good the controller is to lead a plant, this section looks at the effort of the controller to do the work. This effort is summarized by the Integral of the Absolute value of the actuating signal, *IAU*, as defined in Eq. 4.10. A traditional definition of the *IAU* does not include the constant b. But because many steady dynamics correspond to limit cycles, producing actuating signals that may oscillate between two bounds, the b value serves to facilitate the distinction between such behaviors. This constant b corresponds to the value of the actuating signal that produces null error at a constant reference, when the reference equals the continuous component of the sinusoidal reference.

$$IAU = \int_0^{20} |u - b|dt \tag{4.10}$$

In addition to the three indices in Eq. 4.8 through Eq. 4.10, the frequency-domain analysis uses an extra index, the measure of the gain in decibels, as defined in the Eq. 4.11. This index associates the amplitude of the oscillation at the output with the amplitude of the oscillation at the reference. This measure allows to compute the bandwith of the control system, among other features.

$$decibels = 20\log\left(\frac{Aout}{Ain}\right)\ [db] \tag{4.11}$$

Time characterization, as a second analysis of the control performance, also uses *ITAE* and *IAU* indexes, but this time only constant values serve as reference signals instead of sinusoidal waves.

The plant in this section is a DC motor plus a power amplifier, as shown in Fig. 4-25. This figure also depicts the whole setup in laboratory. The process to come up with a model of the plant uses the voltage coming into the transistor and the corresponding motor speed. This data serves as input to the System Identification Toolbox of Matlab, which provides a linear model $H(s)$ equal to $7470/(s^2 + 14s + 37)$ (rad/s)/V. Additional features of the motor include nominal voltage of 12 V, with 4.3 W of power, nominal speed of 4.060 rpm, torque equal to 10 mNm, and commercial reference 1.13.78.xxx of Bühler.

The output of the speed sensor, a tachometer, is a sinusoidal wave with frequency and amplitude variable proportional to the motor speed. The conditioning process of that signal starts when a diode bridge rectifies the wave. A second stage filters the resulted signal using a RC circuit. Finally, the signal passes through a low-pass filter in Simulink, as shown in Fig. 4-25. The model of the tachometer may be summarized as a constant $ks = 0.006$ V/(rad/s).

One of the main components of the emotion-based controller is the reference model, as explained in section 4.2. The concept of anticipation, core of the proposed strategy, requires a model of an ideal dynamic for the plant in order to trigger an emotion. Even though, the analysis in this section does not use it, because it focuses solely on studying the part of the control law given by αA_i, which may be seen as a nonlinear transformation of the traditional concept of error. This transformation allows us to correlate definitions of emotions in humans with traditional concepts in the control of dynamical systems.

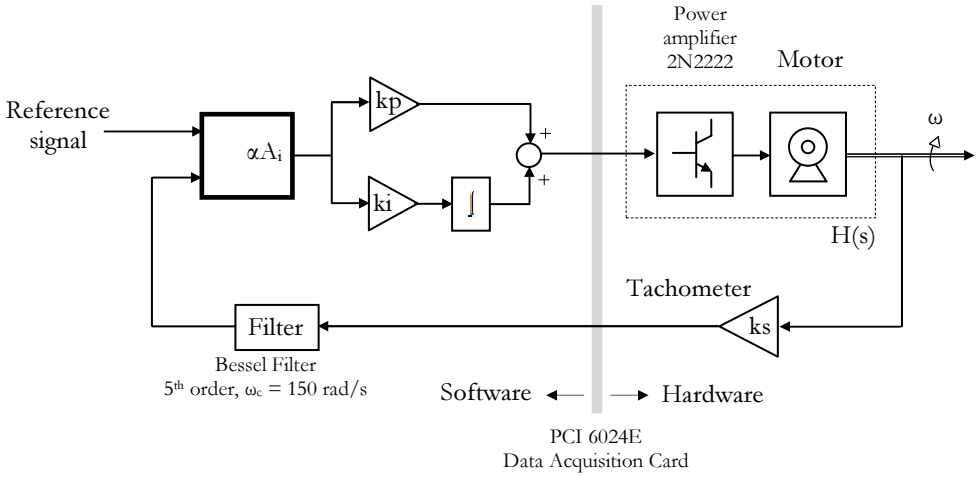


Fig. 4-25 Speed control of a DC motor using the emotion-based controller.

The frequency-domain analysis uses a reference signal $r = a\sin(\omega t) + dc$. The constant dc equals 5.5, corresponding to half the active input range for the motor. Correspondently, the amplitude a varies from 0.5 to 2.5 in order to cover the input range for the motor. The frequency range starts at 0.628 rad/s and finishes at 100 rad/s. The lower bound guarantee two cycles during the 20 s experiment, whereas the upper bound corresponds to the frequency at which the tachometer and data acquisition card reach the lower measurable amplitude. The gains kp and ki in figure 4-25 were defined as 2 and 5, respectively, as proposed by running an auto tuning of a traditional PI controller using SISO tool of Matlab. The results of the frequency characterization are shown in Fig. 4-26.

The left column in Fig. 4-26 shows simulations for the frequency-domain analysis. The right column shows the actual experimental results. The emotion-based controller corresponds to the dark surface, whereas the traditional PI controller corresponds to the light surface. Each simulation and its experimental counterpart use a sampling time *dt* of 2 ms, and a window time prediction *tw* of 10 ms. Each experiment was run three times. Thus the final value in the figure equals the main value of these partial results. The results in the first row use the emotional definition of the error, whereas the second row shows the results for the traditional error definition. The light color at each surface corresponds to the traditional PI controller, and the dark color represents the results for the emotion-based controller.

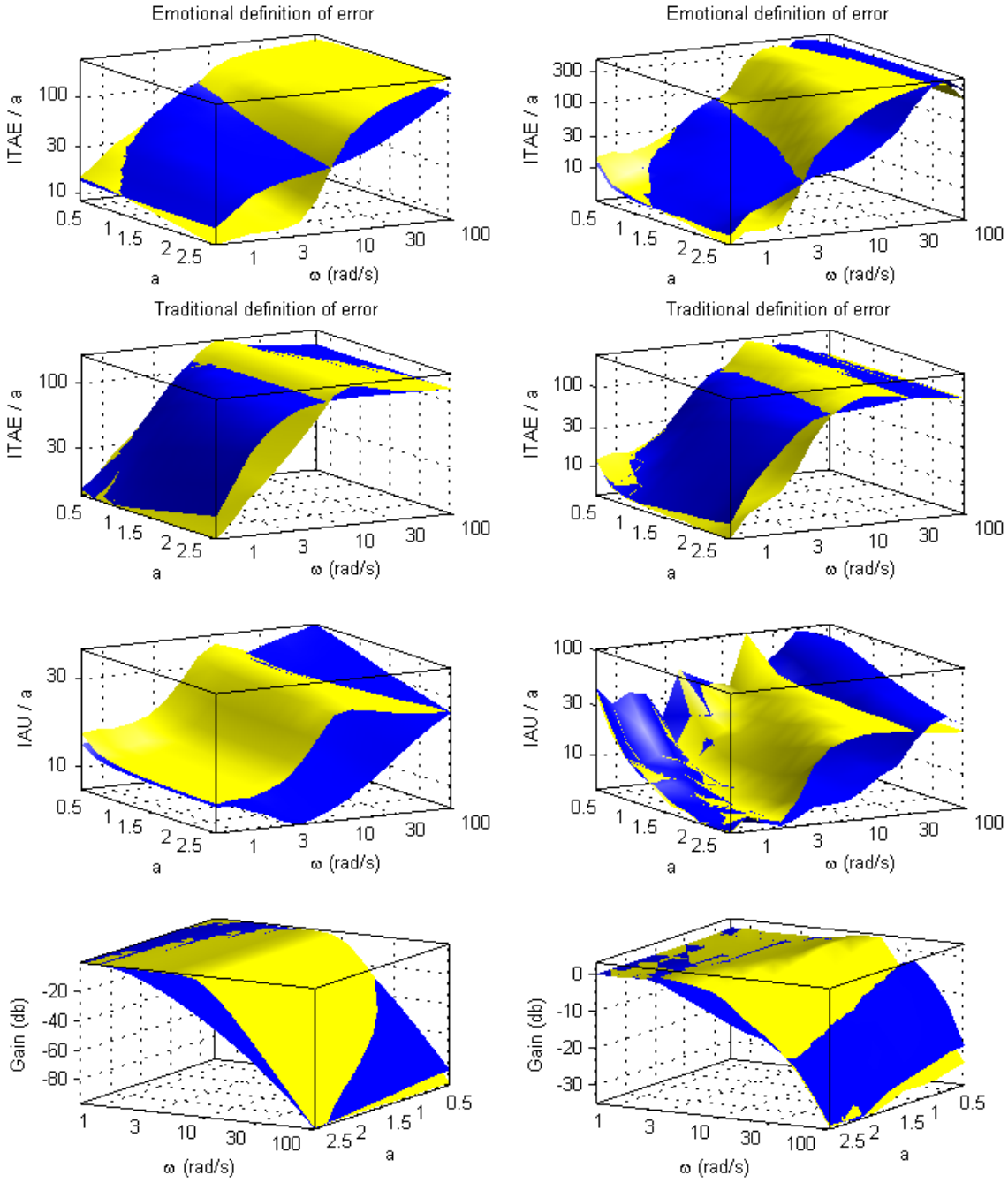


Fig. 4-26 Frequency characterization.

A first observation coming from the results in Fig. 4-26 contrasts the whole left column with the right column, in other words, the simulations versus the experimental results. The similarity between these two columns let us ensure that the model of the plant and sensor properly capture the dynamic evaluated in this experiment and at the same time guarantees the reliability of the data.

An analysis of the first two rows in Fig. 4-26, concerning the *ITAE* indexes relative to the amplitude, shows a quasi-linear dependency of the *ITAE* indexes with respect to the amplitude *a*, as expected for linear plants. In addition, those plots show that increasing the frequency increases also the index value, which means that larger frequencies implies also larger errors for both controllers. A third observation concerns the comparison between controllers. The traditional PI controller performs better for low frequencies, whereas the emotion-based controller, in general, performs better for high frequencies.

The Bode magnitude (the fourth row in the Fig. 4-26,) shows that the cut frequency, dividing low from high frequencies, is around 5 rad/s for the PI controller and 3 rad/s for the emotion-based controller. That lower value for the proposed controller contrasts with the decibels at high frequencies where the emotion-based controller performs better. As concluded in the previous paragraph, the proposed controller outperforms the traditional controller at high frequencies. This is because the emotion definition implies anticipation, and acting ahead seems to be effective to counteract high frequency changes at the reference. On the other hand, in general, the emotion-based controller leads the plant with less effort than a traditional controller, as indicated by lower *IAE* index values for the proposed controller in the third row of Fig. 4-26.

The second analysis, in the time domain, uses a constant reference, $r = 5.5$. Thus, instead of varying frequency and amplitude, as was already done in the frequency domain, this analysis focuses on the effects of changing the controller gains, *kp* and *ki*. This analysis looks at two aspects: 1) the behavior of the control error by means of *ITAE* indexes, and 2) the effort of the controller to lead the plant, using the *IAU* index. The emotion-based controller, in Fig. 4-27, corresponds to the dark surface, whereas the traditional PI controller corresponds to the light surface

A first observation, based on the results in the Fig. 4-27, regards the effect of the proportional gain *kp* on the system behavior. Changes in *kp* produces two main scenarios. One of them, called stable region, corresponds to values of *kp* smaller than 5. The error in this region shrinks as the time passes, as the concept of absolute stability requires. The second section, for *kp* larger than 5, corresponds to limit cycles at the output of the plant. The performance of the emotion-based controller in this second region is visibly better than the traditional controller, passing from *ITAE* equal to 50 for the traditional controller, to 30 for the emotion-based controller. Unfortunately, this performance is not uniform across the whole landscape, but, as shown in the top view in Fig. 4-27 (high and right side), in general, the proposed controller produces smaller *ITAE* indexes.

The influence of the *ki* gain depends on the region under analysis. For the limit cycle region there is almost no effect at all, but for the stable region, the variation of the integral gain, *ki*, forms a bowl shape with minimum at $ki = 25$. The integral gain does not have effect in the limit cycle region, because the magnitude of any instant error override the integral contribution, which comparatively grows slower, and generate values far below, than the instant error. The Fig. 4-28 shows the results for the traditional definition of the error instead of the emotional definition. Results for traditional and emotional definition of error are very similar and vary only in their magnitudes.

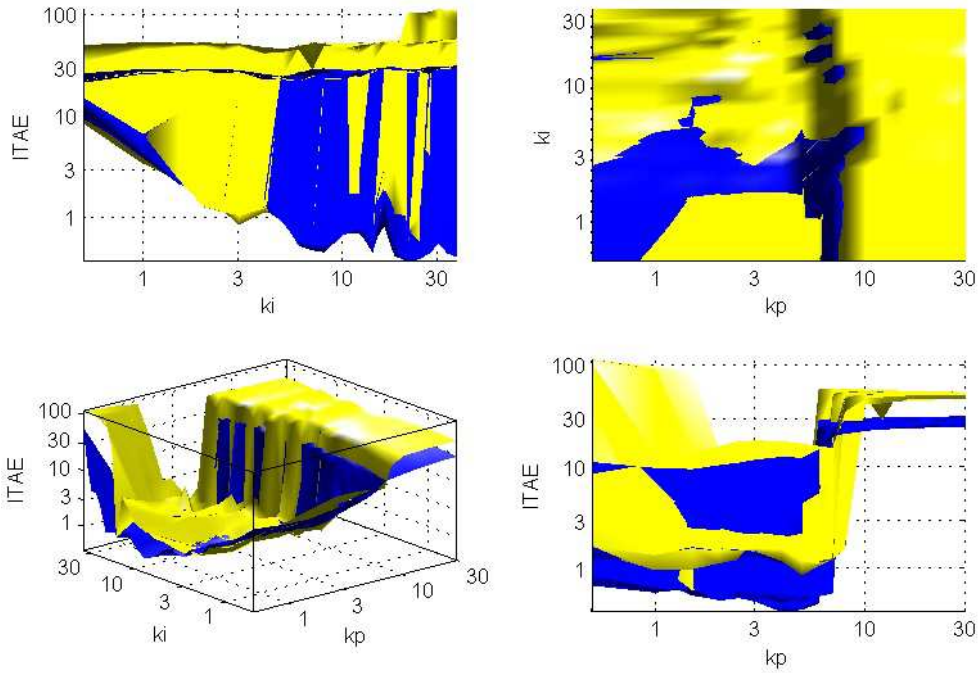


Fig. 4-27 Time characterization. ITAE for an emotional definition of the error.

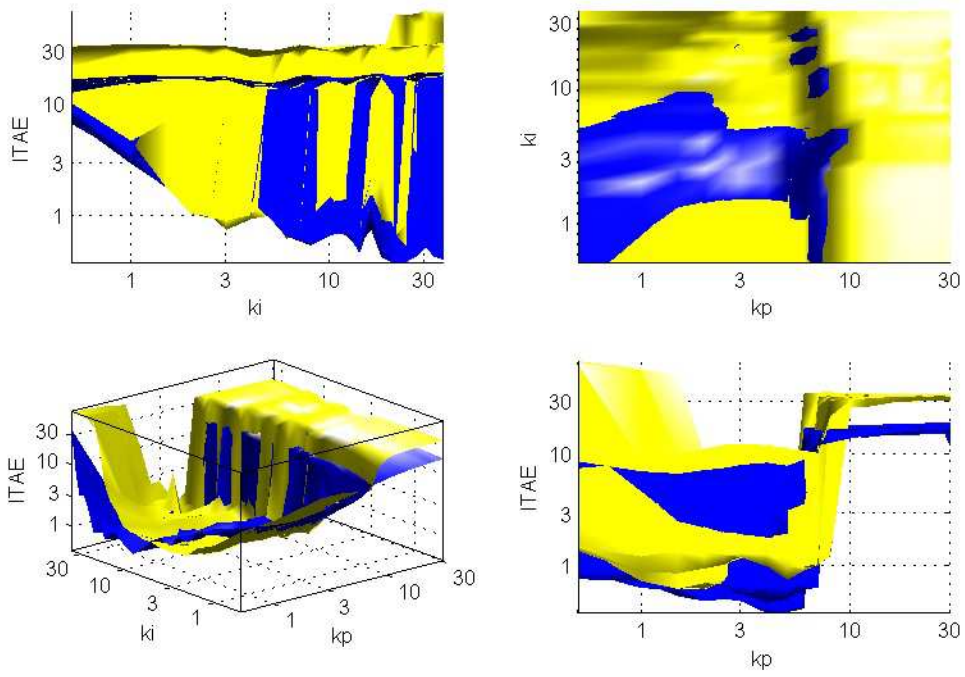


Fig. 4-28 Time characterization. ITAE for the traditional definition of the error.

Given the constant reference, and comparing both controllers, the *IAE* values for the stable region do not differ more than 5%. Thus, both controllers have similar capabilities to handle stable behaviors. On the other hand, the proposed controller requires, in average, 20% less effort to handle the limit cycle region, equating the effort with the *IAU* value, as shown in Fig. 4-29, where the emotion-based controller corresponds to the dark surface, whereas the traditional PI controller corresponds to the light surface. This, added to the reduction in the *ITAE* index, allows us to remark the advantage of using the emotion-based controller at the cycle limit region. Additionally, the proposed controller has an adequate performance at the stable region.

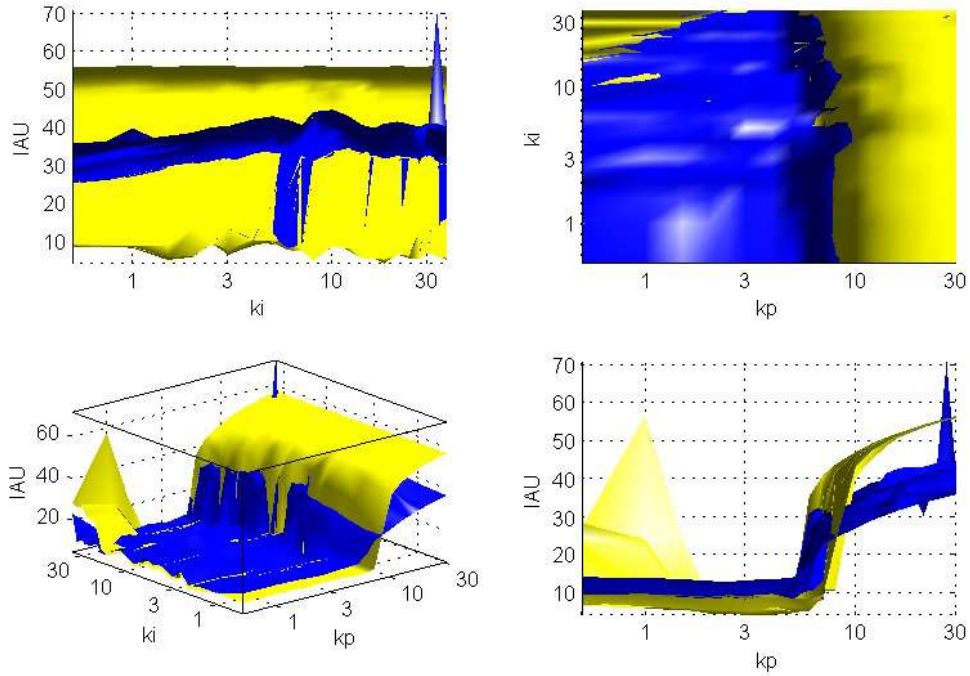


Fig. 4-29 Time characterization for the actuating signal IAU.

5. Conclusions

General conclusions

The proposed model of an emotion-based controller in this thesis applies cognitive science to traditional control of dynamical systems. This important effort brings well known facts from neuroscience and psychology about how humans make decisions, and combines them with computer science and artificial intelligence tools to produce application for control systems, an area traditionally studied from a mathematical perspective. The introduced emotion-based control algorithm illustrates that the human brain can certainly serve as an inspiration for new control algorithms.

This thesis demonstrates the advantages of using anticipation to define the proposed model of emotions. This model has two main elements: first, a predicted scenario to represent the dynamic of the reference model, and second, an anticipated dynamic originating with the plant. A comparison between these two elements defines an emotional state. For instance, good scenarios compared with suitable dynamics trigger calmness. This emotional state has been chosen as the most desirable emotional state given that psychologists have demonstrated that calmness is appropriate to perform highly cognitive tasks.

The simplicity of the Circumplex Model of Affect has proven adequate to propose the emotion-based control algorithm, because the computation of its two parameters (valence and arousal) requires few operations. The values of these parameters were associated with the relationship between predicted scenarios for the control system at every instant in time. More complex models would result in approaches even more similar to human cognition. Nevertheless, the intricacy of these models may increase the computation time, rendering them inappropriate for the control of dynamical systems.

Experiments in this work have demonstrated that anticipation truly serves as a bridge between cognitive concepts that embody emotions' influence on decision-making in humans and the mathematical principles employed in the control of dynamical systems. The simplest implementation of anticipation, used in this document, supposes that the dynamic of a system will continue its current trend. Thus, a discrete derivative multiplied by a time window will predict future behaviors.

The use of a reference model may be one of the main contributions to control systems from this thesis. Seeking to match the dynamic of two systems (reference model and plant) instead of trying to follow a reference signal, as classical control does, results in substantial improvements in the transformation of a plant's dynamic. Even though, the reference model for the experimentation stage of the proposed controller in this thesis was defined in such a way that it properly follows an external reference signal, whereas the controller aims to lead the plant toward the dynamic of the

reference model. This definition of the reference model serves to simplify the analysis of the control performance, because focus the attention in the emotions definition.

Another advantage of the proposed control algorithm consists of the option to include, in addition to the emotional component, a logical component (i.e., a traditional control law), to the new control architecture. The combination of logical plus emotional decision-making models cognitive processes more faithfully than using only one of these processes. At the same time, this combination increases the set of possible dynamics expressed by the proposed controller, enriching the control system.

In general, prediction has proven to be a very difficult problem to solve, as stated in the discussions presented in the design of two new algorithms for anticipating the value of a periodical signal. A closer look at the problem shows that an irrational ratio between sampling rate and period makes it impossible to guarantee the precision of the period's computation from data. Therefore, the estimated value for the period will be just an approximation, which entails errors in the reconstruction of a signal, especially for predictions in the long term.

The measurement principle introduced in this thesis to approximate the rate of change of a signal only requires equidistant samples, as in the case of signals coming from an incremental encoder. The computation of the angular velocity exemplifies the estimation of a discrete derivative, a requirement for prediction. A detailed observation of the pitfalls of traditional algorithms, such as oscillations in the approximation at constant speed, inspired the development of a new algorithm, which minimizes the maximum relative error, and assures non-oscillatory approximations at steady state.

Thesis results

- A set of definitions of emotional states for dynamical systems was presented. The comparison between the anticipated ideal scenario and the anticipated real scenario triggers an emotional state. These emotional states correspond to the definitions in the Circumplex Model of Affect.

- A control architecture was introduced. It combines the definition of emotional states with the agent's decision-making. The proposed architecture flexibly allows the inclusion of a traditional controller, akin to the logical component of cognition in humans.

- A new algorithm was introduced to approximate the rate of change of a signal using encoders. The algorithm has constant output for constant speeds, contrary to traditional algorithms that generate two values. The maximum relative error falls to less than half the error produced by traditional algorithms.

- An algorithm was designed to approximate the period of a periodic function by minimizing the mean squared error between a perfect sine function and samples of the target function. The algorithm requires an iterative process, starting with few data and enlarging the data length as it iterates.

- A procedure was introduced to approximate the Fourier coefficients of a periodic signal using a perceptron neural network. This procedure requires estimating the period and samples from the target signal. The weights of the network correspond to the Fourier coefficients.

- A second algorithm was designed to approximate the period of a periodic signal. It solely requires discrete sampling at a constant rate, and a data length of at least two periods. The algorithm has quadratic computational requirements ($O(n^2)$ in the number of data samples n).

- A framework for a dynamical control system was proposed. This framework emphasizes the relationship between dynamical systems to change the dynamic of a plant. An application of this control system includes the definition of a comfort zone, which may be used in developing environmentally friendly algorithms by reducing energy consumption.

Future work

One of the aspects that may improve the controller performance is the design of the reference model. This thesis reports the use of a quite simple version of this model, but it can be enriched in complexity. On the one hand, it would be interesting to study the application of an auto tuning algorithm to set the parameters of the reference model. On the other hand, it would also be important to have a model evolving according to the relationship between itself and the plant. Such changes would improve the adaptability of the controller, and at the same time, model cognitive aspects that actually occur in humans, such as learning and adaptation.

The core connection between how cognitive sciences see emotions and what control systems theory does in this research is the concept of anticipation. The anticipation computed for the proposed control algorithm supposes that the system will continue its actual trend. However, more complex dynamics may not meet this assumption. Thus, trying different types of anticipation is important in order to extend the application of the new controller. Future work may involve the study of more complex types of prediction, such as statistical analysis or any type of extrapolation depending on the nature of the plant.

Systems with larger complexity than the linear and nonlinear systems that were used in the experimental validation could require a larger set of emotional states. Emotions such as tiredness, depression, sadness and frustration are not included in the current emotion model, because they can be associated with unsatisfactory dynamics. Nevertheless, more complex systems, such as cooperative humanoids may require these emotions given that they are in humans.

The combination of a traditional controller and the proposed emotion-based controller was not rigorously studied in this thesis, because the cooperative relationship that was implemented by adding the output of both controllers was sufficient to validate the proposed controller. However, it is important to test other types of combinations. For instance, competitive, parallel, or cascade connections may be implemented. Another option may include the use of fuzzy logic to correlate the output of both controllers, which may enrich the dynamic of the whole control system.

The effectiveness of the proposed control architecture has been illustrated by controlling well-known problems that can be described by a small set of differential equations. These experiments show that the emotion metaphor truly improves the performance of a control system. Although, the application of emulating emotions to enhance control systems goes beyond these examples, for instance, the metaphor of a system following a track on a landscape may be extended to find alternatives for motion planning. Secondly, the fact that the plant follows a reference model may also be used as a basis in fault tolerance algorithms. Another application may add emulated emotions in human computer interfaces, where a human may be seen as part of a control loop. Finally, the principles of the new control algorithm could also be used over more complex systems, such as biological systems.

6. Appendix A

List of publications

Proceedings

1. Rairán J., Garzón M., "Green Approaches to Process Control in Thermal Rooms" in: Control and applications CA2011, Vancouver, BC, Canada. Jun. 1 – 3, 2011.

2. Rairán J., Niño L., "Algoritmo para la Toma de Decisiones Basado en las Emociones Humanas Aplicado al Control de Sistemas Dinámicos", in: sdsc09 Doctoral Symposium, Jul. 24, 2009.

3. Rairán J., "Control basado en emociones, una revisión" in: Convención internacional de ingeniería en Cuba – CIIC, Jun. 4 – 6, 2008.

Papers

4. Rairán J., "Definition of an emotion-based controller for dynamical systems", in: Tecnura Journal, Num. 33, 2012, pp. 12 – 22. B category.

5. Rairán J., "Algoritmo para la aproximación de la velocidad de giro de un eje mediante un encoder incremental", in: Ingeniería y Universidad Journal, accepted, 2013. A2 category.

6. Rairán J., "An optimal algorithm for speed approximation using incremental encoders", in: Ingeniería e Investigación, in review, 2013. A1 category.

7. Rairán J., Garzón M., "Period estimation for reconstruction of (quasi-) periodic signals", in: IET in Signal processing, in rview, 2012. A2 category.

8. Rairán J., "Reconstruction of Periodic Signals Using Neural Networks", in: Tecnura Journal, Num. 37, pp. 12 – 22, 2012, B category.

9. Rairán J., Chiquiza D., Parra M., "Implementación de nuerocontroladores en línea. Tres configuraciones tres plantas", in: Ingeniería y Universidad Journal, Vol. 16, Num. 2, pp. 163 – 182, 2012, A2 category.

10. Rairán J., López F., Monroy P., "Control de posición de un sistema bola y viga con actuadores magnéticos", in: Tecnura Journal, Vol. 15, Num. 30, 2011, pp. 12 – 23. B category.

11. Rairán J., Fonseca J., “Doble lazo de control para regular la posición y la velocidad en un motor de corriente directa”, in: Ingeniería e Investigación Journal, Vol. 15, Num. 2, 2011, pp. 337 – 357. A2 category.

12. Rairán J., “Definition of emotional states for dynamical systems”, in: Visión electrónica Journal, year 5, Num. 2, 2011, pp. 4 – 18. C category.

13. Rairán J., Fonseca J., “Medición del sentido de giro, velocidad y posición angular de un eje mediante encoders”, in: Respuestas Journal, Universidad Francisco de Paula Santander, Vol. 15 Num.1, pp. 33 – 42, 2010. C category.

14. Rairán J., Guerrero C., Mateus A., “Diseño de controladores de tipo proporcional integral derivativo (PID) y difuso para la posición de un motor de corriente continua”, in: Ingeniería e Investigación Journal, Vol. 14, Num. 1, 2010, pp. 137 – 160. B category.

15. Rairán J., “Control basado en emociones - una revisión”, in: Avances en Sistemas e Informática Journal, Vol. 7, Num. 2, 2010, pp. 27 – 34. C category.

16. Rairán J., “Modelo circumplejo del afecto aplicado al control de sistemas dinámicos”, in: Respuestas Journal, year 14, Num. 1, 2009, pp. 5 – 15. C category.

17. Rairán J., Sierra Y., Moreno N., “Implementación de dos estrategias de control para la velocidad de un motor DC”, in: Ingeniería e Investigación Journal, Vol. 29, Num. 2, pp. 100 – 106, 2009. A2 category.

7. References

[1] Akkaya R., Kulaksiz A., and Aydogdu O., "DSP implementation of a PV system with GA-MLP-NN based MPPT controller supplying BLDC motor drive", ScientDirect Energy conversion and management, Konya Turkey, Jun. 2007, pp. 210 – 218.

[2] Arbid M., and Fellous J., "Emotions: from brain to robot", Trends in Cognitive Sciences, Vol. 8, Num. 12, Dec. 2004, pp. 554 – 561.

[3] Armony J., Servan D., Cohen J., and LeDoux J., "Computational modeling of emotion: explorations through the anatomy and physiology of fear conditioning", Trends in Cognitive Sciences, Vol. 1, Num. 1, 1997, pp. 28 – 34.

[4] Athienitis A., and Santamouris M., "Thermal Analysis and Design of Passive Solar Buildings", James & James (Science Publishers), 2002.

[5] Atthajariyakul S., and Leephakpreeda T., "Neural computing thermal comfort index for HVAC systems ", in Energy and Buildings, Vol. 39, Is. 10, Oct. 2007, pp. 1115 – 1122.

[6] Boggarpu N., and Kavanagh R., "New Learning Algorithm for High-Quality Speed Measurement and Control When Using Low-Cost Optical Encoders", IEEE Trans. on Instrumentation and Measurement, Vol. 59, Num. 3, Mar. 2010, pp. 565 – 574.

[7] Bubnicki Z., "Modern Control Theory", Springer-Verlag, 2005, 320 pp.

[8] Camras L., and Witherington D., "Dynamical systems approaches to emotional development", Elsevier, Developmental review, Vol. 25, 2005, pp. 328 – 350.

[9] Candan C., "A method for fine resolution frequency estimation from three DFT samples", IEEE Trans. on Signal Processing Letters, Vol. 18, Num. 6, Jun. 2011, pp. 351 – 354.

[10] Cañamero L., "Designing Emotions for Activity Selection in Autonomous Agents", Emotions in Humans and Artifacts, Cambridge, MA: The MIT Press, 2003, pp. 115 – 148.

[11] Chen C. T., "Linear System Theory and Design," Oxford Series in Electrical and Computer Engineering. New York Oxford, 1999, Third edition, 334 pp.

[12] Chudamani R., Vasudevan K., and Ramalingam C., "Real-time estimation of power system frequency using nonlinear least squares", IEEE Trans. on Power Delivery, Vol. 24, Num. 3, Jul. 2009, pp. 1021 – 1028.

[13] Damacio A., "Descartes' error: Emotion, Reason, and Human Brain", Ed. Penguin Books, USA, 1994.

[14] Fagerberg P., Ståhl A., and Höök K., "eMoto: emotionally engaging interaction", Personal and Ubiquitous Computing, Vol. 8, 2004, pp. 377 – 381.

[15] Fellous J., Armony J., and LeDoux J., "Emotional Circuits and Computational Neuroscience", The handbook of brain theory and neural networks, Second Edition, The MIT Press preprint, 2002, pp. 1 – 6.

[16] FNET Web Display (2012). Retrieved April 10, 2012, http://fnetpublic.utk.edu/sample_events.html

[17] Fragopanagos N., and Taylor J., "Modelling the interaction of attention and emotion", Neurocomputing, Vol. 69, 2006, pp 1977 – 1983.

[18] Friedman A., (2010, Aug 21-22). “How I (almost) saved the Earth”, The Wall Street Journal [Newspaper article]. p. W1. Available: http://www.wsj.com
[19] Fountain M., and Arens A., “Air Movement and thermal comfort,” ASHRAE Journal, pp. 26-30, Aug. 1993.
[20] Gai X., Liu S., and Zhang N., “Research of a new type intelligent controller and its application”, Intelligent Control and Automation, 2008. WCICA 2008, 7th World Congress, Jun. 2008, pp. 760 – 764.
[21] Ghorbani A., and Owrangh K., “Stacked generalization in neural networks: generalization on statistically neutral problems”, in Proc. of the International joint conference on Neural Networks IJCNN 2001, Washington USA, 2001, Vol. 3, pp. 1715 – 1720.
[22] Goodwin G., Graebe S., and Salgado M., “Control System Design”, Prentice Hall, 2000, pp. 908.
[23] Gore B., and Jarvis P., “Modeling the complexities of human performance”, Systems, Man and Cybernetics, 2005 IEEE International Conference, Vol. 2, Oct. 2005, pp. 1604 – 1609.
[24] Grecucci A., Cooper R., and Rumiati I., “Computational model of action resonance and its modulation by emotional stimulation”, Cognitive Systems Research, Vol. 8, Num. 3, 2007, pp. 143 – 160.
[25] Gupta G., Mukhopadhyay S., and Moi C., “A project based approach to teach mixed-signal embedded microcontroller for DC motor control, Proceedings”, third IEEE International workshop on electronic design, Malaysia, Jan. 2005. pp. 1 – 6.
[26] Gurney K., Prescott T., and Redgrave P., “A computational model of action selection in the basal ganglia. II”, Analysis and simulation of bahaviour, in Biological Cybernetics, Vol. 84, 2001, pp. 411 – 423.
[27] Hathaway D., ‘Solar cycle forecasting’, Space Science Reviews, 2008, 144, (1-4), pp. 401 – 412.
[28] Haykin S., “Neural Networks and Learning Machines”, Pearson, 906 pp., 2009.
[29] “Highlights of Annual 2006 Characteristics of New Housing, in US Census Bureau”. (2010, Aug 27). Available: http://www.census.gov/const/www/highanncharac2006.html
[30] IEEE Standard for Digitizing Waveform Recorders, IEEE Std. 1057TM-2007, 2008. Institute of Electrical and Electronic Engineers, 445 Hoes Lane, Piscataway, NJ 08855, USA.
[31] Ioannou P., Sun J., “Robust Adaptive Control”, Dover Publications, 2012, pp. 821.
[32] Karam M., Fadali M., and White K., “A Fourier/Hopfield neural network for identification of nonlinear periodic systems”, in Proc. of the 35th Southeastern Symposium on System Theory, Morgantown, WV, USA, 2003, pp. 53 – 57.
[33] Lantos P., ‘The skewness of a solar cycle as a precursor of the amplitude of the next’, Soar Physics, Vol. 236, Num. 1, 2006, pp.199 – 205.
[34] LeDoux J., “Emotional networks in the brain”, Handbook of emotions, Michael Lewis JMH editorial, Guilford Press, New York, NY, US, 1993, pp. 109 – 118.
[35] Lucas C., and Shahmirzadi D., “Introducing BELBIC: Brain emotional learning based intelligent controller”, Intelligent Automation and Soft Computing, Vol. 10, Num. 1, 2004, pp. 11 – 22.
[36] Lygouras J., Pachidis T., Tarchanidis K., and Kodogiannis V., “Adaptive High-Performance Speed Evaluation Based on a High-Resolution Time-to-Digital Converter”, IEEE Trans. on Instrumentation and Measurement, Vol. 57, Num. 9, Sep. 2008, pp. 2035 – 2043.
[37] Marinier R., and Laird J., “Toward a Comprehensive Computational Model of Emotions and Feelings”, 6th International Conference on Cognitive Modeling, 2004, pp. 1 – 6.

[38] Marshak A., Johnson D., Johnson J., "A Bessel Rational filter, in Circuits and Systems", IEEE Transactions on Circuit and Systems, Vol. 6, Num. 21, Nov. 1974, pp. 97 – 99.
[39] McKilliam R., Quinn B., Clarkson L., and Moran B., "Frequency estimation by phase unwrapping", IEEE Trans. on Signal Processing, Vol. 58, Num. 6, Jun. 2010, pp. 2953 – 2963.
[40] Mehrabian A., and Lucas C., "Emotional Learning based Intelligent Robust Adaptive Controller for Stable Uncertain Nonlinear Systems", International Journal of Computational Intelligence. Vol. 2, Num. 4, 2005, pp. 246 – 252.
[41] Mendes N., Oliveira G., Araujo H., and Coelho L., "A Matlab-based simulation tool for building thermal performance analysis," in 8th International IBPSA Conf., Eindhoven, Netherlands, Aug. 2003, pp. 855 – 862.
[42] Menon A., Mehrotra K., Mohan C., and Ranka S., "Characterization of a Class of Sigmoid Functions with Applications to Neural Networks", Elsevier Science Neural Networks, Vol. 9, Num. 5, Jul. 1996, pp. 819 – 835.
[43] Merry R., Molengraft R., and Steinbuch M., "Error modeling and improved position estimation for optical incremental encoders by means of time stamping", in Proc. 2007 American Control Conference, New York City, USA, 2007, pp. 3570 – 3575.
[44] Michaud F., "EMIB — Computational Architecture Based on Emotion and Motivation for Intentional Selection and Configuration of Behaviour-Producing Modules", Cognitive Science Quarterly, Cognitive Science Quarterly, Special Issue on Desires, Goals, Intentions, and Values: Computational Architectures, 2002, pp. 1 – 20.
[45] Mirmomeni M., Lucas C., Araabi B., Moshiri B., and Bidar M.R., 'Recursive spectral analysis of natural time series based on eigenvector matrix perturbation for online applications', IET Signal Process., Vol. 5, Num. 6, 2011, pp. 515 – 526.
[46] Morén J., and Balkenius C., "A Computational Model of Emotional Learning in the Amygdala", The MIT Press, Proceedings of the 6th International Conference on the Simulation of Adaptive Behavior, 2000.
[47] Morris F., Braun J., and Treado S., "Experimental and simulated performance of optimal control of building thermal storage", ASHRAE Trans., Vol. 100, Num. 1, 1994, pp. 402 – 414.
[48] Murray R., (2010, Sep 21). Panel report on Future Directions in Control, Dynamics and Systems [Online]. Available: http://www.cds.caltech.edu/%7Emurray/cdspanel/report/cdspanel-15aug02.pdf
[49] Nam S., Lee D., Kang S., Ahn S., and Choi J., 'Fundamental frequency estimation in power systems using complex Prony analysis', Journal of Electrical Engineering & Technology, 2011, 6, (2), pp. 154 – 160.
[50] Ogata K., "Modern Control Engineering", Prentice Hall, 5th Edition, 2009.
[51] Pantazis Y., Rosec O., and Stylianou Y., "Iterative estimation of sinusoidal signal parameters", IEEE Signal Processing Letters, Vol. 17, Num. 5, May. 2010, pp. 461 – 464.
[52] Payne F. Jr., "A Definition of Emotions", Journal of Biblical Ethics in Medicine – Vol. 3, Num. 4, pp 1 – 9.
[53] Peter C., and Herbon A., "Emotion representation and physiology assignments in digital systems", Interacting with Computers, Vol. 18, 2006, pp. 139 – 170.
[54] Petrella R., and Tursini M., "An Embedded System for Position and Speed Measurement Adopting Incremental Encoders", IEEE Trans. on Industry Applications, Vol. 44, Num. 5, Sep. 2008, pp. 1436 – 1444.
[55] Petrovic´ P., Stevanovic M., 'Algorithm for Fourier coefficient estimation', IET Signal Process, 2011, 5, (2), pp. 138 – 149.

[56] Posner J., Russell J., and Peterson B., "The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology", Development and Psychopathology, Vol. 17, 2005, pp. 715 – 734.
[57] Power B., "Behavior: the control of perception", Benchmark publications Inc. 2005, pp. 332.
[58] Provencher S., "Estimation of complex single-tone parameters in the DFT domain", IEEE Trans. on Signal Processing, Vol. 58, Num. 7, Jul. 2010, pp. 3879 – 3883.
[59] Roberts M. "Signals and Systems: Analysis of Signals Through Linear Systems", Mc Graw Hill, 2003, pp. 1054
[60] Rolls E., "Brain mechanisms of emotion and decision-making", International Congress Series - ICS, Elsevier, Vol. 1291, Jun. 2006, pp. 3 – 13.
[61] Russell J., and Barrett F., "Core Affect, Prototypical Emotional Episodes, and Other Things Called Emotion: Dissecting the Elephant", Journal of Personality and Social Psychology, Vol. 76, Num. 5, 1999, pp. 805 – 819.
[62] Sander D., Grandjean D., and Scherer K., "A systems approach to appraisal mechanisms in emotion", Neural Networks, Science Direct, Vol. 18, 2005, pp. 317 – 352.
[63] Scherer K., "The nature of emotion: Fundamental questions", New York: Oxford University Press. 1994, pp. 127 – 130.
[64] Scherer, K. R. Psychological models of emotion. In Joan C. Borod (Ed.), The Neuropsychology of Emotion (pp. 137–162). New York: Oxford University Press. 2000
[65] Scheutz M., "Useful Roles of Emotions in Artificial Agents: A Case Study from Artificial Life", Proceedings of AAAI 2004, AAAI Press, 2004, pp. 31 – 40.
[66] Schumacher B., Bachmann G., and Guebeli M., "Economiser tx2," in *UKACC International Conf. on Control '98,* September 1-4, 1998, Conference publication no. 455, Sep. 1998, pp. 1711 – 1716.
[67] Se-Han L., Lasky A., and Velinsky S., "Improved Speed Estimation for Low-Speed and Transient Regimes Using Low-Resolution Encoders", IEEE/ASME Trans. on Mechatronics, Vol. 9, Num. 3, Sep. 2004, pp. 553 – 560.
[68] SIDC – Solar Influences Data Analysis Center (2012). Monthly and monthly smoothed sunspot number, Retrieved April 10, 2012, http://sidc.oma.be/sunspot-data
[69] So H., and Chan F., Sun W., "Subspace approach for fast and accurate single-tone frequency estimation", IEEE Trans. on Signal Processing, Vol. 59, Num. 2, Feb. 2011, pp. 827 – 831.
[70] Sosnowski, S., Kühnlenz, K., and Buss, M., "EDDIE - An Emotion-Display with Dynamic Intuitive Expressions", The 15th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN06), Hatfield, UK, Sep. 2006, pp. 569 – 574.
[71] Suratgar A., Tavakoli B., and Hoseinabadi A., "Modified Levenberg-Marquardt Method for Neural Networks Training", World Academy of Science, Engineering and Technology, Num. 6, 2007, pp. 636 – 638.
[72] Szederkényi G., Lakner R., Gerzson M. "Intelligent Control Systems: An Introduction with Examples" (Applied Optimization, Volume 60), Kluver Academic Publishers, 2012, pp. 312.
[73] "Thermal Environmental Conditions for Human Occupancy", ANSI /ASHRAE Standard 55-2004.
[74] Tipsuwan Y., Sribye J., and Kamonsantiroj S., "An experiment study of Network – Based DC motor speed control using SANFIS", The 33rd Annual Conference of the IEEE Industrial Electronics Society, Taipei Taiwan, Nov. 2007, pp. 426 – 432.
[75] Tsuji T., Hashimoto T., Kobayashi H., Mizuochi M., and Ohnishi K., "A Wide-Range Speed Measurement Method for Motion Control", IEEE Trans. on Industrial Electronics, Vol. 56, Num. 2, Feb. 2009, pp. 510 – 519.

[76] “US Climate at a Glace”. National Environmental Satellite, Data, and Information Servis. NOAA Satellite and Information Service. 2010. (2010, Aug 27). Available: http://www.ncdc.noaa.gov
[77] “US Energy Information Administration”. Independent Statistics and Analysis. 2005 Residential Energy Consumption Survey. Release in 2008. (2010, Aug 27) Available: http://www.eia.doe.gov/
[78] Yang C., and Wei G., “A noniterative frequency estimator with rational combination of three spectrum lines”, IEEE Trans. on Signal Processing, Vol. 59, Num. 10, Oct. 2011, pp. 5065 – 5070.
[79] Zhdanov A., and Vinokurov A. “Emotions Simulation in Methodology of Autonomous Adaptive Control”, Proceedings, International Symposium on Intelligent Control/lntelligent Systems and Semiotics IEEE, Cambridge, MA, Sep. 1999, pp. 15 – 17.