# IDA-PBC Controller Tuning Using Steepest Descent

J. A. Morales(✉), M. A. Castro, D. Garcia, C. Higuera, and J. Sandoval

Tecnólogico Nacional de México, Instituto Tecnológico de La Paz, La Paz, Mexico
{jmoralesv,mcastro,diego.garcia,cesarh,jsandoval}@itlp.edu.mx

**Abstract.** The optimization of controller parameters or gains is a challenge usually approached using empirical methods that consume valuable time, without the certainty that the obtained gains actually produce the desired behaviour of the controlled plant. There are several analytical and numerical methodologies to find the parameters for PID controllers, however currently there is not enough available information regarding the application of optimization methods for nonlinear controllers. The present work describes the application of the maximum descent method to find the gains of IDA-PBC controller for a ball and beam system. The proposed methodology involves implementing a mathematical model to describe the system's dynamics, the design of a objective function to measure how closely the plant follows the desired behaviour, and finally the evaluation of a set of gains obtained by the numerical method. The dynamic model and the optimization algorithm were implemented in C language in order to reduce the computer time compared to the use of frameworks such as MATLAB. Numerical simulations to validate the effectiveness of the proposed methodology are included.

**Keywords:** Ball and beam system · Controller tuning
Nonlinear control system · Steepest descent · IDA-PBC

## 1 Introduction

Control systems theory deals with the analysis and design of components and their interaction as a system to produce a desired behaviour [1]. A key configuration in control theory is based on the essential notion of feedback, a closed-loop controller uses feedback to control the states or outputs of a dynamical system. Its name comes from the way that information flows through the system: The system's input is usually a reference point $r(t)$, that represents the desired system output, using a feedback the current system output $y(t)$ is obtained and subtracted from $r(t)$ to calculate an error signal $e(t)$ which is used by the controller to obtain a signal $u(t)$ to modify the plant's behaviour, as shown in Fig. 1.

The theory and practice of control has a wide range of applications on the fields of engineering such as aeronautics, chemistry, mechanical systems, ambient management systems, construction, electrical networks among many other disciplines. The advantages of an efficient control on industrial applications are huge,


L. Trujillo et al. (Eds.): NEO 2017, SCI 785, pp. 158–170, 2019.
https://doi.org/10.1007/978-3-319-96104-0_8

and include quality improvements, reduction of waste and energy consumption, enhanced security levels, and pollution control.

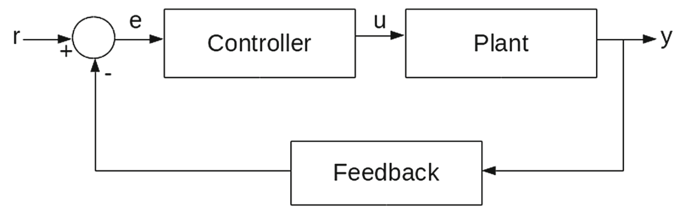



**Fig. 1.** Closed-loop control system.

The starting point of a control system analysis is the mathematical model of the plant, usually presented as an operator between the system's inputs and outputs, or as a set of partial differential equations. Most mathematical models used in control system are linear and there is a vast and robust knowledge corpus on this specific subject. However, current technological advances have generated a huge variety of new problems and applications where it's nonlinear essence is of importance. For instance, nonlinear dynamics are observed commonly on modern engineering applications such as flight command systems, robotic manipulators, automated highways, plane wing structures, boats, reaction engines, turbo-diesel motors, electrical induction motors and high performance fuel injection systems [2–4]. Such systems cannot be properly described by linear models, that is the main reason for the use of nonlinear models and the development of tools and concepts for nonlinear systems control. Interconnection and Damping Assignment (IDA) an extension of Passivity Based Control (PBC), is a successful methodology for the design of nonlinear control systems [5] that provides certain advantages compared to others. The foremost advantage is that controllers produced using this methodology are designed to achieve asymptotic stability of desired equilibrium states on nonlinear systems, without the need of linearisation or decoupling procedures [6]. Secondly, when using the Hamiltonian structure with a desired energy function, this function qualifies as a Lyapunov candidate for the desired equilibrium. Also, the IDA-PBC method is motivated by the principles of energy and damping injection, and it is demonstrated that the energy shaping and the designs of control systems based on passivity are effective to solve problems that involve underactuated mechanical systems [7]. Ultimately, the IDA-PBC method provides free parameters, which increase the number of possible solutions of the partial differential equations [5]. However a problem with this method is that it does not provide a way to adjust or tune the design parameters and the performance of the controllers that produce, depends on this values.

## 2 The IDA-PBC Method

A brief review on the IDA-PBC method applied to control a class of underactuated mechanical systems is presented (see the work done by Ortega et al. [5] for further details). The procedure starts from the Hamiltonian description of the system by means of the total energy function (kinetic plus potential energies) given by

$$H(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}(\mathbf{q})\mathbf{p} + V(\mathbf{q}), \tag{1}$$

where $\mathbf{q}$ and $\mathbf{p} \in \mathbb{R}^n$ are the vectors of generalized position and momenta, respectively, $M(\mathbf{q}) = M^T(\mathbf{q}) > 0$ is the inertia matrix and $V(\mathbf{q})$ is the potential energy [8]. If we assume that the system has not natural damping, the equations of motion can be written as

$$\frac{d}{dt}\begin{bmatrix}\mathbf{q}\\ \mathbf{p}\end{bmatrix} = \begin{bmatrix}0_{n\times n} & I_{n\times n}\\ -I_{n\times n} & 0\end{bmatrix}\begin{bmatrix}\nabla_{\mathbf{q}} H\\ \nabla_{\mathbf{p}} H\end{bmatrix} + \begin{bmatrix}0_{n\times m}\\ G\end{bmatrix}\mathbf{u}, \tag{2}$$

being $\mathbf{u} \in \mathbb{R}^m$, the vector of control inputs and $G \in \mathbb{R}^{n\times m}$, with $\text{rank}(G) = m$ where $n > m$ for underactuated systems.

The IDA-PBC methods assigns a particular desired structure in closed-loop system, with desired energy function given by

$$H_d(\mathbf{q},\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M_d(\mathbf{q})^{-1}\mathbf{p} + V_d(\mathbf{q}), \tag{3}$$

where $M_d(\mathbf{q}) = M_d^T(\mathbf{q}) > 0$ and $V_d(\mathbf{q})$ are the desired inertia matrix and the desired potential energy function, respectively.

The desired closed-loop system is

$$\frac{d}{dt}\begin{bmatrix}\mathbf{q}\\ \mathbf{p}\end{bmatrix} = \begin{bmatrix}0_{n\times n} & M^{-1}M_d\\ -M_d M^{-1} & J_2(q,p) - GK_vG^T\end{bmatrix}\begin{bmatrix}\nabla_q H_d\\ \nabla_p H_d\end{bmatrix}, \tag{4}$$

where $J_2 = -J_2^T$ and $K_v = K_v^T$ are free matrices. Now, for this class of systems, the main challenge of the IDA-PBC method consists in solving the following set of partial differential equations, called matching equations

$$G^{\perp}[\nabla_q(\mathbf{p}^T M^{-1}\mathbf{p}) - M_d M^{-1}\nabla_q(\mathbf{p}^T M_d^{-1}\mathbf{p}) + 2J_2 M_d^{-1}\mathbf{p}] = \mathbf{0}, \tag{5}$$

and

$$G^{\perp}[\nabla_q V - M_d M^{-1}\nabla_q V_d] = \mathbf{0}, \tag{6}$$

where $G^{\perp} \in \mathbb{R}^{(n-m)\times n}$, such as $G^{\perp}G = 0_{(n-m)\times m}$ whose solutions $M_d$ and $V_d$ produce the control law given by

$$\mathbf{u} = (G^TG)^{-1}G^T[\nabla_q H - M_d M^{-1}\nabla_q H_d + J_2 M_d^{-1}\mathbf{p}] - K_v G^T M_d^{-1}\mathbf{p}. \tag{7}$$

Furthermore, if $M_d$ is a positive definite in a neighborhood of $\mathbf{q}^*$ and

$$\mathbf{q}^* = argmin(V_d), \tag{8}$$

then $[\mathbf{q}^T\mathbf{p}^T]^T = [\mathbf{q}^{*T}0^T]^T$ is a stable equilibrium of the closed-loop desired system, with a Lyapunov function $H_d$. This equilibrium is asymptotically stable if it is locally detectable from the output $G^T\nabla_p H_d$.

## 3 Ball and Beam System

The ball and beam system is one of the most popular and important benchmark systems for studying control systems. Several classical and modern control methods have been used to stabilize the ball and beam system [9,10]. The sensor placed on one side of the beam detects the ball role along the beam and its position. An actuator drives the beam to a desired angle by applying a torque at the end of the beam. Figure 2 shows the ball and beam system (Quanser Model SRV02 and BB01) which is utilized in this research work. The controller regulates the ball position by moving the beam using the motor and overcoming the disturbances. The ball and beam system is an inherently unstable system. In other words, the ball position can be changed without limit for a fixed input of beam angle. This property has made the ball and beam system a suitable device to test different control strategies. The ball and beam system has 2 Degrees-of-Freedom (DOFs). The ball is assumed to have friction, rotary moment of inertial and coriolis acceleration during motion on the beam. However, some of the dynamic properties are neglected in most of the research work regarding the ball and beam mechanism in order to simplify the dynamic equation of the system [11,12], a linear approximation is not accurate when the angle of the beam is appreciable. Thus a more advanced control technique such as nonlinear control like IDA-PBC should work better (Table 1).

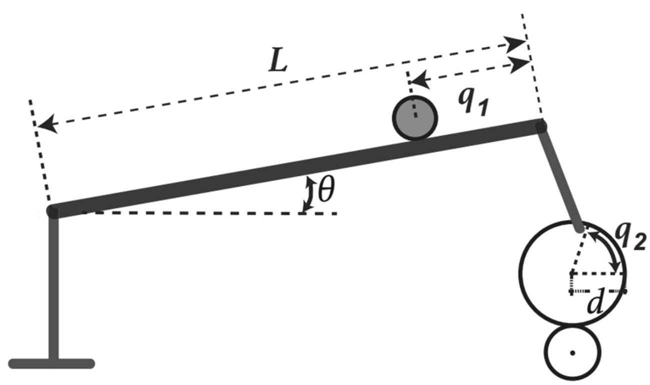



**Fig. 2.** Ball and beam system.

The behaviour of the system in its Hamiltonian structure (2) can be followed in [13,14] and is described by the following definitions, with the values $a_1 = m_B + \frac{J_B}{R_B^2}$, $a_2 = J_b \frac{d^2}{L^2} + J_{fw}$, $a_3 = m_B \frac{d^2}{L^2}$, $a_4 = m_B g$ and $a_5 = m_b g \frac{L}{2}$, where $\mathbf{q} = [q_1, q_2]^T$ $\mathbf{p} = [p_1, p_2]^T$,

$$M(q_1) = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 + m_B \left(\frac{L}{2} + q_1\right)^2 \end{bmatrix}, \tag{9}$$

$$V(\mathbf{q}) = a_4\left(\frac{L}{2}+q_1\right) + sin(q_2) + a_5 sin(q_2) \tag{10}$$

and

$$G = [0, 1]^T. \tag{11}$$

**Table 1.** Ball and beam system physical parameters.

| Symbol | Description | Units |
|---|---|---|
| $m_b$ | Ball mass | 0.15 kg |
| $J_b$ | Inertia of the beam | 0.009 kg m$^2$ |
| $J_{fw}$ | Inertia of the ball | 0.002 kg m$^2$ |
| $L$ | Length of the beam | 0.4255 m |
| $d$ | Coupling union radius | 0.0254 m |
| $R_B$ | Ball radius | 0.0127 m |
| $m_B$ | Ball mass | 0.064 kg |

## 4 An IDA-PBC for the Ball and Beam System

The control objective is to bring the ball to a desired position over the beam, which must remain horizontal. Formally, the control objective is established as follows

$$\lim_{t\to\infty}\begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} = \begin{bmatrix} q_{1d} \\ 0 \end{bmatrix}. \tag{12}$$

Muralidharam et al. [13] designed an IDA-PBC control law to a ball and beam system that satisfies the control objective (12), where $M_d$ y $V_d$ are given by

$$M_d = \begin{bmatrix} k_1 a_1\left(\frac{L}{2}+q_1\right) & a_1\left(a_2 + a_3\left(\frac{L}{2}+q_1\right)^2\right) \\ a_1\left(a_2 + a_3\left(\frac{L}{2}+q_1\right)^2\right) & k_4 \end{bmatrix} \tag{13}$$

and

$$V(\mathbf{q}) = \frac{a_4}{a_1}[1 - cos(q_2)] + \frac{k_p}{2}\left(q_2 - \frac{k_2}{k_1} log\left(\frac{L + 2[q_1 - q_{d1} - q_2]}{L}\right)\right)^2, \tag{14}$$

with $k_1 = \sqrt{2a_1a_3}$, and $k_4$ y $k_p$ are free parameters strictly positive.

$$J_d = \begin{bmatrix} 0 & j_2(\mathbf{q},\mathbf{p}) \\ -j_2(\mathbf{q},\mathbf{p}) & 0 \end{bmatrix}, \tag{15}$$

$$\alpha = [\alpha_1; \alpha_2]^T, \tag{16}$$

$$W = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \tag{17}$$

$$\alpha_1 = 2a_3\lambda_2 \left( q_1 + \frac{L}{2} \right) (\lambda_4 - \lambda_1) \tag{18}$$

and

$$\alpha_2 = a_3 \left( q_1 + \frac{L}{2} \right) \lambda_4^2. \tag{19}$$

Free matrix $J_d$ is defined by (15) where $j_2 = -\mathbf{p}^T M_d^{-1} \alpha(\mathbf{q}, \mathbf{p}) W$, $\alpha$ by (16), $W$ is given by (17), $\alpha_1(\mathbf{q}, \mathbf{p})$ and $\alpha_2(\mathbf{q}, \mathbf{p})$ by (18), (19) respectively.

The functions $\lambda_i$ are as follows

$$\lambda_1 = k_1 \left( q_1 + \frac{L}{2} \right), \lambda_2 = k_2, \lambda_4 = \frac{k_4}{a_2 + a_3 \left( q_1 + \frac{L}{2} \right)^2}, \tag{20}$$

where $k_1 > 0$ y $k_2 = k_1 \sqrt{a_1/(2a_3)}$.

In [14] $q_{d1}$ is introduced in $V_d$ (14), what extends the results of Muralidharan [13] where only $q_{d1} = 0$ is considered.

Using Eqs. (13) and (14), the control input $\mathbf{u}$ can be obtained by (7) and the system (2) can be simulated.

Once the control law is obtained, remains the problem of assigning suitable values to the free parameters $k_4$, $k_p$ and $k_v$ in the control law, to ensure the control objective (12) with good performance, this procedure is known as tuning the controller. A problem that can arise with this kind of controller is that a bad tuning may result in a poor performance or even lead to instability.

## 5 Tuning of the IDA-PBC Controller

There are many tuning methods to determine the gains of PID controllers to obtain good performance and robustness [15]. Some methods with simple formulas use little information of the plant's dynamics resulting in moderate performance and a re-tune process by trial and error depending on those results. More sophisticated tuning method can get rise to considerable improvements in performance, but they are also more computationally demanding and depend on more information of the plant's dynamics [16]. However, there are no methods in the literature for nonlinear controllers, such as the one presented in this work.

We consider the tuning of the control law as an optimization problem. To do this we need the following:

1. Select a performance index to be used as objective function.
2. Select an optimization algorithm.

For the purpose of selecting a performance index we can consider the rise time (the time needed by the control system to reach the desired value after a perturbation), peak overshoot (the highest value reached by the response before reaching the desired value) and settling time.

However, there are other criteria that take into account the transitory regime of the solution, to get a finer tuning and improve the response of the controller. Most of these criteria consider the tracking error as the current value of the position $\mathbf{q}(t)$ minus the desired value $\mathbf{q_d}$, in this particular problem $\mathbf{q_d}, \mathbf{q}(t) \in \mathbb{R}^2$, so the error $\mathbf{e}(t)$ is defined as

$$\mathbf{e}(t) = \begin{bmatrix} q_1(t) - q_{1d} \\ q_2(t) \end{bmatrix}. \tag{21}$$

A performance index commonly applied in control field is the integrated square error or ISE [17], defined as shown below

$$ISE = \int_0^\infty \|\mathbf{e}(t)\|^2 dt. \tag{22}$$

In this performance index, larger errors contribute a lot in the integrate, this tends to give solutions with small overshoots, but unfortunately it is insensitive to small errors. A variation of this index, that was used in this work is the integrate time-square error, given by

$$ISTE = \int_0^\infty \|\mathbf{e}(t)\|^2 t^2 dt, \tag{23}$$

which takes account of errors that remain in time, thus penalizing oscillatory solutions [18].

It was decided to use a numerical method because they are guaranteed to produce at least local optimum when they are used in smooth functions. In contrast, heuristic methods, are designed to produce good answers quickly, but they are not guarantee of finding a true optimum.

The applied numerical method was gradient descent, a first-order iterative algorithm, that consists in taking steps proportional to the negative of the function's gradient at the current point.

The reason to apply this method, in spite of its low order of convergence, is that newtonians or quasinewtonians algorithms would have required the implementation of a numerical estimate of a Hessian matrix or many evaluations of the aptitude functions which was considered expensive and unstable.

Gradient descent is based on the observation that if a multivariable function $F(\mathbf{x})$ is defined and differentiable in a neighborhood of a point $a$, then $F(\mathbf{x})$ decreases fastest if one goes from $a$ to the direction of the negative gradient of $F(\mathbf{x})$ at $a$. It follows that, for small enough values of $\gamma$, (24) is fulfilled and $F(a_n) \geq F(a_{n+1})$ [19].

$$a_{n+1} = a_n - \gamma_n \nabla F(a_n). \tag{24}$$

Taking this observation into account, the algorithm starts with a guess $\mathbf{x}_0$ for a local minimum of $F$, and considers the sequence $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2 ...$ such that the following is fulfilled

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma_n \nabla F(\mathbf{x}_n), \quad n \geq 0, \tag{25}$$

and $F(\mathbf{x}_n)$ in the sequence

$$F(\mathbf{x}_0) \geq F(\mathbf{x}_1) \geq F(\mathbf{x}_2) \geq \ldots \geq F(\mathbf{x}_n) \tag{26}$$

converges to a minimum. Note that the value of step size $\gamma$ is allowed to change at each iteration.

When the step length $\gamma_k$ is computed, a trade-off must be solved. It is desirable to choose $\gamma_k$ to achieve a substantial reduction of $F$, but at the same time, it is important to limit the time consumed making the choice. The ideal choice would be the global minimizer of the univariable function

$$\phi(\gamma) = F(\mathbf{x}_k + \gamma \nabla F(\mathbf{x}_n)), \tag{27}$$

but in general, it is too expensive to identify this value. A simple condition we could impose on $\gamma_k$ is that it provides a reduction in $F(\mathbf{x})$ as shown in

$$F(\mathbf{x}_k + \gamma_k \nabla F(\mathbf{x}_k)) < F(\mathbf{x}_k). \tag{28}$$

To find adequate values to $\gamma$ in each iteration, we used a so-called backtracking approach. In its most basic form, backtracking proceeds as shown in Algorithm 1.

**Algorithm 1.** Backtracking.

```
Choose γ_0 > 0, ρ ∈ (0,1)
γ = γ_0
repeat
   γ = ργ
until F(x_k + γ∇F(x_k)) < F(x_k)
return  γ_k = γ
```

Once solved the problem of finding $\gamma$ in each iteration, we can formulate the gradient descent Algorithm 2.

**Algorithm 2.** Gradient descent.

```
Choose x_0, set ε > 0, k = 1, x_k = x_0.
while ∇F(x_k) >= ε do
   d_k = −∇F(x_k)
   Choose γ using Algorithm 1
   x_{k+1} = x_k + γd_k
end while
return  x_k
```

For the particular problem of tuning the ball and beam controller, an objective function of the form (23) was defined as

$$F(k_4, k_p, k_v) = \int_0^T \|\mathbf{e}(t)\|^2 t^2 dt. \tag{29}$$

Since the control law is continuous when $k_4, k_p, k_v > 0$ and $M_d$, $M$ are invertible, $\mathbf{q}(t)$ is bounded because the control objective (12) is achieved. So we can guess $F$ is a smooth function, thus, can be minimized locally using gradient descent.

However, we can not assure that $F$ has just one optimum point, so a suitable approach consists in generating multiple initial random values for the vector $\mathbf{x}_0 = [k_4, k_p, k_v]$, gradient descent is used to find the local optimum for each of the starting points and the best of them is considered as the solution.

## 6 Implementation

The evaluation of (29) requires to know the values of $\mathbf{q}$ over a time interval $(0, T)$, so a full simulation of the system's dynamic is needed for each vector $\mathbf{x}_i = [k_4, k_p, k_v]^T$. In order to solve the set of differential equations involved, a fourth order Runge-Kutta (RK) numerical method was implemented, this algorithm guarantees an error of $\Delta t^5$ order for each step of size $\Delta t$ [20].

To compute the next vector $\mathbf{x}_{i+1}$, the gradient of (29) must be estimated, given the complexity of the analytical solution, a divided centred differences method was used, this ensures an numerical error proportional to the square of the step size $\epsilon$ used in Algorithm 2.

Since this is a multi-modal problem, it is adequate to perform several executions of the descent using different starting points, a parallel implementation of the algorithm was used to take advantage of the modern processor architecture by evaluation simultaneous starting points.

The program was written in C language and compiled using the GCC 4.8.4 compiler collection, the message passing library OpenMPI 1.8.4 was used to implement process synchronization. The presented results were obtained using a workstation with i7-3770 processor and 4 GB of RAM, running a 64 bits LinuxMint 18.2 installation.

The simulation period limit $T$ was 3 s with initial resting conditions $q_1(0) = 0.2\,m$, $q_2(0) = -1$ rad and a maximum $1.2N$ in each components of the requested control signal vector was used.

The RK parameters used were $\epsilon = 1e^{-5}$ and $\Delta t = 1e^{-4}$. For the backtracking algorithm, the proposed values in [19], $\gamma_0 = 1$ and $\rho = 0.5$ were used.

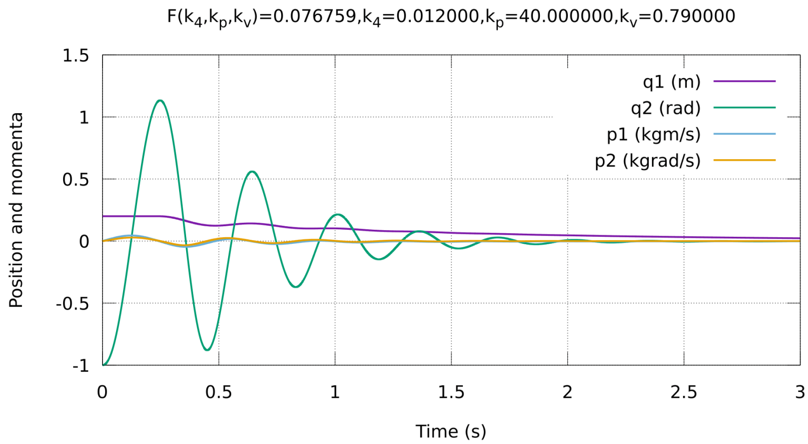
The parallelization strategy was to execute eight processes (one per every processor's virtual core) each one applied the gradient descent to ten different random starting position vectors $\mathbf{x_0}$, finally the best result of each process is consolidated on the rank 0 process and displayed as the final result.

## 7 Results

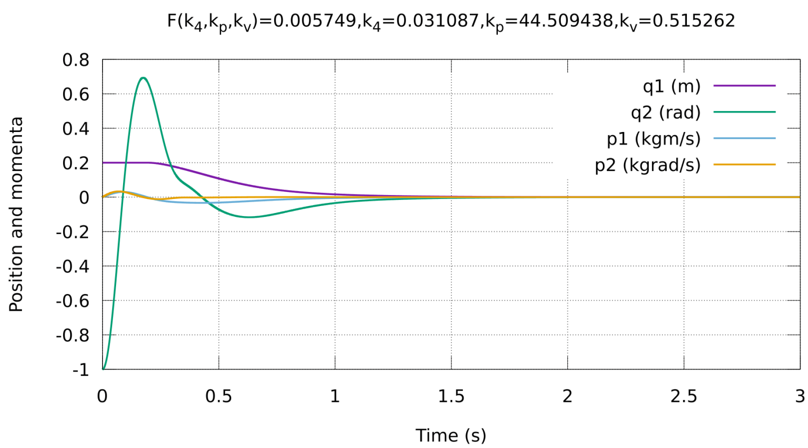
The obtained results were compared against the performance in simulation of the set of gains empirically obtained presented in [14]. Table 2 shows a comparison of the obtained sets of gains using each method.

**Table 2.** Comparison of gains.

| Tuning method | $k_4$ | $k_p$ | $k_v$ | $F(k_4, k_p, k_v)$ |
|---|---|---|---|---|
| Empyrical | 0.012 | 40.0 | 0.795 | 0.076759 |
| Gradient descent | 0.031087 | 44.509438 | 0.515262 | 0.005749 |



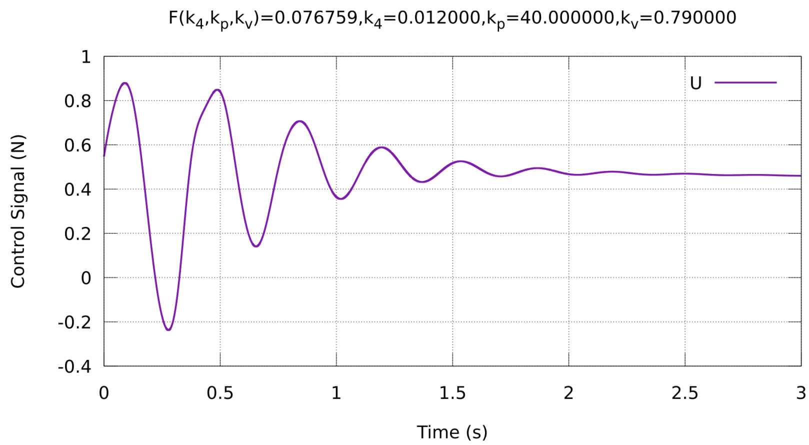

(a) Empirical.



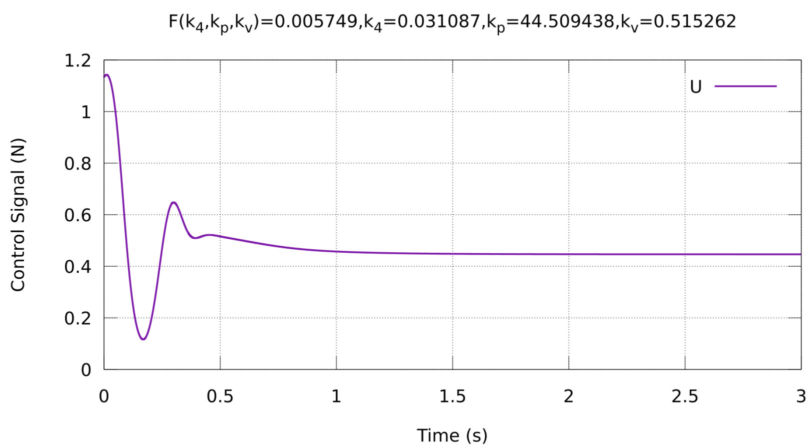

(b) Gradient descent.

**Fig. 3.** Position and momenta behaviour.

The states and momenta behaviour of the closed loop system using the empirically obtained gains in [14] and the proposed methodology is shown on Fig. 3a and b. Also the control signal required by the controller using each set of gains is depicted on Fig. 4a and b respectively.




(a) Empirical.




(b) Gradient descent.

**Fig. 4.** Control signal behaviour.

## 8 Conclusions and Future Work

The proposed methodology obtains a set of gains that produces a superior behaviour of the simulated system in overshoot reduction and setting time with less oscillations than the gains obtained in [14].

It is also of notice that the range of control signals produced by the tuning process does not exceed the nominal value of the actuator.

The methodology of assuming that the tuning problem is equivalent to the optimization of a soft function and that therefore it is suitable to solve by using numerical methods seems to be valid for this particular ball and beam system.

The implementation of this methodology on different plants for its validation remains as future work, as well as the evaluation of different objective functions and optimization strategies.

## References


1. Khalil, H.K.: Nonlinear Systems, 3rd edn. Prentice Hall, Harlow (2002)
2. Zúñiga, J.: Control de un Robot SCARA Basado en Pasividad. Maestría en ciencias en ingeniería electrónica, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Morelos, México (2016)
3. García, D., Sandoval, J., Gutiérrez-Jagüey, J., Bugarin, E.: Control IDA-PBC de un Vehículo Submarino Subactuado. Revista Iberoamericana de Automática e Informática industrial **15**, 36 (2017)
4. González-Cabezas, H.A., Duarte-Mermoud, M.A.: Control de velocidad para motores de inducción usando IDA-PBC. Revista Chilena de Ingeniería **446**, 81–90 (2005)
5. Ortega, R., Spong, M.W., Gómez-Estern, F., Blankenstein, G.: Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment. IEEE Trans. Autom. Control **47**(8), 1218–1233 (2002)
6. Acosta, J.Á., Ortega, R., Astolfi, A., Mahindrakar, A.D.: Interconnection and damping assignment passivity-based control of mechanical systems with underactuation degree one. IEEE Trans. Autom. Control **50**(12), 1936–1955 (2005)
7. Gómez-Estern, F., Van der Schaft, A.: Physical damping in IDA-PBC controlled underactuated mechanical systems. Eur. J. Control **10**(5), 451–468 (2004)
8. De-León-Gómez, V., Santibañez, V., Sandoval, J.: Interconnection and damping assignment passivity-based control for a compass-like biped robot. Int. J. Adv. Rob. Syst. **14**(4), 1–18 (2017). https://doi.org/10.1177/1729881417716593
9. Wang, W.: Control of a ball and beam system. Ph.D. thesis (2007)
10. López, F., Monroy, P., Rairán, J.D.: Control de posición de un sistema bola y viga con actuadores magnéticos. Tecnura **15**, 12–23 (2011)
11. Ortiz, E., Liu, W.Y.: Modelado y control PD-Difuso en tiempo real para el sistema barra-esfera. In: Congreso anual de la AMCA, pp. 1–6 (2004)
12. Meneses Morales, P., Zafra Siancas, H. D.: Diseño e implementación de un módulo educativo para el control de sistema Bola y Varilla. PhD thesis, Pontificia Universidad Católica del Perú (2013)
13. Muralidharan, V., Anantharaman, S., Mahindrakar, A.D.: Asymptotic stabilisation of the ball and beam system: design of energy-based control law and experimental results. Int. J. Control **83**(6), 1193–1198 (2010)
14. Higuera, C., Chargoy, A., Sandoval, J., Bugarin, E., Coria, L.N.: IDA-PBC en Voltaje para la Regulación de un Sistema Barra-Bola Experimental. In: Congreso Nacional de Control Automático, Monterrey, Nuevo León, Mexico, pp. 1–6 (2017)
15. Morales Viscaya, J.A., Rochín Ramírez, A., Castro Liera, M.A., Sandoval Galarza, J.A.: AG y PSO como métodos de sintonía de un PID para el control de velocidad de un motor CD. In: Castro Liera, I., Cortés Larrinaga, M. (eds.) Nuevos avances en robótica y computación, 1st edn. pp. 81–87. ITLP, La Paz (2015)
16. Wu, H., Su, W., Liu, Z.: PID controllers: design and tuning methods. In: Proceedings of the 2014 9th IEEE Conference on Industrial Electronics and Applications, ICIEA 2014 (2014)
17. Ogata, K.: Modern Control Engineering, 5th edn. Pearson, New York (2016)

18. Schultz, W.C., Rideout, V.C.: Control system performance measures: past, present, and future. IRE Trans. Autom. Control **AC–6**, 22–35 (1961)
19. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, New York (2006)
20. Nieves Hurtado, A., Domínguez Sánchez, F.C.: Métodos numéricos: aplicados a la ingeniería. 4th edn. Grupo Editorial Patria (2012)