International Journal of Engineering Education (IJEE)

e-ISSN : 2540-9808

Virtual Laboratories as Strategy for Teaching Improvement in Math Sciences and Engineering in Bolivia

Francisco Vargas

UNIARA Universidade de Araraquara - SP - Brazil

e-mail: trivenoj@yahoo.com.br

Abstract - *The vertiginous technological advancement has made necessary the use of computer software that contributes to the improvement of teaching in math sciences and engineering. It is in this context that the last five years the strategy presented in this article has been disseminated in the main universities of Bolivia, a country where the schools have not yet been able to offer basic disciplines such as calculus, matrix algebra, physics and/or differential equations to solve problems considering applicative aspects. To establish this connection, it is necessary to deduce differential equations associated with practical problems, solve these equations with different numerical algorithms, and establish the concept of simulation to later introduce languages like Python/VPython free of license to elaborate Virtual Laboratories that allow obtaining the solutions in two and three dimensions. The classical problems addressed for this purpose are the satellite of two degrees of freedom and the inverted pendulum.*

Keywords – *Numerical algorithms, Computer software, Simulation, Satellite control.*

1. Introduction

Considering the vertiginous technological advance and looking for a high level of competi- tiveness in the professional market, several universities in the world have introduced computer tools in their teaching programs (Matlab-Simulink®, Mathematica™), what has allowed to address a wide type of problems and increase their level of understanding not only in math sciences and engineering but also in other areas [Gilat, 2006], [Lopez and Sanches, 2011]. In this sense, this paper aims to present the strategy that has been disseminated in the main universities of Bolivia.

1.1 Geographical location

Bolivia is located in the central-western part of the South Amer- ican continent. With an area of 1.098,581 square kilometers, the country has approximately the size of Texas and California together, or twice the size of Spain. The country is bordered by Brazil to the north and east, Paraguay to the southeast, Argentina to the south, Chile to the southwest and Peru to the northwest. It consists of 9 departments with their capitals and provinces. Each of the departments has one state university and around 4 or more privative universities.

1.2 Bolivian problem

More than five years ago, different activities have been executed with several Bolivian universities (advanced courses, symposia and recently, master's degrees courses) [Vargas, 2011], [Vargas, 2013], [Vargas, 2014], [Vargas, 2016], [Vargas, 2017].

As a result of this interaction, it has become clear that some universities face several problems, among them, the lack of funds for the purchase of specialized software, at the same time they have been slow in updating their study programs and in the application of modern techniques in higher education. These and other aspects do not allow universities to be much more innovative. These problems increase due to structural failures in secondary education (such as reduced hours in math sciences) and successive changes of methodology in normal schools where teachers of basic cycle are trained [Vargas, 2018].

The strategy presented in this document was disseminated in some Bolivian universities as a proposal to improve teaching in math sciences and engineering. In the proposal were also considered the comments of students such as, what do I do with so much mathematics?, Calculus I, II, III , differential equations, numerical methods?, where do I apply all this?. These questions are asked because math science disciplines are still administered without a link of

52

applicative character. This work tries to establish this link as the main contribution to Bolivian universities. [Vargas, 2018].

1.3 The proposal strategy

Figure 1 illustrates the chronological sequence of the strategy that starts from the basic cycle that contains disciplines such as calculus and physics, an intermediate cycle that provides knowledge in numerical methods, automatic control laws and software languages such as Python/VPhyton. In sequence there are advanced concepts such as simulation and visualization and interpretation of the obtained results.

For the dissemination of the strategy the problems of the satellite of two degrees of freedom and inverted pendulum are used, from its mathematical conception until the visualization of simulation results in two and three dimensions.

The paper is organized as follows: in the section 2, basic concepts such as Lagrange and the mathematical representation of systems are presented. In the section 3 a summary of the software Python/VPython is made. In the section 4 and the section 5 the problems of the satellite of two degrees of freedom and the inverted pendulum are presented. The control problem is presented in the section 6. Finally, in the section 7 some conclusions are presented.
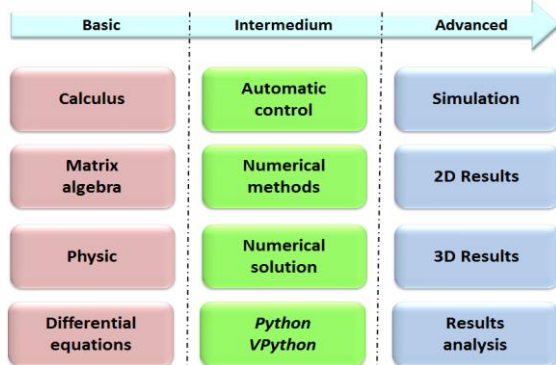

Figure 1. Chronological sequence strategy

2. Basic concepts

Most of the phenomena that occur in nature involve the variation of one quantity in re- lation to another, leading naturally to mathematical models based on differential equations [Kupferman, 2008]. To obtain these equations, physical laws such as: the second New- ton's law, the law of conservation of energy, the Kirchhoff's law, or those corresponding to a particular phenomenon are used. Some of these basic concepts are presented below.

2.1 Euler-Lagrange equations

A dynamic system with n degrees of freedom with generalized coordinates $q \in R^n$ and external forces $Q_l \in R^n$ is described by the following Euler-Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}(\dot{q}, q)\right) - \frac{\partial \mathcal{L}}{\partial q}(\dot{q}, q) = Q_\ell \quad (1)$$

$$\mathcal{L} = \mathcal{T}(\dot{q}, q) - \mathcal{V}(q) \quad (2)$$

Corresponds to Lagrange function, T (q˙, q) is the kinetic energy given by:

$$\mathcal{T}(\dot{q}, q) = \frac{1}{2}\dot{q}^T D(q)\dot{q} \quad (3)$$

where $D(q) \in R^{n \times n}$, is the generalized inertia matrix what satisfies D(q) = DT (q) > 0 and (q), is potential energy limited by (q) c, such that $c \in R$.

The external forces considered are: friction, control action and the interaction of the system with the environment. The control vector considered is $u \in R^n$. Friction forces are represented as $\partial F/\partial \dot{q}$ (q˙), where F(q˙) is the function of Rayleigh that satisfies the following restriction:

$$\dot{q}^T \frac{\partial \mathcal{F}}{\partial \dot{q}}(\dot{q}) \geq 0 \quad (4)$$

In summary, we have the following external forces:

$$Q_\ell = -\dot{q}^T \frac{\partial \mathcal{F}}{\partial \dot{q}}(\dot{q}) + Q_\epsilon + u \quad (5)$$

where Q_ϵ is the signal that contains external perturbations.

2.2 Mathematical representation

The obtained differential equations are represented like as:

$$\begin{aligned}
\dot{x}_1 &= f_1(t, x_1, x_2, \ldots, x_n, u_1, \ldots, u_m) \\
\dot{x}_2 &= f_2(t, x_1, x_2, \ldots, x_n, u_1, \ldots, u_m) \\
&\vdots \\
\dot{x}_n &= f_n(t, x_1, x_2, \ldots, x_n, u_1, \ldots, u_m)
\end{aligned} \quad (6)$$

where x˙i denotes the derivative of xi related at the time t while, u1, u2, . . . , um, are the control variables and x1, x2, . . . , xn, correspond to the state variables; the output $y \in R^p$ is given by:

$$\begin{aligned}
y_1 &= h_1(t, x_1, x_2, \ldots, x_n) \\
y_2 &= h_2(t, x_1, x_2, \ldots, x_n) \\
&\vdots \\
y_p &= h_p(t, x_1, x_2, \ldots, x_n)
\end{aligned} \quad (7)$$

In a compact way, the equation (6) is written as:

$$\dot{x} = f(t, x, u) \quad (8)$$
$$y = h(t, x, u) \quad (9)$$

with $x \in R^n$ and $u \in R^m$.

2.3 Equilibrium points

The equilibrium points are solutions of the equations (8) and (9) such as:

Definition 2.1. The point x = x∗ in the state space is a equilibrium point of x˙ = f (t, x) if:

$$x(t_0) = x^* : x(t) = x^*; \ \forall t \geq t_0 \quad (10)$$

in another form, the equilibrium point is (x∗, u∗).

53

In simple problems (like the pendulum), the solution of the equation (10) is easy [Khalil, 2002]. However, when the degree of difficulty increases, extracting operating conditions is also more complex, an aircraft is an example [Vargas et al., 2016].

2.4 Linearization

The linearization at a point of operation is a common practice in engineering to analyze nonlinear systems. Once the linear model is obtained, its analysis is carried out through the consecrated tools [Ogata, 2011]. However, the approximation around an operation point offers the local behavior of the non-linear system and not the global behavior in which we are interested. One of the most common methods to perform the linearization is the expansion by series of Taylor, effective for small variations around the operation point [Duke et al., 1988], [Gangsaas et al., 2008].

3. Python and VPython Languages

Python is a high-level interpreted language with sequential commands developed at the end of the 1980s by Guido van Rossum at the National Research Institute of Mathematics and Computer Science in the Netherlands [Kusalaas, 2013]. This language is widely used in large organizations due to its multiple programming paradigms. It provides standard libraries that include areas such as: chain operations, internet, web service tools, operational system interfaces and protocols. The Python interpreter is developed under an open source license, which makes it free to use and distribute, even for commercial purposes [Hetland, 2008]. This last characteristic makes it ideal for its use, since it dispenses the license payment reducing costs in universities and industries. Fig. 2 presents a Python code example (corresponds to the pendulum [Vargas and Paglione, 2015]).

Figures 3(a) and 3(b) allow to verify the behavior of a pendulum in two dimensions. In sequence to the development of the language Python in the spring and summer of the year 2000, with the help of David Andersen, Ruth Chabay, Ari Heitner, Ian Peters and Bruce Sherwood, David Scherer creates Visual, a module for Python that was not only relatively simple to use, but also allowed the development of objects in three dimensions. The combination of Python and Visual is called VPython which allows users to create objects such as spheres, cylinders and 3D cones and displays them in a window, which facilitates the creation of three-dimensional visualizations. This last feature has made VPython an essential tool for the creation of virtual laboratories [Villate, 2018]. The three-dimensional representation of the simple pendulum is illustrated in Fig. 4.

```
from numpy import *
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# Parameters
g = 9.8; L = 0.1; m = 0.2; u = 0.0; k = 0.1

def eqs_mov(state, t, p): # dy/dt = f(t,y)
    y1, y2 = state
    dydt = np.zeros_like(state)
    dydt = [y2,
    -(g/L)*np.sin(y1)-(k/m)*y2 + u/(m*L**2)]
    return dydt

# Initial Conditions
p = [g, L, k, u, m]
state = [2., 0]
t = np.linspace(0,10,300)
print(t)
sol = odeint(eqs_mov, state, t, args=(p,))
```

Figure 2. Python code for pendulum model

(a) Angular position and velocity   (b) Phase diagram

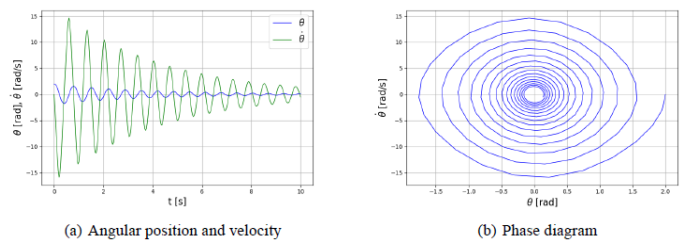Figure 3. Pendulum behavior

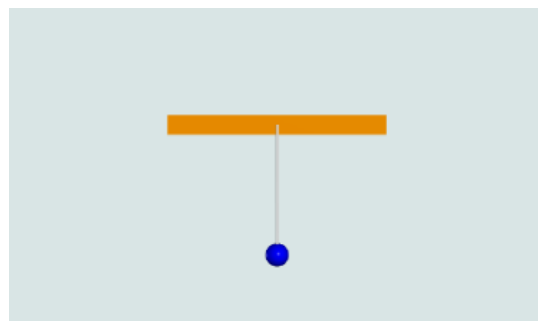Figure 4. Virtual environment for pendulum

4. Inverted Pendulum Problem

The inverted pendulum is an inherently unstable system with a highly non-linear dynamic. It is a system that belongs to the class of sub-actuated mechanical systems. The dynamics of the inverted pendulum emulates the dynamics of other systems, such as: the control of orientation of a rocket or a satellite, the automatic landing system of aircraft among others [Yue et al., 2018]. The problem is illustrated in the Figure. 5.
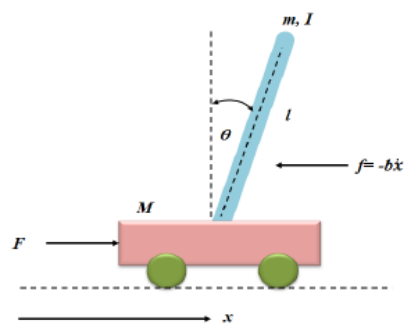

Figure 5. Inverted pendulum

54

The variables correspond to a mass bar m, inertia I and length A, articulated in a M mass cart with wheels that generate friction proportional to the linear velocity $\dot{x}$. The angle θ, measured from the vertical position, in the clockwise direction. A force, F is required to push the cart horizontally.

4.1 Mathematical model

The kinetic energies T1 and T2 of both masses are defined as:

$$T_1 = \frac{1}{2}M\dot{x}^2 \quad (11)$$

$$T_2 = \frac{1}{2}mv_2^2 + \frac{1}{2}I\dot{\theta}^2 \quad (12)$$

In the equation (12), v2 is defined from the horizontal position ph = x + A sin θ and vertical position pv = A cos θ of m given by:

$$v_2^2 = \left(\frac{dp_h}{dt}\right)^2 + \left(\frac{dp_v}{dt}\right)^2 \quad (13)$$

Deriving the horizontal and vertical positions in relation to the time is obtained:

$$v_2^2 = \dot{x}^2 + 2\ell\dot{x}\dot{\theta}\cos(\theta) + \ell^2\dot{\theta}^2 \quad (14)$$

Substituting the equation (14) in the equation (12), the kinetic energy is defined as:

$$T_2 = \frac{1}{2}m\left(\dot{x}^2 + 2\ell\dot{x}\dot{\theta}\cos(\theta) + \ell^2\dot{\theta}^2\right) + \frac{1}{2}I\dot{\theta}^2 \quad (15)$$

The equations (11), (15) and (2) allow to obtain the mathematical representation of inverted pendulum.

4.2 Computational solution

The presented results correspond to those obtained using the nonlinear model of inverted pendulum, described in detail in A.1. Figure 6(a) presents the results for θ and $\dot{\theta}$. It is verified that both are oscillatory.

Figure 6(b) illustrates the results of x and $\dot{x}$. The instability of both variables is verified. In these simulations it is possible to modify the parameters of the pendulum, the integration method, the initial conditions and others, allowing the students simulate different conditions. Finally, Fig. 7 illustrates the virtual environment for the inverted pendulum.
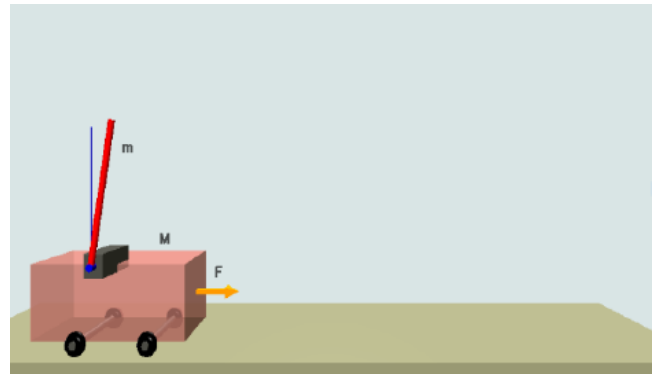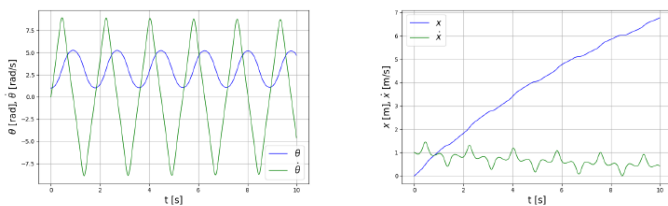

(a) Angular position and velocity   (b) Linear position and velocity

Figure 6. Inverted pendulum state variables

Figure 7. Virtual environment for inverted pendulum

5. Two Degree of Freedom Satellite Problem

The orbit of the satellite of two degrees of freedom with mass M is illustrated in Fig. 8, can be described by two non-linear and coupled equations obtained from the sum of the kinetic energy due to the tangential component vt = r(t)$\dot{\alpha}$(t) and radial component of velocity vr = $\dot{r}$(t). It is assumed that the satellite has traction in these directions, thus the magnitude of the forces are Ft and Fr respectively.

Although this problem is well known in the area of automatic control, it is not studied in the graduation in Bolivia, also, its importance lies in that it is a problem with multiple inputs and multiple outputs [Vargas and Irusta, 2014].

5.1 Mathematical model

The total kinetic energy of the satellite of Figure 8 is given by:

$$T = \frac{1}{2}M[v_t^2 + v_r^2] = \frac{1}{2}M[r^2(t)\dot{\alpha}^2(t) + \dot{r}^2(t)] \quad (16)$$

The equations (16) and (2) allow to obtain the mathematical model of this problem 4 [Wolovich, 1994].
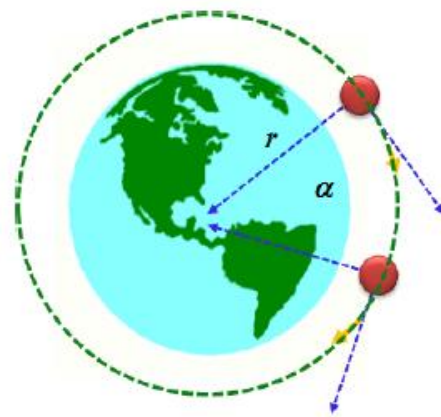

Figure 8. Two degree of freedom satellite

5.2 Computational solution

Figure 9(a) illustrates the results obtained for r(t) and $\dot{r}$(t). It is verified that the position r(t) moves away from its initial condition through a ramp, for this reason the derivative $\dot{r}$ tends to a constant.

55

Figure 9(b) illustrates the results obtained for α and α̇. It is verified that the angle α remains with constant behavior over time, for this reason the derivative α̇ tends to zero. In practice, it is expected that the satellite will describe a trajectory with constant angular velocity α̇ and with position r(t) also constant.

Similar to the inverted pendulum problem, Figure 10 illustrates the virtual environment for the satellite of two degrees of freedom.

(a) Distance and distance variation
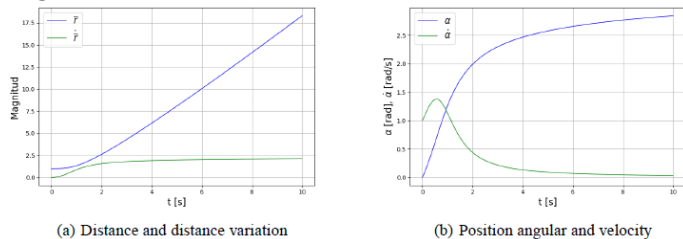
(b) Position angular and velocity

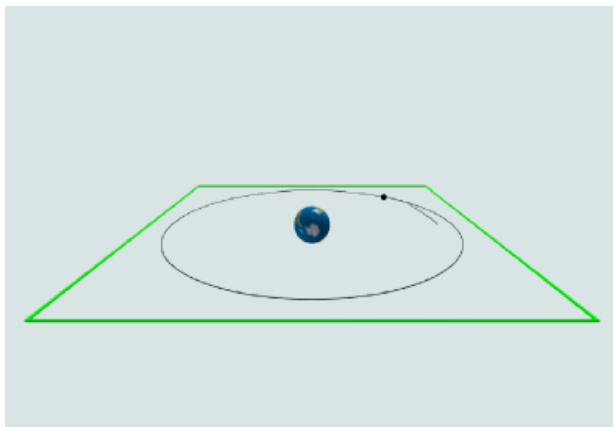Figure 9. Two Degree of Freedom Satellite in two dimensions

Figure 10. Virtual environment for two degree of freedom satellite

After the first simulations, some questions that arise are: why the inverted pendulum is not in the vertical?, why the satellite follows a spiral and not a circular path?. Obvious questions for those who are familiar with the control area, however, this does not happen with the students and professionals who participated in the dissemination activities.

6. The Control Problem

The analysis and synthesis are performed by applying theory of optimal control and the tech- nique of Linear Quadratic Regulator (LQR), method based on the linearization of dynamic sys- tems, represented in the time domain [Brian and Frank, 2003]. The employed control structure is presented in Fig. 11 [Ogata, 2011].
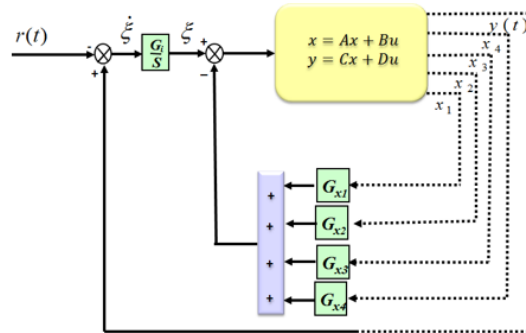

Figure 11. Generic structure control

6.1 Algorithm

The algorithm to find a control law u5 is given by:

$$u = -[G_{x1} \ G_{x2} \ G_{x3} \ G_{x4}]x + G_i\xi \quad (17)$$

$$\dot{\xi} = r - y \quad (18)$$

so that the feedback control law minimizes the performance index given by:

$$J = \int_0^\infty \{x^*Qx + u^*Ru\}dt \quad (19)$$

subject to the following state equation:

$$\dot{x} = Ax + Bu$$

$$y = Cx \quad (20)$$

The gain matrix G = [Gx1 Gx2 Gx3 Gx4 Gi] is obtained from the solution of Riccati's equation.

$$A_{aum}^T P + P A_{aum} - P B_{aum} R^{-1} B_{aum}^T P + Q = 0 \quad (21)$$

where Aaum and Baum are the augmented matrices. The optimal gain G can be obtained quickly using the Python control toolbox (it works very well on the Linux platform). The routine used to calculate the gains is presented in Figure 12.

6.2 Inverted pendulum control

The controlled variable is the linear position of the car. The desired reference for both variables is zero. Initial condition of the integrator is [10o 0 0.1 0]T . The gain vector G obtained is:

$$\begin{bmatrix} 70.1790 & 15.7800 & 6.3411 & 7.8731 & 2.2361 \end{bmatrix} \quad (22)$$

Figure 13(a) illustrates the results obtained for the angular position of the inverted pendulum, it is observed that after the transitory regime the angular position reaches the reference cor- responding to the zero value. In Fig. 13(b) the result for the linear velocity of the cart x is illustrated, just as the angle after the transient regime reaches the permanent regime value (The B.1 presents the differential equations including the control variable for this case.).

56

```
from control import *
from slycot import *
from control.matlab import *
import numpy as np

#Dates
Jm = 1.2; Bm = 0.1; J1 = 0.3; B1 = 0.05; m = 1; Kr = 0.5; 1 = 0.5; g = 9.8;

A = [[0 , 1],[-(g*1*m)/(J1+Jm*Kr**2),
              -(B1+Bm*Kr**2)/(J1+Jm*Kr**2)]];
B = [[0] ,[1/(Jm+(J1/Kr**2))]];
C = [[1 , 0],[0 ,1]]
D = [[0],[0]]

Aaum = [[A],[[0],[0]],[B],[1]]
Baum = [[B],[0]]

Q = [[20, 0, 0],[0, 20, 0],[0, 0, 20]]
R = 0.01
G, S, E = lqr(Aaum,Baum, Q, R)
```

Figure 12. Gains calculation Python code

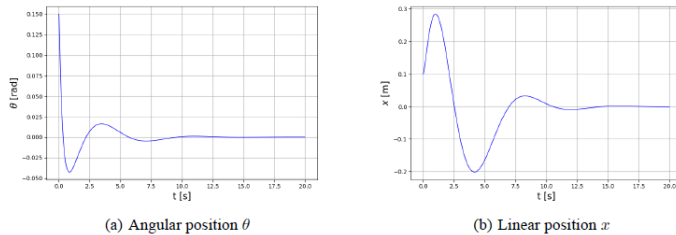(a) Angular position θ (b) Linear position x

Figure 13. Closed loop results inverted pendulum

6.3 Two degree of freedom satellite control

The controlled variables correspond to the position in relation to the center of the earth and the angular position of the satellite. The desired references are the unit for |r| and a ramp 0.5t for α. Initials conditions of the integrator are [0.8 0 0 1]T . The gain matrix G (r, r˙, α, α˙) obtained is:

$$\begin{bmatrix} 15.0466 & 8.3576 & -3.3905 & 0.0601 & -6.0658 & 1.7907 \\ 3.5564 & 0.0601 & 11.4848 & 7.9502 & -1.7907 & -6.0658 \end{bmatrix} \quad (23)$$

Figure 14(a) shown the result of radial component r(t), where is verified that after of transient, the reference is reached.
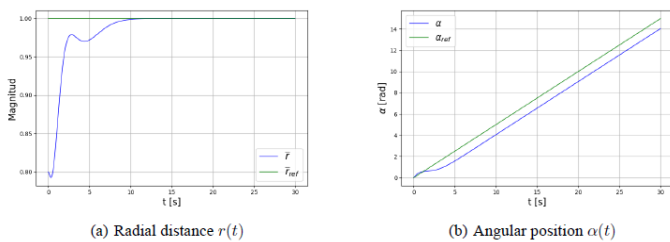

(a) Radial distance $r(t)$ (b) Angular position $\alpha(t)$

Figure 14. Closed loop results satellite TDOF

Fig. 14(b) illustrates the result obtained for the angular position α(t), which follows the specified ramp with an offset, offset is characteristic of LQR control for ramp reference. To solve this problem it is necessary to study other control strategy [Wolovich, 1994].

Once the control projects were developed, the students verified in three dimensions that there was no more instability in both problems.

7. Conclusions

One of the main benefits of the presented strategy is that the Python/Vpython software is free; additionally, it should be emphasized that in the medium and long term the strategy will help in the training and motivation of the new

Bolivian scientists, including improvements of the strategy discussed here.

The strategy clearly shows the complete cycle that begins with the deduction of models until the visualization of results in two and three dimensions. Attendees evidenced and felt how the knowledge acquired in calculus, algebra, numerical methods, automatic control and others are associated with practical problems.

To the participating universities, the extensive use of the strategy has been proposed, in addition to the change of their teaching plans in math sciences and engineering. As a result some of them, although very slowly, are making important changes. Also, it should be mentioned that classical problems such as the inverted pendulum and the two degree of freedom satellite were used to transmit the strategy, the same happened to expose the control problem.

In addition to these aspects, the questionnaire presented in Table 1 related with the method- ology practiced in the different universities was done.

Table 1. Type of methodology practiced

Methodology	< 25%	< 50%	< 75%	< 100%
Non applied method	√		√	
Applied method				

The result was that until 75% still administer subjects such as calculus, algebra and others disciplines isolated from practical applications, this aspect allowing the implementation of proposal strategy.

Finally, during the dissemination of the strategy, two books were published, one in Bolivia and another in Brazil [Vargas and Irusta, 2014], [Vargas and Paglione, 2015].

References

[Brian and Frank, 2003] Brian, S. and Frank, L. (2003). Aircraft Control and Simulation. John Wiley Inc. [Duke et al., 1988] Duke, E. L., Antoniewicz, R. F., and Krambeer, K. D. (1988). Derivation and Definition of a Linear Aircraft Model. NASA Reference Publication 1207, first edition.

[Gangsaas et al., 2008] Gangsaas, D., Hodgkinson, J., and Harden, C. (2008). Multidisciplinary control law design and flight test demonstration on a business jet. pages 1–25. AIAA Guidance, Navigation and Control Conference and Exhibit.

[Gilat, 2006] Gilat, A. (2006). Matlab com Aplicações em Engenharia. Bookman.

[Hetland, 2008] Hetland, M. L. (2008). Beginning Python: From Novice to Professional, Second Edition. Springer-Verlag New York.

[Khalil, 2002] Khalil, H. K. (2002). Nonlinear Systems. Prentice Hall Inc.

57

[Kupferman, 2008] Kupferman, R. (2008). Lecture Notes in Classical Mechanics. The Hebrew University.

[Kusalaas, 2013] Kusalaas, J. (2013). Numerical Methods in Engineering with Python 3. Cambridge University Press.

[Lopez and Sanches, 2011] Lopez, J. C. C. and Sanches, L. A. J. (2011). Introducción al Mathematica 7. Uni- versidad Politecnica de Valencia, first edition.

[Ogata, 2011] Ogata, K. (2011). Engenharia de Controle Moderno. Pearson Education.

[Vargas, 2011] Vargas, F. J. T. (2011). Aeronáutica y robótica industrial. https://lpz.ucb.edu.bo/ Forms/Noticias/NoticiasUCB.aspx?NSNoticia=74441.

[Vargas, 2013] Vargas, F. J. T. (2013). Experto cochabambino en robótica dictará charla sobre drones. http:

//www.telecombol.com/2013/11/experto-cochabambino-en-robotica.html.

[Vargas, 2014] Vargas, F. J. T. (2014). Drones controlan fronteras y bolivia tiene limitaciones. https://www. opinion.com.bo/opinion/articulos/2014/0504/noticias.php?id=1 27049.

[Vargas, 2016] Vargas, F. J. T. (2016). Un equipo de bolivianos desarrolla algorit- mos para controlar drones. https://www.paginasiete.bo/gente/2016/10/27/ equipo-bolivianos-desarrolla-algoritmos-para-controlar-drones.

[Vargas, 2017] Vargas, F. J. T. (2017). Técnicas lineales y no lineales aplicadas al control de aeronaves. https://www.upb.edu/es/contenido/conferencia-%C2%A8t%C3% A9cnicas-lineales-y-no-lineales-aplicadas-al-control-de-aeronaves%C2%A8. [Vargas, 2018] Vargas, F. J. T. (2018). Universidad: No quieren dar un paso adelante. https://www.lostiempos.com/oh/actualidad/20180312/ universidad-no-quieren-dar-paso-adelante.

[Vargas and Irusta, 2014] Vargas, F. J. T. and Irusta, E. O. P. (2014). Modelado, Simulación y Control de Sistemas Mecatronicos y Aeroespaciales con Matlab®y MathematicaT M . Editorial Serrano.

[Vargas et al., 2016] Vargas, F. J. T., Oliveira Moreira, F. J., and Paglione, P. (2016). Longitudinal stability and control augmentation with robustness and handling qualities requirements using the two degree of freedom controller. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 38(7):1843–1853.

[Vargas and Paglione, 2015] Vargas, F. J. T. and Paglione, P. (2015). Aplicações de Modelagem, Simulação e Controle Com Ferramentas de Algebra Computacional. LTC.

[Villate, 2018] Villate, J. E. (2018). Simulações com vpython. https://def.fe.up.pt/python/ vpython.html.

[Wolovich, 1994] Wolovich, W. A. (1994). Automatic Control Systems: Basic Analysis and Design. Oxford University Press, second edition.

[Yue et al., 2018] Yue, M., An, C., and Sun, J.-Z. (2018). An efficient model predictive control for trajectory tracking of wheeled inverted pendulum vehicles with various physical constraints. International Journal of Control, Automation and Systems, 16(1):265–274.

58

APPENDIX A. MATHEMATICAL REPRESENTATION STAGES

A.1. **Nonlinear model of inverted pendulum**. The Fig. 15 shows the computational solution to obtain the nonlinear model of inverted pendulum [Vargas and Paglione, 2015].

A.2. **Operation points of inverted pendulum**. The computational solution to obtain the operation points of inverted pendulum is illustrated in the Fig. 16.

$$Ph[t] = x[t] + L\text{Sin}[\theta[t]];$$
$$Pv[t] = L\text{Cos}[\theta[t]];$$
$$vx[t] = \partial_t Ph[t];$$
$$vy[t] = \partial_t Pv[t];$$
$$Tm[t] = \tfrac{1}{2}m\text{Simplify}\left[(vx[t]^2 + vy[t]^2)\right] + \tfrac{1}{2}J\theta'[t]^2;$$
$$TM[t] = \tfrac{1}{2}Mx'[t]^2;$$
$$Lg[t] = \text{Simplify}[Tm[t] + TM[t]];$$
$$dL_{x'} = \text{Simplify}\left[\partial_{x'[t]}Lg[t]\right];$$
$$dL_{x} = \text{Simplify}\left[\partial_{x[t]}Lg[t]\right];$$
$$dLx'_t = \text{Simplify}\left[\partial_t dL_{x'}\right];$$
$$Res1 = dLx'_t - dL_x - u + bx'[t];$$
$$dL_{\theta'} = \text{Simplify}\left[\partial_{\theta'[t]}Lg[t]\right];$$
$$dL_{\theta} = \text{Simplify}\left[\partial_{\theta[t]}Lg[t]\right];$$
$$dL\theta'_t = \text{Simplify}\left[\partial_t dL_{\theta'}\right];$$
$$Res2 = dL\theta'_t - dL_\theta - mgL\text{Sin}[\theta[t]];$$
$$\text{Simplify}\left[\text{Solve}\left[\{Res1 == 0, Res2 == 0\}, \{\theta''[t], x''[t]\}\right]\right];$$
$$x''[t] \rightarrow$$
$$\left(Ju + L^2mu - gL^2m^2\text{Cos}[\theta[t]]\text{Sin}[\theta[t]] - b\left(J + L^2m\right)x'[t] + Lm\left(J + L^2m\right)\text{Sin}[\theta[t]]\theta'[t]^2\right) / \left((J + L^2m)(m + M) - L^2m^2\text{Cos}[\theta[t]]^2\right);$$
$$\theta''[t] \rightarrow$$
$$-\frac{Lm\left(g(m + M)\text{Sin}[\theta[t]] - \text{Cos}[\theta[t]]\left(u - bx'[t] + Lm\text{Sin}[\theta[t]]\theta'[t]^2\right)\right)}{-(J + L^2m)(m + M) + L^2m^2\text{Cos}[\theta[t]]^2};$$

Figure 15. Nonlinear model of inverted pendulum

$$\theta[t] = x1[t]; \theta'[t] = x2[t]; x[t] = x3[t]; x'[t] = x4[t];$$
$$x1p[t] = x2[t];$$
$$x2p[t] = -\frac{Lm\left(g(m + M)\text{Sin}[\theta[t]] - \text{Cos}[\theta[t]]\left(u - bx'[t] + Lm\text{Sin}[\theta[t]]\theta'[t]^2\right)\right)}{(-J - L^2m)(m + M) + L^2m^2\text{Cos}[\theta[t]]^2};$$
$$x3p[t] = x4[t];$$
$$x4p[t] =$$
$$\left(Ju + L^2mu - gL^2m^2\text{Cos}[\theta[t]]\text{Sin}[\theta[t]] - b\left(J + L^2m\right)x'[t] + Lm\left(J + L^2m\right)\text{Sin}[\theta[t]]\theta'[t]^2\right) / \left((J + L^2m)(m + M) - L^2m^2\text{Cos}[\theta[t]]^2\right);$$
$$eqs = \{x1p[t], x2p[t], x3p[t], x4p[t]\}/.$$
$$\{M \rightarrow 1, m \rightarrow 1, u \rightarrow 0, L \rightarrow 1, g \rightarrow 9.8, b \rightarrow 0.1, J \rightarrow 0.1\}\ eqs = eqs/.\{x2[t] \rightarrow 0, x4[t] \rightarrow 0\}$$
$$\text{Solve}[eqs[[2]] == 0, x1[t]]$$
$$\text{Solve}[eqs[[4]] == 0, x1[t]]$$

Figure 16. Operation points inverted pendulum

In this figure it is verified that the operation points of the variables x2 and x4 are equal to zero. This does not happen with x3 variable that is not in evidence, for this reason is assumed that the origin is the initial position of the car (x3 = 0). Finally, the operation points for x1 are calculating obtaining [0 0 0 0]T .

A.3. Linear model of inverted pendulum. The matrices Ae and Be are presented in the Figure 17.

```
x1p[t] = x2[t];
x2p[t] =
```
$$-\frac{Lm\left(g(m+M)\mathrm{Sin}[\theta[t]]-\mathrm{Cos}[\theta[t]]\left(u-bx'[t]+Lm\mathrm{Sin}[\theta[t]]\theta'[t]^2\right)\right)}{(-J-L^2m)(m+M)+L^2m^2\mathrm{Cos}[\theta[t]]^2};$$
```
x3p[t] = x4[t];
x4p[t] =
```
$$(Ju+L^2mu-gL^2m^2\mathrm{Cos}[\theta[t]]\mathrm{Sin}[\theta[t]]-b\left(J+L^2m\right)x'[t]+$$
$$Lm\left(J+L^2m\right)\mathrm{Sin}[\theta[t]]\theta'[t]^2\right)/\left((J+L^2m)(m+M)-L^2m^2\mathrm{Cos}[\theta[t]]^2\right);$$
```
equi = {x1p[t], x2p[t], x3p[t], x4p[t]};
x = {x1[t], x2[t], x3[t], x4[t]};
A = 0IdentityMatrix[4];
```
$$\mathrm{Do}\left[A[[i,j]]=\mathrm{FullSimplify}\left[\partial_{x[[j]]}\mathrm{equi}[[i]]\right],\{j,1,4\},\{i,1,4\}\right];$$
```
MatrixForm[A];
```
$$A_e = \mathrm{Simplify}[A/.\{x1[t]\to 0, x2[t]\to 0, x3[t]\to 0, x4[t]\to 0\}]//$$
```
MatrixForm
```
$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{gLm(m+M)}{L^2mM+J(m+M)} & 0 & 0 & \frac{bLm}{L^2mM+J(m+M)} \\ 0 & 0 & 0 & 1 \\ -\frac{gL^2m^2}{L^2mM+J(m+M)} & 0 & 0 & -\frac{b(J+L^2m)}{L^2mM+J(m+M)} \end{pmatrix}$$
```
B = {0, 0, 0, 0};
```
$$\mathrm{Do}\left[B[[i]]=\mathrm{FullSimplify}\left[\partial_u\mathrm{equi}[[i]]\right],\{i,1,4\}\right];$$
```
MatrixForm[B];
```
$$B_e = \mathrm{Simplify}[B/.\{x1[t]\to 0, x2[t]\to 0, x3[t]\to 0, x4[t]\to 0\}]//$$
```
MatrixForm
```
$$\begin{pmatrix} 0 \\ -\frac{Lm}{L^2mM+J(m+M)} \\ 0 \\ \frac{J+L^2m}{L^2mM+J(m+M)} \end{pmatrix}$$

Figure 17. Linear model inverted pendulum

APPENDIX B. CONTROL PROBLEM STAGE

B.1. Closed loop simulation inverted pendulum. Figure 18 contains the nonlinear model of inverted pendulum that includes the control variable u = cktr.

```python
from numpy import *
from scipy.integrate import odeint
from matplotlib.font_manager import FontProperties
import matplotlib.pyplot as plt

M = 1; m = .159; l = 0.4; g = 9.81; J = m*l**2/12; b = 0.1;

Kint = -1.73      # Integral gain
KI = -1*Kint

def derivs(state, t, p):#dy/dt = f(x1,y1,x2,y2)
        x1, y1, x2, y2 = state
        dydx = np.zeros_like(state)

        aux3 = KI*x2
        aux4 = x1*44.5351+y1*9.1050+x2*7.3385+y2*6.9858
        cktr = aux3+aux4

        a1 = l*m*(1/np.cos(x1))*(cktr-g*m*np.tan(x1)
        a2 = -g*M*np.tan(x1)+l*m*np.sin(x1)*y1**2-b*y2)
        a3 = -l**2*m**2+(J+l**2*m)*(m+M)*(1/np.cos(x1)**2)
        a4 = -g*l**2*m**2*np.cos(x1)*np.sin(x1)+(J+l**2*m)
        a5 = (cktr+l*m*np.sin(x1)*y1**2-b*y2)
        a6 = l**2*m**2*np.cos(x1)**2+(m+M)*(-J-l**2*m)
        aux1 = -((a1-a2)/a3)
        aux2 = -(a4*a5/a6)

        dydx = [y1,
        aux1,
        y2,
        aux2]
        return dydx
# initial conditions
dt = 0.02
t = np.arange(0.0, 10, dt)
p = [M, m, l, g, J, b]
state = [1, 0, 0, 1]

# Call ODE module
r = odeint(derivs, state, t, args=(p,))
```

Figure 18. Python code closed loop inverted pendulum

B.2. Closed loop simulation two degree of freedom satellite. Fig. 19 contains the nonlinear model of two degree of freedom satellite code that includes the control variables u = [0] and u[1].

```
import numpy as np
from scipy.integrate import odeint
from pylab import figure, plot, xlabel, grid, legend, title, ylabel
import matplotlib.pyplot as plt
from numpy import *

Fn = 0; Ft = 1; m = 1; k = 1

def derivs(state, t, p):#f = (x1',y1',x2',y2')
        x1, y1, x2, y2, x3, y3 = state
        dydx = np.zeros_like(state)
        ref = 0.5*t

        # Control I
        Kint = matrix([[-6.0658,1.7907],[-1.7907,-6.0658]])
        KI = -1*Kint
        aux1 = KI*matrix([[x3],[y3]])
        Kest = matrix([[15.0466,8.3576,-3.3905,0.0601],
        [3.5564,0.0601,11.4848,7.9502]])
        eE = matrix([[x1-.8],[y1],[x2],[y2-1]])
        aux2 = Kest*eE
        u = aux1-aux2
        print(u)

        dydx = [y1,
        x1*y2**2-k/x1**2 + u[0],
        y2,
        -2*y1*y2/x1 + u[1]/(m*x1**2),
        (1-x1),
        (ref-x2)]
        return dydx

# ODE parameters
abserr = 1.0e-6; relerr = 1.0e-6
dt = 0.01
t = np.arange(0.0, 30, dt) # Time vecto
p = [k, m, Fn, Ft] # Parameters and initial conditions:
state = [.8, 0, 0, 1, 0, 0]

r = odeint(derivs, state, t, args=(p,), atol=abserr, rtol=relerr) # Call ODE
ref = 0.5*t # Angular position reference
```

Figure 19. Python code closed loop two degree of freedom