State Machine Controller for Control of the Nonlinear Pendubot

Hamid Keshmiri Neghab
Faculty of Electrical Engineering, University of West Bohemia, Pilsen, Czech Republic
Email: neghab@fel.zcu.cz

Hamed Keshmiri Neghab
Faculty of control engineering Ferdowsi University of Mashhad, Mashhad, Iran
Email: keshmirineghab.hamed@mail.um.ac.ir

Abstract

Pendulum Robot (Pendubot) is made up of a DC motor that is linked to a physical rigid link (elbow) that is connected to a free spinning link (arm) that does not have any. The control goal is to keep the free link vertical while balancing the controller link (elbow) in either the extended position (both links are up) or the extended position (both links are down). A State Feedback Control is utilized at certain operational points to maintain the Up-Up or Down-Up posture. The controller can also use a state machine to transition between Down-Down positions and apply additional control rules to arrive at the desired place. In this paper, partial feedback linearization and PD controllers are the techniques utilized for this process using LabVIEW control design and simulation module, were used to control the state of a pendulum robot. This paper also uses a state-machine to switch between states. Moreover, another controller is used to push the elbow to the stationary position to avoid a disorderly fall of the pendulum if it cannot reach the position (elbow and a free link pointing down). This paper shows how to utilize Linearization, how to construct LQR and pole placement controllers, and how to apply it to a switching controller.

Keywords: Pendubot, Elbow, State Feedback, Feedback Linearization, State Machine.

Introduction

A Pendulum Robot in the field of control represents a standard problem. It is used to show various concepts in linear and nonlinear control such as stabilization or swinging up of unstable systems [1]. This framework has a place in the course of non-minimum stage frameworks, since of its inner flow. The Pendubot is a curious case of the modified pendulum. It is an under-actuated framework since the number of its degree of opportunity is greater than the number of its control inputs, which makes it troublesome to control. Moreover, controlling such frameworks is challenging due to nonlinear flow, non-holonomic conduct, the need for linearizability displayed by these frameworks [2]. Some nonlinear optimization techniques are discussed in [3] to calibrate a nonlinear DC motor under uncertainty. Cruz and et al. present the plan of a direct state-input controller for the stabilization of two rearranged pendulums, specifically, Furuta pendulum and pendubot. Such a controller plan permits disposing of the constrain cycle that shows up within the frameworks due to the effect of nonlinearity, that's, dead-zone, which is initiated by inactive contact at the engine shaft [4]. The novel control technique is proposed based on Fourier change and clever optimization for a planar Pendubot with an inactive moment interface, which can be treated as a second-order nonholonomic framework whose control has been an open and challenging issue [5].

Chen and et al. amplified a few of the controllability that comes about for a two-link or three-link even pendubot to a more common case, by considering a reasonable N-link flat pendubot show and abusing Lie brackets to think about openness and STLC for the demonstrate [6]. A fuzzy-sliding control for a Pendubot framework is proposed. The controller is planned from a modern thought of application of a fuzzy calculation for optioning control parameters. The reaction of the framework on the Beat position beneath the fuzzy sliding control calculation is demonstrated [7]. A modified differential evolution (MDE) calculation is proposed in [8], to distinguish the parametric energetic show of a Pendubot framework with grinding. Within the MDE calculation, the change is to center on the change stage with a modern transformation plot in which multi-mutation administrators are utilized. Also, a plan of a novel block-backstepping-based nonlinear stabilizing control law of a Pendubot is displayed in [9].
In [10], proposed a modified quantum-behaved particle swarm optimization (MQPSO) approach to planning an ideal fluffy PID controller for asymptotical stabilization of a Pendubot framework. Within the fluffy PID controller, parameters are decided by utilizing MQPSO calculation. The MQPSO strategy and other PSO strategies are at that point connected to plan an ideal fluffy PID controller in a Pendubot framework. Cisneros and et al. Also, the MPC can be investigated with fuzzy logic to make a reliable controller like the subject purposed in [11]. In [12], represent a test approval of a nonlinear discerning management law supported quasi-Linear Parameter variable (q-LPV) demonstrating. q-LPV displaying is utilized to set up the plant demonstrates, later turning the nonlinear optimization direct however time-varying, by forcing a coming up with grouping that is found iteratively. The Pendubot could be a curious benchmark issue because the loci of equilibria lie on a nonlinear complex; furthermore, the plant gets to be additional, tough to manage the more the operating condition approaches one among the 2 wild equilibria. A strategy for the computation of a stabilizing compensator for a twofold altered pendulum known as the Pendubot is proposed in [13]. The method depends on a computational calculation based on different comes about of “the polynomial network approach” and in specific comes about for the arrangement of polynomial framework Diophantine conditions required for the computation and parametrization of legitimate “denominator assigning” and inside stabilizing compensators for straight time-invariant multivariable (LTI) frameworks. In this paper, partial feedback linearization and PD controllers are the techniques utilized for this process using LabVIEW control design and simulation module, were used to control the state of a pendulum robot. This paper also uses a state-machine to switch between states. Moreover, another controller is used to push the elbow to the stationary position to avoid a disorderly fall of the pendulum if it cannot reach the position (elbow and a free link pointing down). This paper shows how to utilize Linearization, how to construct LQR and pole placement controllers, and how to apply it to a switching controller**.** In the following, design of a controller for a controlling the states of a Pendubot, is investigated.

Description

The Pendubot is a two-link robot as shown in Fig1. It consists of a DC motor, an elbow, and an arm. Here elbow is a physically rigid link that is connected to the DC motor, and the arm is referred to as a second rigid link which has no actuation. The goal is to control the free-spinning link in the vertical position whenever we can balance the elbow (controller link) in the extended position or the counteracted position. Extended position means both links are up, and opposite position means elbow is down and arm is up, or elbow is up and arm is down.

The system which aimed to control in this paper is the Nonlinear Pendubot. The control system should have the following features: the outer link of the Pendubot will swing up to the upright position, When the inner link of Pendubot will going downwards, as shown in Fig1 (b), this is the Down-Up control.
Up-Up control is when the inner link of the Pendubot is pointing upwards, as shown in Fig1 (c). While the inner link is moving slowly from one peripheral position to the other, the links in these configurations

will be stabilized. The system should be controlled to their hanging down position when it faces an unsafe state or both links get disturbed.

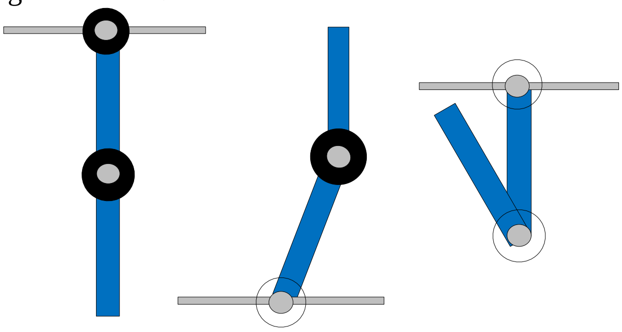


Fig1: Pendubot system from left to right, (a) Down-Down Position, (b) Up-Up Position, (c) Down -Up Position.

The elbow and free link are pointing down in Down-Down Position, as previously stated. This is a stable position as well, and simulation begins here. In the Down-Up Position, the elbow is pointing down and the free link is controlled in the up position. In the Up-Up Position, the elbow and free link are controlled in the up position, similar to an inverted pendulum. And finally, designing the controller parameter to bring the position back to Down-Down.

System Model

At the begining, all variables from the nonlinear subsystem were gathered and the variable's initial values for linearization were determined. The nonlinear Pendubot model was then linearized, and states were reorganized.

$$\text{States: } \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \text{angle controlled link} \\ \text{angle free link} \\ \text{angular velocity controlled link} \\ \text{angular velocity free link} \end{bmatrix}$$

The two-link underactuated planar robot was considered. The general notations and parameters of the Pendubot are described in Fig2.

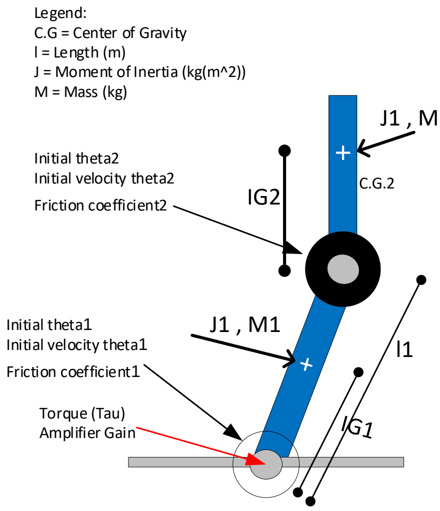


Fig.2: Nonlinear Pendubot Model

Fig 2 shows the variables that are utilized for the controller and regulating the elbow and arm for spinning. The torque (Tau) is the voltage applied to the controller. Here Theta1 is an encoder position of the controlled link (initial Theta1 (elbow)), and Theta2 is an encoder position of the free link (initial Theta2 (arm)). Encoders that have zero reference point down must be specified.

In Fig2, C.G is the Centre of gravity, IG1 and IG2 is the length of the arm to the centre of gravity, L is Length (m), J is referred to as Moment of Inertia(kg(m^2) and M is Mass(kg). The following equations are used for this case:

$$((L1)^{٢} * M2) + J1 = C1 \tag{1}$$

$$((IG2)^{٢} * M2) + J2 = C2 \tag{2}$$

$$L1 * M2 * IG2 = C3 \tag{3}$$

$$\left((L1 * M2) + (IG1 * M1)\right) * g = C4 \tag{4}$$

$$IG2 * M2 * g = C5 \tag{5}$$

$$\ddot{\theta}_{١} = \frac{١}{٢C_{١}}\left[\ddot{\theta}_{٢} C_{٣} cos(\theta_{١} - \theta_{٢}) - C_{٤} sin\theta_{١} - \dot{\theta}_{٢}^{٢} C_{٣} \sin(\theta_{١} - \theta_{٢})\right] \tag{6}$$

$$\ddot{\theta}_{٢} = \frac{١}{٢\square_{٢}}\left[\dot{\theta}_{١}^{٢} C_{٣} \sin(\theta_{١} - \theta_{٢}) - C_{٥} sin\theta_{٢} - \ddot{\theta}_{١} C_{٣} cos(\theta_{١} - \theta_{٢})\right] \tag{7}$$

The initial value for the parameters are as follow:

Table 1. Initial Values

g	**9.8 (m/s^2)**
$\boldsymbol{\theta_1}$	3.1416 (m)
$\boldsymbol{\dot{\theta}_1}$	0 (m/s)
Friction coefficient1	0.02
$\boldsymbol{\theta_2}$	0 (m)
$\boldsymbol{\dot{\theta}_2}$	0 (m/s)
Friction coefficient2	0.00109561
J1	0.00894463 (kg(m^2))
J2	0.00251451 (kg(m^2))
M1	0.463901 (kg)
M2	0.364069 (kg)
Torque (Tau)	0 (N·m)
Amplifier Gain	0.4
IG1	0.116 (m)
IG2	0.134 (m)
L1	0.21 (m)

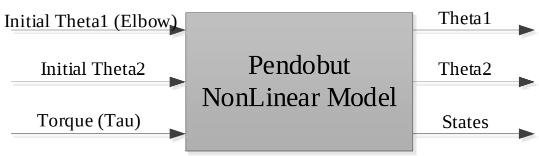


Fig.3: Pendubot Nonlinear Function

Fig 3, illustrates the Pendubot nonlinear function, which described by equations 1 to 7. The below parameters are derived with the Pendubot Nonlinear Function as follow:

Table 2. Derived Parameters

C1	**0.0177673**
C3	0.00621331
C4	1.00394
C2	0.00577668
C5	0.29025

$$\text{States:}\begin{bmatrix}\theta_{\text{١}}\\ \theta_{\text{٢}}\\ \dot{\theta}_{\text{١}}\\ \dot{\theta}_{\text{٢}}\end{bmatrix}=\begin{bmatrix}x_{\text{١}}\\ x_{\text{٢}}\\ x_{\text{٣}}\\ x_{\text{٤}}\end{bmatrix}=\begin{bmatrix}\text{٠.١١٦٧٦٩}\\ -\text{٠.٣٥٧٨٥٦}\\ -\text{٠.١٩١٧٥٨}\\ \text{٠.٣٩١٢٦٦}\end{bmatrix}$$

The nonlinear model can't be used directly. Because it is vital to regulate the Operating Point, it is necessary to linearize the equation. If $[\theta_{\text{١}}\ \theta_{\text{٢}}\ \dot{\theta}_{\text{١}}\dot{\theta}_{\text{٢}}\ ]^T$ denotes the values of the state variables in an equilibrium point, the linearization around this point will be derived by using Taylor's expansion.

Where A and B is the Jacobian matrix and defined as follow:

$$A=\begin{bmatrix}\text{٠} & \text{٠} & \text{١} & \text{٠}\\ \text{٠} & \text{٠} & \text{٠} & \text{١}\\ -\text{٥٤.٩٢٤٧} & \text{٤٠.٤٠٨٨} & -\text{١.٤٩٢٠١} & \text{٠}\\ \text{٤٥.٦٣٢} & \text{٥٨.١٩٨٦} & \text{١.٤٩٢٠١} & -\text{٠.٢٢٥٧٤}\end{bmatrix}$$

$$B=\begin{bmatrix}\text{٠}\\ \text{٠}\\ \text{٢٩.٨٤٠٢}\\ \text{٣.٩٣٣٤٤}\end{bmatrix}$$

For each study point, A and B must be calculated. Those points will be defined when choosing the controllers. The linearized equation for operating point. At last, the final equation that will be used in the study is:

$$\frac{dx}{dt}=\begin{bmatrix}\text{٠} & \text{٠} & \text{١} & \text{٠}\\ \text{٠} & \text{٠} & \text{٠} & \text{١}\\ -\text{٥٤.٩٢٤٧} & \text{٤٠.٤٠٨٨} & -\text{١.٤٩٢٠١} & \text{٠}\\ \text{٤٥.٦٣٢} & \text{٥٨.١٩٨٦} & \text{١.٤٩٢٠١} & -\text{٠.٢٢٥٧٤}\end{bmatrix}x(t)+\begin{bmatrix}\text{٠}\\ \text{٠}\\ \text{٢٩.٨٤٠٢}\\ \text{٣.٩٣٣٤٤}\end{bmatrix}u(t) \quad (8)$$

$$y(t)=\begin{bmatrix}\text{١} & \text{٠} & \text{٠} & \text{٠}\\ \text{٠} & \text{١} & \text{٠} & \text{٠}\end{bmatrix}x(t)+\begin{bmatrix}\text{٠}\\ \text{٠}\end{bmatrix}u(t) \quad (9)$$

Control Problem

Balancing and Swinging-Up are two major problems in the control of a Pendubot. These issues will be solved with various techniques.

A. **LINEAR QUADRATIC REGULATOR METHOD**

To stabilize the Pendubot in position, state-feedback with LQR control is utilized. It is used to drive all states to zero (home position). For swinging Up-Up and swinging Down-Up, PD Controller used. The PD-controller, which uses linearized internal states in the state-feedback control rule, is the simplest form of controller in the outer loop. The equation below, better describes the PD controller:

$$K_p*\left(\dot{\theta}_{ref}-\dot{\theta}_{\text{١}}\right)+K_d*\left(\dot{\theta}_{ref}-\dot{\theta}_{\text{٢}}\right) \quad (10)$$

In our case, for Up-Up Swing K_p up=14.2, and K_d up=1.1, and for swinging Down-Up K_p down=15, K_d down= 0.9 can be initialized. The controller switched to Balance if the bottom link is near zero and the top link is near π. Up-Up movement is the best to execute since an inactive reference flag is sufficient to swing the framework from its Down-Down position to its Up-Up position. With the partial feedback linearization, it is possible to set the precise position of $\theta_{\text{١}}$. Thus, it is not difficult to achieve the up-position for the inner arm. This has been accomplished by setting the reference signal to π/2. However, by switching to the balancing controller, it needed to be able to reach the Pendubot in its Up-Up position. Subsequently, the PD controller (K_p, K_d) has been tuned due to achieve the Up-Up position with an angular speed of the outer arm as small as possible.

LQR Calculates the optimum steady-state feedback gain matrix **K** that reduces a linear quadratic cost function which is specified. This function can be used to calculate **K** for a continuous or discrete state-space model. **Q** specifies an asymmetric, positive semi-definite matrix that penalizes the state vector **x** in the cost function. **R** specifies a symmetric, positive definite matrix that assesses the input vector **u** in the cost function. **N** specifies a matrix of appropriate dimensions that penalizes the cross-product between the input and state vectors. Fig *4, illustrates the block diagram of Linear Quadratic Regulator.*

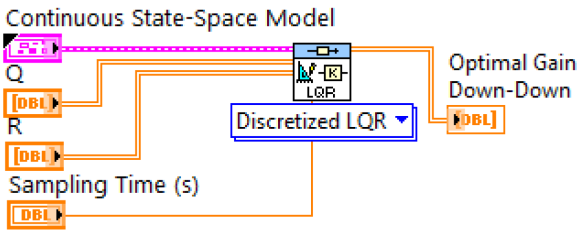


Fig.4: Block diagram LQR Control Technique

As for the Q and R matrices, the following values have been chosen after several experiments. For Swinging Up-Up $Q = diag[١٠,٦٥,١,١]$, R=40 and for swinging Down-Up $Q = diag[١٠,١٠,١٠٠,١]$, R=50 are assumed.

B. STATE MACHINE CONTROLLER

To control the system of such a complex machine, it needs to develop control techniques based on different states. Here, there are 7 different controllers that are controlled by a state-machine that switches whenever the condition is stabilized, Fig 5.

A state machine to calculate and control the program is used. The Down-Down position is considered to be the initial state, and whenever the system doesn't meet specific command, the state machine will bring the system to the down-down position.

To transition between the Down-Down position to Down-Up and Up-Up position, it is used a nonlinear control technique called Feedback Linearization. The implementation of the controller is using a Proportional-Derivative (PD) controller, which is tuned by experimentation in this case. The PD controller describe in previous section. The same LQR technique is applied to obtain the controller around the Up-Up and the Down-Up position. The gain that makes the controller stable and resistant to disturbance, chosen experimentally.

State machine controller calculated optimal gain for swing up (optimal gain Up-Up) or swing down (optimal gain Down-Up). It is shown in Fig 6.

The whole system of the state feedback controller is shown in Fig 7. In summary, there is a DC motor with the state-space model equations. It will be linearized as shown and the system will acquire the encoder measurements. By using control techniques, here: LQR and Pole Placement, the Pendubot will be controlled using a state feedback controller with a state machine.

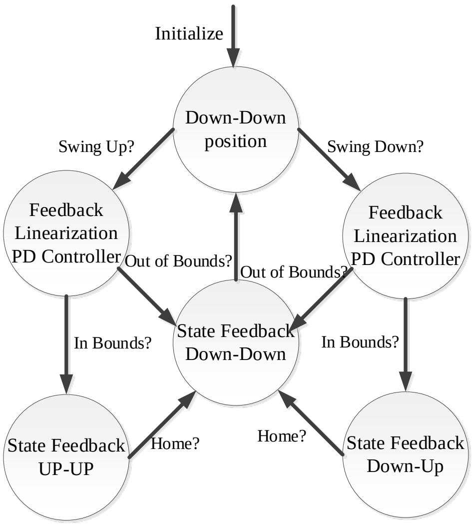


Fig.5: state feedback controller with a state machine

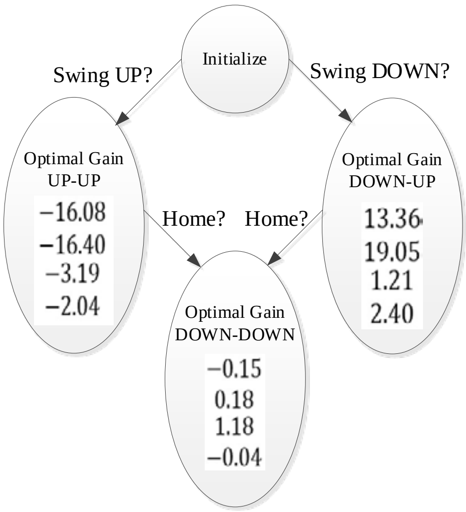


Fig.6: Optimal Gain for Swing Up, Swing Down

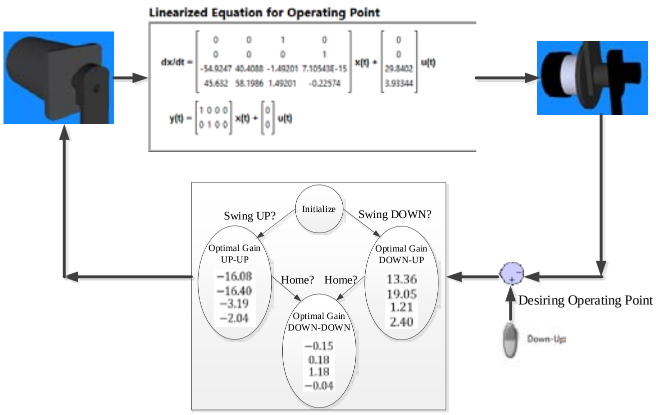


Fig.7: Block diagram of State Feedback Controller with State Machine

C. RESULTS ANALYSIS

Optimal Gain for Down-Down= $diag[١٠٠,١,١٠٠,١]$, $R = ٤٠$, and Sampling Time: 0.01. It can be seen in Fig 8, the block diagram for Down-Down (Home) swinging.

Parameters for the controller to control position to UP-UP, designed as follow. Fig 9, indicates the block diagram of the PD controller for UP Swing.

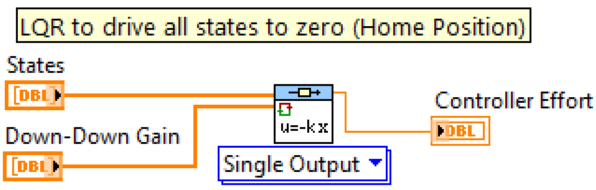


Fig.8: Bring Home all states

Optimal Gain Up-Up= $diag[١٠,٦٥,١,١]$,$R = ٤٠$, Sampling Time: 0.01.

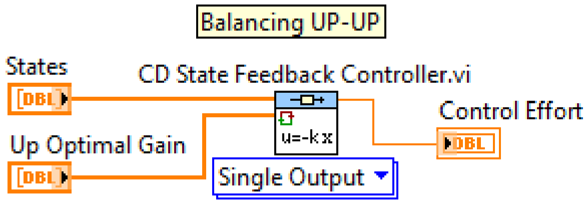


Fig.9: Balancing UP-UP

The optimal gain for controlling the position to DOWN-UP and the block diagram, Fig 10, of the controller for balancing the DOWN-UP position is as follows.

Optimal Gain Down-Up= $diag[١٠,١٠,١٠٠,١], R = ٥٠$, and Sampling Time: 0.01.

Swinging Down-UP: PD Controller for Down Swing.

PD Gains: (K_p Up = 16.55, K_p Down = 10.2), (K_d Up = 5.1567, K_d Down = 0.974).

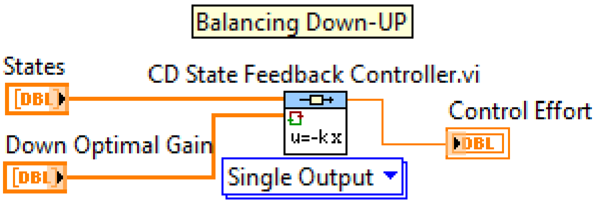
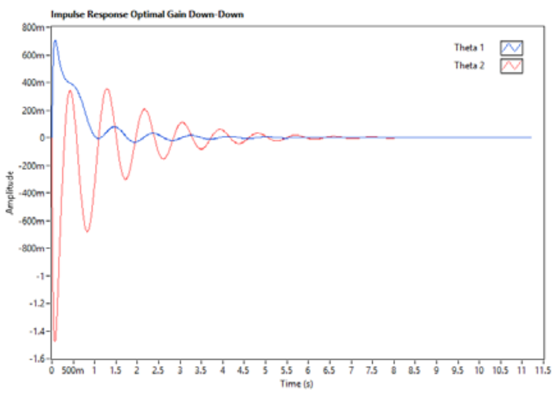


Fig.10: Balancing DOWN-UP

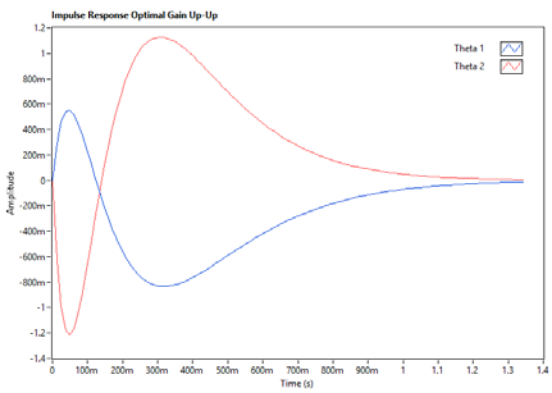
After initialization, by running the simulation, the results are as follow. Fig11 shows the impulse responses of optimal gains for Down-Up, Up-Up, and Down-Down. It can be seen that by running the simulation and applying the swing-up and swing-down, how both theta1 and theta2 changed.

Closed-loop poles are the s-plane locations of a closed-loop transfer function's poles (or eigenvalues). In Fig 12, it can be seen that the position of poles in different states, Down-Down, Up-Up, and Down-up.

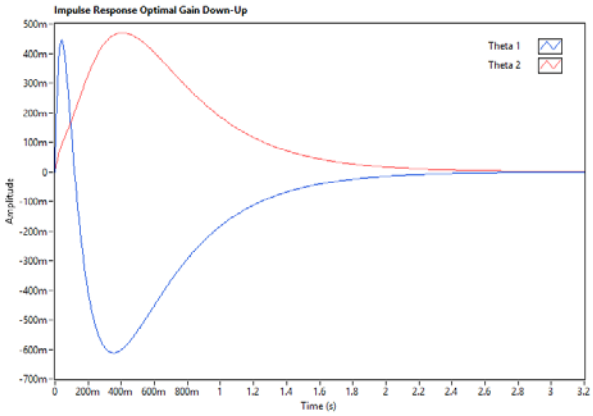
Frequency response is a quantitative measure of a system's or device's output spectrum in response to a stimulus that is used to define the system's dynamics. In contrast to the input, it is a measure of the output's amplitude and phase as a function of frequency. Fig 13, and Fig 14 illustrates the bode magnitude and phase response in different states.



a) Impulse Response Down-Down



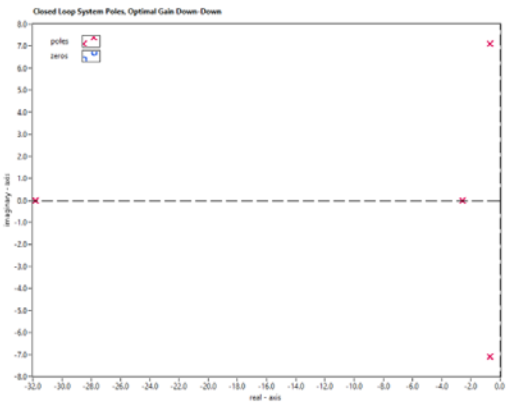
b) Impulse Response Up-Up



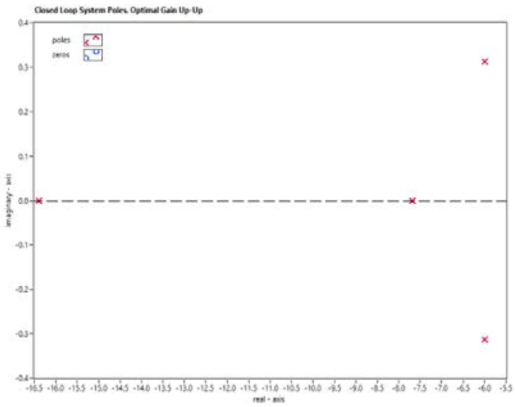
c) Impulse Response Down-Up

Fig.11: Impulse Response

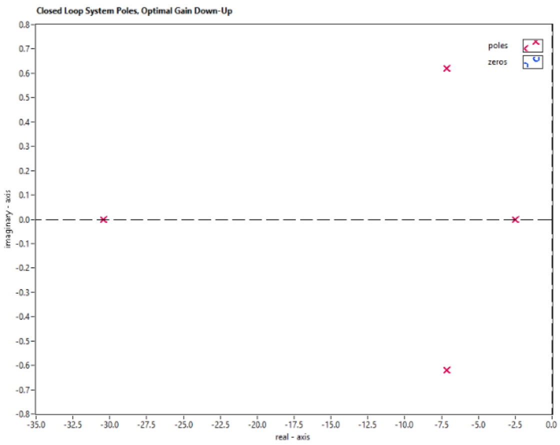
When the simulation started to run, for swinging Up-Up the parameters are set. The procedure for bringing the position to zero (Home) is shown in Fig 15. It can be seen the behavior of the Theta1, Theta2, and control effort. The control effort try to bring the swinging to home. It means Swinging UP-UP, Balancing UP-UP, and finally bringing home.



a) Closed Loop Poles Dow-Down



b) Closed Loop Poles Up-Up

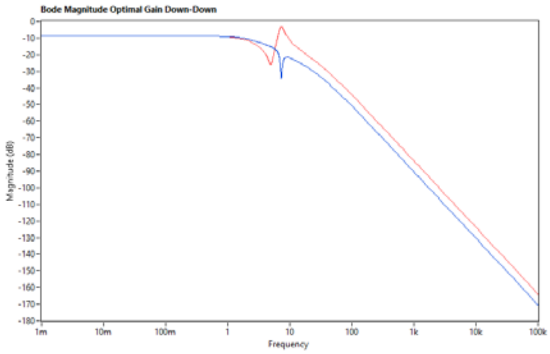


c) Closed Loop Poles Down-Up

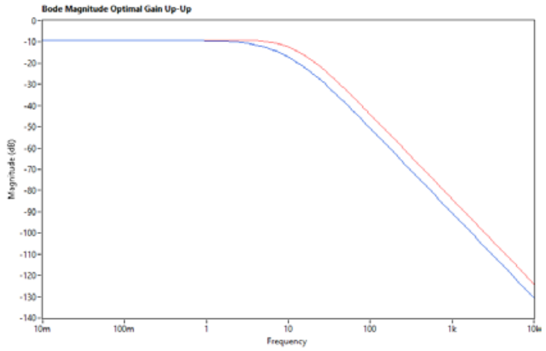
Fig.12: Closed Loop System Poles

For Swinging Down-Up, parameters are set and the results are as for Fig 16. The elbow is set

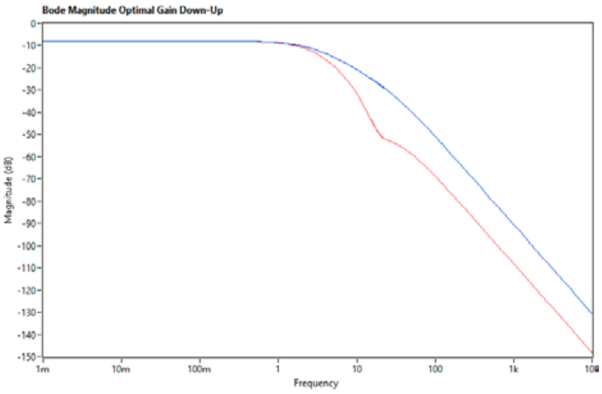
Down and the arm is set Up. After swinging, the controller tried to balance it and then bring it to the home position.



a) Bode Magnitude Down-Down

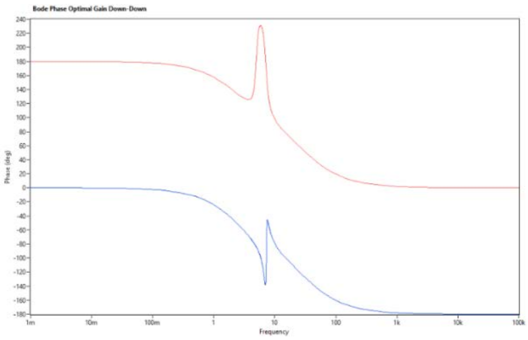


b) Bode Magnitude Up-Up

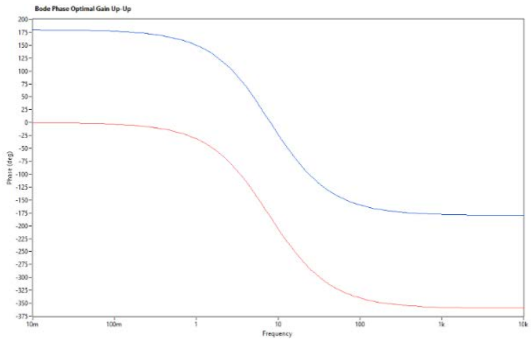


c) Bode Magnitude Down-Up

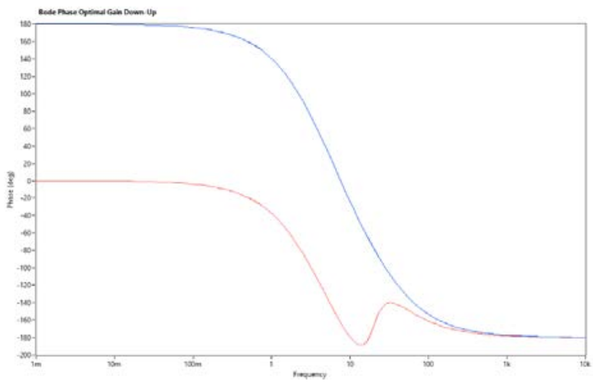
Fig.13: Frequency response of the system



a) Phase Response Down-Down

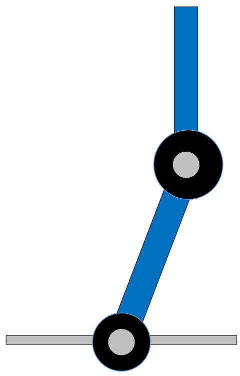


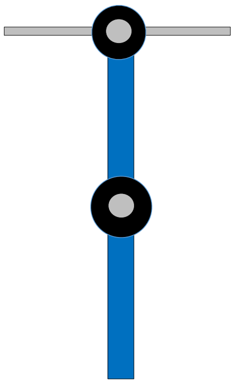
b) Phase Response Up-Up

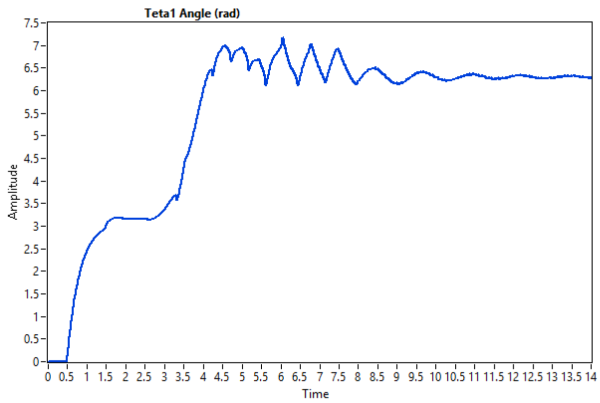


c) Phase Response Down-Up

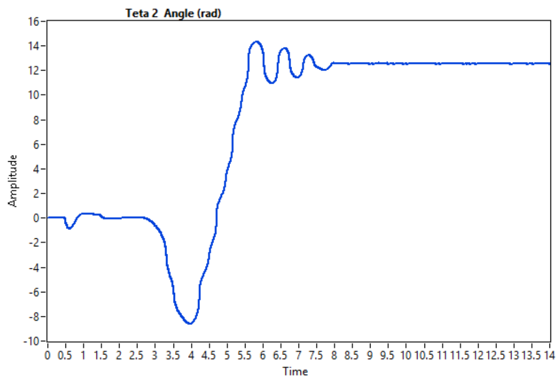
Fig.14: Phase response of the system

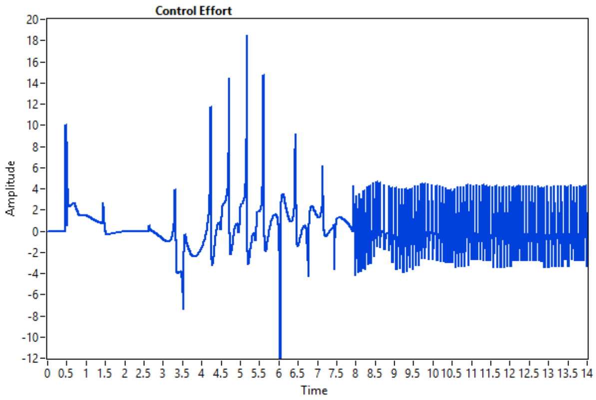






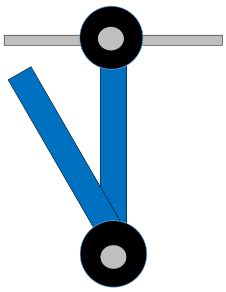
a) Swinging Up-Up b) Bringing home c) Theta۱ Angle

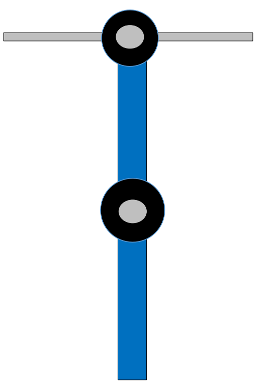


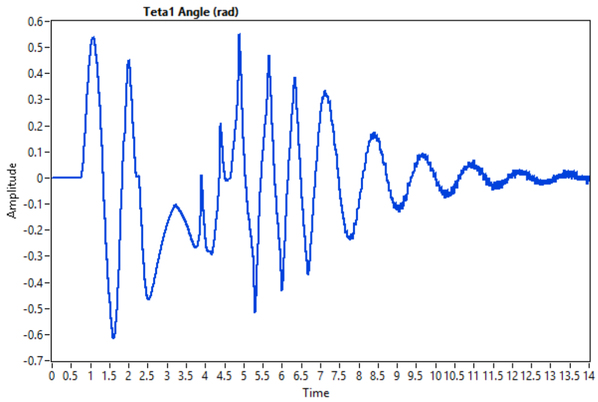


d) Theta۲ Angle f) Control Effort

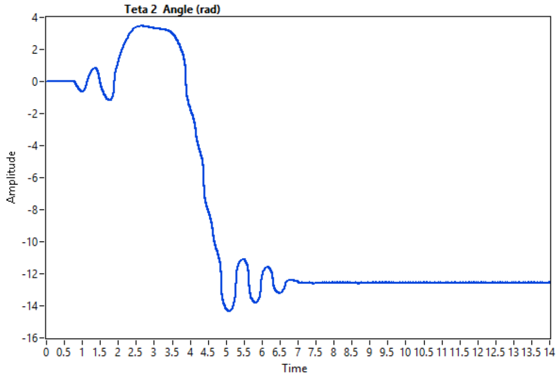
Fig.15: Control of swinging and balancing Up-Up



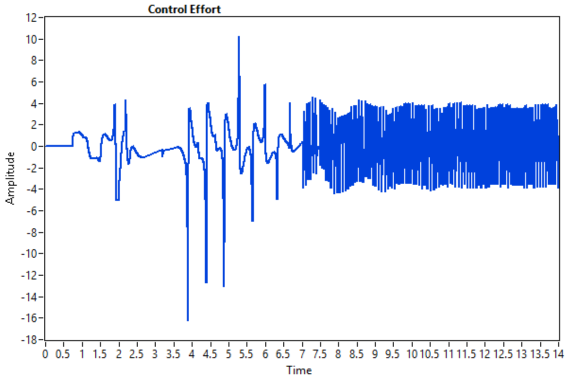




a) Swinging Down-Up b) Bringing home c) Theta۱ Angle



d) Theta۲ Angle



f) Control Effort

Fig.16: Control of swinging and balancing Down-Up

Conclusion

This paper represents the concept of two links under the actuated robot, which is called Pendubot. This system is valuable both in the investigation and for instruction in controls. Different aspects of the control hypothesis from local linear state feedback to balance the Pendubot in one position to more complicate global nonlinear controllers to swing up the Pendubot are easily illustrated.

References

Touati, N. and A. Chemori, Predictive control for the stabilization of a fast mechatronic system: from simulation to real-time experiments. IFAC Proceedings Volumes, 2013. **46**(5): p. 237-242.

Xin, X., et al., New analytical results of energy-based swing-up control for the Pendubot. International Journal of Non-Linear Mechanics, 2013. **52**: p. 110-118.

Neghab, H.K. and H.K. Neghab, Calibration of a Nonlinear DC Motor under Uncertainty Using Nonlinear Optimization Techniques. Periodica Polytechnica Electrical Engineering and Computer Science, 2021. **65**(1): p. 42-52.

Antonio-Cruz, M., V.M. Hernández-Guzmán, and R. Silva-Ortigoza, Limit cycle elimination in inverted pendulums: Furuta pendulum and pendubot. IEEE Access, 2018. **6**: p. 30317-30332.

Wu, J., et al., Control strategy based on Fourier transformation and intelligent optimization for planar Pendubot. Information Sciences, 2019. **491**: p. 279-288.

Chen, T. and B. Goodwine. Controllability and Accessibility Results for an $ N $-Link Horizontal Planar Pendubot. in 2019 18th European control conference (ECC). 2019. IEEE.

Huynh, X.D., et al., Application of fuzzy algorithm in optimizing hierarchical sliding mode control for pendubot system. Robotica and Management, 2017. **22**(2): p. 8-12.

Nguyen, N.S. and D.K. Nguyen, Parameter estimation of Pendubot model using modified differential evolution algorithm. International Journal of Modelling and Simulation, 2019. **39**(3): p. 157-165.

Rudra, S. and R.K. Barai. Design of block backstepping based nonlinear state feedback controller for pendubot. in 2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI). 2016. IEEE.

Lai, L.-C., Y.-Y. Fu, and C.-N. Ko. MQPSO Algorithm Based Fuzzy PID Control for a Pendubot System. in The 2017 International Conference on Artificial Life and Robotics. 2017.

Keshmiri Neghab, H., H. Keshmiri Neghab, and N. Pariz. Designing a Fuzzy Logic Controller for Automated Backward Parking Car. in 2019 IEEE 4th Conference on Technology In Electrical and Computer Engineering (ETECH 2019) Information and Communication Technology. 2019.

Cisneros, P.S. and H. Werner, Wide range stabilization of a pendubot using quasi-LPV predictive control. IFAC-PapersOnLine, 2019. **52**(28): p. 164-169.

Vardulakis, A. and C. Wei. Balance Control of the Pendubot via the Polynomial Matrix Approach. in 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT). 2018. IEEE.