EPORTFOLIO

Rotary inverted pendulum

Abstract

The design and realization of balancing controller in order to stabilize a rotary inverted pendulum is discussed

Rajagugan Abelaash

220213286

Introduction

The rotary inverted pendulum is a benchmark model for controlling non-linear system dynamins in control engineering. The design problem could be used to model and design controllers for real world stabilization problems such as balancing rockets during takeoff , automatic aircraft landing systems , aircraft stabilization in turbulent airflow etc. This report describes design of multiple robust controller for the balancing of a rotary inverted pendulum. The background and current solutions are discussed first and the problem is formulated in the next section. The system is modelled mathematically and then using simulink / Solidworks . The sensor selection process is discussed next , which discuss the suitable options for the required measurement and the possible problems. The controller design methodologies are described next after which the filtering requirements and procedures are discussed. Then the results and outcomes of the project a observed , and discussions are made based on these outcome , finally the design process is concluded with the better controller option.

Literature review

The rotary inverted pendulum system was introduced to feedback control community in 1992 by Katsuhisa Furuta, Professor at the Tokyo Institute of Technology, and hence it is often referred to as the Furuta pendulum. It was designed because a cart based inverted pendulum has movement limitations causing a restriction in the control system. A rotary inverted pendulum requires less space and less unmodelled dynamics. Because of this the study has found that there are less uncertainties in a rotary inverted pendulum compared to a traditional inverted pendulum as the rotor arm is connected directly to the drive motor shaft as opposed to other transmission mechanisms, hence is considered robust.**Invalid source specified.**

A research was conducted on the performance comparison of PID and SMC controllers , the team implemented the system using PLC . OMRON PLC capable of high speed counter and pulse output was used. The PID was tuned using the Ziegler Nichols first method and the SMC controller was based on the system dynamic model. The team found that the amplitude of fluctuation of the balancing angle of the pendulum was less when the PID controller is used , however the SMC controller provide longer stabilization time **Invalid source specified.**

Dual PID controller , LQR and FSF were compared in a study to determine the performance of each method. All methods were implemented on the Quanser ROTPEN trainer system. The performance of each controller was based on settling time , overshoot , range of pedulum oscillations , range of arm oscillations , motor voltage and robustness. Both FSF and LQR performed better overall compared to the Dual PID controller . . LQR performed the best in all of the sections with highest efficiency and was the most robust out the 3 . The controllers were implemented for range of +-30 degree from the vertical upright position as the linear model work best for this range.**Invalid source specified.**

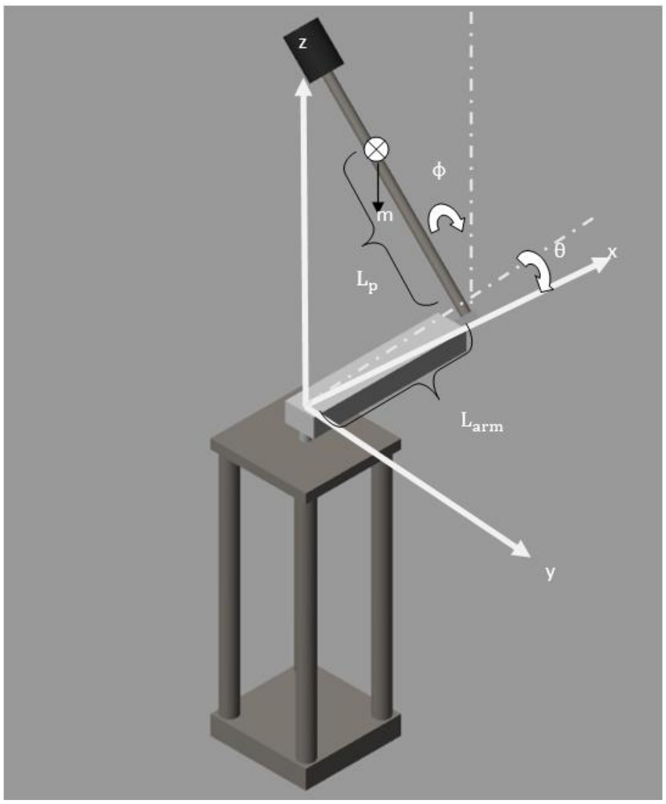
A study was conducted to implement LQR control on a rotary inverted pendulum by refining the system using PID control and swing up control. The system was introduced with time varying uncertainty . It was found that the system ended up being very robust to outside interference.**Invalid source specified.**

Problem formulation

The problem tasked to handle is the development of a robust controller that is capable of maintaining a rotary inverted pendulum in its upright equilibrium position . The rotary inverted pendulum consists of a free moving pendulum , which is controlled to its upright position by the means of torque input to the rotor arm . The torque input to the system controls the direction and speed of the rotational motion of the rotor arm. The rotational movement will transfer energy to the pendulum , allowing it for rotary movements . The system is non-linear and highly dynamic. The controller must be able to balance the forces in the pendulum by computing and predicting the motion depending on the magnitude and direction of the torque input to the system. The controller must robust enough to reliably balance the pendulum. Sensors are used to monitor the current state of the system continuously.

System modelling

The physical model of the rotary inverted pendulum was built using Solid works , material properties were added to each component to give them realistic masses and moment of inertia . The Solid works model was then imported into Simulink with smimport function which imports all of the relations alone with mass properties , densities and moment of inertia of the model and translate them to a Simulink model .



m	0.03918kg
L_p	92.85mm
L_{arm}	80mm
J_{arm}	26.197kgmm²
J_p	54.057kgmm²

Assumptions

Assuming that there is no damping on the joints , the model is considered as a ideal implementation and there is no external forces or internal energy losses in the system model.

Assuming that the pendulum mass is a simple cylinder, the rod is considered as a long thin uniform rod and the coupling joint is considered to be a rectangular block. All these components are assumed to be 1 infinitely stiff rod.

The rotor arm is considered to be a rectangular rod , with infinite stiffness .

Mathematical modelling

Mathematical modelling is done in order to characterize the fundamental problem in terms of a the mathematical dynamics occurring in the system. The dynamic equations of the system could be achieved using Newton mechanics , however the rotary inverted pendulum is a complex dynamic problem and it is quite challenging to obtain the models for the system using the classical approaches. The Euler Lagrange method of analysis is used to obtain these equations which uses the energy conversions in the system to obtain a good model of the equations.

The kinetic energy of the pendulum could be described as follows , as it contains rotational and translational components to its kinetic energy.

$$K_{\text{pendulum}} = \frac{1}{2}mv^2 + \frac{1}{2}\mathrm{I}\omega^2 = \frac{1}{2}\,mv^2 + \frac{1}{2}\mathrm{J_p}\dot{\emptyset}^2$$

The kinetic energy of the arm is described by :

$$K_{\text{arm}} = \frac{1}{2}\mathrm{I}\omega^2 = \frac{1}{2}\mathrm{J_{arm}}\dot{\theta}^2$$

The total kinetic energy would be descried by the following equation :

$$K_{\text{total}} = \frac{1}{2}\,mv^2 + \frac{1}{2}\mathrm{J_p}\dot{\emptyset}^2 + \frac{1}{2}\mathrm{J_{arm}}\dot{\theta}^2$$

The rotor arm is considered to be at zero potential , hence the only gravitational energy in the system is in the pendulum which is described as follows:

$$Potential\ energy\ P = mgL_p(cos(\phi) - 1)$$

The pendulum is considered as a rigid body and treated as a single contraption, hence in order to calculate the velocity , the positional components of the center mass of the pendulum must be obtained :

$$X_x = L_{arm}cos(\theta) - L_p sin(\emptyset)sin(\theta)$$

$$X_y = L_{arm}sin(\theta) + L_p sin(\emptyset)cos(\theta)$$

$$X_y = L_p cos(\emptyset)$$

The velocity components of the center of mass will be the derivative of its positional vector :

$$\dot{X}_x = -\dot{\theta}L_{arm}sin(\theta) - L_p(\dot{\theta}sin(\emptyset)sin(\theta) + \dot{\emptyset}cos(\emptyset)sin(\theta)$$

$$\dot{X}_y = \dot{\theta}L_{arm}cos(\theta) + L_p(\dot{\emptyset}cos(\emptyset)cos(\theta) + \dot{\theta}sin(\emptyset)cos(\theta)$$

$$\dot{X}_z = -\dot{\emptyset}\, L_p sin(\emptyset)$$

By squaring the terms and adding them up together we can get the velocity of the center of mass of the pendulum , which is described by the following expression:

$$v^2 = \dot{\theta}^2 L_{arm}{}^2 + L_p{}^2(\dot{\emptyset}^2 + \dot{\theta}^2 sin^2(\emptyset)) + 2\dot{\emptyset}\dot{\theta}L_p L_{arm}cos(\emptyset)$$

Hence substituting this back into the K.E equation we get the :

$$K.E_{\,total}\, T = \frac{1}{2}J_{arm}\dot{\theta}^2 + \frac{1}{2}m(\dot{\theta}^2 L_{arm}{}^2 + L_p{}^2(\dot{\emptyset}^2 + \dot{\theta}^2 sin^2(\emptyset)) + 2\dot{\emptyset}\dot{\theta}L_p L_{arm}cos(\emptyset)) + \frac{1}{2}J_p\dot{\phi}^2$$

The lagrangian function is described below , where T is the total kinetic energy of the system and P is the total potential energy in the system :

$$L = T - P$$

Substituting the Kinetic energy expression and the potential energy expressions that were derived previously we get :

$$L = \frac{1}{2}J_{arm}\dot{\theta}^2 + \frac{1}{2}m(\dot{\theta}^2 L_{arm}{}^2 + L_p{}^2(\dot{\emptyset}^2 + \dot{\theta}^2 sin^2(\emptyset)) + 2\dot{\emptyset}\dot{\theta}L_p L_{arm}cos(\emptyset)) + \frac{1}{2}J_p\dot{\phi}^2 + mgL_p(1 - cos(\phi))$$

Euler-Lagrangian with respect to $\dot{\theta}$, where τ is the torque input by the motor . It is assumed that no energy is lost in the system.

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \tau$$

$$J_{arm}\ddot{\theta} + \ddot{\theta}\, mL_{arm}{}^2 + \{\, mL_p{}^2(\ddot{\theta}sin^2(\emptyset) + 2\ddot{\theta}\ddot{\emptyset}sin(\emptyset)cos(\emptyset))\} + \{L_{arm}L_p m(\ddot{\emptyset}cos(\emptyset) - \dot{\emptyset}^2 sin(\emptyset))\} = \tau$$

Euler-Lagrangian with respect to $\dot{\phi}$

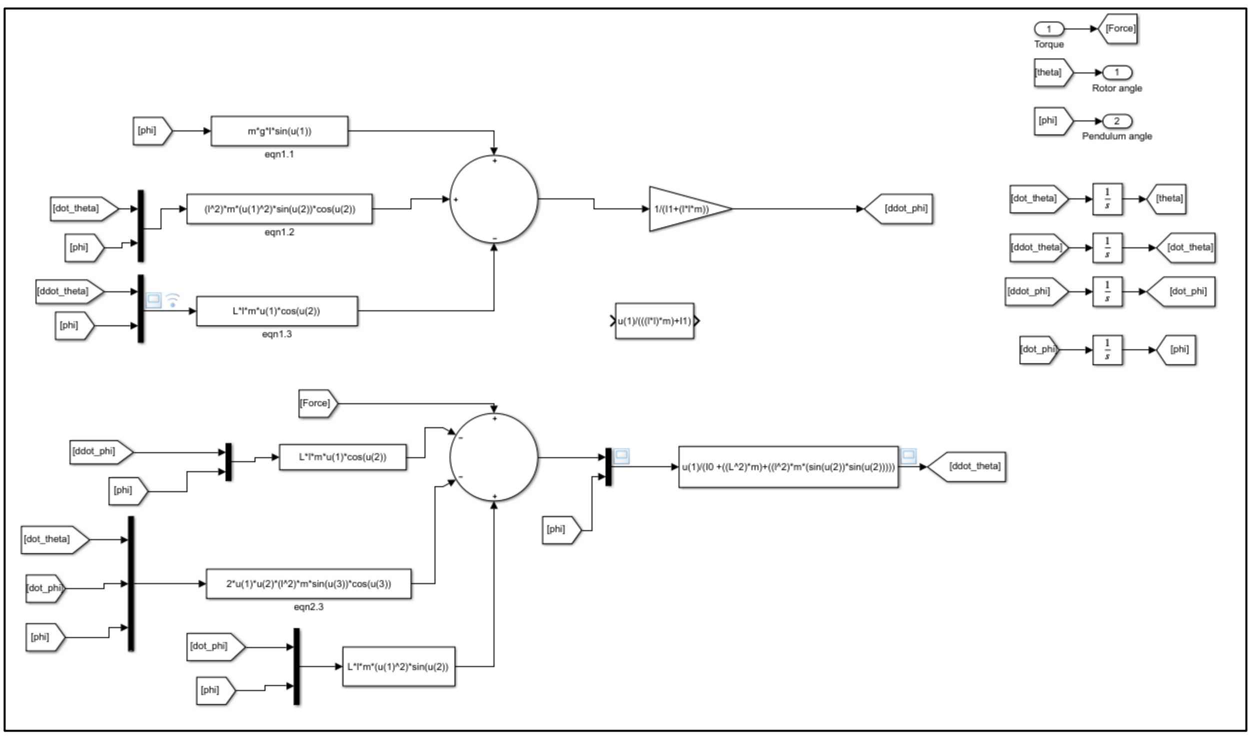
$$\frac{\partial}{\partial T}\left(\frac{\partial L}{\partial \dot{\phi}}\right) - \frac{\partial L}{\partial \phi} = 0$$

$$J_p\ddot{\emptyset} \, + \ddot{\emptyset}\, mL_p{}^2 + L_{arm}L_p m(\ddot{\theta}cos(\emptyset) \, - \, L_p{}^2 m\,\dot{\theta}^2 sin(\emptyset)cos(\theta) \, - \, mgL_p sin(\emptyset) \;\; = \; 0$$

The non-linear model was obtained above , for engineering analysis the system must linearized as this makes it easier for calculations and simulations . This is done by assuming that the system is in equilibrium , this happens when the pendulum is in its most upright position , where ϕ approaches 0 , hence cos(ϕ) -> 1 , sin(ϕ) -> ϕ and $\dot{\phi}^2$ -> 0 . Applying this will result in the following linearized equations.

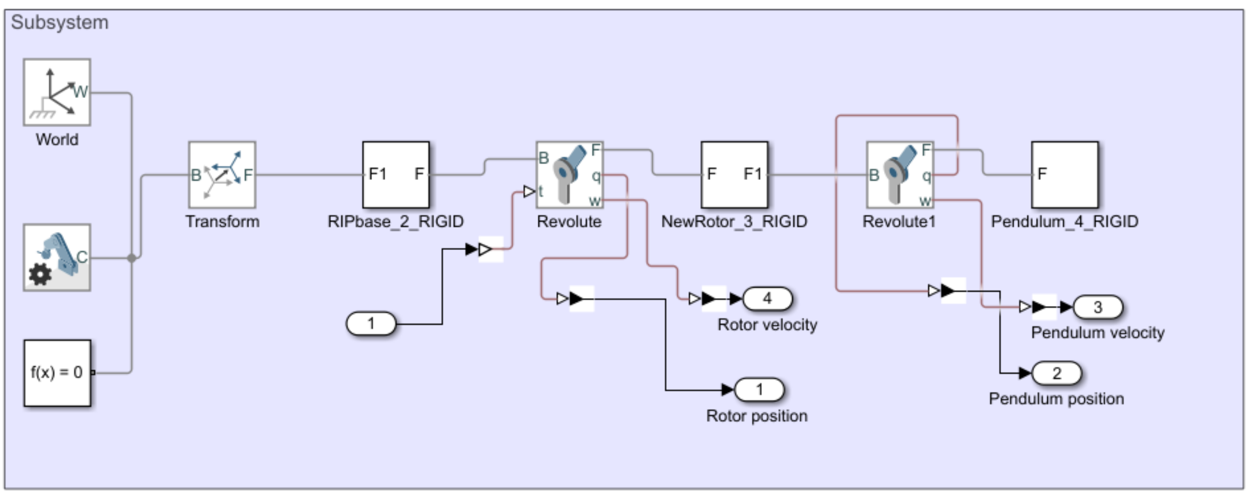
$$J_{arm}\ddot{\theta} \, + \ddot{\theta}\, mL_{arm}{}^2 + \{\, mL_p{}^2(\ddot{\theta}\emptyset^2)\} \, + \{L_{arm}L_p m(\ddot{\emptyset})\} = \, \tau$$

$$J_p\ddot{\emptyset} \, + \ddot{\emptyset}\, mL_p{}^2 + L_{arm}L_p m(\ddot{\theta} - \, L_p{}^2 m\,\dot{\theta}^2 \emptyset cos(\theta) \, - \, mgL_p\emptyset \;\; = \; 0$$



The mathematical model obtained above was implemented in Simulink to try and verify the accuracy of it when relating it to the simulation of the physical model by comparing the results obtained. However unfortunately the design failed , this could be caused due to a multitude of reasons. The designed physical model is much more complex compared to the derived mathematical model , one of the reasons is that the mathematical model assumes that the system is simple with very simple joints connecting them together however the actual model has more complex joints where the bodies connect to each other by more complex methods , the pendulum is connected to the rotor via a block that acts like a coupling joint that physically sits on the rotor , the same is to be said with how the rotor is connected to the base , components interconnect between each other and these interactions between them are not considered in the mathematical model. However Simulink is a much more accurate tool which considers these complex interbody relationships when simulating the output. Another source of uncertainty occurs due to the moment of inertia calculations, it is assumed that the pendulum contraption is rigid body where the moment of inertia is calculated using simple composite shapes of a rod (the pendulum extender rod),cylinder (pendulum mass)and a simple block (coupling joint between the pendulum and the rotor) , however this is not the case as in reality the each of these components interconnect between each other , example the coupler is not a simple block but has hole where the pendulum rod physically sits inside and is connected to the rotor where it also sits inside the coupler joint. The pendulum rod also sits physically inside of the cylindrical pendulum mass. The rotor is assumed to be a simple rectangular block , but in reality it has a long elongation at the end of it to allow for connections to the pendulum which in turn affects how it would be modelled in a real world scenario. Hence these uncertainties and possible unmodelled dynamics in the mathematical model is assumed to be the problem as to why the model is failing.

For the further completion of the project with higher reliability, accurate state spaces models of the physical design are acquired by using MATLAB to linearize the system. This way any possible unmodelled dynamics and uncertainties are resolved. The findop function finds the operating specifications of a model at a defined time , in the case of this design , when the pendulum is in the most upright position , which the stable point of the system. This operating specifications report is then linearized in order to obtain the state space models for the system using the MATLAB linearize function.




```
A =

         0    1.0000         0         0
         0         0  440.5470         0
         0         0         0    1.0000
         0         0  443.3680         0
```

```
B =

   1.0e+04 *

         0
    1.5425
         0
    1.2338
```

Sensing

Optical encoders detect light passing through a designated disc, and the PCB converts the data into pulses. Magnetic encoders use a magnetic sensor instead of an optical sensor, and the rotating discs include a series of magnetic poles that are detected. The movement direction is based on the phase connection between Channels A and B, on which one leads the other at the sensing point. The code disc has two tracks, Channel A and Channel B, which are coded exactly 90 degrees out of phase. If higher resolutions are required, the sensor can count the signal's leading and trailing edges in each channel, quadrupling the pulses per revolution.

The OMRON E6B2-CWZ6C is a fantastic rotary encoder ; it has a base resolution of 2000ppr, which means increments of 0.18 degrees may be detected, and this can be raised further utilising encoder methods. It also has a very fast rising time of only one second.

Item Model	E6B2-CWZ6C
Power supply voltage	5 VDC –5% to 24 VDC +15%, ripple (p-p): 5% max.
Current consumption *1	80 mA max.
Resolution (pulses/rotation)	10, 20, 30, 40, 50, 60, 100, 200, 300, 360, 400, 500, 600, 720, 800, 1,000, 1,024, 1,200, 1,500, 1,800, 2,000
Output phases	Phases A, B, and Z
Phase difference between outputs	90°±45° between A and B (1/4
Output configuration	NPN open-collector output
Output capacity	Applied voltage: 30 VDC max. Sink current: 35 mA max. Residual voltage: 0.4 V max. (at sink current of 35 mA)
Maximum response frequency *3	100 kHz
Rise and fall times of output	1 μs max. (Control output voltage: 5 V, Load resistance: 1 kΩ, Cable length: 2 m max.)

A rotary potentiometer sensor could be used to detect the position of the rotor arm, as it is a simple yet effective, low-cost, and relatively straightforward solution for the needed objective. Potentiometer sensors are notorious for their excessive noise and lack of robustness, however this is a tradeoff taken here to keep costs down while maintaining a simple solution. The sensor must be able to detect motion in all directions and be compact. For this circumstance, the HP-16 Series from MIDORI PRECISIONS is a good choice; it has enough mechanical travel for the job at 10 revolutions with 3600 degree detection. It also has a high resolution of 0.016 percent and a linearity of 0.25 independently.

Effective Electrical Travel	3600	°
Total Resistance Tolerance	±5	%
Independent Linearity	±0.25	%
Rated Dissipation	2(40℃)	W
Insulation Resistance	MIN. 100/DC1000V	MΩ
Dielectric Strength	AC1000/1 Minute	V
Temperature Coefficient of Resistance	MAX. 20	ppm/K
End Output Voltage	MAX. 0.25	%
Equivalent Noise Resistance	MAX. 100	Ω

The aspects to be measured must be carefully stated in order to ensure that the controller receives sufficient feedback in order to achieve the project's goals. The type of sensing required is dependent on the type of controller to be used as well and the overall goals of the project. It has already been established that position sensing for both the rotor and the pendulum is required to ensure sufficient feedback to the controller design to control the parameters of the system in the design proposal report. In addition to position sensing , a newer controller design that is more robust requires the velocity sensing as well , this to facilitate for better state estimations and filtering of noisy signals . As this a simulation based project only the sensor noise and possible counteractive methods for noise via filtering is discussed. Noise is introduced into the system to better simulate real world scenarios . The matlab band limited white noise is applied to all the sensor input at a power level of 0.0001 , this causes the position sensors to have a +-2% uncertainty , the same is done with the velocity sensors.

.

Methodology

Controller development

In order to satisfy the requirements of the project a good feedback controller must be established in order to maintain the pendulum in balance even when subject to noise and disturbances because a open-loop system is incapable producing the desired result. An open loop system controller design is based on predetermined calculations in-order achieve the output required, this however is unreliable in the real world as there are disturbances and uncertainties involved that affects the system model in a way which are not accounted for in the preliminary calculations this will in-turn affect the required parameters from the system to achieve the desired output which the controller cannot provide as it is not accounted for. Open loop controllers are also very difficult to implement on plants which have model uncertainties or are open loop unstable, the rotary inverted pendulum is a highly non-linear system with a lot of uncertainties which is very unstable, as the model is only stable when the pendulum is in full upright position. Hence the need for a feedback controller is justified. A negative feedback controller will be perfect for the purposes of the project. A negative feedback controller utilizes its effective input as the difference between the reference input and the feedback signal, while a positive feedback controller is the sum. In a negative feedback loop the gain decreases, while stability, accuracy and bandwidth increases where as a position feedback controller pushes the system towards instability. Two different negative feedback controllers were implemented in the system which are based on 2 different design methodologies in order achieve the required output from the system to compare performance.

PID (Proportional-integral-derivative) controller

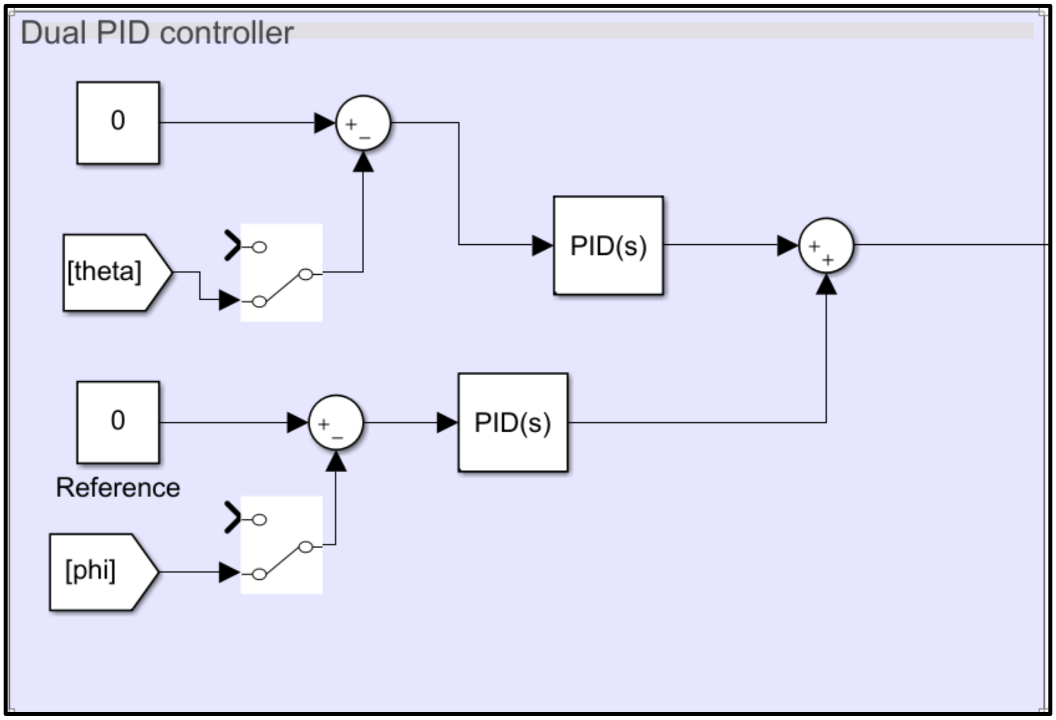
A proportional-integral-derivative (PID) based controller design was established in order achieve the desired output from the system without the need for a model. A PID controller design was found to be the most common among classical control approaches and is used quite commonly in real world industrial problems. The basis of control is dependent on a error signal which is obtained by the difference of the required value and the actual value continuously , correction is then applied to the error depending on the proportional, integral and derivative coefficients ,the output is then transferred to the model to achieve an actual value reading closer to the required value.

Transfer function of a PID controller

$$u(t) = K_p . e(t) + K_i \int_0^t e(t)\, dt + K_d \frac{d}{dt} e(t)$$

The implementation of this controller was using the PID block which is built into Simulink. After rigorous tuning of the PID it was found that each of the parameters refer to a different aspect in control, the gain Kp controls amplification of the speed of the movement, the integral term Ki controls the steady state accuracy of the system and the derivative term Kd controls the overshoot of the arm. These parameters were tuned first for the pendulum angle PID controller , Kp was tuned until the arm is capable of moving to or just narrowly pass the upright position , then depending on the amount of overshoot , Kd was increased and tuned until the overshoot of the was acceptable , finally Ki was introduced until there was no steady state error on the pendulum angle . However this was much more challenging process to get the desired response of the system which is robust enough negate disturbances. The filter coefficient was increased in pendulum PID until it was large enough to approximate Kd*(N*s/s+N) to Kd*s. Initially design consisted of a single PID design which focused only on bringing the pendulum to a fully upright , zero reference that after continuous tuning of the coefficients of PID transfer function was capable of balancing the pendulum in the upright position however was not capable of stabilizing the system into a fixed point rotor angle point regardless of the tuning , initial designs of the PID achieved balancing but contained continuous spinning of the rotor arm, hence the parameters were further tuned . It was found that the reason for this was due to steady error in the pendulum angle, in further designs it was possible slow the continuous spinning a lot more. The best achieved with the single PID still contained very slow movements of the rotor hence the need for a Dual PID controller was clear as the single PID couldn't negate this steady state error by its own. The logical reasoning for introducing an additional PID controller was simple, it introduces additional zeros on the root locus plot to stabilize the system faster and much efficiently with use of lower gain. It also gives controller the readings and errors related to the rotor angle .

Hence the second PID was implemented for the rotor angle , and the output of which is summed together with pendulum PID and sent to the model as a torque input. The procedures followed for tuning were the same however it was much more complex as the stabilization affected both system interconnectedly until the systems was stable enough to tune better. The PID tuning for the rotor was initially done by trial and error , and with a little support from the Matlab inbuilt PID tuning function. After minimum stabilization was reached , the parameters were tuned as described above for achieving the desired response from the controllers. The dual PID controller is implemented in Simulink as follows where output of the summation block is connected directly to the model , and the rotor angle is denoted by theta while the pendulum angle is denoted by phi . The error signals for each of the PIDs are the difference between the reference and their respective angles.



LQG (Linear quadratic gaussian) controller

A LQG controller has been implemented which follows the model based design procedure. It is a combination of the multivariate feedback controller LQR and the state estimator Kalman filter. Kalman filter could be used to estimate any unmeasured variables or for filtering the variables that are directly measured. In the controller design implemented , we assume a full state feedback and hence the Kalman filter is mainly used to estimate states but also to filter out any white gaussian noise in the system. The LQR controller is a well established feedback control strategy which is then used to stabilize the system after state estimations are taken. The system is linearized to obtain the state space models of the system at equilibrium position . The system is in the standard state space form given below where $\dot{x}$ in which A and B refer to the state of the system and u is the control input to the system and also a measurement equation y is formed where C represents the measurements taken and V represents the noises from measurements ;

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

$$y = Cx + V$$

The main objective of the LQR design is to find a state feedback law $u = -Kx$ that minimizes the cost function given by the following equation , where Q matrix defines the weights on the states and R matrix defines the weights on the on control input of the cost function.

$$J = \int_0^{\infty} (x-r)^T Q (x-r) + u^T R u + 2(x-r)^T N u \, dt$$

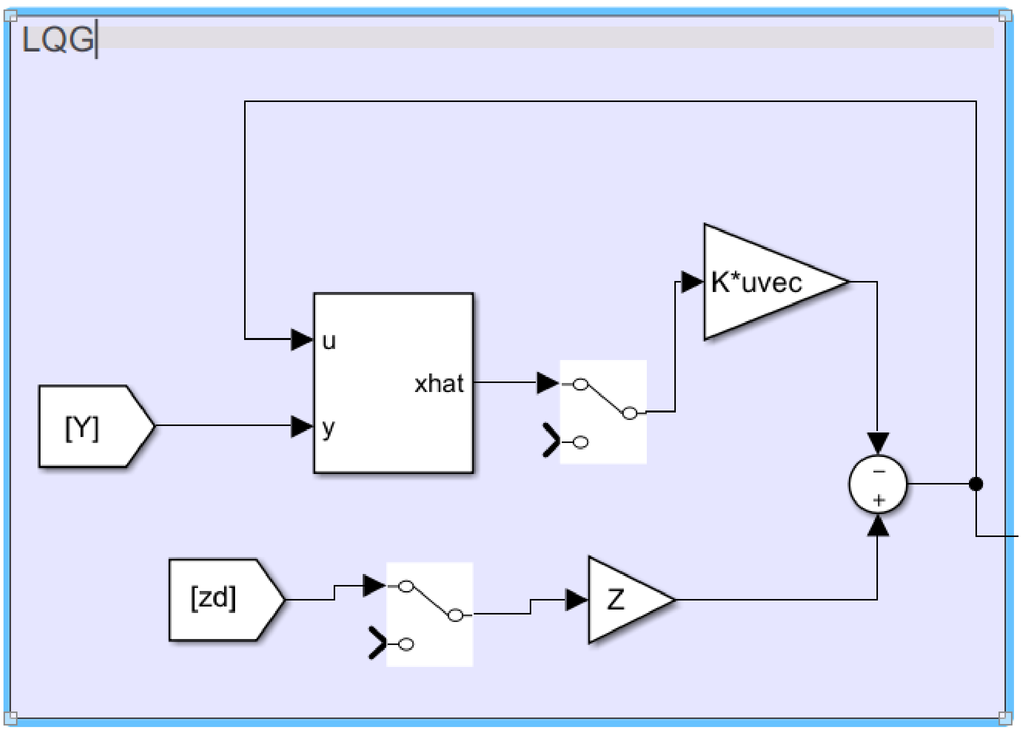
K in the control law is given by :

$$K = R^{-1}(B^{T}P + N^{T})$$

And P is found by solving the Riccati equation

$$A^{T}P + PA - (PB + N)R^{-1}(B^{T}P + N^{T}) + Q = 0$$

The controller is implemented in Simulink by first linearizing the system and obtaining the state space models using the MATLAB built in functions . after which the Q and R matrix are defined and initialized which are then in the later stages of development tuned to get the desired functionality of the controller. The gain K is then found using the MATLAB built in function lqr using the A,B,Q and R matrices . The closed loop system matrix is then found and verified to ensure the stability of the system. The first element of the of the K matrix defines the optimal gain to stabilize the system. The obtained values are then used in the Simulink environment to design the LQG controller. The LQG implementation is far simpler compared to the PID implementation as there no process of rigorous tuning involved. Also mainly because the Simulink interface allows for easy implementation by the use of in built functions and blocks. The implementation diagram is given below , where Y is vector that contains 4 measurements which is the angle and velocity of both the pendulum and the rotor , this design a full state feedback design. Which is compared against the desired state of the system inorder to achieve the optimal movement response. The gain Z corresponds to the optimal gain to achieve rotor balance.



Filtering discussions

Unwanted noise in electrical and electromechanical systems is a common problem in the real world , often the effects of said noise could be detrimental to the system design and maybe harmful due to the way they introduce unwanted erratic behavior into systems. The generation of noises are unpredictable and could be caused due a number factors in complex system , at which point it is challenging to figure out which component is causing the noise or if it caused by external factors due to emfs and other non visible sources. Hence filtering techniques are implemented on these systems to minimize the errors and produce more reliable and robust results. In the case of the rotary inverted pendulum described in this project , noises are artificially introduced via sensor readings to determine the robustness of the system to combat these noises when implemented in the real world. Filtering techniques are introduced to increases the model's robustness , as the filtering techniques will remove the erratic behavior of the noise. The rotary inverted pendulum is a highly non linear system which when affected by noises would cause erratic behavior and imbalance of the system. Two different filtering methodologies are implemented depending on the type of controller being used. PID implementation used a first order low pass filter to help with signal processing however , it was found that the PID implementation is not compatible with more noise filtering due to the delays introduced in the angle sensing because of filtering. This is discussed further in the results and discussions.

Model based filtering is a procedure available in LQG based controller design because of the use of Kalman filter for state estimation. If the Kalman filter is set up with full state feedback control we are able to use this filtering method , hence a design consideration was made to employ full state feedback and use the prowess of Kalman filter for white gaussian noise filtering. In addition to being able to measure and estimate states, Kalman filter is capable of generating a optimal estimate of desired quantities given a set of measurements , which means it is capable of neglecting any noise and only the required data is obtained from the input. Kalman filter is a recursive data processing algorithm , hence the speed of filtering is excellent and no delays are involved in the filtering process.

Results/Functionality and outcomes

The overall performance of each of the control methodologies is compared in terms of the criterion set out to achieve and the overall success of the design in terms of robustness. Both the pendulum and the rotor start at angle of 10 degrees from the origin , this is to ensure that the linearization of the pendulum does not fail , as closer angle to the upright position mean higher stability

Balancing of the pendulum

The first major objective of the controller is to actually stabilize the system in the upright position , and having minimal movement after the actual stabilization of the pendulum arm.

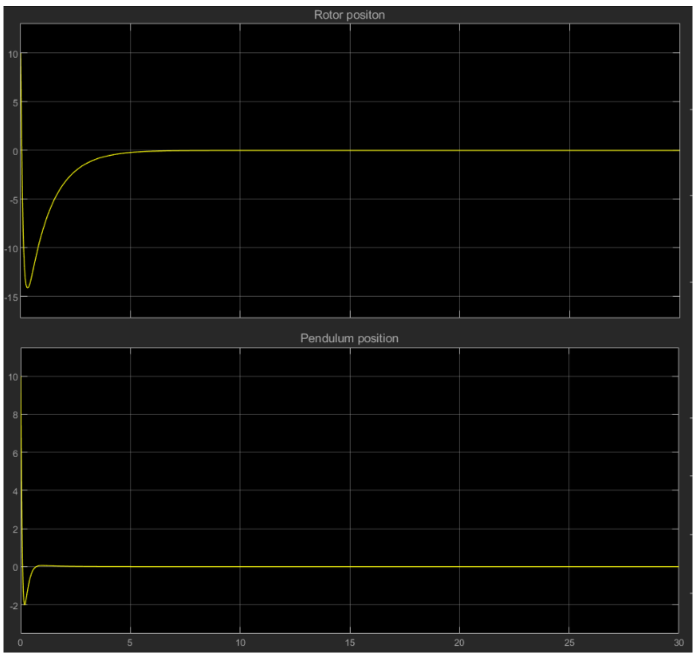


Figure 2 : LQG implementation

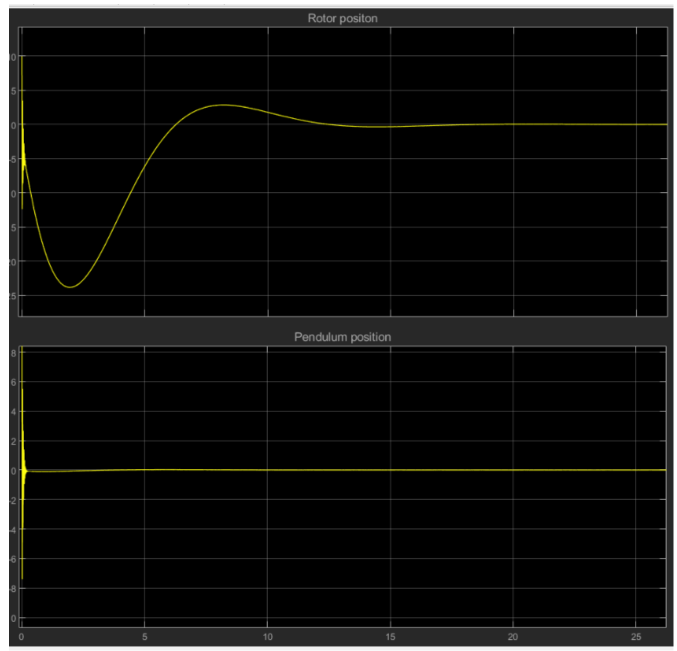
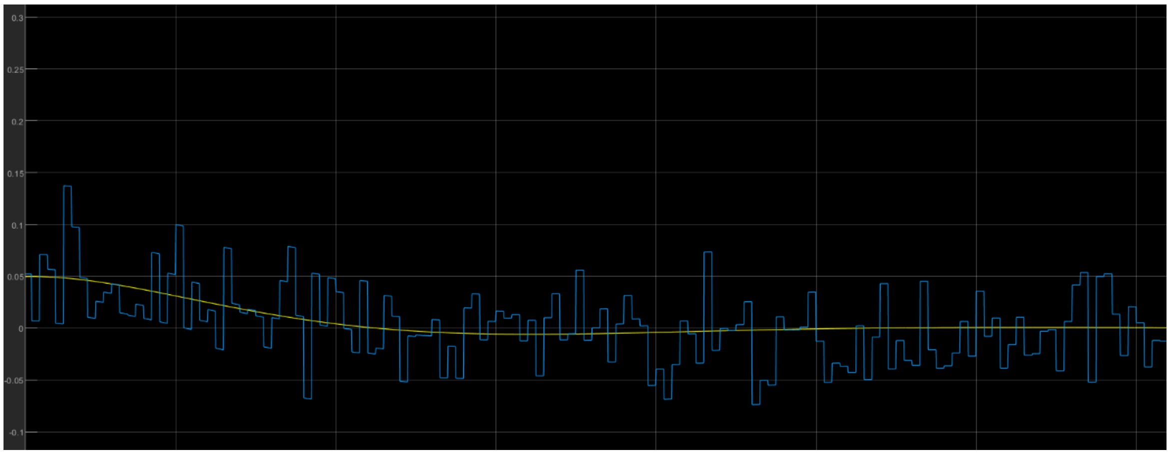


Figure 1 PID implementation

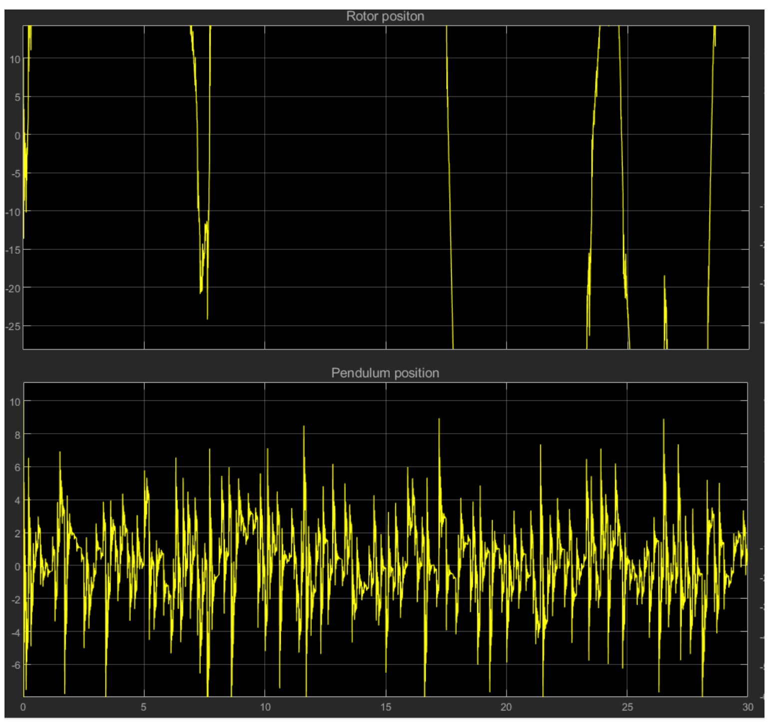
Both controllers were able to successfully stabilize the pendulum and reach steady state, however it is found that the LQG controller performed better than the designed PID controller in terms having minimal overshoot , lower settling time and also reaching stabilization faster . The PID took about 14 seconds to stabilize where as the LQG only took about 5 seconds with minimal wasted movements. It is also noted that line of the LQG is much smoother hence contain less noises in the actual signal processing during the actual stabilization.

Performance of the controller with noises and filtering

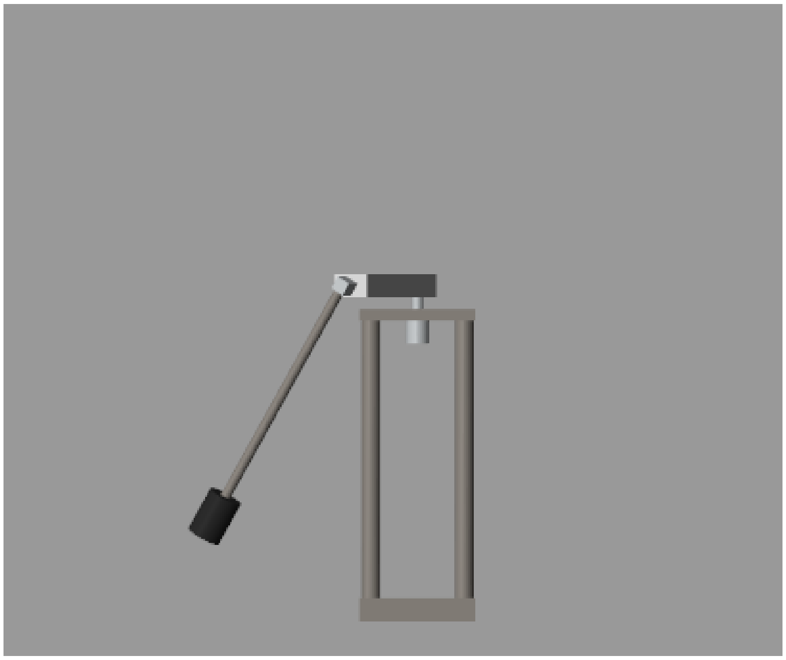
Noises are introduced into the system via the sensor outputs , these noises are then filtered out using different technique depending on the controller being used. The type of noise used is band limited white gaussian noise at a power level of 0.0001 . This gives the sensor a uncertainty of 4% .



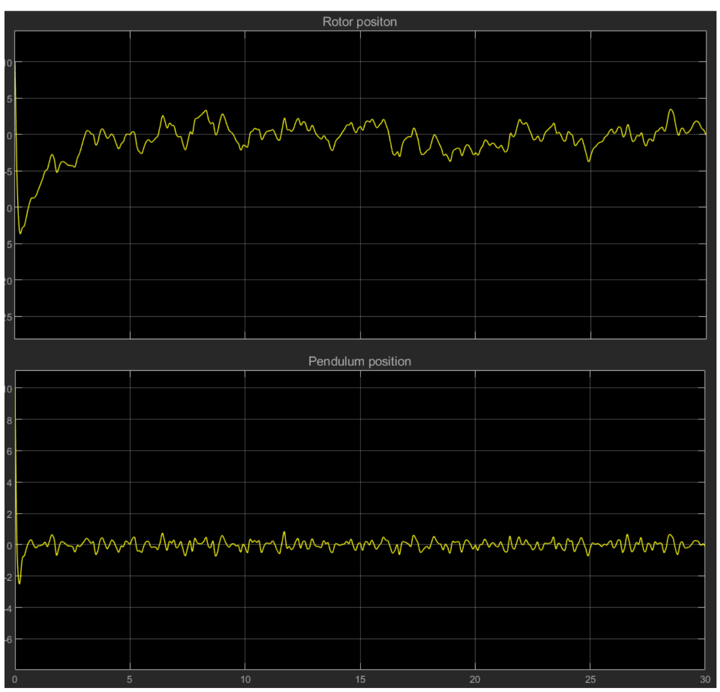
The PID controller is given this unfiltered signal to observe the controller response. The following response is observed. The pendulum is still able to be balanced without falling over , however there is jittery motion and the rotor arm is spun around erratically to prevent the pendulum from falling over.



Now a first order low pass filter with a time constant of 0.01s is implemented to try and prevent the noise from reaching the controller. The pendulum fails to keep up stabilization and falls over , then moves in a erratic motion. The same was repeated for a lower time constant and repeated multiple time , however the pendulum fails every time , until the time constant is so low that there is no difference between the filtered and unfiltered signals , hence this is considered unsuccessful.

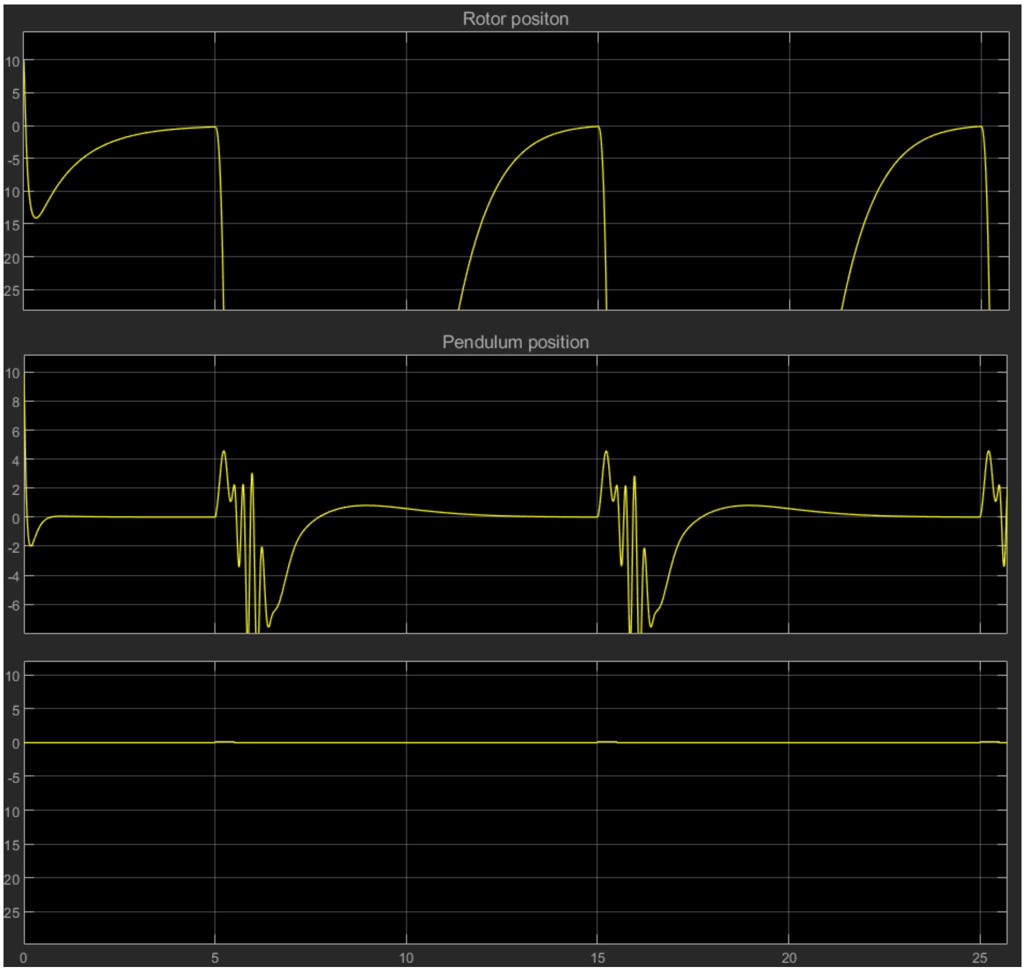


The same level of noise is applied to the LQG controller with the Kalman filter and the response is observed without any external filtering. The response observed was impressive , the controller was able to minimize a lot of the noise with only having small wobbles or jitters in the stabilization , not only was the pendulum stabilized but also the rotor was hovering over its reference point as well.

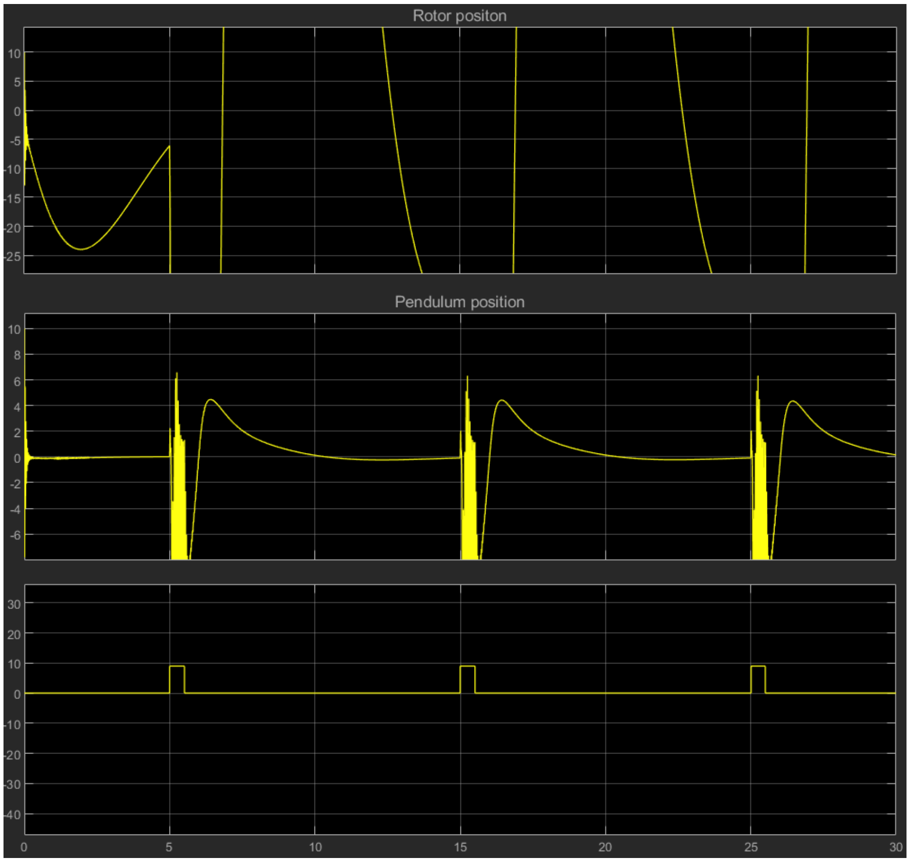


Disturbances

External disturbance is added to the system in the form of a force acting on the top of the pendulum , this is simulate a touch or a push and the response of each system is observed , all noises and filters are turned for this part of the simulation a better visualization of how robust each controller is to external forces. The force applied is put into the system using a signal generator to simulate consecutive external disturbance to further clarify the capabilities of each controller. Initial testing started at 0.050N of force. The LQG response is observed first . The LQG is capable of holding the pendulum upright and restabilize the system . After multiple iterations of testing by gradually increasing the force applied , it was found that the LQG fails at a disturbance force of 0.14N as the pendulum is knocked over and it is unable to pick it back up



he PID controller is now added external disturbance to observe the response. The PID controller was capable of handling an external distrubance of up to 9N



Discussions

Both controllers were successful in terms of balancing the pendulum to its most upright position , however it is quite clear that the PID needs better tuning in order to improve its performance specifications , the LQG controller performed better overall in all testing methods except for disturbances. This version of the PID failed in terms of being a viable option for environments with high noise , however its disturbance performance is much better than the LQG , this makes it quite a hardy controller which justifies its overall preference in industrial environments. LQG on the other hand is excellent in noise filtering , and would be a viable option in high sensitivity area of implementation.

Conclusions

It is found that the LQG controller performs better overall and is the better controller out the 2 , however the PID does need better tuning for optimum results. The LQG is a much faster approach to implement if performance is the key goal in the stabilization problem, as further tuning of the PID require much more time investment .

Bibliography

Anon., n.d. s.l., s.n.

B. Prakash, B. K. R. ,. R. K. B., 2016. *Design, implementation and comparison of different controllers for a rotary Inverted Pendulum.* Delhi , India, IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES).

G. Sainzaya, F. Y. T. H. ,. C. Y., 2017. *LQR control with refined PID to balance rotary inverted pendulum with time-varying uncertainty.* Pingtung, Taiwan, International Conference on Fuzzy Theory and Its Applications (iFUZZY).

K.Furuta, ,. M. Y. ,. S. K., 1991. *Swing up control of inverted pendulum.* s.l., International Conference on Industrial Electronics, Control and Instrumentation.

S. Howimanporn, S. C. ,. S., 2020. *Comparison between PID and Sliding Mode Controllers for Rotary Inverted Pendulum Using PLC.* s.l., International Conference on Automation, Control and Robots (ICACR).