Edukit Rotary Inverted Pendulum

An Open and Configurable System With Digital Motor Controller Technology

Revision 2.0
January 12, 2021

Table of Contents

1. Edukit Quick Start and Edukit System Web Site

This Edukit Manual is available at the **Edukit System Website**. Please access the Web site for system resources and updates.

These resources include Edukit Quickstart with

1. System Assembly Video Guidance Sequence
2. *Installation of Firmware and Startup on Mac OSX Catalina or Big Sur*
3. *Installation of Firmware and Startup on Windows*
4. After the Assembly and Firmware installation steps above, the Edukit system will start automatically.
5. Please Appendix A: Edukit Quick Start Guide for a convenient Serial Interface for access and control of the Edukit
6. **Real Time Control System Workbench**:
    a. The Real Time Control System Workbench displays real time data within the control loop. The user can configure and adjust the control system in real time and observe system response. This system and its usage is described in this manual.
    b. This is available in Matlab and Octave versions
7. Control System Design Scripts guiding:
    a. Dual PID
    b. Full State Control Systems
8. Frequent updates with new tools and tutorials

2. Introduction

Control systems engineering increases in importance ever more rapidly with the appearance of self-driving vehicles, novel aerial vehicles, medical robotics, and countless consumer products. These applications require not only engineers with hands-on experience in control systems, but also very importantly, in the management and control of electric motor actuators and precise position sensors.

Digital control has evolved to advance nearly every electronic system. In contrast, *direct digital electric Motor Controllers* has most recently appeared. Integrated system on chip power electronics now provides accurate electric motor actuation control with combined sensing of motor shaft angle. This presents critical new capabilities and requirements for control systems. Training and instruction in direct digital Motor Controllers is essential for engineers today with the appearance of these systems in every form of mechatronics and robotics.

Control systems engineering design skill relies on understanding of both theory and practice. This is important to both experienced, professional engineers addressing a new control problem or students learning control systems engineering for the first time. Engineering education for fields, including electronics and computer engineering have taken advantage of hands-on instruction for all students for many decades. However, understanding of control systems practice, requiring hands-on-experience, has not been available for control systems students due to the cost of platforms.

The Edukit Rotary Inverted Pendulum has been designed to provide every engineer learning control systems with a flexible, low cost, *personal* design platform. A diverse and fascinating set of design challenges are available. These include the unique characteristics of state-of-the art motor control systems. This introduces the concepts of motor control and power electronics with classic and important control problems.

The Edukit Rotary Inverted Pendulum has been developed to provide engineers with experience in both Control Systems as well as state-of-the-art Motor Control interfaces. This includes opportunities for both development of stable control, with a suspended pendulum mode, and the classic unstable control problem with the inverted pendulum. This also includes a state-of-the-art high speed, precise, and configurable electric Motor Controller integrated with a processor hosting user composed controller systems.

An important feature of the Edukit Rotary Inverted Pendulum system is the opportunity to demonstrate the limitations of classic control and introduce *modern control.* Thus, this system includes stable and unstable plants as well as conventional PID control, parallel PID control structures, and LQR control based on design optimization.

Recent advances in state-of-the-art motor control systems by STMicroelectronics now enables the development of powerful control system kits that are sufficiently low in cost to be owned by every student. Further, it is possible for an individual to purchase and use this system for a cost less than a textbook. It can further be used in multiple courses from introductory level through senior and graduate level.

Hands-on experience with the Edukit Rotary Inverted Pendulum can be gained with many paths:

1) **Introduction to Control and Motor Control:** An introduction to control systems can be gained by students and users who access the Edukit Rotary Inverted Pendulum from any computer over a serial interface to adjust a real-time controller, configure control architectures, acquire data and both experience control as well as design advanced control methods.
2) **Matlab, Terminal Command Line Access, or C-Code Real Time Development:** Real time controllers are included along with demonstration systems. This also includes methods for configuration by Matlab or Terminal session command line interfaces.
3) **Open Source C-Code Real Time Development:** The real time C-Code control systems may be examined, modified, or replaced by users, offering a very wide range of instruction and research opportunities in advanced control systems.
4) **Matlab Real Time Viewer:** Live data is displayed via a Matlab "scope" interface of all physical system and control system parameters and signals.
5) **Example Instruction Sequences:** The Edukit Rotary Inverted Pendulum control systems can also be used in the Control Systems course curriculum with a sequence of projects focusing on PID control of stable systems, PID control of unstable systems, and State Space LQR control of unstable systems. This includes reference simulation and design methods based on Mathworks Matlab.
6) **Example Advanced Instruction Sequences:** The Edukit Rotary Inverted Pendulum control systems can also be used in advanced Control Systems course, mechatronics courses, and embedded systems courses where users have direct access to the real-time C-code implementation of the control system and the ability to view the control system template, modify and replace any or all components. This is supported by the STM32 system and open source Integrated Development Environment (IDE) tools compatible with Windows and Apple OS X operating system platforms.
7) **Individualization for Each Student:** An important characteristic of the Edukit Rotary Inverted Pendulum is its adaptability to enable *individualization for each student and for each assignment. This ensures that individual students or selected student groups may operate independently with assurance of independent results when assessing assignment results. This is described further in Appendix C.*
8) **Shared and Open Environment:** *First, our objective is to continuously adapt and advance the Edukit Rotary Inverted Pendulum to address the needs of students and instructors. All requests for new features, new analyses, and new instructional approaches will be pursued. Also, all components are open and available for any innovations at any level by all users.*

   This description, intended for Instructors who wish to learn about the Integrated Rotary Inverted Pendulum, describes system configuration, system dynamics, control system design examples, and experimental measurements.

Overview of This Manual

1) *Introduction*
    a. This describes the motivation for the Integrated Rotary Inverted Pendulum, the introduction of direct digital actuators and the difference between this platform and prior systems.
2) *Edukit Rotary Inverted Pendulum System Overview*
    a. This describes the system physical architecture
3) *Edukit Rotary Inverted Pendulum System Characteristics*
    a. This describes the system critical dimensions and motor control parameters
4) *Integrated Rotary Inverted Pendulum: Rotor System Model*
    a. This describes the second order model developed for the response of the Motor Controller
5) *Suspended Pendulum Dynamic Response*
    a. This describes the dynamic response characteristic of the Pendulum when operating in a suspended, and stable mode.
    b. This permits an introduction to control system development with an inherently stable plant.
6) *Suspended Pendulum Single PID Controller*
    a. This describes the design and implementation of a Single Input Single Output (SISO) PID controller stabilizing Pendulum Angle
7) *Suspended Pendulum Dual PID Controller*
    a. This describes the design and implementation of dual, parallel, controllers stabilizing Pendulum Angle and Rotor Angle.
    b. This approach of dual PID controllers enables a method for implementing stable control. However, a strategy for optimal control is not available with this method.
8) *Inverted Pendulum Single PID Controller*
    a. This describes the design and implementation of a Single Input Single Output (SISO) PID controller stabilizing Pendulum Angle for the Inverted Pendulum operating mode.
    b. The Inverted Pendulum operating presents an inherently unstable control challenge ideal for gaining experience in modern control.
9) *Suspended Pendulum Dual PID Controller*
    a. This describes the design and implementation of dual, parallel, controllers stabilizing Pendulum Angle and Rotor Angle for the Inverted Pendulum operating mode.
    b. This approach of dual PID controllers enables a method for implementing stable control. However, a strategy for optimal control is not available with this method.
10) *State Space Modern Control: The Linear Quadratic Regulator for Inverted Pendulum Operation*
    a. This describes the design and implementation of a Multiple Input Multiple Output (MIMO) State Space Linear Quadratic Regulator (LQR) control system for the Inverted Pendulum operating mode.
    b. This provides an introduction and experience in Modern Controls and optimal control methods.
11) *State Space Modern Control: The Linear Quadratic Regulator for Suspended Pendulum Operation*
    a. This describes the design and implementation of a Linear Quadratic Regulator (LQR) control system for the Suspended Pendulum operating mode.
    b. This provides additional experience in optimal control methods and the limitations of classic control for MIMO systems.

3. Edukit Rotary Inverted Pendulum System Overview

The Edukit Rotary Inverted Pendulum System includes an Inverted Rotary Pendulum [Furuta_1992] structure with a rotor system based on a Stepper Motor, Motor Control System based on the STMicroelectronics L6474 Motor Controller [ST_L6474_2019] and IHM01A1 [ST_IHM01A1_2019] support board, and finally a Nucleo F401RE board supporting the STM32 processor[ST_Nucleo_2019]. Figure 1 shows the system architecture with the following definitions for dimensions and angle values in Table 1 and Table 2. Figure 2 through Figure 4 show images of the system.

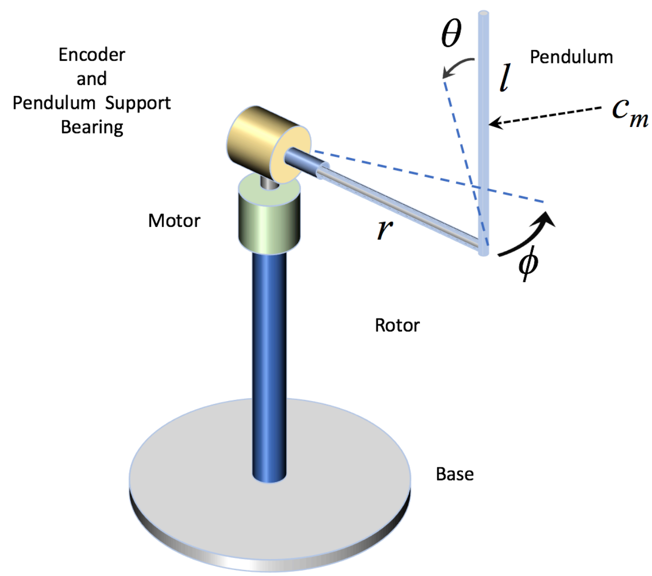


Figure 1. The Edukit Rotary Inverted Pendulum System shown with pendulum inverted. This shows its components including Motor providing rotation actuation of the rotor, Encoder providing sensing of rotor axial rotation angle, and the pendulum upright component with its supported mass. The definitions of dimensions and rotation angles defining pendulum position are also shown.

System Structure Characteristic	*Definition*
l	Pendulum Vertical length from Rotor Arm to supported Mass
r	Rotor Arm length
ϕ	Angle of Rotor relative to start reference angle
θ	Angle of pendulum relative to gravity vertical

Table 1. The Edukit Rotary Inverted Pendulum System Structure Characteristics

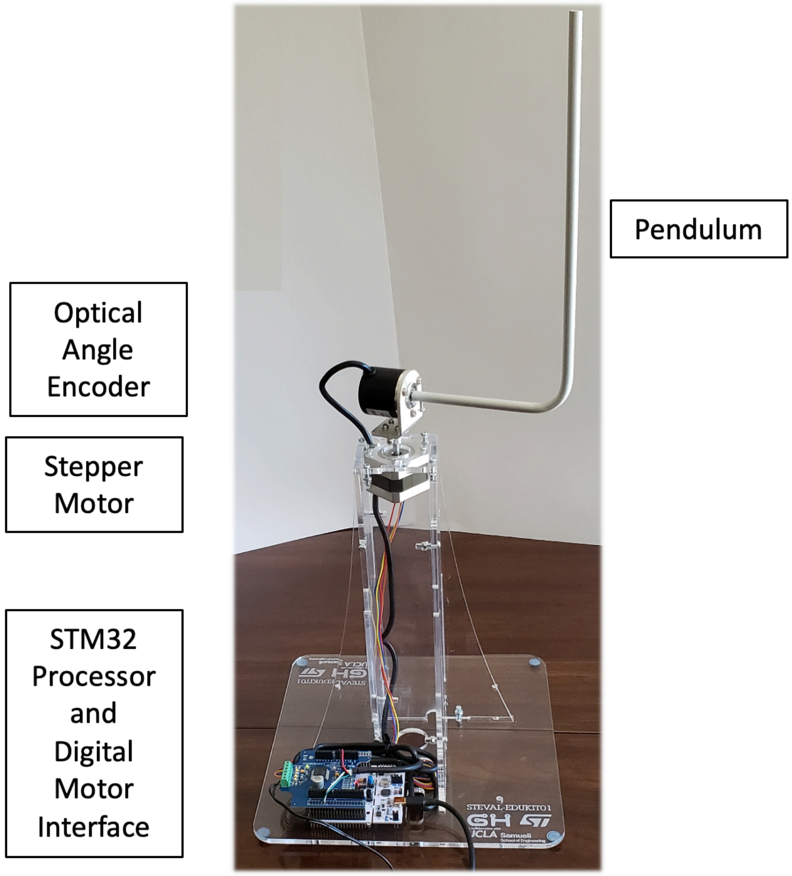


Figure 2. Edukit Rotary Inverted Pendulum System view during operation with active control and pendulum stabilized in vertical orientation

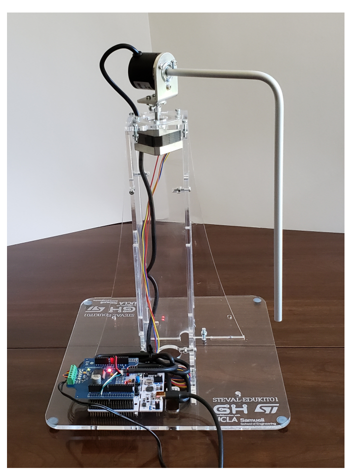


Figure 3. Edukit Rotary Inverted Pendulum System view during operation with active control and pendulum stabilized in suspended orientation

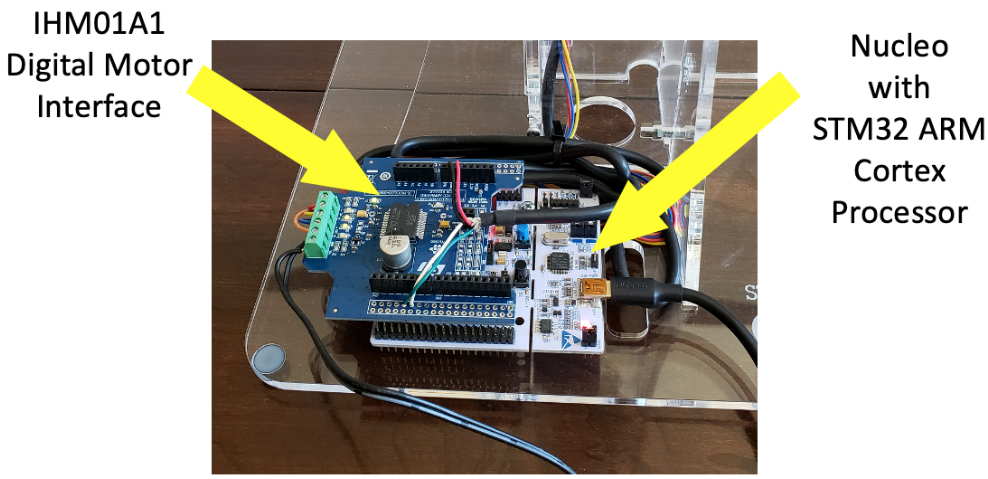


Figure 4. IHM01A1 Motor Controller and the Edukit Rotary Inverted Pendulum STM32 ARM Cortex system processor.

4. Edukit Rotary Inverted Pendulum System

The Edukit Rotary Inverted Pendulum System includes the STM32 Processor (integrated on the Nucleo F401RE PCB) and the L6474 Digital Motor Controller integrated circuit include on the IHM01A1 PCB itself mounted on the Nucleo PCB.

All sensing, computation, and control are performed on these components and are independent of any external computing and shown in *Figure 5*.

This approach affords complete visibility and control of the Edukit mechatronics real time system.

At the same time, the Edukit algorithms hosted on the STM32 Processor include methods for presenting real time data and accepting configuration commands from a remote Windows or Mac OSX system. This includes the Edukit Control Systems Real Time Workbench providing visualization and configuration of the complete control system along with its measurement capabilities that include the fundamental Sensitivity Functions as described in

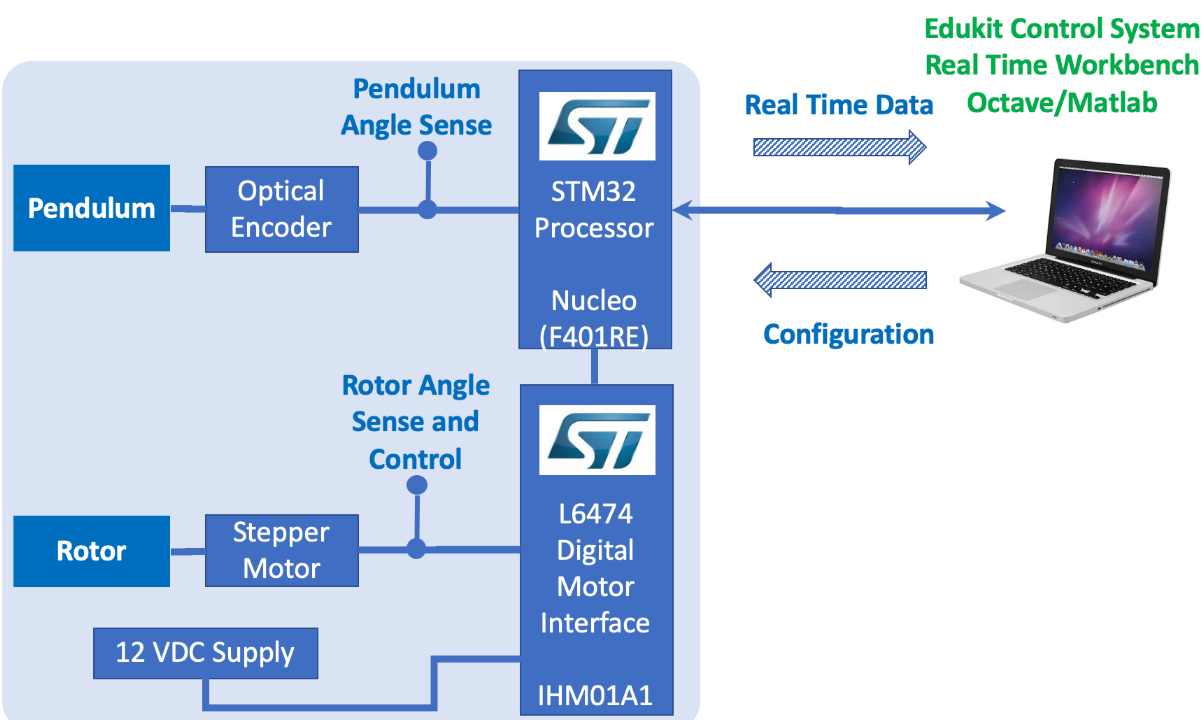


Figure 5. Edukit Rotary Inverted Pendulum System. System feedback control and sensing is performed in real time by the STM32 Processor with L5474 Digital Motor Controller. A Window or Mac OSX platform may be applied for acquisition and display of real time data.

Edukit Rotary Inverted Pendulum Components

Table 2 describes system characteristics of the Integrated Rotary Inverted Pendulum. The Rotor actuator is based on a NEMA-17 Stepper Motor controlled by the L6474 – IHM01A1. Each incremental step command to the motor controller produces an effective 1/16th microstep for a 200 step per rotation motor. This results in an angular displacement of 9.817 x 10^{-4} radians per step command or 0.0563 degrees per step and a Rotor Angle Control Gain of 17.778 step commands per degree. This gain is applicable for steady state, DC response. Motor control dynamics at high control frequency is described in Section 6.

This is configured to operate at an adjustable step rate M_{Speed}. The value of M_{Speed} are set to 200 steps per second and also to 100 steps per second in the examples in this review.

Each incremental change in stepper motor position is read with a factor of two greater gain for an effective 1/16th microstep for a 200 step per rotation. This results in angular displacement measurement of 1.963 x 10^{-3} radians per step measured or 0.1126 degrees per step and a Rotor Angle Measurement Gain of 8.889 steps measured per degree.

Pendulum Angle, θ, is measured with a quadrature optical encoder (LPD3806-600BM-G5-24C) providing a resolution of 2400 step count values per revolution equal to 6.667 step counts per degree.

System Characteristic	*System Characteristic Description*	*System Characteristic Value*
l	Pendulum Length measured as vertical separation from Rotor Arm to Pendulum Mass	0.235 m
r	Rotor Arm Length	0.14 m
ϕ	Angle of Rotor relative to initial angle reference.	
θ	Angle of Pendulum relative to gravity vector	
Rotor Angle Measurement Gain	Measurement gain of rotor angle provided by the motor and motor control interface	8.889 Steps per Degree
Rotor Angle Control Gain	Rotor angle actuation gain provided by the motor and motor control interface	17.778 Steps per Degree
Pendulum Angle Measurement Gain	Measurement gain of Pendulum Angle provided by the optical encoder and its interface	6.667 Steps per Degree

Table 2. The Edukit Rotary Inverted Pendulum System Characteristics

5. Edukit Rotary Inverted Pendulum Real Time Control

The Edukit control system is developed to enable a wide range of control architectures including signal and dual PID control, state-space modern control, and others. The control system showing dual PID control of Pendulum Angle and Rotor Angle is shown in Figure 6.

Two reference tracking signalsand , $\theta_{\mathrm{Pend-Command}}$ and $\phi_{\mathrm{Rotor-Command}}$ are provided to Pendulum and Rotor control, respectively. Two configurable control algorithms operate on the error signal between reference tracking signals and Pendulum Angle , θ_{Pend} and Rotor Angle, $\phi_{\mathrm{Rotor-Command}}$, respectively. The controller outputs are then summed and applied to the input of the Rotor Plant. The Rotor Plant, G_{Rotor} determines response of Rotor Angle to the control input, of $\phi_{\mathrm{Rotor-Control}}$. Finally, the Rotor Angle is an input to the Pendulum transfer function, G_{Pend} that determines Pendulum Angle.

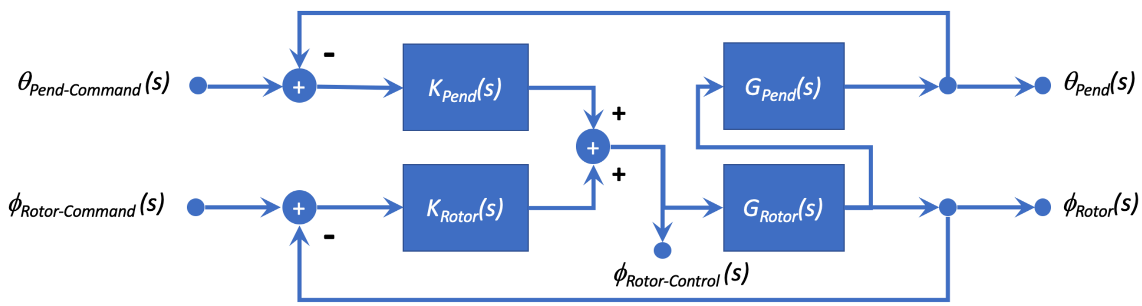


Figure 6. The Edukit Real Time Control System is shown. Two reference tracking signalsand , $\theta_{Pend-Command}$ and $\phi_{Rotor-Command}$ are provided to Pendulum and Rotor control, respectively. Two controllers operate on errors between reference tracking signals and Pendulum Angle, , θ_{Pend}, and Rotor Angle, $\phi_{Rotor-Command}$, respectively. The controller outputs are summed. The Rotor Plant, G_{Rotor}, determines response of Rotor Angle to the control system sum of $\phi_{Rotor-Control}$. The Rotor Angle is an input to the Pendulum transfer function, G_{Pend} that determines Pendulum Angle.

6. Edukit Rotor System Plant

Motor Control Overview

Motor control systems appearing in the previous generation of actuators and robotic systems relied on simplified control and measurement. For example, analog methods for control of motor current resulted in control of motor torque only. Measurement of motor shaft angle relied on separate angle encoder systems that were often limited in resolution.

State of the art motor control systems are based on integrated systems combining high speed current switching devices and circuits with current sensing to provide precise motor control. This enables the replacement of torque control of the motor system with direct control of motor operation schedule.

An example is an integrated Stepper Motor control system that provides control of motor speed, acceleration, and deceleration. In addition, an integrated Stepper Motor controller includes direct detection of motor position.

Also, most importantly, the Stepper Motor enables a precise control of motion at a fixed speed, M_{Speed}, rather than a control of torque with a resulting response dependent on instantaneous system state and associated inertial and gravitational forces. The Stepper Motor control system initiates motion of the Stepper Motor rotor towards a target angle at the speed, M_{Speed}. Motion continues at M_{Speed} until the target is reached or the motor control system receives a new target value.

Previous inverted pendulum research and available inverted pendulum systems were based on analog motor control. The Edukit Rotary Inverted Pendulum includes a Stepper Motor Controller based on the STMicroelectronics L6474 integrated Motor Controller. This is supported by open source software systems developed for the Integrated Rotary Inverted Pendulum.

The powerful motor control systems available require operation models as was required for simple motor control systems in the past.

Multiple motor configurations and corresponding models have been developed. This offers an ideal opportunity for instruction since instructors and students may now select a motor configuration and its dynamic response and implement new control systems adapted to each motor characteristic.

A new digital motor control method has been developed for Edukit to provide high performance in motor acceleration, motor velocity, and motor position control. In contrast to conventional systems that have been limited to operation at fixed motor step rate, the Edukit system provides rapid response in step rate from zero to over 10,000 microsteps per second. This is described in Appendix E: Rotor Actuator Plant.

Motor Controller Response and Rotor Plant Transfer Function

The Rotor Control Plant, G_{Rotor}, accepts a control input, of $\phi_{\mathrm{Rotor-Control}}$, and yields output angle of ϕ_{Rotor} , as shown in Figure 7.

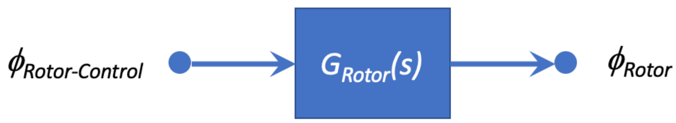


Figure 7. Rotor Control Plant with input control signal of $\phi_{Rotor-Control}$, and output angle, of ϕ_{Rotor}.

During each cycle of control system operation for the Edukit Rotary Inverted Pendulum, new target signals are supplied to the Rotor Plant. The Rotor Control system supported by the Stepper Motor Controller is described in detail in Appendix E: Rotor Actuator Plant.

The Rotor Control, stepper motor control system, accepts a control input setting *acceleration* of the Rotor. Thus, *position* of the Rotor is the double-integral with respect to time of the control input.

The G_{Rotor} plant transfer function was determined directly by system identification. As described in Appendix F: Rotor Actuator Plant Transfer Function.

Rotor Angle Response Transfer Function, G_{Rotor}, is

$$G_{Rotor} = \frac{\phi_{Rotor}}{\phi_{Rotor-Control}} = \frac{1}{s^2} \tag{1}$$

Accuracy and validation of this model is described in Appendix F: Rotor Actuator Plant Transfer Function.

7. Suspended Pendulum Dynamic Response

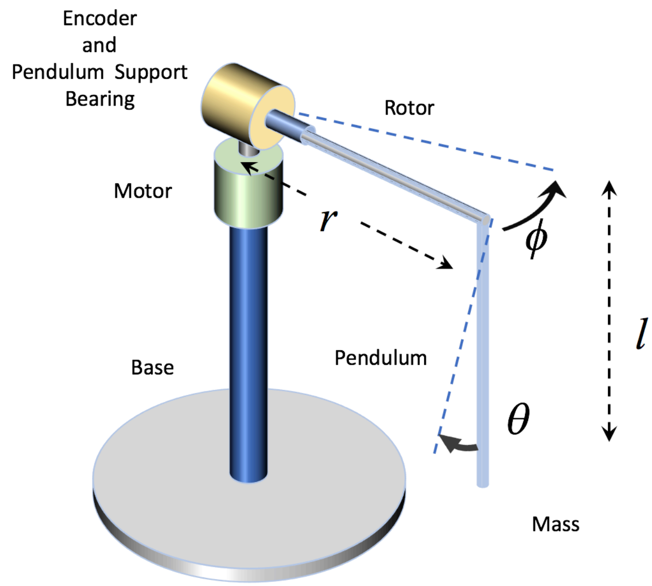


Figure 8. The Edukit Rotary Inverted Pendulum System shown with pendulum suspended. The definitions of dimensions and rotation angles defining pendulum position are also shown. This system is configured with a uniform pendulum rod of mass, m.

The suspended pendulum configuration is shown in Figure 8. This is a stable system where Pendulum angle, θ remains zero in the absence of a control input of rotor angle, ϕ.

The response of the suspended pendulum is determined by rotor angle, ϕ, gravitational acceleration, and friction force. The derivation of the transfer function between $\phi(s)$ and $\theta(s)$ is provided in Appendix D: Integrated Rotary Inverted Pendulum: Pendulum Dynamics. This includes description of System Identification applied to determine Q. The $G_{pend}(s)$ transfer function for the suspended pendulum is:

$$G_{pend}(s) = \frac{\theta(s)}{\phi(s)} = \frac{{\omega_r}^2 s^2}{s^2 + {}^{\omega_g}/_{Q}\, s + {\omega_g}^2}$$

With

$${\omega_r}^2 = \frac{r}{L\left(\frac{2}{3} + \frac{2I_{rotor}}{mL^2}\right)} = \frac{r}{L(0.6805)} = 0.803(rad/sec)^2$$

$${\omega_g}^2 = \frac{g}{L\left(\frac{2}{3} + \frac{2I_{rotor}}{mL^2}\right)} = \frac{g}{L(0.6805)} = 55.85(rad/sec)^2$$

$$Q = 20$$

8. Suspended Pendulum Single PID Controller: Frequency Response Design

This section describes the development of a Single Input Single Output (SISO) PID Controller controlling only Pendulum Angle. This is followed by a Section that describes a Dual PID Controller architecture that combines two SISO systems. Both systems will be designed using Frequency Response methods.

This design and development provides valuable background in PID control and its limitations associated with design guidance. Later Sections introduce Multiple Input Multiple Output (MIMO) Control based on the Linear Quadratic Regulator. This permits development of optimal control for parameters including a specified cost function.

The Suspended Mode feedback control system supplies an output to Rotor Control, ϕ, to control the difference between Pendulum angle, θ, and a reference tracking command, $\theta_{reference}$, as shown in *Figure 9*.

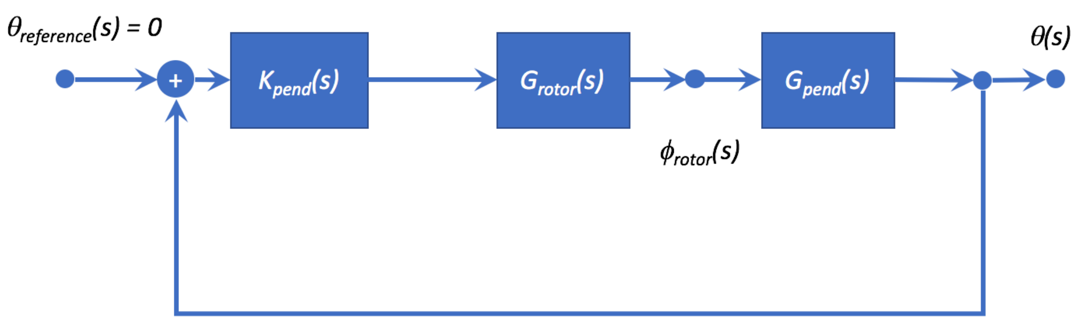


Figure 9. Pendulum Angle stabilization control system for Suspended Mode Pendulum operation.

Examining, *Figure 9*, the transfer function from $\theta_{reference}$ to θ is

$$\frac{\theta(s)}{\theta_{reference}(s)} = \frac{G_{rotor}(s)G_{pendulum}(s)K_{pendulum}(s)}{1 + G_{rotor}(s)G_{pendulum}(s)K_{pendulum}(s)}$$

As described in Section 6, development of the PID control system first requires development of a model for the Stepper Motor system providing Rotor control of Rotor Angle, ϕ, in response to Rotor Angle Control input, $\phi_{Rotor-Control}$.

Thus, the Rotor response transfer function of *Figure 9* is defined as in Section 6, Edukit Rotor System Plant,

$$G_{rotor}(s) = \frac{\phi(s)}{\phi_{Rotor-Control}(s)} = \frac{1}{s^2}$$

First, the stability of the plant including the Pendulum and Rotor response transfer functions is evaluated. The response of the Pendulum, G_{pend}, is shown in the Bode plot of *Figure 10*. The resonance in its response at $\omega_g = \sqrt{55.85(rad/sec)^2} = 7.47\ rad/sec$ is observed.

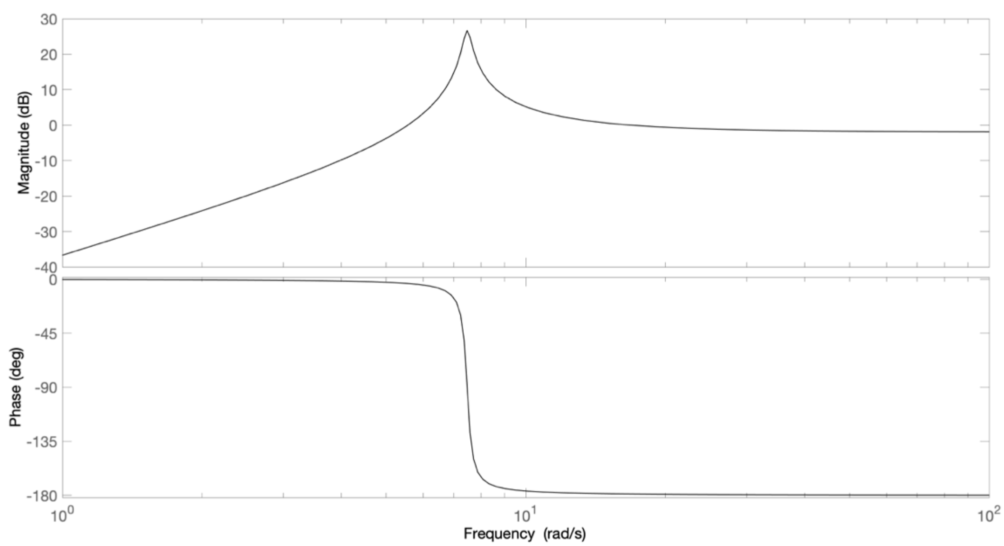


Figure 10. The Bode plot of the Pendulum transfer function between Pendulum Angle and Rotor Angle is shown. The resonance appears at $\omega_g = 7.47\ rad/sec$.

The product of the transfer functions, $G_{pendulum}$ and G_{rotor}, forms the control system plant. The Bode plot for the open loop plant, $G_{pendulum}G_{rotor}$, is shown in *Figure 11*.

The step response for this plant for an input of Rotor Angle, and an output of Pendulum Angle is shown in *Figure 12*. The underdamped response, is clear.

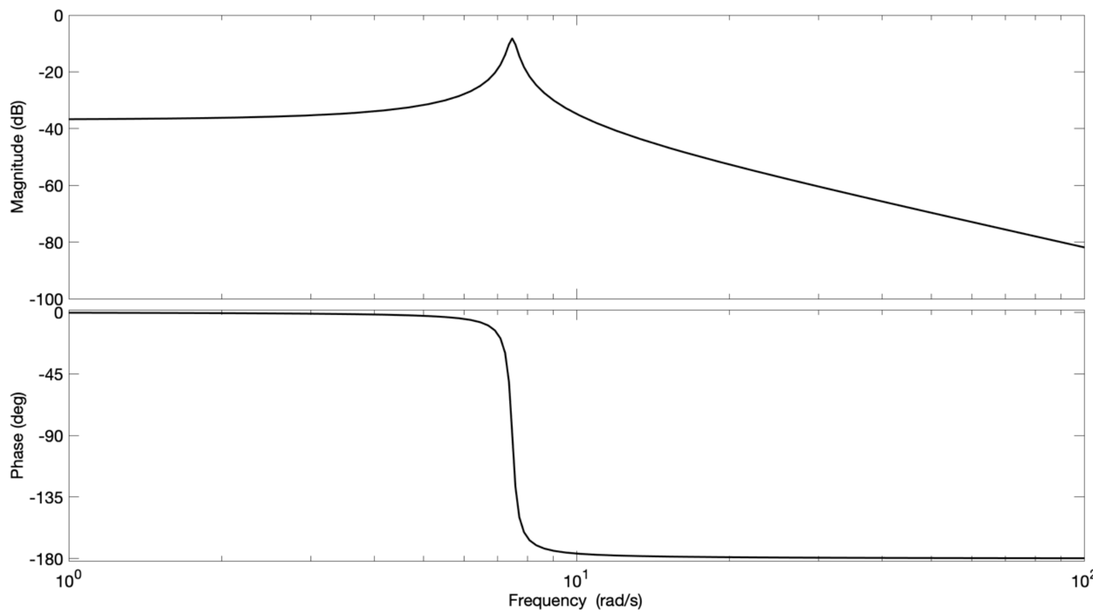


Figure 11. The Bode plot of the open loop Pendulum and Rotor Plant is shown. The resonance appears at $\omega_g = 7.47\ rad/sec$.

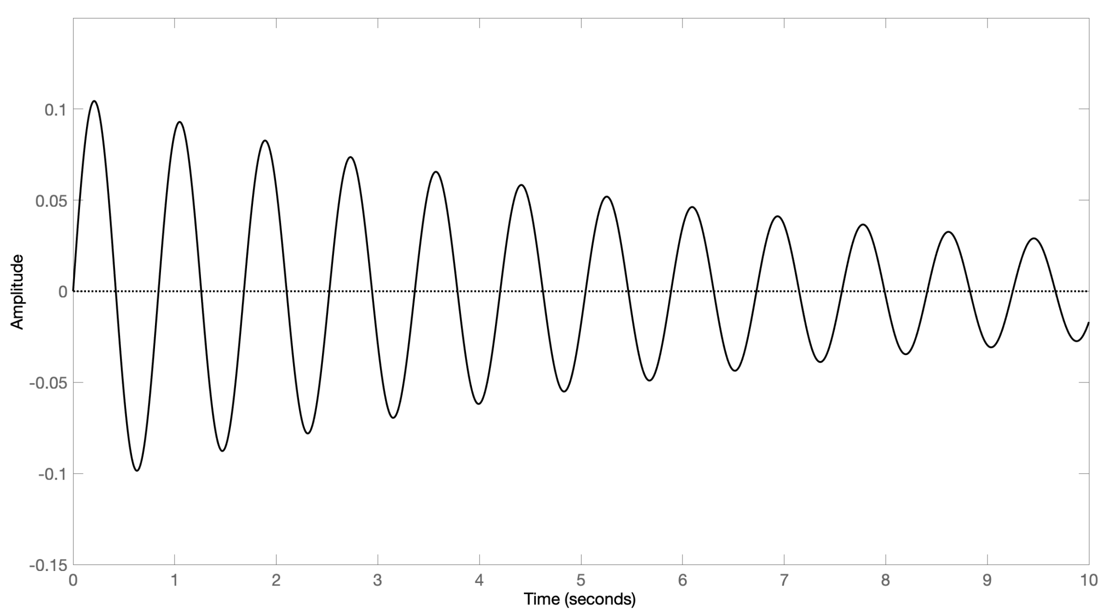


Figure 12. The step response of the open loop plant corresponding to the product of Pendulum Angle and Rotor Angle transfer functions is shown. The resonance, "ringing" behavior appears at $\omega_g = 7.47\ rad/sec$.

It is clear that proportional and derivative control will be required to provide properly damped control of pendulum angle. Integral control is also added to ensure accurate settling of pendulum angle at DC.

The design of a PID controller proceeds first with selection of Integral Frequency and Derivative Frequency values. Then, system gain will be determined by Frequency Response design methods subject to stability and performance constraints.

The poles appearing for the open loop plant are

0.0000 + 0.0000i
0.0000 + 0.0000i
-0.1384 + 7.4720i
-0.1384 - 7.4720i

Also two zeroes appear for the open loop plant both at the origin.

Design of a SISO PID controller proceeds using frequency response methods. The design objectives for this example will include:

1) Verification of stability by examination of both Bode and Nyquist characteristics.

2) Phase margin is greater than 30 degrees [Seborg_1989]

3) Selection of Design Based on Minimizing Maximum Values of Magnitude of Sensitivity Functions and selecting for minimum Settline Time.

Stable and robust control design can be obtained by ensuring that the maximum values of Sensitivity and Complimentary Sensitivity Functions are less than a threshold. [Hast_2013] [Yaniv_2004]

The maximum values of Sensitivity and Complimentary Sensitivity Functions are computed for each design iteration. The design selection will occur with parameters that minimize these maximum values as will be discussed.

The first step in design is directed to development of a PD controller with derivative time, T_d, and proportional gain, K.

The form of a PD controller is,

$$K_{pend}(s) = K(1 + T_d s)$$

The process for selection of K, T_i. and T_d will include the following steps with election of initial estimates for K, T_i. and T_d.

Iterative adjustment of K, T_i. and T_d values will then continue to obtain system performance meeting design characteristics. At each iteration, the following considerations will apply:

1) Verification of Stability by Nyquist analysis
2) Verification of Gain and Phase Margin by Bode Stability analysis

Iteration will continue with the objectives to select K, T_i. and T_d to

1) Minimize impulse response settling time system
2) Minimize the maximum values of the Sensitivity Functions to select a design meeting specified objectives.

An illustration of this process begins with a selection of $T_d = 0.5s$. This will be adjusted to explore the influence of T_d values on performance. Also the gain, K, values will be adjusted. It is important to note that this is an illustration of one approach and is not intended to imply that this is an optimal solution for a specified performance objective.

The Bode plot of the Pendulum and Rotor plant $G_{rotor}G_{pend}$, Plant and Controller, $G_{rotor}G_{pend}K_{pend}$, and the open loop controller, K_{pend} , are shown for gain, *K = 1* and T_d = 0.5 sec in *Figure 13*.

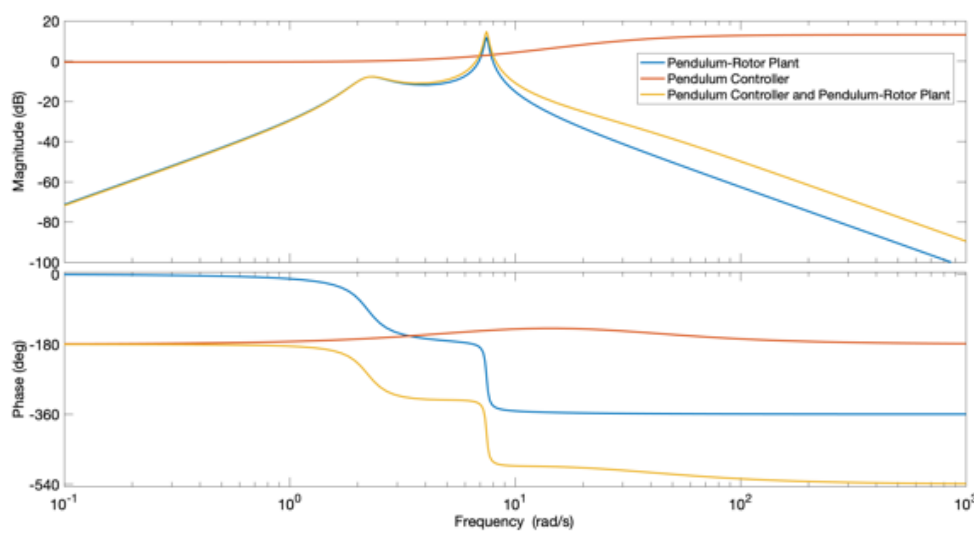


Figure 13. The Bode plot of the Pendulum and Rotor plant $G_{rotor}G_{pend}$, Plant and Controller, $G_{rotor}G_{pend}K_{pend}$, and the open loop controller, K_{pend} , are shown for gain, K = 1 and T_d = 0.5 sec.

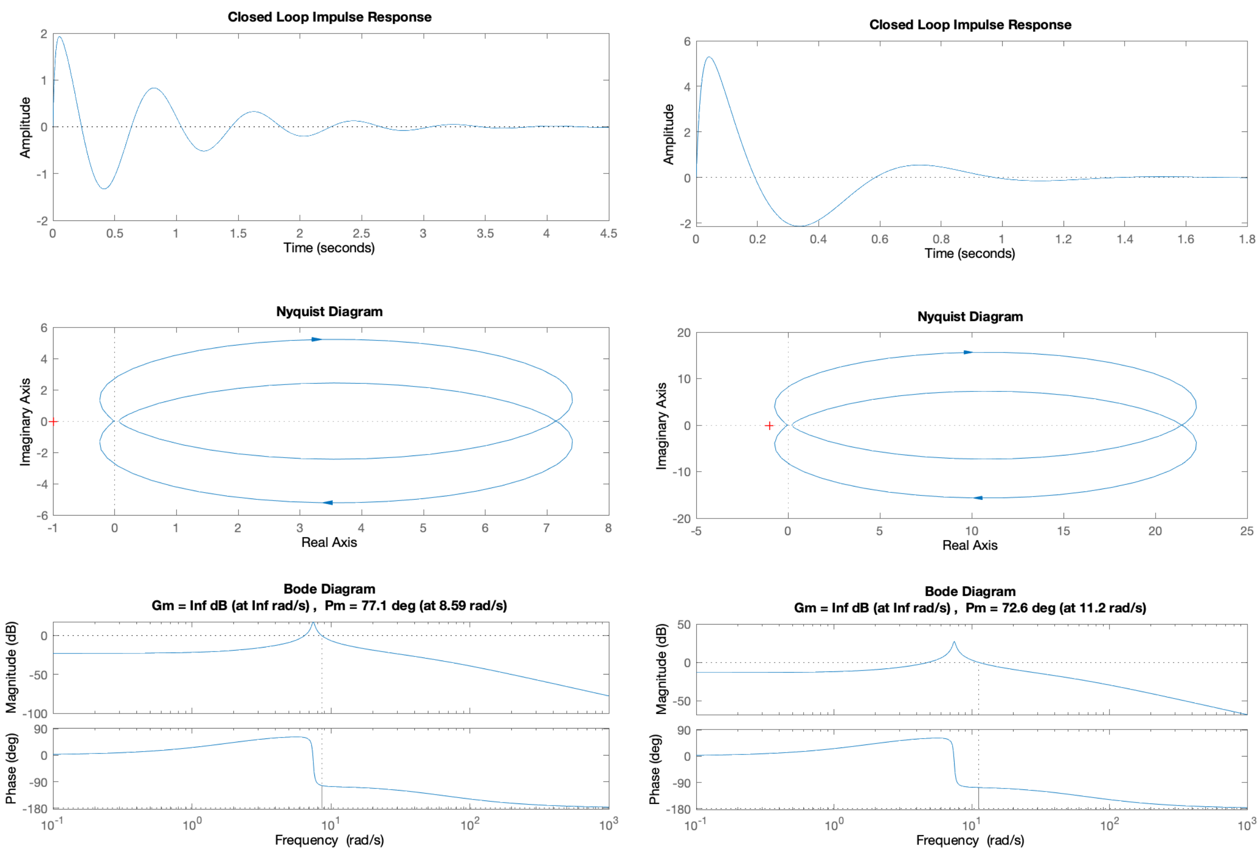


Figure 14. The Impulse Response, Nyquist Diagram, and Bode Margin plot of $G_{rotor}G_{pend}K_{pend}$ *are shown for K = 5 and* T_d *= 0.5 sec (left) and K = 15 and* T_d *= 0.5 sec (right).*

The Impulse Response, Nyquist Characteristics, and Bode Margin Characteristics are shown for *K = 5*, *K = 15* and *K = 10* in *Figure 14* and *Figure 15*, respectively. These characteristics show variation in Impulse Response Settling Time (to 2 percent of the peak value of the Impulse Response). However, each *K* value results in stablility as evaluated by the Nyquist Characteristics, and Bode Margin Characteristics.

For example, for *K = 10* the Nyquist Criterion may be tested for verification of stability. First, the poles of the closed loop system, zeroes of $1 + G_{rotor}G_{pend}K_{pend}(s)$ are

0.0000 + 0.0000i
0.0000 + 0.0000i
-58.5848 + 0.0000i
-2.2619 + 7.9621i
-2.2619 - 7.9621i

Then, the poles of $G_{rotor}G_{pend}K_{pend}(s)$ are

0.0000 + 0.0000i
0.0000 + 0.0000i
-62.8319 + 0.0000i
-0.1384 + 7.4720i
-0.1384 - 7.4720i

The number of zeroes of $1 + G_{rotor}G_{pend}K_{pend}(s)$ in the right half plane, Z = 0. Also, the number of poles, of $G_{rotor}G_{pend}K_{pend}(s)$ in the right half plane, P = 0. Then, the number of clockwise encirclements of the point, s = -1, is zero, as observed.

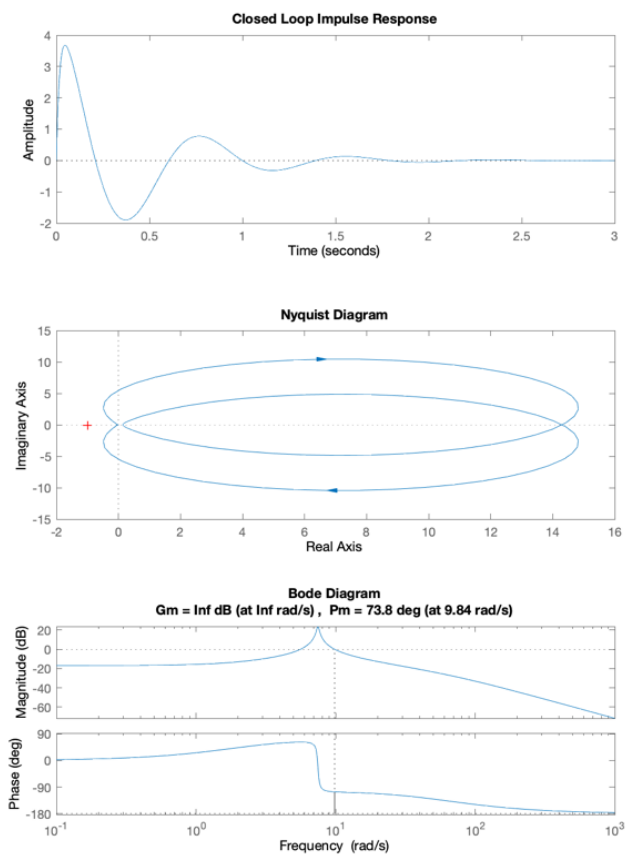


Figure 15. The Impulse Response, Nyquist Diagram, and Bode Margin plot of $G_{rotor}G_{pend}K_{pend}$ are shown for K = 10 and T_d = 0.5 sec.

The maximum values and frequency response of the Sensitivity Functions may be determined by measuring the real-time response of each Sensitivity Function transfer function to a step function input. First, the control system configuration enabling the measurement of Sensitivity Functions is shown in *Figure 16*.[Astrom_2019]

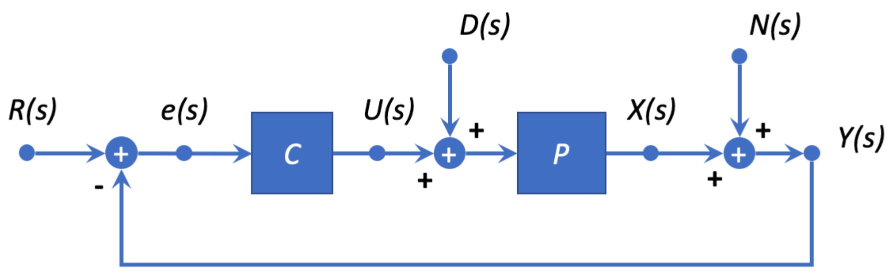


Figure 16.The Edukit control system architecture for either Pendulum or Rotor control shown with signal inputs for measurement of Sensitivity Functions. Sensitivity Functions are measured by applying Reference Tracking signals, $R(s)$, Load Disturbance signals, $D(s)$, and Noise signals, $N(s)$, Measured response is obtained at $e(s)$, $U(s)$, and $Y(s)$.

The Sensitivity Functions defined in terms of the input Reference Tracking signals, $R(s)$, Load Disturbance signals, $D(s)$, and Noise signals, $N(s)$, and measured response of $e(s)$, $U(s)$, and $Y(s)$ are:

a. Sensitivity Function

$$S(s) = \frac{e(s)}{R(s)} = \frac{1}{1 + T_{rotor}(s)K_{rotor}(s)}$$

b. Complimentary Sensitivity Function

$$T(s) = \frac{Y(s)}{R(s)} = \frac{T_{rotor}(s)K_{rotor}(s)}{1 + T_{rotor}(s)K_{rotor}(s)}$$

c. Load Disturbance Rejection Sensitivity Function

$$LD(s) = \frac{Y(s)}{D(s)} = \frac{T_{rotor}(s)}{1 + T_{rotor}(s)K_{rotor}(s)}$$

d. Noise Rejection Sensitivity Function

$$NS(s) = \frac{U(s)}{N(s)} = \frac{K_{rotor}(s)}{1 + T_{rotor}(s)K_{rotor}(s)}$$

The maximum value for the range of $s = 0$ to $s = \infty$ will be computed for each with

$$M_S = \max_{0 \le \omega < \infty} |S(j\omega)|$$
$$M_T = \max_{0 \le \omega < \infty} |T(j\omega)|$$
$$M_{LD} = \max_{0 \le \omega < \infty} |LD(j\omega)|$$
$$M_{NS} = \max_{0 \le \omega < \infty} |NS(j\omega)|$$

The Edukit Real Time Control System Workbench (described in Appendix B Section 19) includes methods for configuring the Edukit real time control system to introduce each of the $R(s)$, $D(s)$, and, $N(s)$ signals while measuring $e(s)$, $U(s)$, and $Y(s)$ in real time.

For each of the three gain values the system response characteristics have been computed and included in *Table 3*.

K	T_d (sec)	Phase Margin (degrees)	Gain Margin (dB)	2% Settling Time (secs)	M_s	M_T	M_{LD}	M_{NS}	$NS(j\omega)$ at $\omega = 1000$
5	0.5	77.06	Inf	3.41	1.05	0.94	0.046	162.1	161.8
10	0.5	73.76	Inf	1.85	1.07	1.00	0.024	324.2	323.6
15	0.5	72.6	Inf	1.11	1.09	1;02	0.016	486.2	485.5

Table 3. Pendulum Controller Gain Margin, Phase Margin, Settling Time, and Maximum Values of Sensitivity Functions for each gain, K, value and $T_d = 0.5$.

The value of *K = 10* is chosen for this example as the meeting the objectives for minimizing Sensitivity Function values as well as providing an intermediate settling time value.

Then, for this value of *K*, the influence of T_d selection is evaluated. The results are shown in *Table 4*.

K	T_d (sec)	Phase Margin (degrees)	Gain Margin (dB)	2% Settling Time (secs)	M_s	M_T	M_{LD}	M_{NS}	$NS(j\omega)$ at $\omega = 1000$
10	0.25	65.65	Inf	3.74	1.11	1.04	0.044	167.1	166.8
10	0.5	73.76	Inf	1.85	1.07	1.00	0.024	324.2	323.6
10	0.75	75.87	Inf	1.14	1.08	1;00	0.017	481.2	480.5

Table 4. Pendulum Controller Gain Margin, Phase Margin, Settling Time, and Maximum Values of Sensitivity Functions for each T_d *value and gain,* $K = 10$.

Finally, the value of *K = 10* and $T_d = 0.5$ is chosen for this example as the meeting the objectives for minimizing Sensitivity Function values as well as providing an intermediate settling time value. It is important to note that this is an example illustration of one method of controller design. Many other methods can now be used to meet specific objectives.

The value of $K = 10$ and T_d = 0.5 sec. yields these gain values:

$$K_P = 10.0, \quad K_I = 0.0, \quad K_D = KT_d = 5.0$$

The Edukit control system also includes an impulse generation method. This introduces a pulse of amplitude, 50 degrees for a duration of 2 milliseconds in the Pendulum Angle Reference Tracking signal, $\theta_{reference}$. The Pendulum Angle Response is shown in *Figure 17*.

Note that the discrete steps in the output of the digital rotary encoder are visible – each corresponds to 0.15 degrees of rotation.

An important limitation of this Single Input Single Output (SISO) controller appears. Specifically, while the controller stabilizes Pendulum Angle, Rotor Angle does not appear as an input and is not stabilized. Thus, a disturbance in Rotor position is not compensated for by this controller.

The limitations of SISO control may be characterized directly. The response of Rotor Angle, ϕ_{rotor}, to a signal applied to $\theta_{reference}$ may be computed. Examining *Figure 9*, the transfer function from $\theta_{reference}(s)$ to $\phi_{rotor}(s)$ is

$$T_{Rotor-Response} = \frac{\phi_{rotor}(s)}{\theta_{reference}(s)} = \frac{G_{rotor}K_{pend}}{1 + G_{pend}G_{rotor}K_{pend}}$$

Thus, an impulse in $\theta_{reference}(s)$ induces a response in Rotor Angle.

This SISO system does not provide control of Rotor Angle. Thus, Rotor Angle may drift as a result of the Pendulum Angle Reference impulse.

As an experimental example, the Pendulum position for the Edukit Rotary Inverted Pendulum was disturbed, leading to a drift in Rotor position shown in *Figure 18*.

The next section describes the addition of a second PID controller that provides Rotor Angle stabilization relative to a Rotor Angle Reference tracking command.

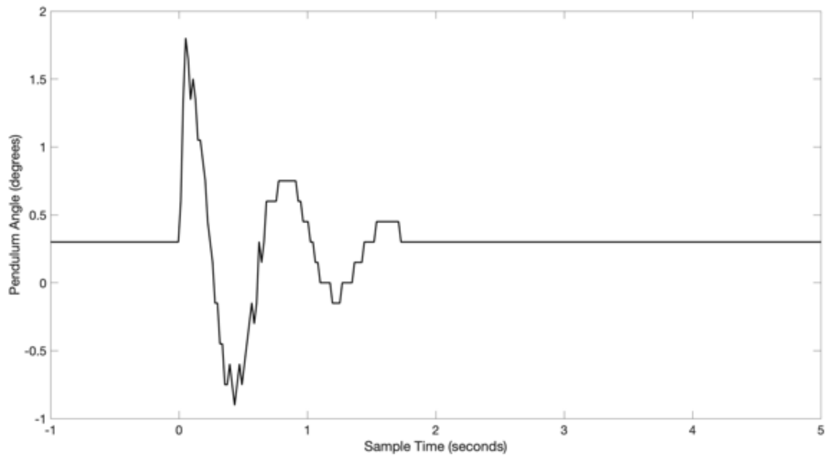


Figure 17. Impulse response of Pendulum angle for a reference impulse applied at t = 0.

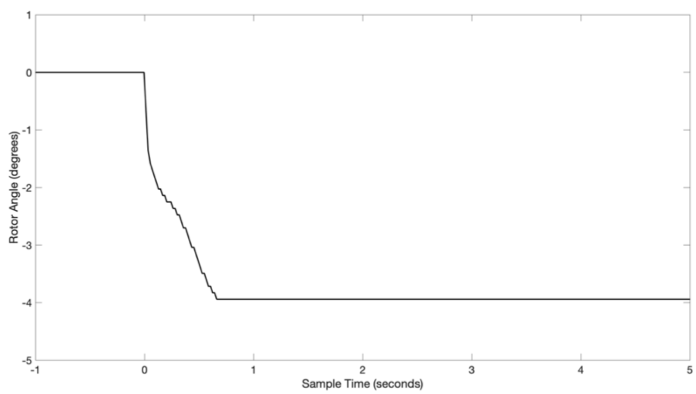


Figure 18. Response of Rotor Angle, ϕ_{rotor}, to an impulse signal applied to Pendulum angle reference, $\theta_{reference}$, at t = 0. Note the drift in Rotor Angle. During this period, Pendulum Angle remains constant and near zero.

9. Suspended Pendulum Dual PID Controller: Frequency Response Design

The previous Section 8, introduced a method for design of a Pendulum Angle Controller. While Pendulum Angle control was established, Rotor Angle was not constrained or provided with a control capability. Thus, Rotor Angleis subject to drift.

The introduction of a second PID controller will enable stabilization of Rotor Angle. This will be accomplished by introducing an Inner Loop and Outer Loop control system.

The design process will include the steps of:

1) Design of Single PID Controller, $K_{pend}(s)$
2) Definition of Inner Control Loop including $K_{pend}(s)$
3) Definition of Outer Control Loop integrating the Inner Control Loop and a new controller, $K_{rotor}(s)$
4) Design of the Rotor Control system, $K_{rotor}(s)$.

The Inner Loop controller will depend on the design of $K_{pend}(s)$ in Section 8. This design will exploit as designed $K_{pend}(s)$ while a new Outer Loop controller of Rotor Angle, $K_{rotor}(s)$, is added

The Inner Loop Controller defines a transfer function from a Rotor Angle Control signal, $\phi_{Rotor-Control}$ to the Rotor Angle, ϕ_{Rotor} as shown in *Figure 19*.

The Outer Loop Controller provides control of Rotor Angle, Rotor Angle, ϕ_{Rotor} to a Rotor Angle Tracking Command, $\phi_{Rotor-Command}$ as shown in *Figure 20*.

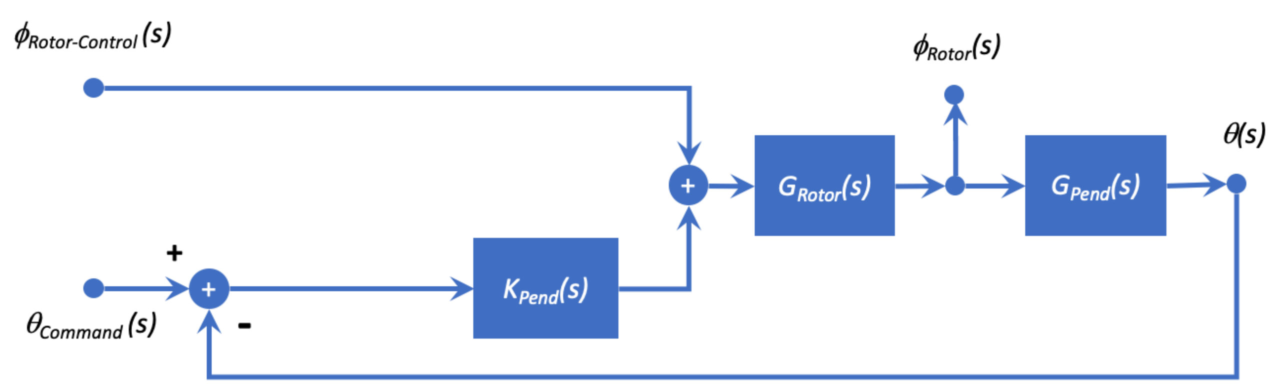


Figure 19. Dual PID Control System Inner Control Loop

The SISO Pendulum Angle control system architecture of *Figure 19* will be now modified to create an Inner Loop controller.

Now, the Inner Loop Rotor Angle Control transfer function, $T_{Rotor}(s)$ between $\phi_{Rotor-Control}$ and ϕ_{Rotor} will be determined. The Pendulum Angle reference input signal, $\theta_{command}(s)$, will be held zero.

$$\phi_{Rotor}(s) = \left(\phi_{Rotor-Control}(s) - \phi_{Rotor}(s)G_{pendulum}(s)K_{pendulum}\right) G_{Rotor}(s)$$

$$T_{rotor}(s) = \frac{\phi_{Rotor}(s)}{\phi_{Rotor-Control}(s)}$$

and

$$T_{rotor}(s) = \frac{G_{rotor}(s)}{1 + G_{pend}(s)K_{pend}(s)\, G_{rotor}(s)}$$

This Rotor Angle Control system now introduces the Outer Loop controller to enable control of Rotor Angle. Here, the difference between a Rotor Angle tracking command, $\phi_{Rotor-Command}(s)$ and Rotor Angle, $\phi_{Rotor}(s)$, is supplied as an error signal to a new controller, K_{rotor}, and summed with the output of the Pendulum Angle Controller, $K_{pendulum}$ as shown in *Figure 20*.

The control system block diagram can also be drawn with the Inner Loop represented by $T_{rotor}(s)$ as shown in *Figure 21*.

During subsequent design steps, the configuration of $K_{pend}(s)$ will remain fixed and as specified in the previous design steps.

The Outer Loop controls ϕ_{Rotor} with the goal of tracking the reference input, $\phi_{Rotor-Command}(s)$.

By definition

$$\phi_{rotor} = \phi_{Rotor-Control}(s)T_{rotor}(s)$$

Thus, the transfer function from $\phi_{Rotor-Command}$ to ϕ_{Rotor} is

$$T_{rotor-tracking}(s) = \frac{\phi_{rotor}(s)}{\phi_{Rotor-Command}(s)} = \frac{T_{rotor}(s)K_{rotor}(s)}{1 + T_{rotor}(s)K_{rotor}(s)}$$

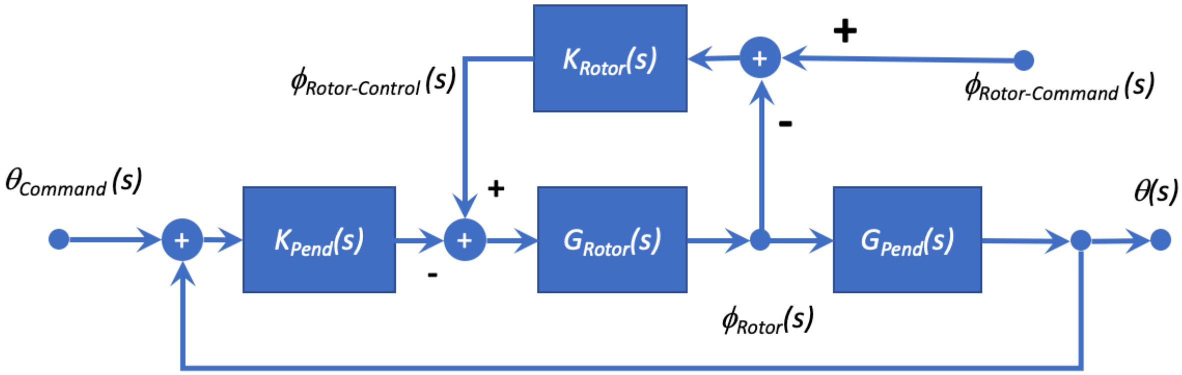


Figure 20. Dual PID Control System with Inner and Outer Control Loop

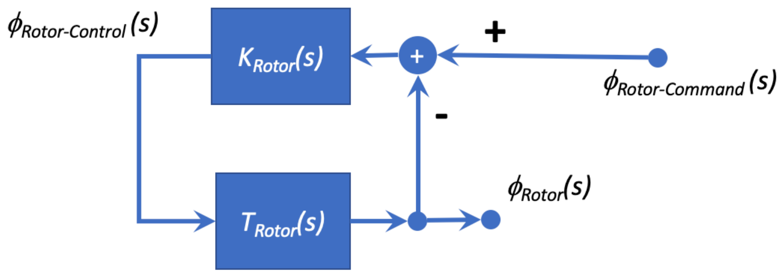


Figure 21. Dual PID Control System with Inner Control Loop represented by $T_{rotor}(s)$

Design of the Rotor PID controller proceeds using frequency response methods. The design objectives for this example will include:

1) Verification of stability by examination of both Bode and Nyquist characteristics.

2) Phase margin is greater than 30 degrees [Seborg_1989]

3) Selection of Design Based on Minimizing Maximum Values of Magnitude of Sensitivity Functions and selecting for minimum Settline Time.

Stable and robust control design can be obtained by ensuring that the maximum values of Sensitivity and Complimentary Sensitivity Functions are less than a threshold. [Hast_2013] [Yaniv_2004]

The maximum values of Sensitivity and Complimentary Sensitivity Functions are computed for each design iteration. The design selection will occur with parameters that minimize these maximum values as will be discussed.

The first step in design is directed to development of a PD controller with derivative time constant, T_{d-r}, and proportional gain, K.

The form of the PD controller is,

$$K_{rotor}(s) = K(1 + T_{d-r}s)$$

The process for selection of K and T_{d-r} will include the following steps with election of initial estimates for K and T_{d-r}.

Iterative adjustment of K, T_i. and T_d values will then continue to obtain system performance meeting design characteristics. At each iteration, the following considerations will apply:

1) Verification of Stability by Nyquist analysis
2) Verification of Gain and Phase Margin by Bode Stability analysis

Iteration will continue with the objectives to select $K$and T_{d-r} to

3) Minimize impulse response settling time system
4) Minimize the maximum values of the Sensitivity Functions to select a design meeting specified objectives.

An illustration of this process begins with a selection of $T_{d-r} = 1\ sec$. This will be adjusted to explore the influence of T_{d-r} values on performance. Also the gain, K, values will be adjusted. It is important to note that this is an illustration of one approach and is not intended to imply that this is an optimal solution for a specified performance objective.

The Bode plot of the Pendulum and Rotor plant $T_{Rotor}(s)$ Plant and Controller, $T_{Rotor}(s)K_{Rotor}(s)$, and the open loop controller, K_{Rotor} , are shown for gain, *K = 1* and T_d = 0.5 sec in *Figure 22*.

The Impulse Response, Nyquist Characteristics, and Bode Margin Characteristics are shown for *K = 1.5*, *K = 2.5* and *K = 2.0* in *Figure 23* and *Figure 24*, respectively. These characteristics show variation in Impulse Response Settling Time (to 2 percent of the peak value of the Impulse Response). However, each *K* value results in stablility as evaluated by the Nyquist Characteristics, and Bode Margin Characteristics.

For example, for *K = 2* the Nyquist Criterion may be tested for verification of stability. First, the poles of the closed loop system, zeroes of $1 + T_{Rotor}K_{Rotor}(s)$ are

-3.1165 + 0.0000i
-0.5833 + 0.0000i
-0.0244 + 0.0720i
-0.0244 - 0.0720i
-0.0121 + 0.0108i
-0.0121 - 0.0108i

Then, the poles of $T_{Rotor}(s)K_{Rotor}(s)$ are

0.0000 + 0.0000i
0.0000 + 0.0000i
-3.1416 + 0.0000i
-0.5858 + 0.0000i
-0.0226 + 0.0796i
-0.0226 - 0.0796i

The number of zeroes of $1 + T_{Rotor}(s)K_{Rotor}(s)$in the right half plane, Z = 0. Also, the number of poles, of $T_{Rotor}(s)K_{Rotor}(s)$ in the right half plane, P = 0. Then, the number of clockwise encirclements of the point, s = -1, is zero, as observed.

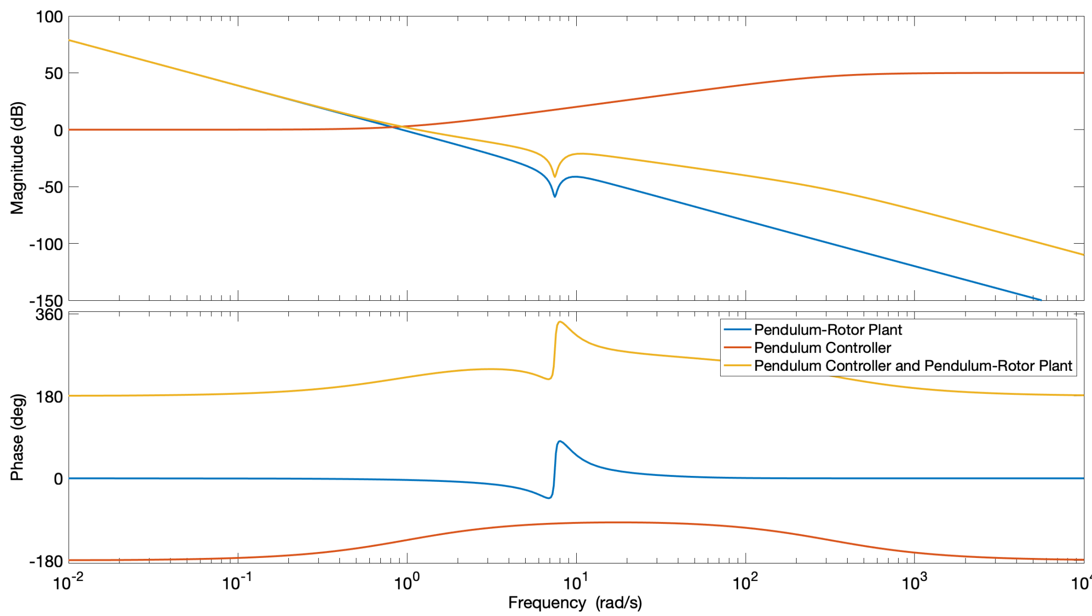


Figure 22.The Bode plot for the controller, K_{rotor}, the Inner Loop Plant, T_{rotor}, and for the open loop combined plant and controller, $K_{rotor}T_{rotor}$, are shown for K = 1 and T_d = 1 sec.

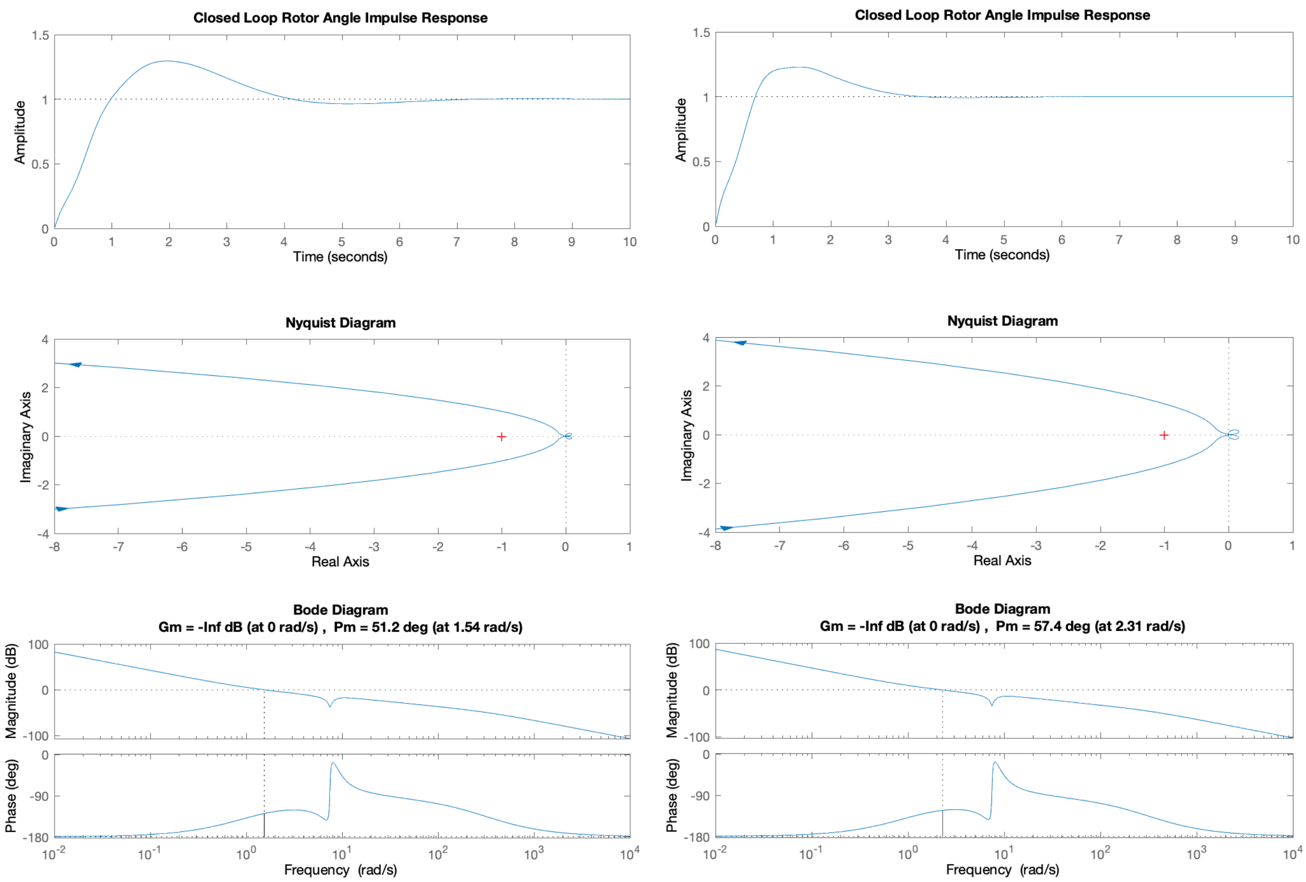


Figure 23. Closed Loop Impulse Response, Nyquist Characteristic and Bode margin plot for $K = 1.5$ *(left), for* $K = 2.5$ *(right) and* $T_{d-r} = 1\ sec.$ *Note that each characteristic indicates stability.*

As described in, Section 8, the Sensitivity Functions defined in terms of the input Reference Tracking signals, $R(s)$, Load Disturbance signals, $D(s)$, and Noise signals, $N(s)$, and measured response of $e(s)$, $U(s)$, and $Y(s)$ may be computed. Also, the maximum values and frequency response of the Sensitivity Functions may be determined by measuring the real-time response of each Sensitivity Function transfer function to a step function input.

For each of the three gain values the system response characteristics have been computed and included in *Table 5*.

K	T_d (sec)	Phase Margin (degrees)	Gain Margin (dB)	2% Settling Time (secs)	M_s	M_T	M_{LD}	M_{NS}	$NS(j\omega)$ at $\omega = 1000$
1.5	1.0	51.20	Inf	6.31	1.19	1.44	0.719	472.5	451.2
2.0	1.0	55.12	Inf	3.44	1.17	1.36	0.505	630.0	601.7
2.5	1.0	57.42	Inf	3.23	1.20	1.31	0.400	787.5	752.2

Table 5. Rotor Controller Gain Margin, Phase Margin, Settling Time, and Maximum Values of Sensitivity Functions for each gain, K, value and $T_d = 0.5$.

The value of *K = 2* is chosen for this example as the meeting the objectives for minimizing Sensitivity Function values as well as providing an intermediate settling time value.

Then, for this value of *K*, the influence of T_{d-r} selection is evaluated. The results are shown in *Table 6*.

K	T_d (sec)	Phase Margin (degrees)	Gain Margin (dB)	2% Settling Time (secs)	M_s	M_T	M_{LD}	M_{NS}	$NS(j\omega)$ at $\omega = 1000$
2.0	0.75	44.91	Inf	5.51	1.33	1.58	0.601	473.0	451.7
2.0	1.0	55.12	Inf	3.44	1.17	1.36	0.505	630.0	601.7
2.0	1.25	61.51	Inf	3.69	1.18	1.25	0.500	787.0	751.7

Table 6. Rotor Controller Gain Margin, Phase Margin, Settling Time, and Maximum Values of Sensitivity Functions for each T_{d-r} *value and gain,* $K = 10$.

Finally, the value of *K = 2* and $T_d = 1.0$ is chosen for this example design as the meeting the objectives for minimizing Sensitivity Function values as well as providing an intermediate settling time value. It is important to note that this is an example illustration of one method of controller design. Many other methods can now be used to meet specific objectives.

The value of *K = 2* and T_{d-r} = 1.0 sec. yields these gain values:

$$K_P = 2.0, \qquad K_I = 0.0, \qquad K_{d-r} = KT_{d-r} = 2.0$$

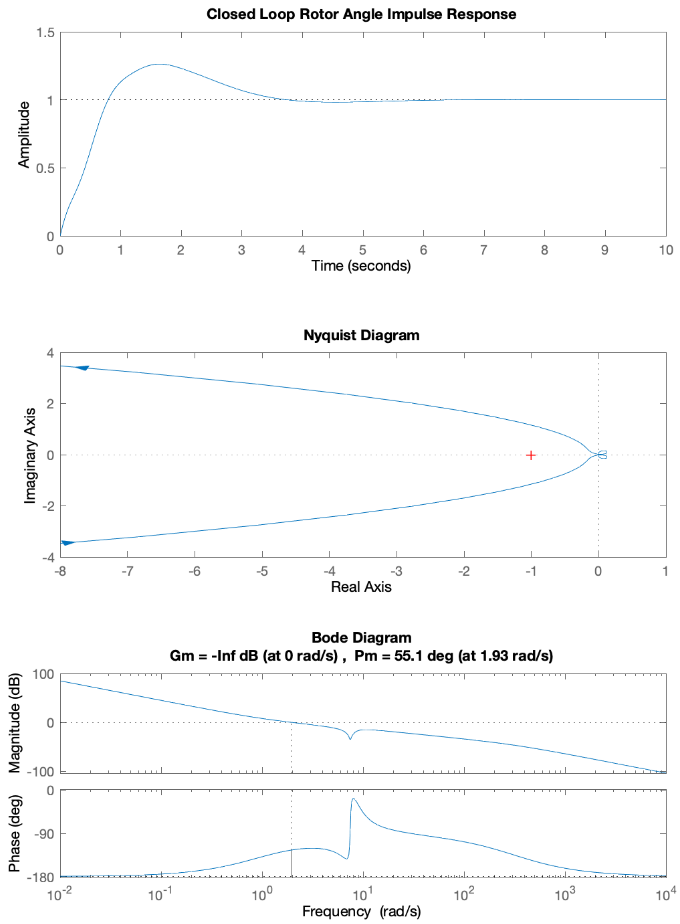


Figure 24. Closed Loop Impulse Response, Nyquist Characteristic and Bode margin plot for $K = 2.0$ *and* $T_{d-r} = 1\ sec.$ *Note that each characteristic indicates stability.*

The conditions for stability may be evaluated by examining Gain and Phase margins. *Figure 24* displays the Bode margin plot for $K = 2.0$ and $T_d = 1$ sec. The Gain and Phase margin values are consistent with stability for this controller design.

The Nyquist Criterion may also be tested for indication of stability. First, the poles of the closed loop system, zeroes of $1 + T_{rotor}K_{rotor}(s)$ are

-3.0833 + 0.0000i
-0.6187 + 0.0000i
-0.0084 + 0.0737i
-0.0084 - 0.0737i
-0.0310 + 0.0395i
-0.0310 - 0.0395i

Then, the poles of $T_{rotor}K_{rotot}(s)$ are

-3.1416 + 0.0000i
-0.6198 + 0.0000i
-0.0064 + 0.0796i
-0.0064 - 0.0796i
-0.0032 + 0.0206i
-0.0032 - 0.0206i

There are no RHP poles, confirming stability. The number of zeroes of $1 + T_{rotor}K_{rotor}(s)$ in the right half plane, Z = 0. Also, the number of poles, of $T_{rotor}K_{rotot}(s)$ in the right half plane, P = 0. Then, the number of clockwise encirclements of the point, s = -1, is zero as observed in the Nyquist Characteristic of *Figure 24*.

The frequency response of the Sensitivity Functions may be determined by measuring the real-time response of each Sensitivity Function transfer function to a step function input. The Edukit System includes a Sensitivity Function Measurement system (described in Section 20) that includes methods for configuring the Edukit real time control system to introduce each of the $R(s)$, $D(s)$, and, $N(s)$ signals while measuring $e(s)$, $U(s)$, and $Y(s)$ in real time.

The input signals of $R(s)$, $D(s)$, and, $N(s)$ are applied as step pulse signals of amplitude that may be configures and with step pulse period and frequency that may also be configured. For the results here, the step pulse amplitude was 40 degrees and the period was 20 seconds at a frequency of 0.05 Hz. In order to measure each Sensitivity Function, one is selected and data is acquired for a period of at least 60 seconds. Finally, the Fourier transform of both the input signal and response are computed. The ratio of these transforms is computed over the spectral region of 0.1 to 1000 rad/sec. Finally, the magnitude of this ratio provides the magnitude of the transfer function frequency response.

The Edukit system response to a step input to the reference tracking signal, $\phi_{Rotor-Command}(s)$ is shown in *Figure 25*. Close agreement between simulated and measured response is observed. Also, Sensitivity Function, Complimentary Sensitivity Function, Load Disturbance Rejection Sensitiivty Function, and Noise Rejection Sensitivity Function results are shown in *Figure 26*, *Figure 27*, *Figure 28*, *Figure 29*, respectively. Now, a wide range of many design objectives may proceed with capability to accurately determine each of the fundamental control system performance characteristics.

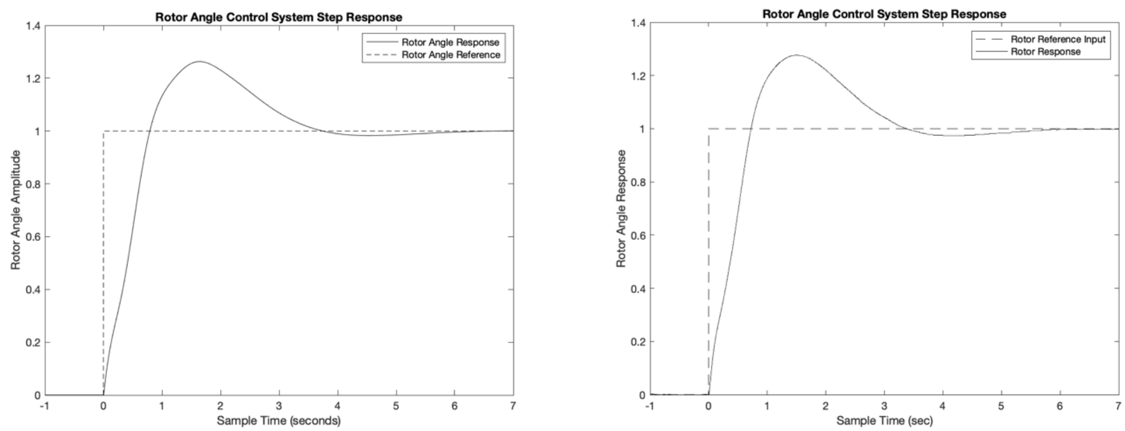


Figure 25. Rotor Angle Step Response for Edukit System with PD controller with Pendulum Controller Gain of K = 10.0 and T_d = 0.5 sec and also with Rotor Controller Gain of K = 10.0 and T_{d-r} = 0.5 sec. Step response is normalized to a step amplitude of 40 degrees applied to $\phi_{Rotor-Command}$ at t = 0. Simulated response is shown at left and experimental system measurement is shown at right.

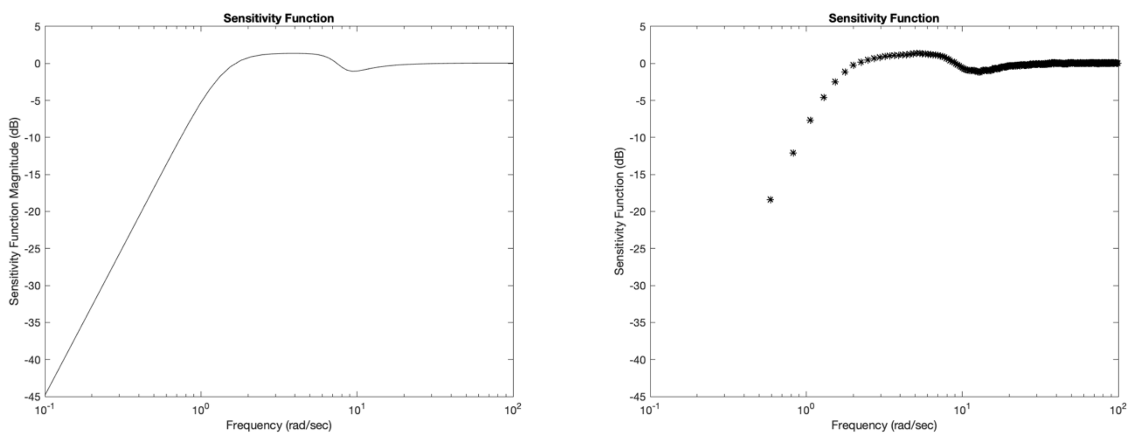


Figure 26. Sensitivity Function Frequency Response for Edukit System with PD controller with Pendulum Controller Gain of K = 10.0 and T_d = 0.5 sec and also with Rotor Controller Gain of K = 10.0 and T_{d-r} = 0.5 sec. Simulated response is shown at left and experimental system measurement is shown at right.

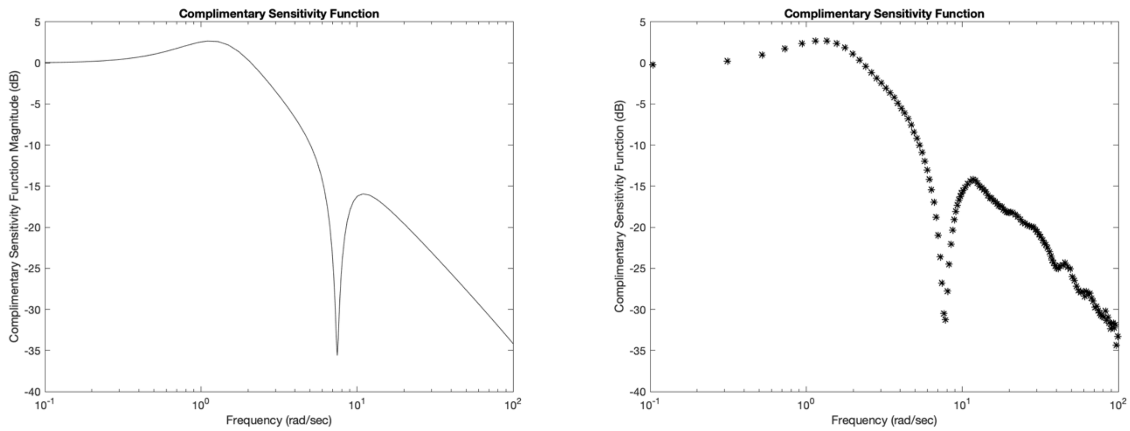


Figure 27. Complimentary Sensitivity Function Frequency Response for Edukit System with PD controller with Pendulum Controller Gain of K = 10.0 and T_d *= 0.5 sec and also with Rotor Controller Gain of K = 10.0 and* T_{d-r} *= 0.5 sec. Simulated response is shown at left and experimental system measurement is shown at right.*

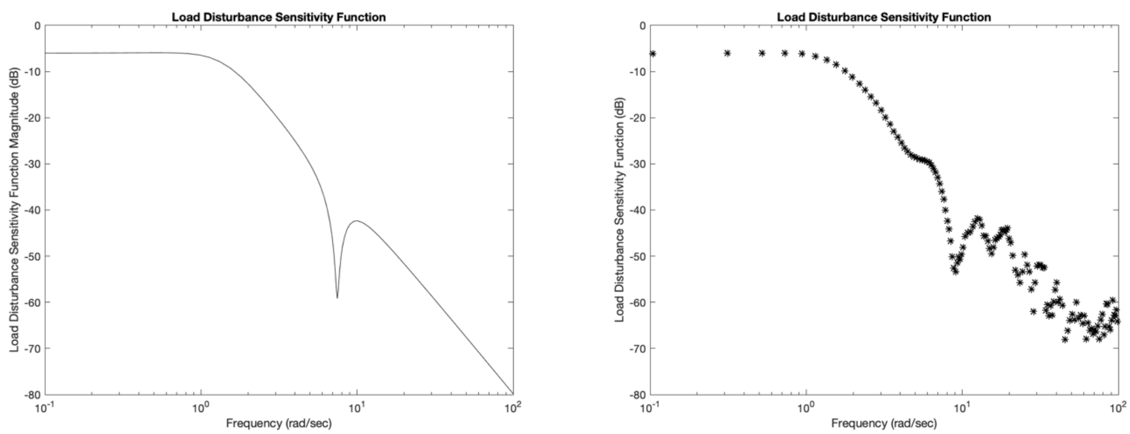


Figure 28. Load Disturbance Sensitivity Function Frequency Response for Edukit System with PD controller with Pendulum Controller Gain of K = 10.0 and T_d *= 0.5 sec and also with Rotor Controller Gain of K = 10.0 and* T_{d-r} *= 0.5 sec. Simulated response is shown at left and experimental system measurement is shown at right.*

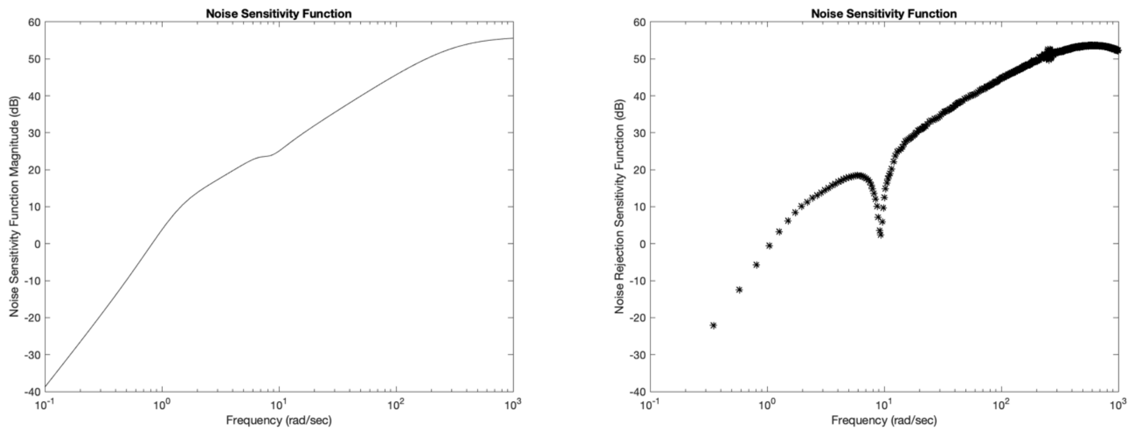


Figure 29. Noise Sensitivity Function Frequency Response for Edukit System with PD controller with Pendulum Controller Gain of K = 10.0 and T_d *= 0.5 sec and also with Rotor Controller Gain of K = 10.0 and* T_{d-r} *= 0.5 sec. Simulated response is shown at left and experimental system measurement is shown at right.*

10. Inverted Pendulum Dynamic Response

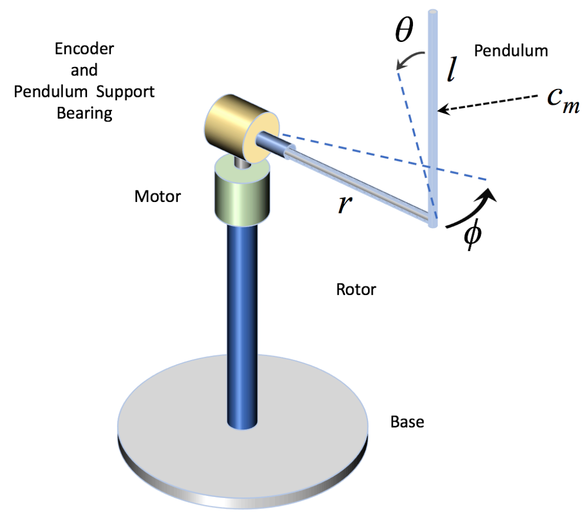


Figure 30. The Edukit Rotary Inverted Pendulum System shown with pendulum inverted. The definitions of dimensions and rotation angles defining pendulum position are also shown. This system is configured with a uniform pendulum rod of mass, m.

The inverted pendulum configuration is shown in *Figure 30*. This is an inherently *unstable* system where Pendulum angle, θ remains zero in the absence of a control input of rotor angle, ϕ.

The response of the inverted pendulum is determined by rotor angle, ϕ, gravitational acceleration, and friction force. The derivation of the transfer function between $\phi(s)$ and $\theta(s)$ is provided in Appendix D: Integrated Rotary Inverted Pendulum: Pendulum Dynamics. This includes description of System Identification applied to determine Q. The $G_{pend}(s)$ transfer function for the inverted pendulum is:

$$G_{pend}(s) = \frac{\theta(s)}{\phi(s)} = \frac{{\omega_r}^2 s^2}{s^2 + {}^{\omega_g}/_{Q}\, s - {\omega_g}^2}$$

With

$${\omega_r}^2 = \frac{r}{L\left(\frac{2}{3} + \frac{2I_{rotor}}{mL^2}\right)} = \frac{r}{L(0.6805)} = 0.803(rad/sec)^2$$

$${\omega_g}^2 = \frac{g}{L\left(\frac{2}{3} + \frac{2I_{rotor}}{mL^2}\right)} = \frac{g}{L(0.6805)} = 55.85(rad/sec)^2$$

$$Q = 20$$

11. Inverted Pendulum Single PID Controller: Frequency Response Design

This section describes the development of a Single Input Single Output (SISO) PID Controller controlling only Pendulum Angle. This is followed by a Section that describes a Dual PID Controller architecture that combines two SISO systems. Both systems will be designed using Frequency Response methods.

This design and development provides valuable background in PID control and its limitations associated with design guidance. Later Sections introduce Multiple Input Multiple Output (MIMO) Control based on the Linear Quadratic Regulator. This permits development of optimal control for parameters including a specified cost function.

The Suspended Mode feedback control system supplies an output to Rotor Control, ϕ, to control the difference between Pendulum angle, θ, and a reference tracking command, $\theta_{reference}$, as shown in *Figure 31*.

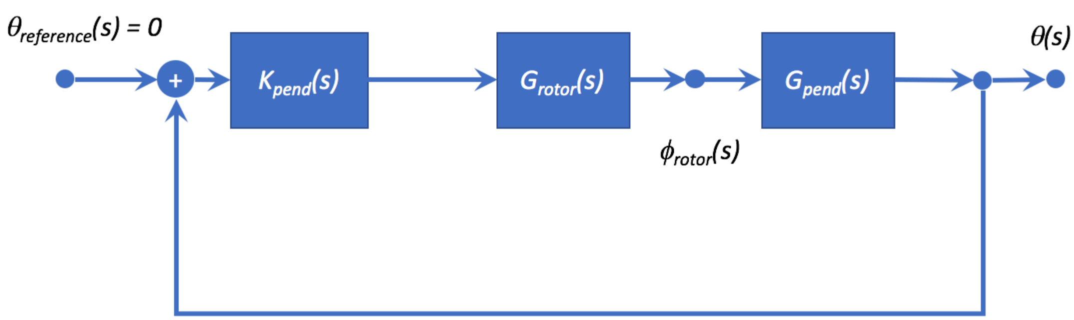


Figure 31. Pendulum Angle stabilization control system for Suspended Mode Pendulum operation.

Examining, *Figure 31*, the transfer function from $\theta_{reference}$ to θ is

$$\frac{\theta(s)}{\theta_{reference}(s)} = \frac{G_{rotor}(s)G_{pendulum}(s)K_{pendulum}(s)}{1 + G_{rotor}(s)G_{pendulum}(s)K_{pendulum}(s)}$$

As described in Section 6, development of the PID control system first requires development of a model for the Stepper Motor system providing Rotor control of Rotor Angle, ϕ, in response to Rotor Angle Control input, $\phi_{Rotor-Control}$ The Medium Speed System example of Table 2 is applied to the system in this section.

Thus, the Rotor response transfer function of *Figure 9* is defined as in Section 6, Edukit Rotor System Plant,

$$G_{rotor}(s) = \frac{\phi(s)}{\phi_{Rotor-Control}(s)} = \frac{1}{s^2}$$

First, the inherent instablility of the Pendulum Plant is observed. The Pendulum Plant poles are

-7.6130
7.3362

The Pendulum Plant Zeroes are

0
0

Second, the stability of the plant including the Pendulum and Rotor response transfer functions is evaluated. The response of the $G_{pend}(s)G_{rotor}(s)$, is shown in the Bode plot of *Figure 32*. This plant is unstable as observed with its phase of -180 degrees.

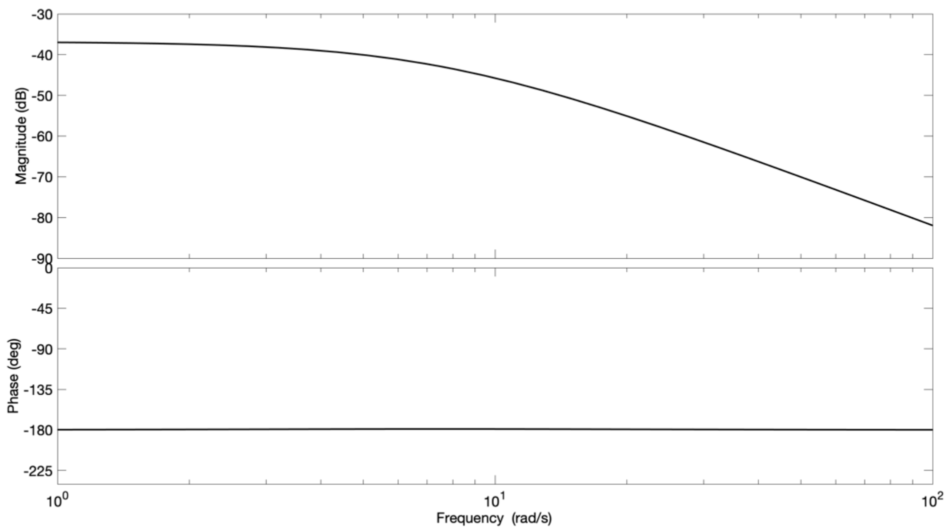


Figure 32. The Bode plot of Pendulum-Rotor plant, $G_{pend}(s)G_{rotor}(s)$, is shown.

The poles appearing for $G_{pend}(s)G_{rotor}(s)$ are

0
0
-7.6130
7.3362

The zeroes appearing for $G_{pend}(s)G_{rotor}(s)$ are both at the origin.

0
0

Design of a SISO PID controller proceeds using frequency response methods. The design objectives for this example will include:

4) Verification of stability by examination of both Bode and Nyquist characteristics.

5) Phase margin is greater than 30 degrees [Seborg_1989]

6) Selection of Design Based on Minimizing Maximum Values of Magnitude of Sensitivity Functions and selecting for minimum Settline Time.

Stable and robust control design can be obtained by ensuring that the maximum values of Sensitivity and Complimentary Sensitivity Functions are less than a threshold. [Hast_2013] [Yaniv_2004]

The maximum values of Sensitivity and Complimentary Sensitivity Functions are computed for each design iteration. The design selection will occur with parameters that minimize these maximum values as will be discussed.

The first step in design is directed to development of a PD controller with derivative time, T_d, and proportional gain, K.

The form of a PD controller is,

$$K_{pend}(s) = K(1 + T_d s)$$

The process for selection of K, T_i. and T_d will include the following steps with election of initial estimates for K, T_i. and T_d.

Iterative adjustment of K, T_i. and T_d values will then continue to obtain system performance meeting design characteristics. At each iteration, the following considerations will apply:

1) Verification of Stability by Nyquist analysis
2) Verification of Gain and Phase Margin by Bode Stability analysis

Iteration will continue with the objectives to select K, T_i. and T_d to

3) Minimize impulse response settling time system
4) Minimize the maximum values of the Sensitivity Functions to select a design meeting specified objectives.

An illustration of this process begins with a selection of $T_d = 0.1s$. This will be adjusted to explore the influence of T_d values on performance. Also the gain, K, values will be adjusted. It is important to note that this is an illustration of one approach and is not intended to imply that this is an optimal solution for a specified performance objective.

The Bode plot of the Pendulum and Rotor plant $G_{rotor}G_{pend}$, Plant and Controller, $G_{rotor}G_{pend}K_{pend}$, and the open loop controller, K_{pend} , are shown for gain, *K* = *1* and T_d = 0.1 sec in *Figure 33*.

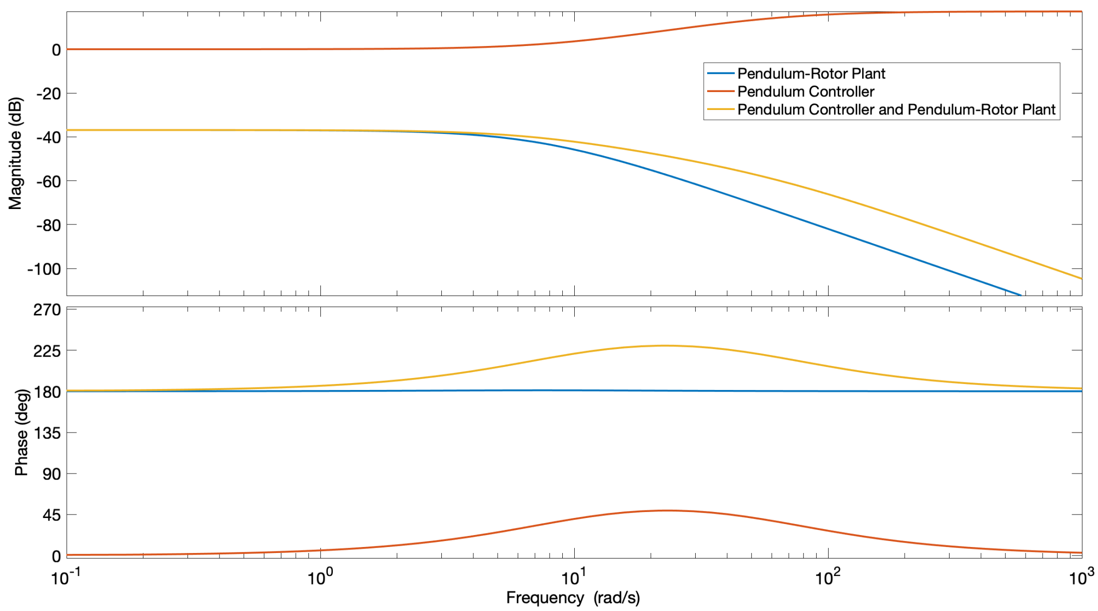


Figure 33. The Bode plot of the Pendulum and Rotor plant $G_{rotor}G_{pend}$, Plant and Controller, $G_{rotor}G_{pend}K_{pend}$, and the open loop controller, K_{pend}, are shown for gain, K = 1 and T_d = 0.1 sec.

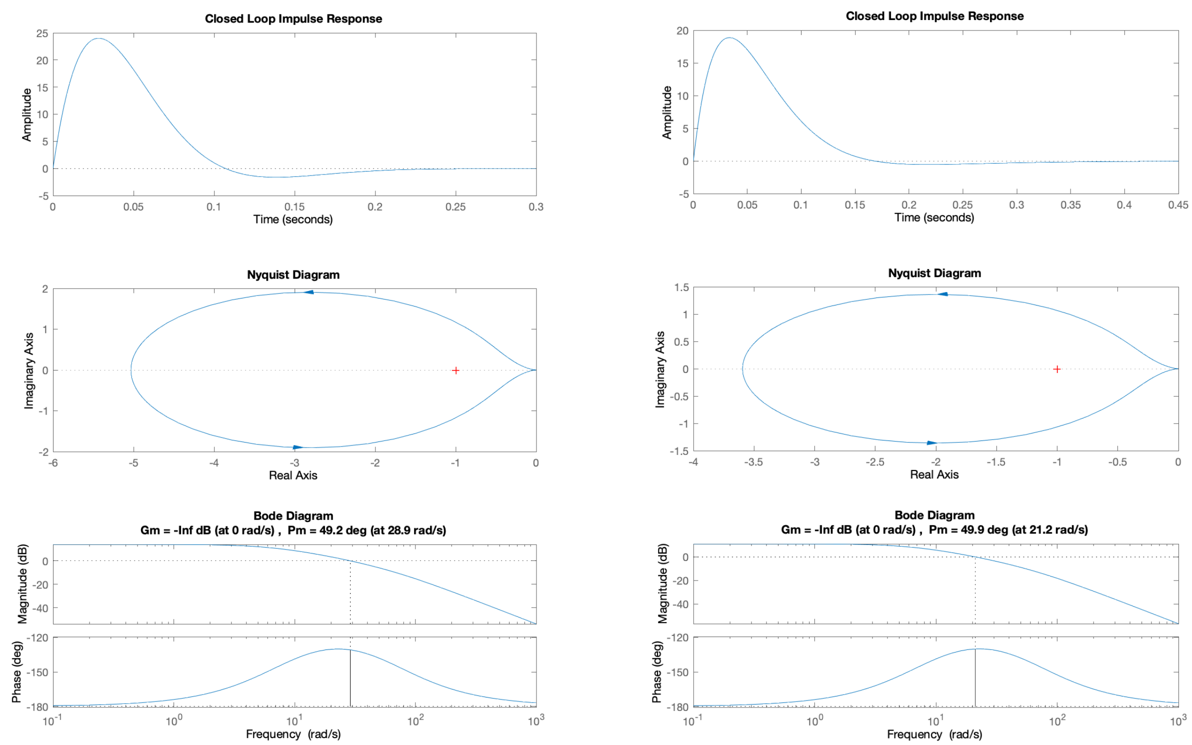


Figure 34. The Impulse Response, Nyquist Diagram, and Bode Margin plot of $G_{rotor}G_{pend}K_{pend}$ *are shown for K = 250 and* T_d *= 0.1 sec (left) and K = 350 and* $T_d = 0.1$ *sec (right).*

The Impulse Response, Nyquist Characteristics, and Bode Margin Characteristics are shown for *K = 250*, *K = 350* and *K = 300* in *Figure 34* and *Figure 35*, respectively. These characteristics show variation in Impulse Response Settling Time (to 2 percent of the peak value of the Impulse Response). However, each *K* value results in stablility as evaluated by the Nyquist Characteristics, and Bode Margin Characteristics.

For example, for *K = 300* the Nyquist Criterion may be tested for verification of stability. First, the poles of the closed loop system, zeroes of $1 + G_{rotor}G_{pend}K_{pend}(s)$ are

0.0000 + 0.0000i
0.0000 + 0.0000i
-26.7001 +22.0167i
-26.7001 -22.0167i
-9.7084 + 0.0000i

Then, the poles of $G_{rotor}G_{pend}K_{pend}(s)$ are

0
0
-62.8319
-7.6130
7.3362

The number of zeroes of $1 + G_{rotor}G_{pend}K_{pend}(s)$ in the right half plane, Z = 0. Also, the number of poles, of $G_{rotor}G_{pend}K_{pend}(s)$ in the right half plane, P = 1. Then, the number of counterclockwise encirclements of the point, s = -1, is one, as observed.

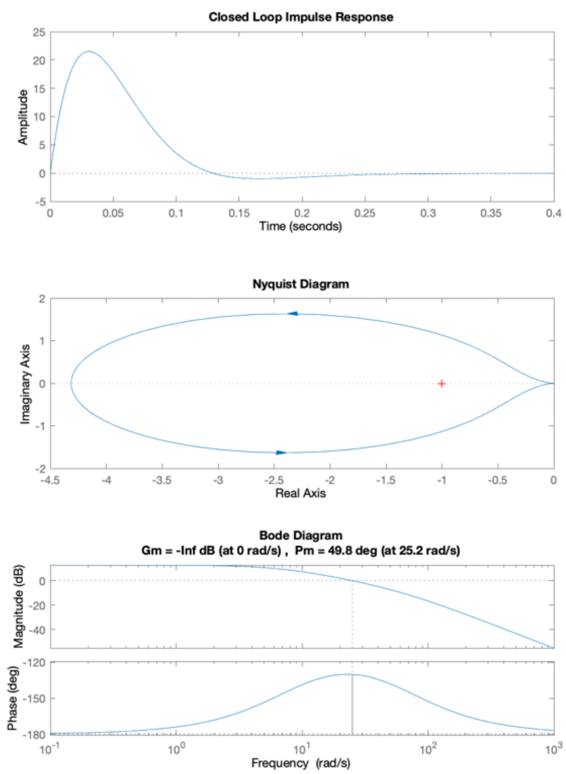


Figure 35. The Impulse Response, Nyquist Diagram, and Bode Margin plot of $G_{rotor}G_{pend}K_{pend}$ are shown for K = 300 and T_d = 0.1 sec.

The maximum values and frequency response of the Sensitivity Functions may be determined by measuring the real-time response of each Sensitivity Function transfer function to a step function input. First, the control system configuration enabling the measurement of Sensitivity Functions is shown in *Figure 36*.[Astrom_2019]

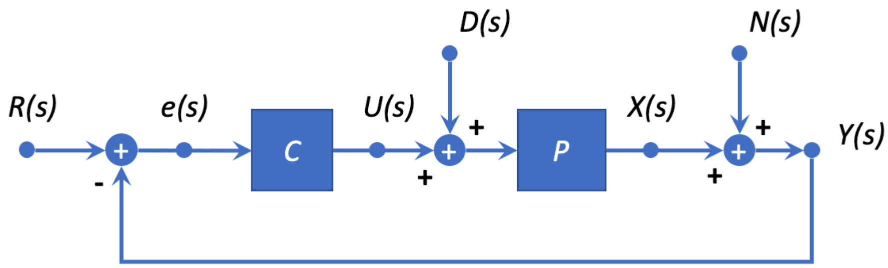


Figure 36.The Edukit control system architecture for either Pendulum or Rotor control shown with signal inputs for measurement of Sensitivity Functions. Sensitivity Functions are measured by applying Reference Tracking signals, $R(s)$, Load Disturbance signals, $D(s)$, and Noise signals, $N(s)$, Measured response is obtained at $e(s)$, $U(s)$, and $Y(s)$.

The Sensitivity Functions defined in terms of the input Reference Tracking signals, $R(s)$, Load Disturbance signals, $D(s)$, and Noise signals, $N(s)$, and measured response of $e(s)$, $U(s)$, and $Y(s)$ are:

a. Sensitivity Function

$$S(s) = \frac{e(s)}{R(s)} = \frac{1}{1 + T_{rotor}(s)K_{rotor}(s)}$$

b. Complimentary Sensitivity Function

$$T(s) = \frac{Y(s)}{R(s)} = \frac{T_{rotor}(s)K_{rotor}(s)}{1 + T_{rotor}(s)K_{rotor}(s)}$$

c. Load Disturbance Rejection Sensitivity Function

$$LD(s) = \frac{Y(s)}{D(s)} = \frac{T_{rotor}(s)}{1 + T_{rotor}(s)K_{rotor}(s)}$$

d. Noise Rejection Sensitivity Function

$$NS(s) = \frac{U(s)}{N(s)} = \frac{K_{rotor}(s)}{1 + T_{rotor}(s)K_{rotor}(s)}$$

The maximum value for the range of $s = 0$ to $s = \infty$ will be computed for each with

$$M_S = \max_{0 \le \omega < \infty} |S(j\omega)|$$
$$M_T = \max_{0 \le \omega < \infty} |T(j\omega)|$$
$$M_{LD} = \max_{0 \le \omega < \infty} |LD(j\omega)|$$
$$M_{NS} = \max_{0 \le \omega < \infty} |NS(j\omega)|$$

The Edukit Real Time Control System Workbench (described in Section 19, Appendix B and Appendix C) includes methods for configuring the Edukit real time control system to introduce each of the $R(s)$, $D(s)$, and, $N(s)$ signals while measuring $e(s)$, $U(s)$, and $Y(s)$ in real time.

For each of the three gain values the system response characteristics have been computed and included in *Table 3*.

K	T_d (sec)	Phase Margin (degrees)	Gain Margin (dB)	2% Settling Time (secs)	M_s	M_T	M_{LD}	M_{NS}	$NS(j\omega)$ *at* $\omega = 1000$
250	0.1	49.9	Inf	0.48	1.34	1.43	0.0055	1819.9	1819.9
300	0.1	49.8	Inf	0.47	1.38	1.36	0.0043	2184.6	2184.6
350	0.1	49.2	Inf	0.45	1.42	1.32	0.0036	2555.9	2549.4

Table 7. Pendulum Controller Gain Margin, Phase Margin, Settling Time, and Maximum Values of Sensitivity Functions for each gain, K, value and $T_d = 0.1$.

The value of *K = 10* is chosen for this example as the meeting the objectives for minimizing Sensitivity Function values as well as providing an intermediate settling time value.

Then, for this value of *K*, the influence of T_d selection is evaluated. The results are shown in *Table 4*.

K	T_d (sec)	Phase Margin (degrees)	Gain Margin (dB)	2% Settling Time (secs)	M_s	M_T	M_{LD}	M_{NS}	$NS(j\omega)$ *at* $\omega = 1000$
300	0.05	34.0	Inf	0.63	1.72	1.93	0.0051	1241.4	1241.4
300	0.1	49.8	Inf	0.47	1.38	1.36	0.0043	2184.6	2184.6
300	0.15	51.96	Inf	0.73	1.41	1.30	0.0043	3233.2	3129.1

Table 8. Pendulum Controller Gain Margin, Phase Margin, Settling Time, and Maximum Values of Sensitivity Functions for each T_d *value and gain,* $K = 10$.

Finally, the value of *K = 300* and $T_d = 0.1$ is chosen for this example as the meeting the objectives for minimizing Sensitivity Function values as well as providing an intermediate settling time value. It is important to note that this is an example illustration of one method of controller design. Many other methods can now be used to meet specific objectives.

The value of K = *300* and T_d = 0.1 sec. yields these gain values:

$$K_P = 300.0, \qquad K_I = 0.0, \qquad K_D = KT_d = 30.0$$

The Edukit control system also includes an impulse generation method. This introduces a pulse of amplitude, 10 degrees for a duration of 4 milliseconds in the Pendulum Angle Reference Tracking signal, $\theta_{reference}$. The Pendulum Angle Response is shown in *Figure 37*.

Note that the discrete steps in the output of the digital rotary encoder are visible – each corresponds to 0.15 degrees of rotation.

An important limitation of this Single Input Single Output (SISO) controller appears. Specifically, while the controller stabilizes Pendulum Angle, Rotor Angle does not appear as an input and is not stabilized. Thus, a disturbance in Rotor position is not compensated for by this controller.

The limitations of SISO control may be characterized directly. The response of Rotor Angle, ϕ_{rotor}, to a signal applied to $\theta_{reference}$ may be computed. Examining *Figure 9*, the transfer function from $\theta_{reference}(s)$ to $\phi_{rotor}(s)$ is

$$T_{Rotor-Response} = \frac{\phi_{rotor}(s)}{\theta_{reference}(s)} = \frac{G_{rotor}K_{pend}}{1 + G_{pend}G_{rotor}K_{pend}}$$

Thus, an impulse in $\theta_{reference}(s)$ induces a response in Rotor Angle.

This SISO system does not provide control of Rotor Angle. Thus, Rotor Angle may drift as a result of the Pendulum Angle Reference impulse.

As an experimental example, the Pendulum position for the Edukit Rotary Inverted Pendulum was disturbed, leading to a drift in Rotor position shown in *Figure 38*.

The next section describes the addition of a second PID controller that provides Rotor Angle stabilization relative to a Rotor Angle Reference tracking command.

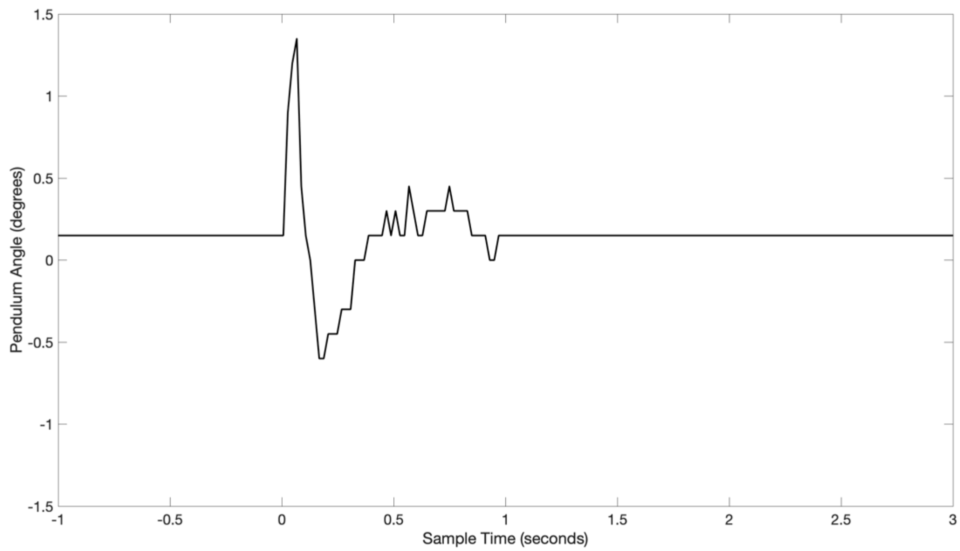


Figure 37. Impulse response of Pendulum angle for a Pendulum Angle Reference Tracking signal, $\theta_{reference}$ of amplitude 10 degrees applied at t = 0.

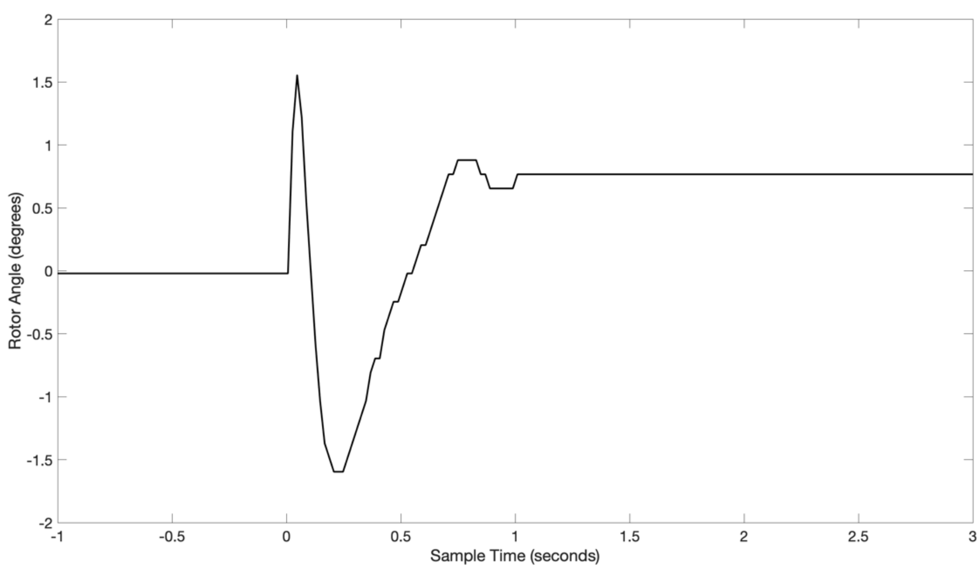


Figure 38. Response of Rotor Angle, ϕ_{rotor}, to an impulse signal applied to Pendulum angle reference, $\theta_{reference}$, at t = 0. Note the uncompensated change in Rotor Angle. During this period, Pendulum Angle remains constant and near zero.

12. Inverted Pendulum Dual PID Controller: Frequency Response Design

The previous Section 8, introduced a method for design of a Pendulum Angle Controller. While Pendulum Angle control was established, Rotor Angle was not constrained or provided with a control capability. Thus, Rotor Angleis subject to drift.

The introduction of a second PID controller will enable stabilization of Rotor Angle. This will be accomplished by introducing an Inner Loop and Outer Loop control system.

The design process will include the steps of:

5) Design of Single PID Controller, $K_{pend}(s)$
6) Definition of Inner Control Loop including $K_{pend}(s)$
7) Definition of Outer Control Loop integrating the Inner Control Loop and a new controller, $K_{rotor}(s)$
8) Design of the Rotor Control system, $K_{rotor}(s)$.

The Inner Loop controller will depend on the design of $K_{pend}(s)$ in Section 8. This design will exploit as designed $K_{pend}(s)$ while a new Outer Loop controller of Rotor Angle, $K_{rotor}(s)$, is added

The Inner Loop Controller defines a transfer function from a Rotor Angle Control signal, $\phi_{Rotor-Control}$ to the Rotor Angle, ϕ_{Rotor} as shown in *Figure 39*.

The Outer Loop Controller provides control of Rotor Angle, Rotor Angle, ϕ_{Rotor} to a Rotor Angle Tracking Command, $\phi_{Rotor-Command}$ as shown in *Figure 40*.

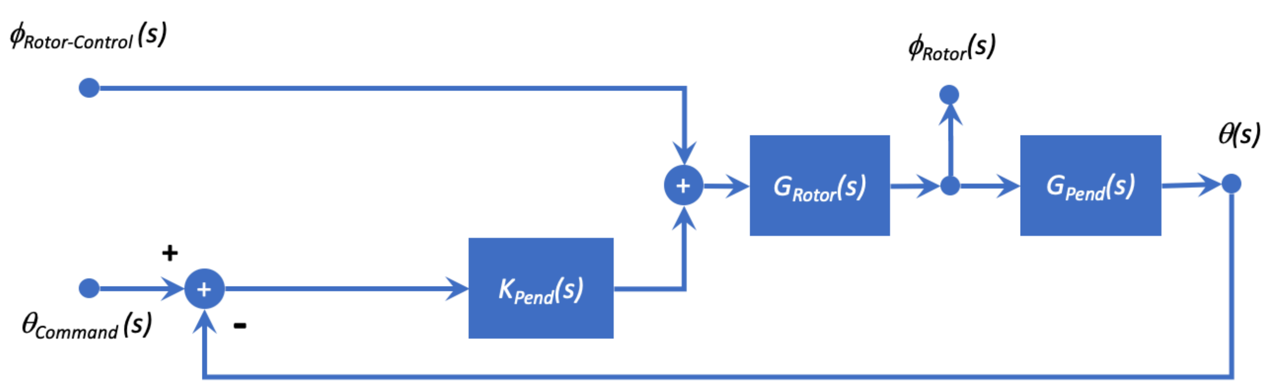


Figure 39. Dual PID Control System Inner Control Loop

The SISO Pendulum Angle control system architecture of *Figure 19* will be now modified to create an Inner Loop controller.

Now, the Inner Loop Rotor Angle Control transfer function, $T_{Rotor}(s)$ between $\phi_{Rotor-Control}$ and ϕ_{Rotor} will be determined. The Pendulum Angle reference input signal, $\theta_{command}(s)$, will be held zero.

$$\phi_{Rotor}(s) = \left(\phi_{Rotor-Control}(s) - \phi_{Rotor}(s)G_{pendulum}(s)K_{pendulum}\right) G_{Rotor}(s)$$

$$T_{rotor}(s) = \frac{\phi_{Rotor}(s)}{\phi_{Rotor-Control}(s)}$$

and

$$T_{rotor}(s) = \frac{G_{rotor}(s)}{1 + G_{pend}(s)K_{pend}(s)\, G_{rotor}(s)}$$

This Rotor Angle Control system now introduces the Outer Loop controller to enable control of Rotor Angle. Here, the difference between a Rotor Angle tracking command, $\phi_{Rotor-Command}(s)$ and Rotor Angle, $\phi_{Rotor}(s)$, is supplied as an error signal to a new controller, K_{rotor}, and summed with the output of the Pendulum Angle Controller, $K_{pendulum}$ as shown in *Figure 20*.

The control system block diagram can also be drawn with the Inner Loop represented by $T_{rotor}(s)$ as shown in *Figure 41*.

During subsequent design steps, the configuration of $K_{pend}(s)$ will remain fixed and as specified in the previous design steps.

The Outer Loop controls ϕ_{Rotor} with the goal of tracking the reference input, $\phi_{Rotor-Command}(s)$.

By definition

$$\phi_{rotor} = \phi_{Rotor-Control}(s)T_{rotor}(s)$$

Thus, the transfer function from $\phi_{Rotor-Command}$ to ϕ_{Rotor} is

$$T_{rotor-tracking}(s) = \frac{\phi_{rotor}(s)}{\phi_{Rotor-Command}(s)} = \frac{T_{rotor}(s)K_{rotor}(s)}{1 + T_{rotor}(s)K_{rotor}(s)}$$

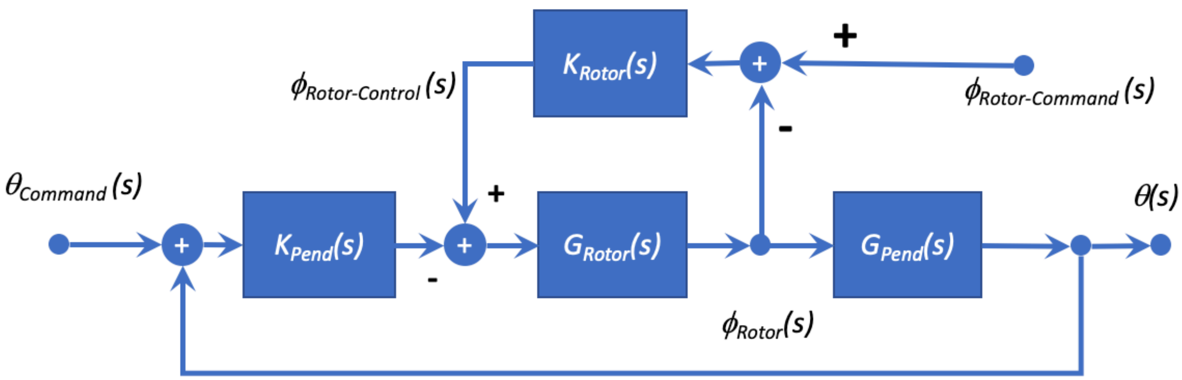


Figure 40. Dual PID Control System with Inner and Outer Control Loop

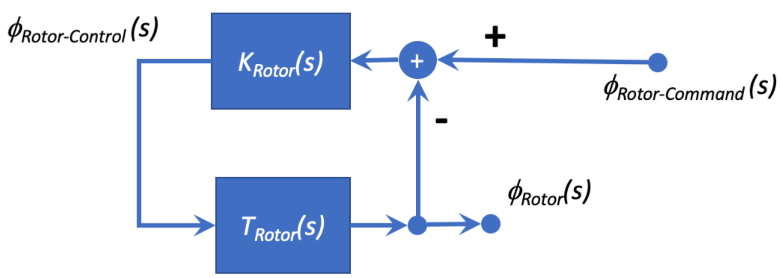


Figure 41. Dual PID Control System with Inner Control Loop represented by $T_{rotor}(s)$

Design of the Rotor PID controller proceeds using frequency response methods. The design objectives for this example will include:

4) Verification of stability by examination of both Bode and Nyquist characteristics.

5) Phase margin is greater than 30 degrees [Seborg_1989]

6) Selection of Design Based on Minimizing Maximum Values of Magnitude of Sensitivity Functions and selecting for minimum Settline Time.

Stable and robust control design can be obtained by ensuring that the maximum values of Sensitivity and Complimentary Sensitivity Functions are less than a threshold. [Hast_2013] [Yaniv_2004]

The maximum values of Sensitivity and Complimentary Sensitivity Functions are computed for each design iteration. The design selection will occur with parameters that minimize these maximum values as will be discussed.

The first step in design is directed to development of a PD controller with derivative time constant, T_{d-r}, and proportional gain, K.

The form of the PD controller is,

$$K_{rotor}(s) = K(1 + T_{d-r}s)$$

The process for selection of K and T_{d-r} will include the following steps with election of initial estimates for K and T_{d-r}.

Iterative adjustment of K, T_i. and T_d values will then continue to obtain system performance meeting design characteristics. At each iteration, the following considerations will apply:

1) Verification of Stability by Nyquist analysis
2) Verification of Gain and Phase Margin by Bode Stability analysis

Iteration will continue with the objectives to select $K$and T_{d-r} to

5) Minimize impulse response settling time system
6) Minimize the maximum values of the Sensitivity Functions to select a design meeting specified objectives.

An illustration of this process begins with a selection of $T_{d-r} = 0.5\ sec$. This will be adjusted to explore the influence of T_{d-r} values on performance. Also the gain, K, values will be adjusted. It is important to note that this is an illustration of one approach and is not intended to imply that this is an optimal solution for a specified performance objective.

The Bode plot of the Pendulum and Rotor plant $T_{Rotor}(s)$ Plant and Controller, $T_{Rotor}(s)K_{Rotor}(s)$, and the open loop controller, K_{Rotor}, are shown for gain, $K = 1$ and T_{d-r} = 0.5 sec in *Figure 42*.

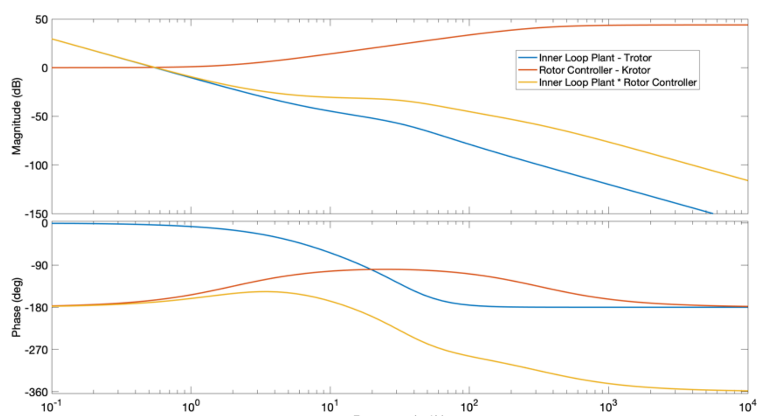


Figure 42.The Bode plot for the controller, K_{rotor}, the Inner Loop Plant, T_{rotor}, and for the open loop combined plant and controller, $K_{rotor}T_{rotor}$, are shown for K = 1 and T_{d-r} = 0.5 sec.

The Impulse Response, Nyquist Characteristics, and Bode Margin Characteristics are shown for *K = 10.0*, *K = 20.0* and *K = 15.0* in *Figure 43* and *Figure 44*, respectively. These characteristics show variation in Impulse Response Settling Time (to 2 percent of the peak value of the Impulse Response). However, each *K* value results in stablility as evaluated by the Nyquist Characteristics, and Bode Margin Characteristics.

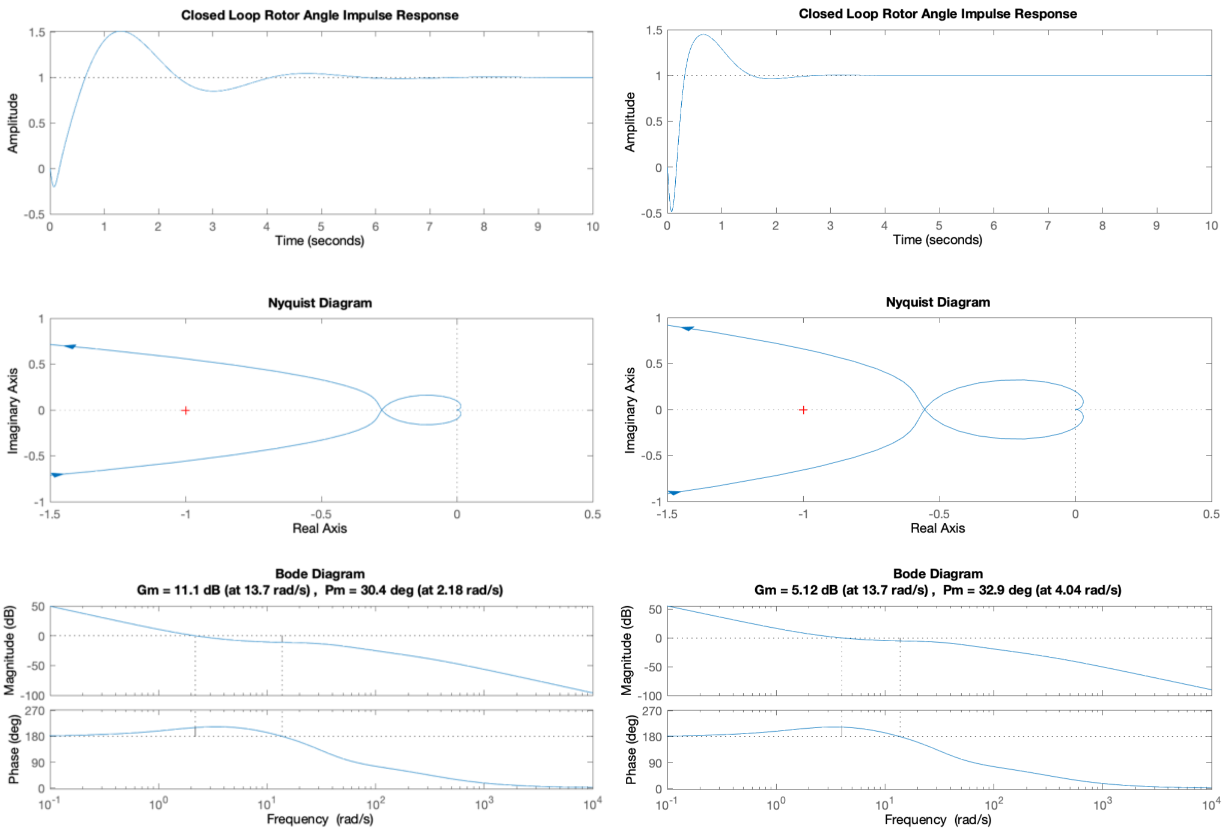


Figure 43. Closed Loop Impulse Response, Nyquist Characteristic and Bode margin plot for $K = 10.0$ *(left), for* $K = 20.0$ *(right) and* $T_{d-r} = 0.5\ sec.$ *Note that each characteristic indicates stability.*

As described in the previous Section, the Sensitivity Functions defined in terms of the input Reference Tracking signals, $R(s)$, Load Disturbance signals, $D(s)$, and Noise signals, $N(s)$, and measured response of $e(s)$, $U(s)$, and $Y(s)$ may be computed. Also, the maximum values and frequency response of the Sensitivity Functions may be determined by measuring the real-time response of each Sensitivity Function transfer function to a step function input.

For each of the three gain values the system response characteristics have been computed and included in *Table 9*.

K	T_d (sec)	Phase Margin (degrees)	Gain Margin (dB)	2% Settling Time (secs)	M_s	M_T	M_{LD}	M_{NS}	$NS(j\omega)$ at $\omega = 1000$
10.0	0.5	30.42	3.61	5.32	1.92	2.07	0.154	1580.0	1506.0
15.0	0.5	33.12	2.40	3.06	1.82	1.89	0.083	2370.0	2257.7
20.0	0.5	32.92	1.8	2.16	2.29	1.82	0.054	3159.9	3007.6

Table 9. Rotor Controller Gain Margin, Phase Margin, Settling Time, and Maximum Values of Sensitivity Functions for each gain, K, value and $T_d = 0.5$.

The value of *K = 15* is chosen for this example as the meeting the objectives for minimizing Sensitivity Function values as well as providing an intermediate settling time value.

Then, for this value of *K*, the influence of T_{d-r} selection is evaluated. The results are shown in *Table 10*.

K	T_d (sec)	Phase Margin (degrees)	Gain Margin (dB)	2% Settling Time (secs)	M_s	M_T	M_{LD}	M_{NS}	$NS(j\omega)$ at $\omega = 1000$
15.0	0.4	44.91	2.95	5.20	2.21	12.33	0.114	1899.0	1809.5
15.0	0.5	33.12	2.40	3.06	1.82	1.89	0.083	2370.0	2257.7
15.0	0.6	37.64	2.02	1.77	2.01	1.65	0.069	2841.0	2704.7

Table 10. Rotor Controller Gain Margin, Phase Margin, Settling Time, and Maximum Values of Sensitivity Functions for each T_{d-r} *value and gain,* $K = 15$.

Finally, the value of *K = 15* and $T_{d-r} = 0.5$ is chosen for this example design as the meeting the objectives for minimizing Sensitivity Function values as well as providing an intermediate settling time value. It is important to note that this is an example illustration of one method of controller design. Many other methods can now be used to meet specific objectives.

The value of *K = 15* and T_{d-r} = 0.5 sec. yields these gain values:

$$K_P = 15.0, \qquad K_I = 0.0, \qquad K_{d-r} = KT_{d-r} = 7.5$$

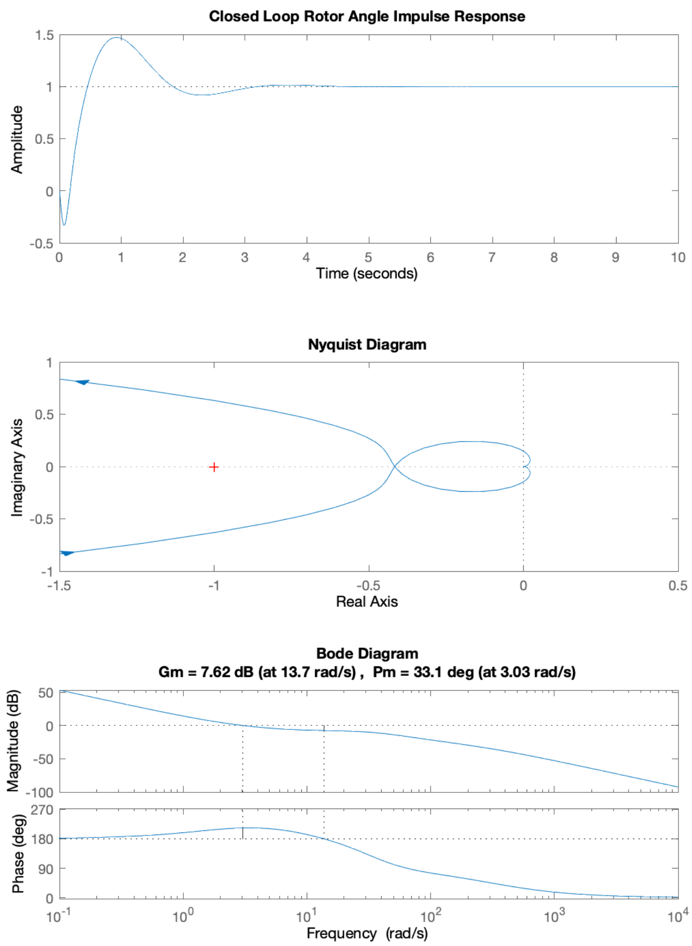


Figure 44. Closed Loop Impulse Response, Nyquist Characteristic and Bode margin plot for $K = 15.0$ *and* $T_{d-r} = 0.5\ sec.$ *Note that each characteristic indicates stability.*

The conditions for stability may be evaluated by examining Gain and Phase margins. *Figure 24* displays the Bode margin plot for *K =15.0* and T_{d-r} = 0.5 sec. The Gain and Phase margin values are consistent with stability for this controller design.

The Nyquist Criterion may also be tested for indication of stability. First, the poles of the closed loop system, zeroes of $1 + T_{rotor}K_{rotor}(s)$ are

-3.2134 + 0.0000i
-0.1760 + 0.1043i
-0.1760 - 0.1043i
-0.1822 + 0.0000i
-0.0126 + 0.0227i
-0.0126 - 0.0227i

Then, the poles of $T_{rotor}K_{rotot}(s)$ are

0.0000 + 0.0000i
0.0000 + 0.0000i
-3.1416 + 0.0000i
-0.2670 + 0.2202i
-0.2670 - 0.2202i
-0.0971 + 0.0000i

There are no RHP poles, confirming stability. The number of zeroes of $1 + T_{rotor}K_{rotor}(s)$ in the right half plane, Z = 0. Also, the number of poles, of $T_{rotor}K_{rotot}(s)$ in the right half plane, P = 0. Then, the number of clockwise encirclements of the point, s = -1, is zero as observed in the Nyquist Characteristic of *Figure 44*.

The frequency response of the Sensitivity Functions may be determined by measuring the real-time response of each Sensitivity Function transfer function to a step function input. The Edukit System includes a Sensitivity Function Measurement system (described in Section 20) that includes methods for configuring the Edukit real time control system to introduce each of the $R(s)$, $D(s)$, and, $N(s)$ signals while measuring $e(s)$, $U(s)$, and $Y(s)$ in real time.

The input signals of $R(s)$, $D(s)$, and, $N(s)$ are applied as step pulse signals of amplitude that may be configures and with step pulse period and frequency that may also be configured. For the results here, the step pulse amplitude was 40 degrees and the period was 20 seconds at a frequency of 0.05 Hz. In order to measure each Sensitivity Function, one is selected and data is acquired for a period of at least 60 seconds. Finally, the Fourier transform of both the input signal and response are computed. The ratio of these transforms is computed over the spectral region of 0.1 to 1000 rad/sec. Finally, the magnitude of this ratio provides the magnitude of the transfer function frequency response.

The Edukit system response to a step input to the reference tracking signal, $\phi_{Rotor-Command}(s)$ is shown in *Figure 25*. Close agreement between simulated and measured response is observed. Also, Sensitivity Function, Complimentary Sensitivity Function, Load Disturbance Rejection Sensitiivty Function, and Noise Rejection Sensitivity Function results are shown in *Figure 26*, *Figure 27*, *Figure 28*, *Figure 29*, respectively. Now, a wide range of many design objectives may proceed with capability to accurately determine each of the fundamental control system performance characteristics.

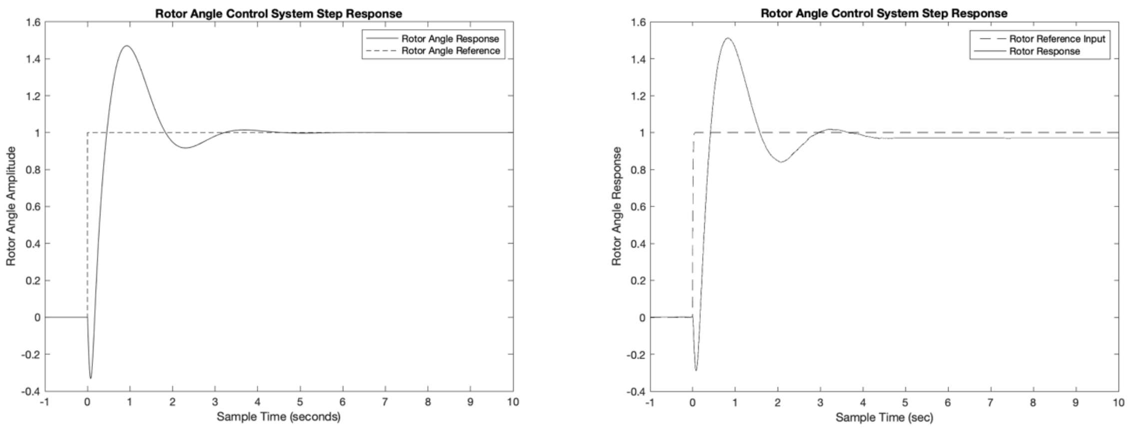


Figure 45. Rotor Angle Step Response for Edukit System with PD controller with Pendulum Controller Gain of K = 300.0 and T_{d-r} = 0.1 sec and also with Rotor Controller Gain of K = 15.0 and T_{d-r} = 0.5 sec. Step response is normalized to a step amplitude of 40 degrees applied to $\phi_{Rotor-Command}$ at t = 0. Simulated response is shown at left and experimental system measurement is shown at right.

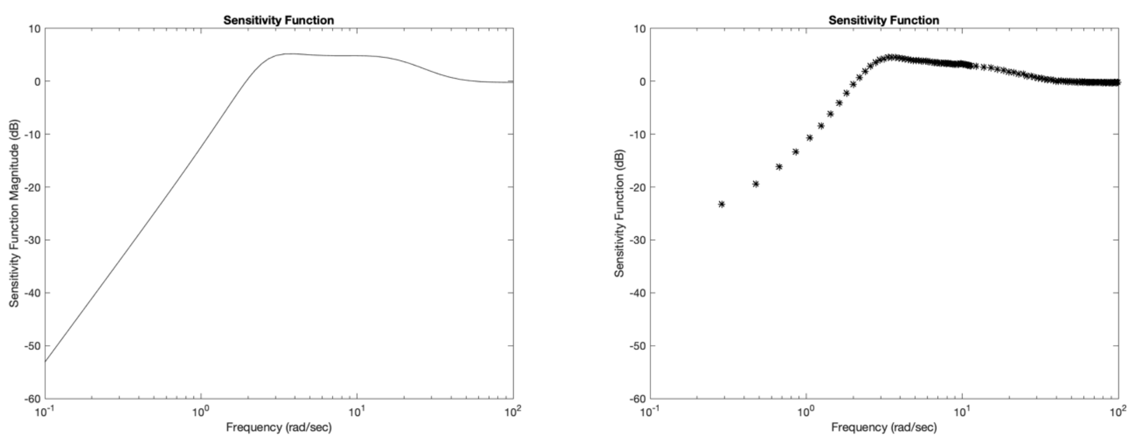


Figure 46. Sensitivity Function Frequency Response for Edukit System with PD controller with Pendulum Controller Gain of K = 300.0 and T_{d-r} = 0.1 sec and also with Rotor Controller Gain of K = 15.0 and T_{d-r} = 0.5 sec. Step response is normalized to a step amplitude of 40 degrees applied to $\phi_{Rotor-Command}$ at t = 0. Simulated response is shown at left and experimental system measurement is shown at right.

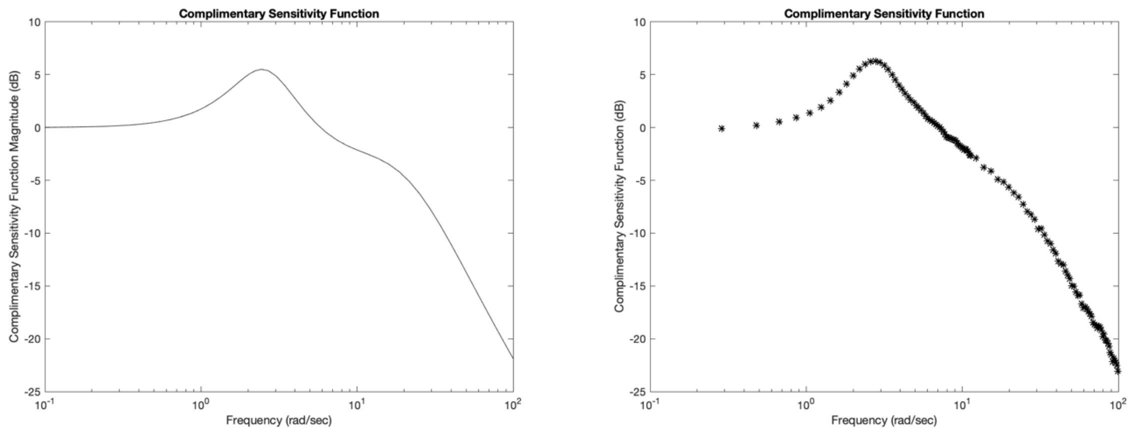


Figure 47. Complimentary Sensitivity Function Frequency Response for Edukit System with PD controller with Pendulum Controller Gain of K = 300.0 and T_{d-r} *= 0.1 sec and also with Rotor Controller Gain of K = 15.0 and* T_{d-r} *= 0.5 sec. Step response is normalized to a step amplitude of 40 degrees applied to* $\phi_{Rotor-Command}$ *at t = 0. Simulated response is shown at left and experimental system measurement is shown at right.*

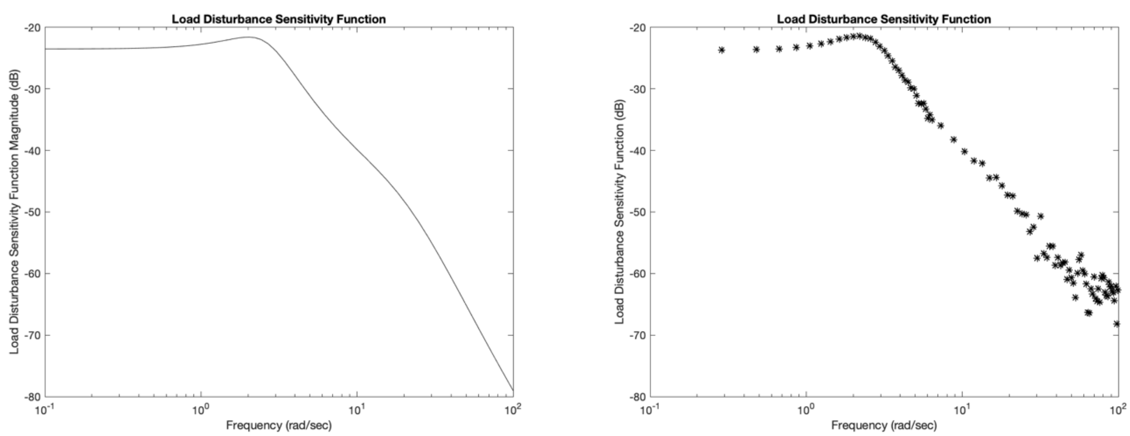


Figure 48. Load Disturbance Sensitivity Function Frequency Response for Edukit System with PD controller with Pendulum Controller Gain of K = 300.0 and T_{d-r} *= 0.1 sec and also with Rotor Controller Gain of K = 15.0 and* T_{d-r} *= 0.5 sec. Step response is normalized to a step amplitude of 40 degrees applied to* $\phi_{Rotor-Command}$ *at t = 0. Simulated response is shown at left and experimental system measurement is shown at right.*

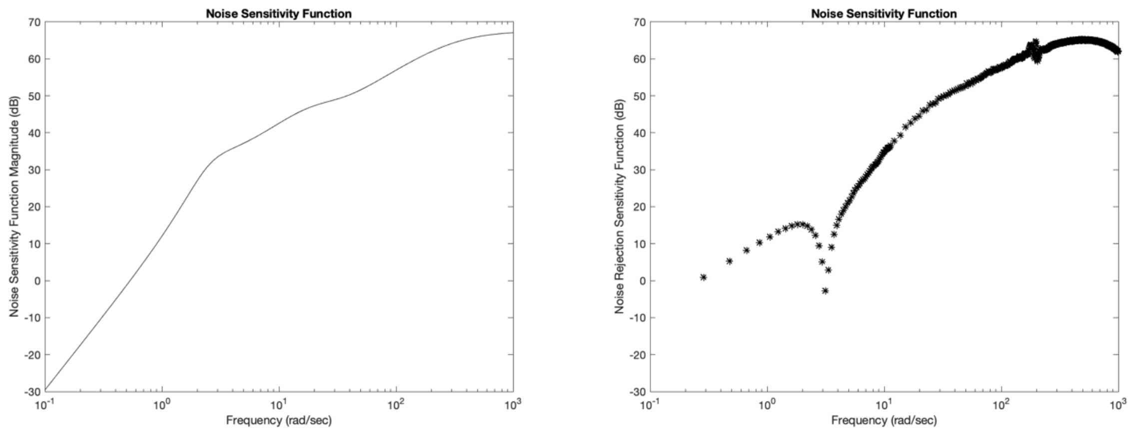


Figure 49. Noise Sensitivity Function Frequency Response for Edukit System with PD controller with Pendulum Controller Gain of K = 300.0 and T_{d-r} = 0.1 sec and also with Rotor Controller Gain of K = 15.0 and T_{d-r} = 0.5 sec. Step response is normalized to a step amplitude of 40 degrees applied to $\phi_{Rotor-Command}$ at t = 0. Simulated response is shown at left and experimental system measurement is shown at right.

13. Inverted Pendulum Single PID Controller: Control of an Unstable Plant

The previous sections have described methods for introduction of Dual PID controller systems for control of the MIMO Edukit Rotary Inverted Pendulum requiring stabilization of both Pendulum Angle and Rotor Angle.

An additional design method for the Edukit Rotary Inverted Pendulum is based on the development of a Rotor Angle Controller with specified and static Controller configuration. Then, design of Pendulum Angle stabilizing controllers is directed to PID controller design including a specified Rotor Angle Transfer Function.

First, the conditions for stabilizing the unstable Inverted Pendulum plant are introduced.

A control system for stabilizing the Pendulum Angle, $\theta_{Pendulum}(s)$ in an inverted position relies on actuation of Rotor Angle, $\phi_{Rotor}(s)$. A controller of Pendulum Angle, $K_{Pend}(s)$, is introduced as shown in *Figure 50*.

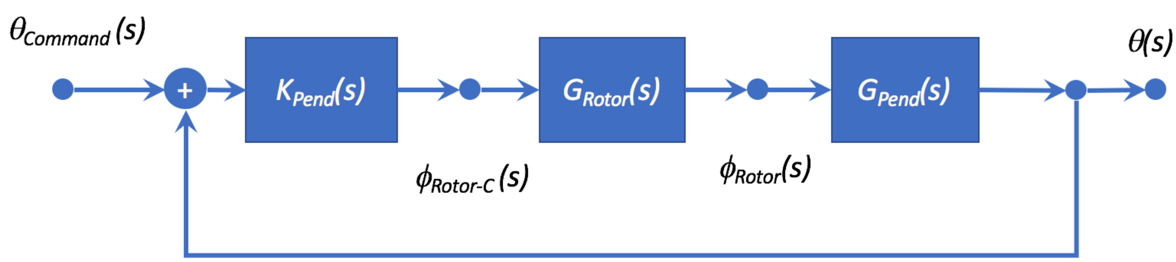


Figure 50.. The Rotary Inverted Pendulum feedback controller of Pendulum Angle.

Evaluating conditions for stability relies on the theory of Strongly Stabilizable systems by Youla. [Youla_1974]. This states that:

A plant is strongly stabilizable if and only if it has an even number of real poles between every pair of real zeros for $Re(s) \geq 0$ including zeroes at $s = \infty$.

Recalling the $G_{Rotor}(s)G_{Pendulum}(s)$ plant of Section 11, the poles of this plant are:

0
0
-7.6130
7.3362

Also, two zeroes exist for this plant at $s = 0$.

The $G_{Rotor}(s)G_{Pendulum}(s)$ plant does not satisfy the condition since first, there is one RHP pole in the region of $Re\ (s)\ \geq 0$ and the zero at $s = \infty$. (Please see the Bode plot of $G_{Rotor}(s)G_{Pendulum}(s)$ in *Figure 51*.

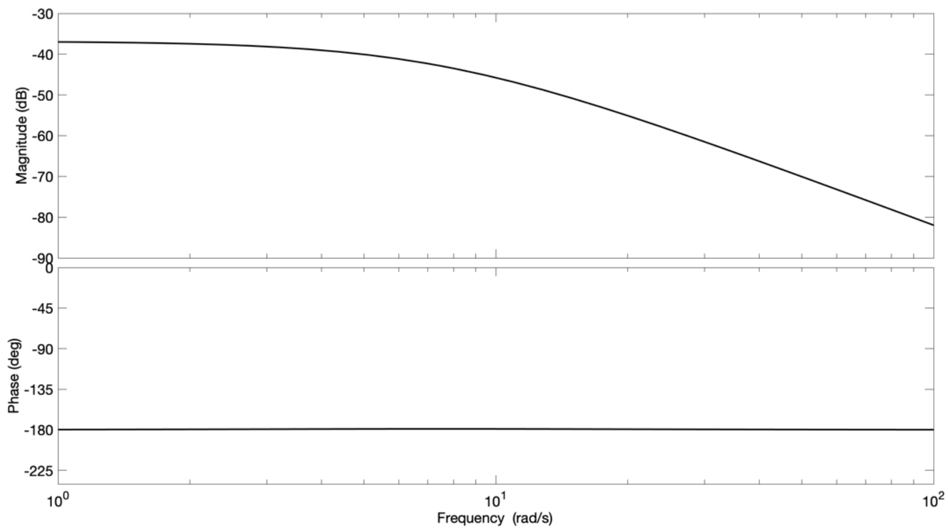


Figure 51. The Bode plot of Pendulum-Rotor plant, $G_{Rotor}(s)G_{Pendulum}(s)$*, is shown.*

One solution for achieving stability is to modify the system to include dual controllers forming Inner Loop and Outer Loop Controllers as shown in Figure 52. Here, $K_{pend}(s)$ of *Figure 50* forms the Outer Loop Controller. The addition of a second Inner Loop controller to the system architecture permits addition of a new RHP pole.

It is important to note that since the new Inner Loop controller introduces a RHP pole, it is *unstable.* However, its introduction, along with a stable Outer Loop controller will produce a stable, close loop system.

The Inner Loop operates on Rotor Angle and the Outer Loop on Pendulum Angle as shown in Figure 52.

The Inner Loop transfer function from Rotor Angle Control input to Rotor Angle, is

$$T_{Rotor}(s) = \frac{\phi_{Rotor}(s)}{\phi_{Rotor-C}(s)} = \left(\frac{G_{Rotor}(s)}{1 + K_{Rotor}(s)G_{Rotor}(s)}\right)$$

The transfer function, $T_{Rotor}(s)$, now replaces the actuator transfer function, $G_{Rotor}(s)$, to produce the new architecture of Figure 53.

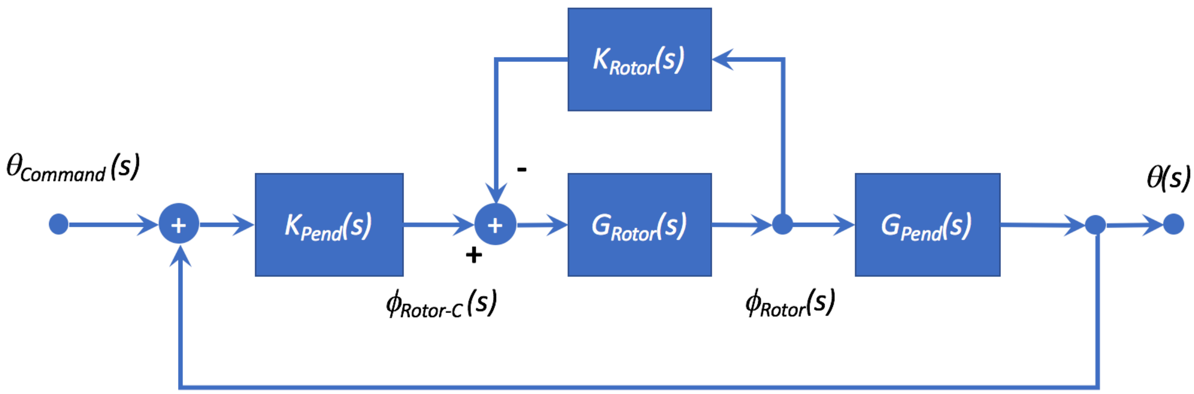


Figure 52. The Inner and Outer Loop control system introduced to create a Strongly Stabilizable system.

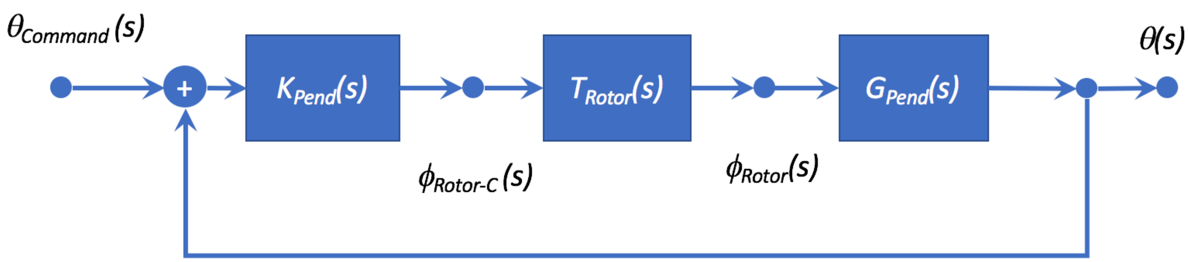


Figure 53. The Inner and Outer Loop Control system with the Inner Loop transfer function represented by $T_{Rotor}(s)$.

The transfer function, $T_{Rotor}(s)$ may be designed separately as described in Section 11. For illustration this is specified as:

$$T_{Rotor}(s) = \frac{\phi_{Rotor}(s)}{\phi_{Rotor-C}(s)} = \left(\frac{a_{Rotor}}{s^2 + b_{Rotor}s + c_{Rotor}}\right)$$

with a_{Rotor} = 0.977, b_{Rotor} = -7.33, c_{Rotor} = -14.66.

Poles of T_{Rotor} are:

8.9652
-1.6352

Inspection of T_{Rotor} shows that this introduces an unstable RHP pole at $s = 8.9652$.

Thus, in combination with $G_{Pendulum}(s)$, the plant of $G_{Pendulum}(s)T_{Rotor}(s)$ now includes these poles:

8.9652
7.3362
-7.6130
-1.6352

This system now satisfies the criteria for strongly stabilizable systems since there are now an even number even number of real poles between every pair of real zeros for $\text{Re}(s) \geq 0$ including zeroes at $s = \infty$.

Then, Frequency Response design methods may proceed following methods illustrated in Section 11.

A PD controller is selected of the form

$$K_{pend}(s) = K(1 + T_d s)$$

A design solution results in a gain, $K = 300$.

$$K_p = 300 \text{ and } K_d = KT_d = 30$$

The Bode plot for this controller, K_{pend}, the Pendulum and Rotor Plant, $G_{pendulum}T_{Rotor}$, and for the open loop combined plant and controller, $K_{pend}G_{pendulum}T_{Rotor}$, is shown in *Figure 54* for $K = 1$.

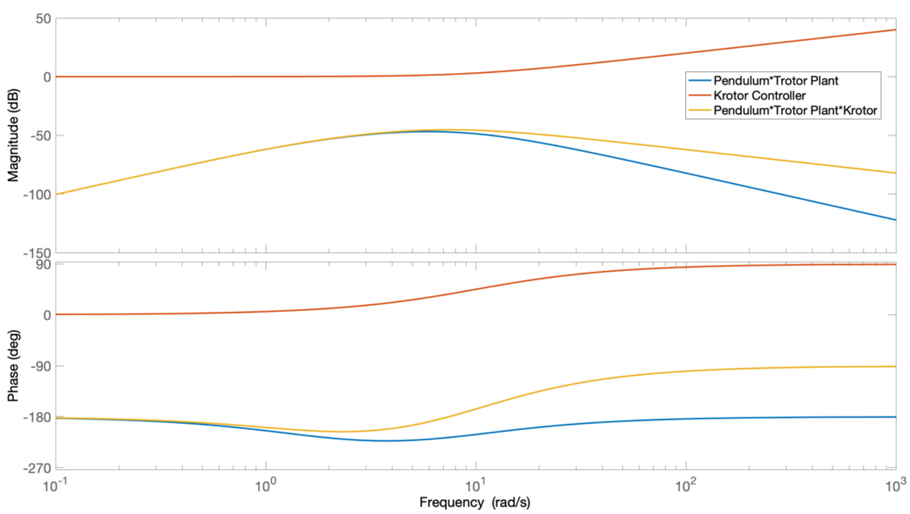


Figure 54. The Bode plot of the Pendulum and T_{Rotor} plant, Pendulum controller, and the open loop controller – plant product transfer function is shown for gain, K, of unity.

The poles appearing for the open loop transfer function, $T_{rotor}G_{pend}K_{pend}(s)$ are:

8.9652
7.3362
-7.6130
-1.6352

The closed loop poles, zeroes of $1 + T_{rotor}G_{pend}K_{pend}(s)$ are

8.3372
7.8815
-7.6056
-1.6383

An evaluation of stability is provided by the Bode plot of $K_{Pendulum}(s)G_{Pendulum}(s)T_{Rotor}(s)$ with the above gains as shown in *Figure 55*. The Phase Margin of 45.2 degrees meets the stability objective.

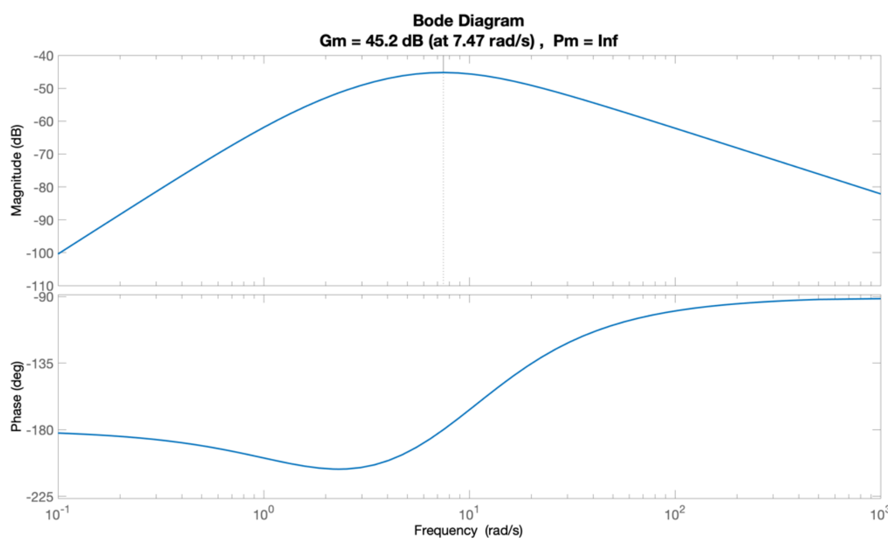


Figure 55. The Bode plot of the open loop control system including controller and plant product transfer function. The phase margin indicates stability with Phase Margin equaling design objectives.

The closed loop transfer function is

$$T_{Pend\_cl}(s) = \frac{\theta(s)}{\theta_{Command}(s)} = \left(\frac{K_{Pend}(s)T_{Rotor}(s)G_{Pend}(s)}{1 + K_{Pend}(s)T_{Rotor}(s)G_{Pend}(s)}\right)$$

This may now be applied to determine the closed loop response of the Pendulum Angle control system.

The Edukit control system also includes an impulse generation method. This introduces a pulse of amplitude, 10 degrees for a duration of 4 milliseconds in the Pendulum Angle Reference Tracking signal, $\theta_{reference}$.

The simulated impulse response is shown in *Figure 56*. This is a further confirmation of stability.

The experimentally measured closed loop response of the Pendulum Angle control system is shown in *Figure 57*.

Together, these results demonstrate the accurate agreement between simulated and experimental systems.

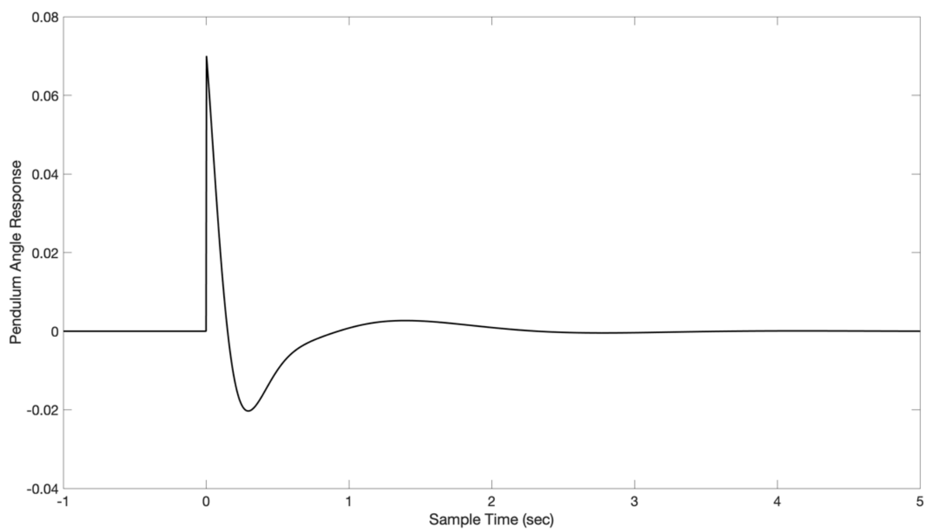


Figure 56. Simulated response of Pendulum Angle, $\theta(s)$, to an impulse signal applied to Pendulum angle tracking command, $\theta_{Command}(s)$, at t = 0.

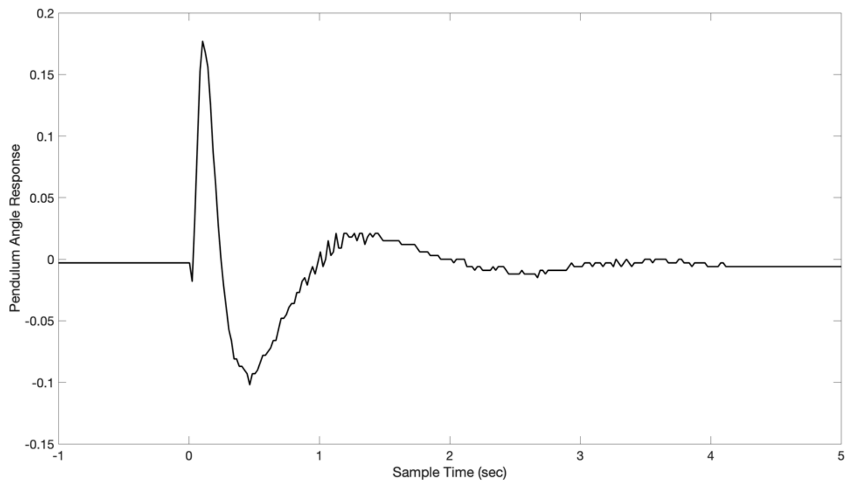


Figure 57. Experimental response of Pendulum Angle, $\theta(s)$, to an impulse signal applied to Pendulum angle tracking command, $\theta_{Command}(s)$, at t = 0.

14. State Space Modern Control: Linear Quadratic Regulator for Inverted Pendulum

The PID controller has been designed by frequency domain methods as well as by root locus techniques. However, these methods are not suited for design of the multi-input multi-output (MIMO) methods required for the rotary inverted pendulum where both pendulum angle and rotor angle require control. The design of a compensator and the placement of its closed loop poles for MIMO systems is challenging. Also, for MIMO systems, the controller systems that produce a set of closed loop poles are not unique. [Speyer_2010]

The Edukit Rotary Inverted Pendulum provides a method for directly comparing classical control, as in the previous section, and methods based on Modern Control described in this section. Modern Control introduces state space methods that enable description of MIMO systems. This also enable development of algorithms providing systematic methods for optimal design. This optimal design also permits optimization methods that permit weighting of response characteristics as well as control effort. This capability, in turn, allows for separate optimization of control system components including the Motor Controller.

A Full State Feedback system has been developed for the Edukit Rotary Inverted Pendulum with one input control signal for Rotor Angle Control and two outputs for Rotor Angle and Pendulum Angle. This is developed enabling Rotor Angle tracking of a command signal while maintaining Pendulum Angle at its nominal vertical, zero value.

The full state feedback control loop is shown in Figure 101. Computation of system dynamics follows.

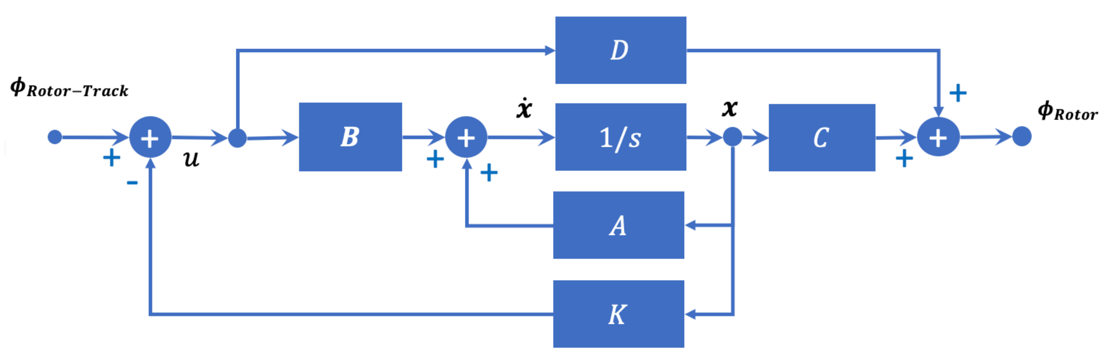


Figure 58. Full State Feedback Rotor Position Control loop with input control signal of $\phi_{Rotor-Track}$, and output angle, of ϕ_{Rotor}.

First, the Rotor angle response transfer function differential equation is:

$$\ddot{\phi} = -b\dot{\phi} - c\phi + a\boldsymbol{u}$$

where the input control signal, $\boldsymbol{u}$, is, $\phi_{Rotor-Command}$ and

$$a = 1.0$$
$$b = 0.0$$
$$c = 0.0$$

While the coefficients are determined as by system identification described in Appendix F: Rotor Actuator Plant Transfer Function, it is important to note that these may also be adjusted by plant modifications. Thus, the complete second order model is included in the development that follows.

Also, the transfer function between Rotor angle and Pendulum angle is defined by a differential equation:

$$\ddot{\theta} = e\ddot{\phi} - f\dot{\theta} + d\theta$$

where

$$e = 0.803(rad/sec)^2$$

$$d = 55.85(rad/sec)^2$$

$$f = \sqrt{55.85(rad/sec)^2}\Big/Q = 0.374\ (rad/sec)$$

Now, substituting for $\ddot{\phi}$ above, pendulum angle is related to input control signal, $\boldsymbol{u}$.

$$\ddot{\theta} = e\left(-b\dot{\phi} - c\phi + a\boldsymbol{u}\right) - f\dot{\theta} + d\theta$$

or

$$\ddot{\theta} = -be\dot{\phi} - ce\phi - f\dot{\theta} + d\theta + ae\boldsymbol{u}$$

Thus, the state-space form of this system of differential equations relates the Rotor and Pendulum angle state vector, $\boldsymbol{x}$, to control vector, $\boldsymbol{u}$.

First, the state vector $\boldsymbol{x}$ is defined

$$\boldsymbol{x} = \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

and

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

with

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix}$$

and the state matrix, $\boldsymbol{A}$

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -c & -b & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -ce & -be & d & -f \end{bmatrix}$$

with the input matrix, $\boldsymbol{B}$

$$B = \begin{bmatrix} 0 \\ a \\ 0 \\ ae \end{bmatrix}$$

then,

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -c & -b & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -ce & -be & d & -f \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ a \\ 0 \\ ae \end{bmatrix} \boldsymbol{u}$$

The system output vector, $\boldsymbol{y}$, including the Rotor Angle, ϕ, and Pendulum Angle, θ, is defined,

$$\boldsymbol{y} = \boldsymbol{Cx} + \boldsymbol{Du}$$

with the output matrix, $\boldsymbol{C}$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and the feedforward matrix, $\boldsymbol{D}$

$$\boldsymbol{D} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

then

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u = \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

The state feedback system is defined as

$$\boldsymbol{u} = \boldsymbol{Kx}$$

The Linear Quadratic Regulator (LQR) design method finds the optimal gain matrix, $\boldsymbol{K}$, that minimizes the cost function of output $\boldsymbol{u}$,

$$J(\boldsymbol{u}) = \int_{\boldsymbol{0}}^{\infty} (\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{Qx} + \boldsymbol{u}^{\boldsymbol{T}}\boldsymbol{Ru})dt$$

The matrix, determines the relative weights of state value contributions to the cost function. For the purpose of this design, the weights of Pendulum Angle, θ, and Rotor Angle, ϕ, are equal thus resulting in a control law that minimizes both with equal weight. The value of $\boldsymbol{R}$, determines the ratio of weight of control effort to control error. Increasing value of $\boldsymbol{R}$ increases the weight of control error at the expense of control effort.

The weight matrix, $\boldsymbol{Q}$, is selected with unit diagonal values, and $\boldsymbol{R}$ is set to 1. Thus,

$$\boldsymbol{Q} = \boldsymbol{C} * \boldsymbol{C}^{\boldsymbol{T}}$$

and

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Before proceeding, the controllability of the Edukit Rotary Inverted Pendulum system plant may be evaluated.

$\boldsymbol{A}$ is a 4 x 4 matrix while $\boldsymbol{B}$ is a 1 x 4 matrix. Then the controllability matrix, $\boldsymbol{M_C}$, is

$$\boldsymbol{M_C} = [\boldsymbol{B} \;\; \boldsymbol{AB} \;\; \boldsymbol{A}^2\boldsymbol{B} \;\; \boldsymbol{A}^3\boldsymbol{B}]$$

Applying the numerical values of a, b, e, and d, the controllability matrix, $\boldsymbol{M_C}$, is

$$M_C = [B \;\; AB \;\; A^2B \;\; A^3B] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0.803 & -0.3003 & 44.7352 \\ 0.803 & -0.3003 & 44.7352 & -33.5040 \end{bmatrix}$$

Computation of $\boldsymbol{M_C}$ above shows that the full row rank of $\boldsymbol{M_C}$ is 4 and that the system is controllable.

The computation of the gain vector, ***K***, by LQR methods produces

$$K_\theta = 174.82; \quad K_{\dot{\theta}} = 22.84; \;\; K_\phi = 1.00; \;\; K_{\dot{\phi}} = 2.01$$

Step response of the Edukit system was characterized by application of a step command of amplitude Rotor Angle Command input. The simulated response and experimental response of Rotor Angle to this step command is shown in *Figure 59*. The simulated response and experimental response of Pendulum Angle is shown in *Figure 60*.

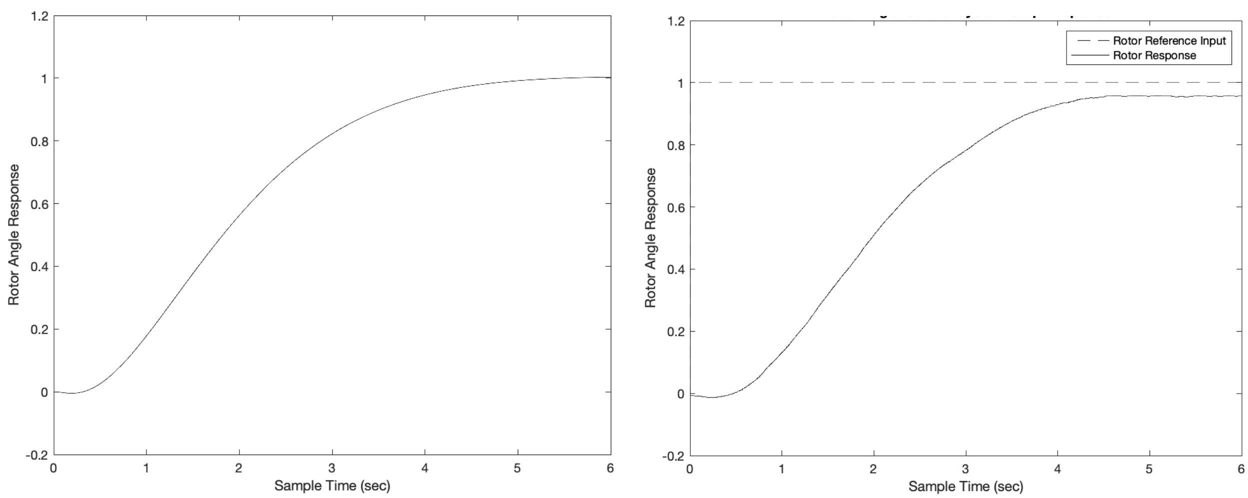


Figure 59. Simulated step response of Rotor Angle (left) and experimentally measured step response (right) for the Edukit Rotary Inverted Pendulum with Full State Feedback Controller for a unit amplitude step in Rotor Angle Control step signal applied at t = 0s.

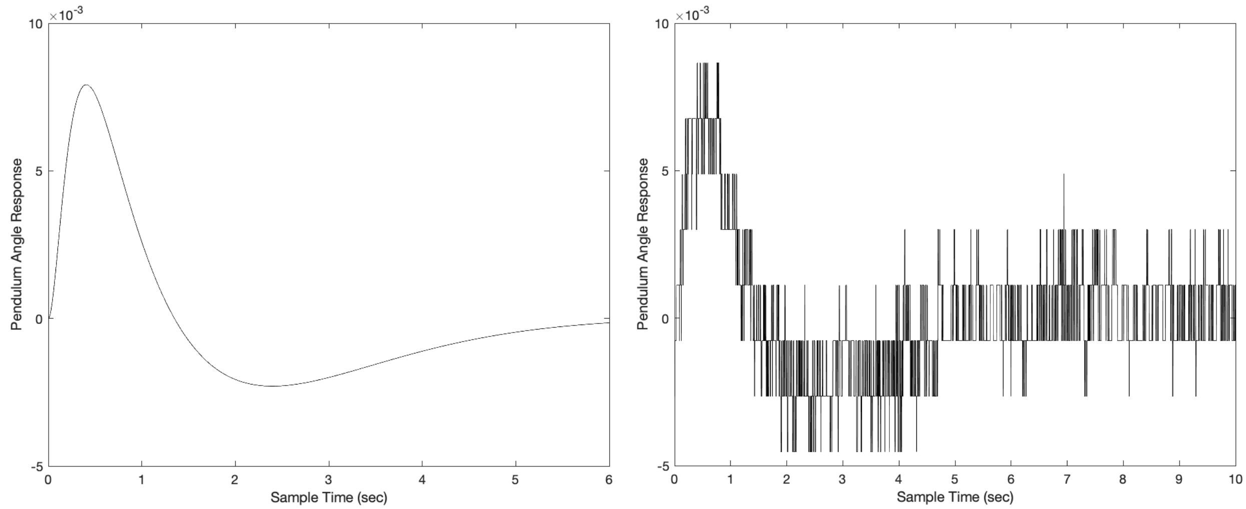


Figure 60. Simulated response of Pendulum Angle (left) and experimentally measured response of Pendulum Angle (right) for the Edukit Rotary Inverted Pendulum with Full State Feedback Controller for a unit amplitude step in Rotor Angle Control step signal applied at t = 0s. Note that the quantization of angle steps 0f 0.16 degrees is visible.

The Full State Feedback Controller may also be extended with integral action applied to the error difference between Rotor Track command and Rotor Angle as shown in Figure 61.

The state vector defined above is augmented with a third state, the integral of the difference between the state output and the reference track command:

$$\dot{\boldsymbol{x}}_{\boldsymbol{I}} = \boldsymbol{r} - \boldsymbol{C}\boldsymbol{x}$$

where $\boldsymbol{r} = \ \phi_{Rotor-Track}$

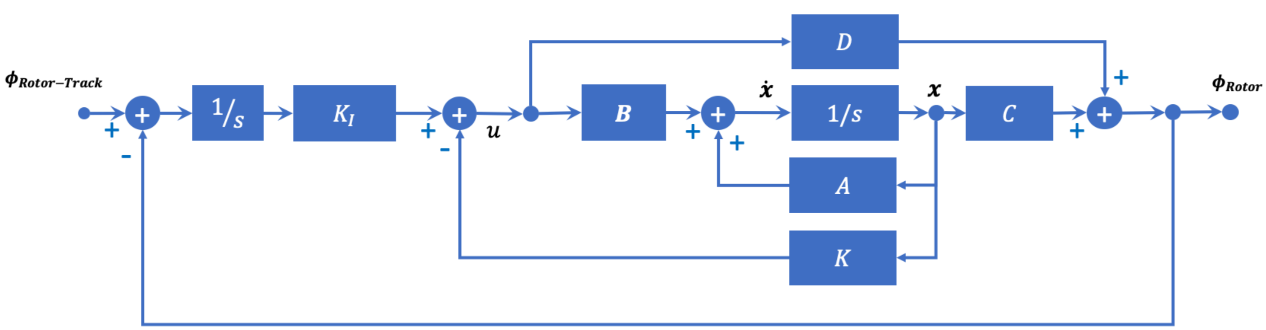


Figure 61. Full State Feedback Rotor Position Control loop including integral action with input control signal of $\phi_{Rotor-Track}$, and output angle, of ϕ_{Rotor}.

Then, the state dynamics becomes

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{x}}_{\boldsymbol{I}} \end{bmatrix} = \boldsymbol{A_I x} + \boldsymbol{B_I u} + \boldsymbol{B_R r}$$

With

$$\boldsymbol{A_I} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -c & -b & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -ce & -be & d & -f & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

with the input matrix, $\boldsymbol{B}$

$$\boldsymbol{B_I} = \begin{bmatrix} 0 \\ a \\ 0 \\ ae \\ 0 \end{bmatrix}$$

And

$$\boldsymbol{B_R} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

then,

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\boldsymbol{x}}_{\boldsymbol{\iota}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -c & -b & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -ce & -be & d & -f & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \boldsymbol{x}_{\boldsymbol{i}} \end{bmatrix} + \begin{bmatrix} 0 \\ a \\ 0 \\ ae \\ 0 \end{bmatrix} \boldsymbol{u} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} r$$

The system output vector, $\boldsymbol{y}$, including the Rotor Angle, ϕ, and Pendulum Angle, θ, is defined,

$$\boldsymbol{y} = \boldsymbol{Cx} + \boldsymbol{Du}$$

with the output matrix, $\boldsymbol{C}$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and the feedforward matrix, $\boldsymbol{D}$

$$\boldsymbol{D} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

then

$$\boldsymbol{y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \boldsymbol{x_i} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \boldsymbol{u}$$

The state feedback system is defined as

$$\boldsymbol{u} = [\boldsymbol{K} \quad \boldsymbol{K_I}]\boldsymbol{x}$$

The Linear Quadratic Regulator (LQR) design method finds the optimal gain matrix, $\boldsymbol{K}$, that minimizes the cost function of output $\boldsymbol{u}$,

$$J(\boldsymbol{u}) = \int_{\boldsymbol{0}}^{\infty} (\boldsymbol{x^T Q x} + \boldsymbol{u^T R u})dt$$

The matrix, determines the relative weights of state value contributions to the cost function. For the purpose of this design, the weights of Pendulum Angle, θ, and Rotor Angle, ϕ, are equal thus resulting in a control law that minimizes both with equal weight. The value of $\boldsymbol{R}$, determines the ratio of weight of control effort to control error. Increasing value of $\boldsymbol{R}$ increases the weight of control error at the expense of control effort.

The weight matrix, $\boldsymbol{Q}$, is selected with unit diagonal values, and $\boldsymbol{R}$ is set to 1. Thus,

$$\boldsymbol{Q} = \boldsymbol{C} * \boldsymbol{C^T}$$

and

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Before proceeding, the controllability of the Edukit Rotary Inverted Pendulum system plant may be evaluated.

$\boldsymbol{A_I}$ is a 5 x 5 matrix while $\boldsymbol{B_I}$ is a 1 x 5 matrix. Then the controllability matrix, $\boldsymbol{M_C}$, is

$$\boldsymbol{M_C} = [\boldsymbol{B}\ \ \boldsymbol{AB}\ \ \boldsymbol{A^2B}\ \ \boldsymbol{A^3B}\ \boldsymbol{A^4B}]$$

Applying the numerical values of a, b, e, and d, the controllability matrix, $\boldsymbol{M_C}$, is

$$\boldsymbol{M_C} = [\boldsymbol{B}\ \ \boldsymbol{AB}\ \ \boldsymbol{A^2B}\ \ \boldsymbol{A^3B}] = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0.803 & -0.3003 & 44.7352 & -33.5040 \\ 0.803 & -0.3003 & 44.7352 & -33.5040 & 2485.9 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Computation of $\boldsymbol{M_C}$ above shows that the full row rank of $\boldsymbol{M_C}$ is 5 and that the system is controllable.

The computation of the gain vector, $\boldsymbol{K}$, by LQR methods produces

$$K_\theta = 191.92;\quad K_{\dot\theta} = 25.07;\ K_\phi = 2.69;\ K_{\dot\phi} = 3.12;\ K_I = 1.00$$

Step response of the Edukit system was characterized by application of a step command of amplitude Rotor Angle Command input. The simulated response and experimental response of Rotor Angle to this step command is shown in *Figure 62*. The simulated response and experimental response of Pendulum Angle is shown in *Figure 63*.

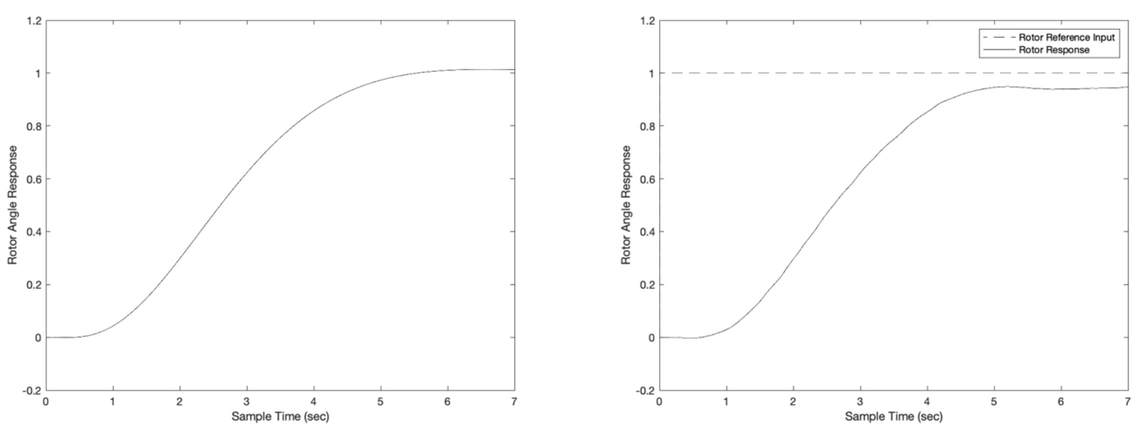


Figure 62. Simulated response of Rotor Angle for the Edukit Rotary Inverted Pendulum with Full State Feedback Controller with Integral Action for a unit amplitude step in Rotor Angle Control step signal applied at t = 0s

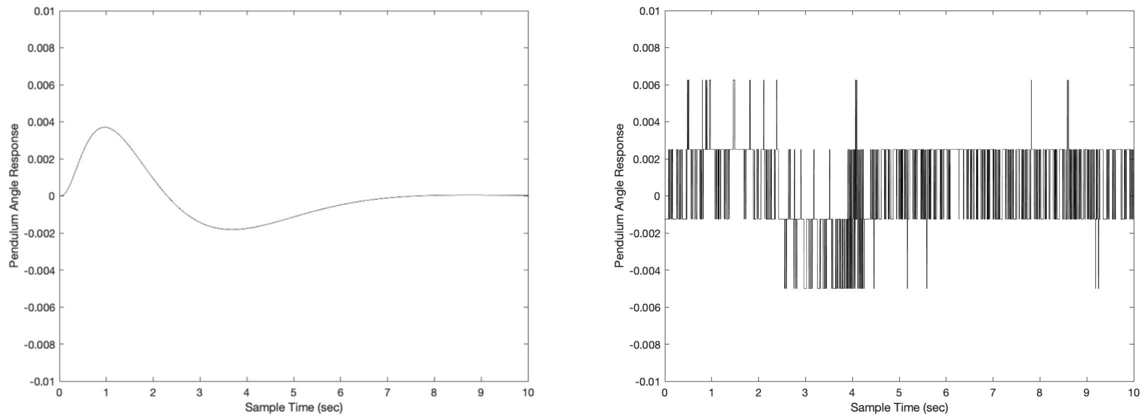


Figure 63. Simulated response of Pendulum Angle for the Edukit Rotary Inverted Pendulum with Full State Feedback Controller with Integral Action for a unit amplitude step in Rotor Angle Control step signal applied at t = 0s. Note that the quantization of angle steps 0f 0.16 degrees is visible.

15. State Space Modern Control: Linear Quadratic Regulator for Stable Suspended Pendulum

The PID controller has been designed by frequency domain methods as well as by root locus techniques. However, these methods are not suited for design of the multi-input multi-output (MIMO) methods required for the rotary inverted pendulum where both pendulum angle and rotor angle require control. The design of a compensator and the placement of its closed loop poles for MIMO systems is challenging. Also, for MIMO systems, the controller systems that produce a set of closed loop poles are not unique. [Speyer_2010]

The Edukit Rotary Inverted Pendulum provides a method for directly comparing classical control, as in the previous section, and methods based on Modern Control described in this section. Modern Control introduces state space methods that enable description of MIMO systems. This also enable development of algorithms providing systematic methods for optimal design. This optimal design also permits optimization methods that permit weighting of response characteristics as well as control effort. This capability, in turn, allows for separate optimization of control system components including the Motor Controller.

A Full State Feedback system has been developed for the Edukit Rotary Inverted Pendulum with one input control signal for Rotor Angle Control and two outputs for Rotor Angle and Pendulum Angle. This is developed enabling Rotor Angle tracking of a command signal while maintaining Pendulum Angle at its nominal vertical, zero value.

The full state feedback control loop is shown in Figure 101. Computation of system dynamics follows.

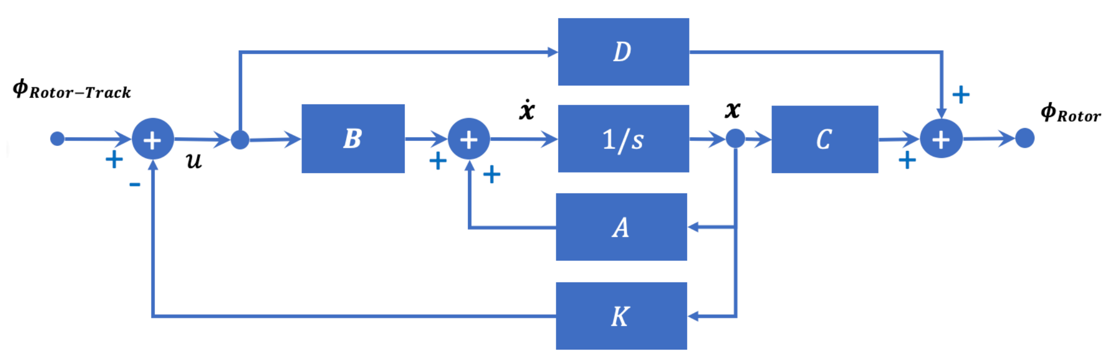


Figure 64. Full State Feedback Rotor Position Control loop with input control signal of $\phi_{Rotor-Track}$, and output angle, of ϕ_{Rotor}.

First, the Rotor angle response transfer function differential equation is:

$$\ddot{\phi} = -b\dot{\phi} - c\phi + a\boldsymbol{u}$$

where the input control signal, $\boldsymbol{u}$, is, $\phi_{Rotor-Command}$ and

$$a = 1.0$$
$$b = 0.0$$
$$c = 0.0$$

While the coefficients are determined as by system identification described in Appendix F: Rotor Actuator Plant Transfer Function, it is important to note that these may also be adjusted by plant modifications. Thus, the complete second order model is included in the development that follows.

Also, the transfer function between Rotor angle and Pendulum angle is defined by a differential equation:

$$\ddot{\theta} = e\ddot{\phi} - f\dot{\theta} - d\theta$$

where

$$e = 0.803(rad/sec)^2$$

$$d = 55.85(rad/sec)^2$$

$$f = \sqrt{55.85(rad/sec)^2}\Big/Q = 0.374\ (rad/sec)$$

Now, substituting for $\ddot{\phi}$ above, pendulum angle is related to input control signal, $\boldsymbol{u}$.

$$\ddot{\theta} = e(-b\dot{\phi} - c\phi + a\boldsymbol{u}) - f\dot{\theta} - d\theta$$

or

$$\ddot{\theta} = -be\dot{\phi} - ce\phi - f\dot{\theta} - d\theta + ae\boldsymbol{u}$$

Thus, the state-space form of this system of differential equations relates the Rotor and Pendulum angle state vector, $\boldsymbol{x}$, to control vector, $\boldsymbol{u}$.

First, the state vector $\boldsymbol{x}$ is defined

$$\boldsymbol{x} = \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

and

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

with

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix}$$

and the state matrix, ***A***

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -c & -b & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -ce & -be & -d & -f \end{bmatrix}$$

with the input matrix, ***B***

$$B = \begin{bmatrix} 0 \\ a \\ 0 \\ ae \end{bmatrix}$$

then,

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -c & -b & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -ce & -be & -d & -f \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ a \\ 0 \\ ae \end{bmatrix} \boldsymbol{u}$$

The system output vector, $\boldsymbol{y}$, including the Rotor Angle, ϕ, and Pendulum Angle, θ**,** is defined,

$$\boldsymbol{y} = \boldsymbol{Cx} + \boldsymbol{Du}$$

with the output matrix, ***C***

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and the feedforward matrix, ***D***

$$\boldsymbol{D} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

then

$$\boldsymbol{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \boldsymbol{u} = \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

The state feedback system is defined as

$$\boldsymbol{u} = \boldsymbol{Kx}$$

The Linear Quadratic Regulator (LQR) design method finds the optimal gain matrix, $\boldsymbol{K}$, that minimizes the cost function of output $\boldsymbol{u}$*,*

$$J(\boldsymbol{u}) = \int_{\boldsymbol{0}}^{\infty} (\boldsymbol{x}^{\boldsymbol{T}} \boldsymbol{Qx} + \boldsymbol{u}^{\boldsymbol{T}} \boldsymbol{Ru}) dt$$

The matrix, determines the relative weights of state value contributions to the cost function. For the purpose of this design, the weights of Pendulum Angle, θ, and Rotor Angle, ϕ, are equal thus resulting in a control law that minimizes both with equal weight. The value of $\boldsymbol{R}$, determines the ratio of weight of control effort to control error. Increasing value of $\boldsymbol{R}$ increases the weight of control error at the expense of control effort.

The weight matrix, $\boldsymbol{Q}$, is selected with unit diagonal values, and $\boldsymbol{R}$ is set to 1. Thus,

$$\boldsymbol{Q} = \boldsymbol{C} * \boldsymbol{C}^{\boldsymbol{T}}$$

and

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Before proceeding, the controllability of the Edukit Rotary Inverted Pendulum system plant may be evaluated.

$\boldsymbol{A}$ is a 4 x 4 matrix while $\boldsymbol{B}$ is a 1 x 4 matrix. Then the controllability matrix, $\boldsymbol{M_C}$, is

$$\boldsymbol{M_C} = [\boldsymbol{B} \ \ \boldsymbol{AB} \ \ \boldsymbol{A^2B} \ \ \boldsymbol{A^3B}]$$

Applying the numerical values of a, b, e, and d, the controllability matrix, $\boldsymbol{M_C}$, is

$$\boldsymbol{M_C} = [\boldsymbol{B} \ \ \boldsymbol{AB} \ \ \boldsymbol{A^2B} \ \ \boldsymbol{A^3B}] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0.803 & -0.3003 & -44.7352 \\ 0.803 & -0.3003 & -44.7352 & 33.5040 \end{bmatrix}$$

Computation of $\boldsymbol{M_C}$ above shows that the full row rank of $\boldsymbol{M_C}$ is 4 and that the system is controllable.

The computation of the gain vector, $\boldsymbol{K}$, by LQR methods produces

$$K_\theta = 1.10; \quad K_{\dot{\theta}} = 0.63; \ \ K_\phi = 1.00; \ K_{\dot{\phi}} = 1.74$$

Step response of the Edukit system was characterized by application of a step command of amplitude Rotor Angle Command input. The simulated response and experimental response of Rotor Angle to this step command is shown in *Figure 65*. The simulated response and experimental response of Pendulum Angle is shown in *Figure 66*.

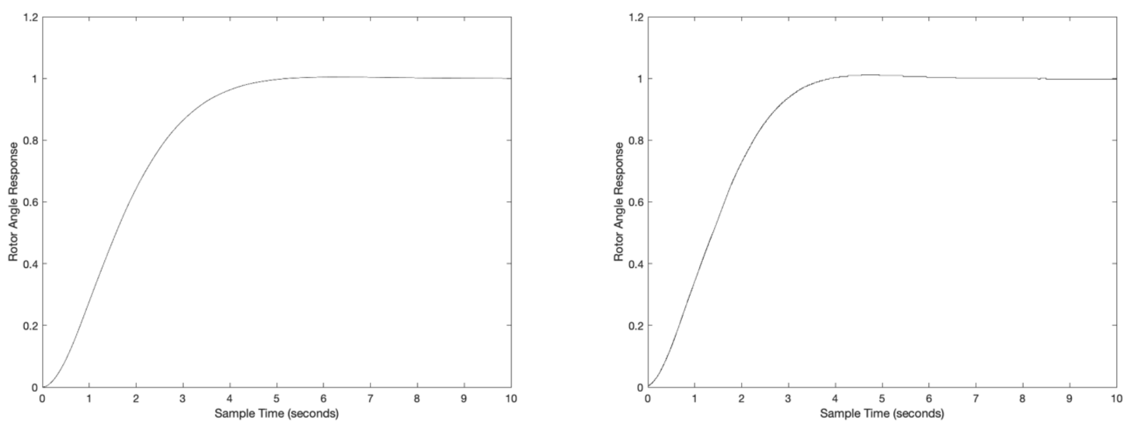


Figure 65. Suspended Pendulum Mode Simulated step response of Rotor Angle (left) and experimentally measured step response (right) for the Edukit Rotary Inverted Pendulum with Full State Feedback Controller for a unit amplitude step in Rotor Angle Control step signal applied at t = 0s.

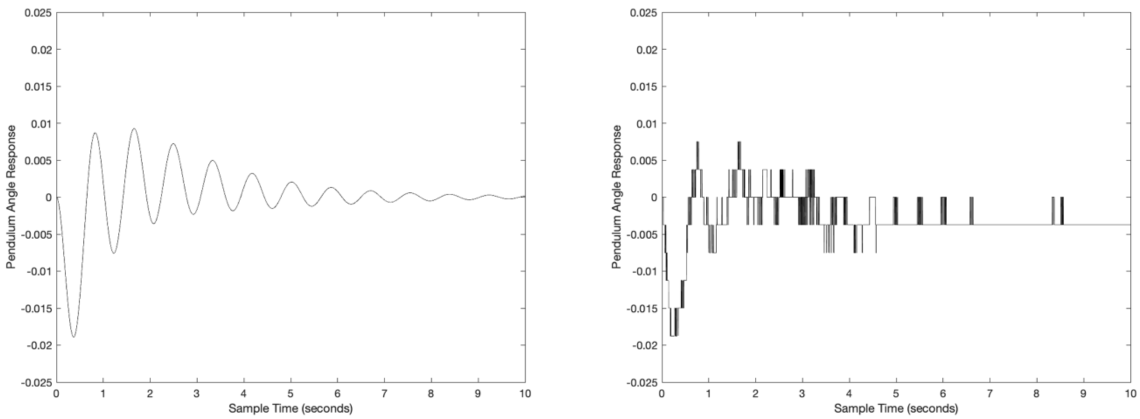


Figure 66. Suspended Pendulum Mode Simulated response of Pendulum Angle (left) and experimentally measured response of Pendulum Angle (right) for the Edukit Rotary Inverted Pendulum with Full State Feedback Controller for a unit amplitude step in Rotor Angle Control step signal applied at t = 0s. Note that the quantization of angle steps 0f 0.16 degrees is visible.

The Full State Feedback Controller may also be extended with integral action applied to the error difference between Rotor Track command and Rotor Angle as shown in Figure 67.

The state vector defined above is augmented with a third state, the integral of the difference between the state output and the reference track command:

$$\dot{\boldsymbol{x}}_{\boldsymbol{I}} = \boldsymbol{r} - \boldsymbol{Cx}$$

where $\boldsymbol{r} = \phi_{Rotor-Track}$

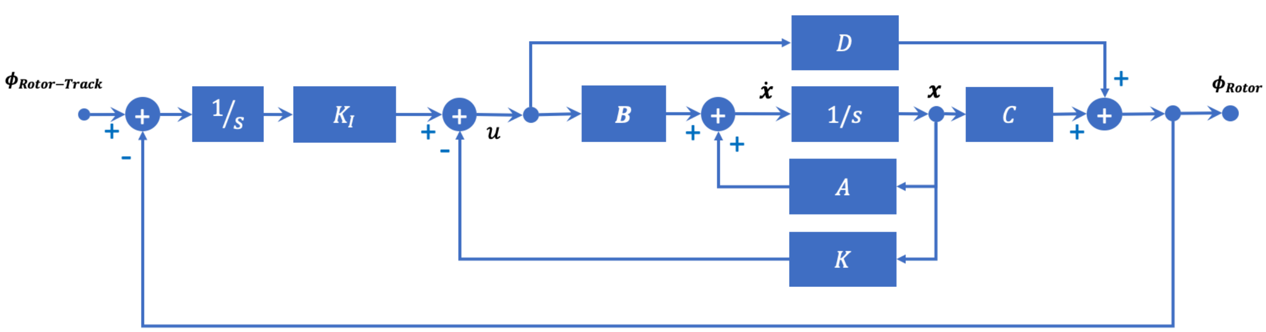


Figure 67. Full State Feedback Rotor Position Control loop including integral action with input control signal of $\phi_{Rotor-Track}$, and output angle, of ϕ_{Rotor}.

Then, the state dynamics becomes

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{x}}_{\boldsymbol{I}} \end{bmatrix} = \boldsymbol{A}_{\boldsymbol{I}}\boldsymbol{x} + \boldsymbol{B}_{\boldsymbol{I}}\boldsymbol{u} + \boldsymbol{B}_{\boldsymbol{R}}\boldsymbol{r}$$

With

$$\boldsymbol{A_I} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -c & -b & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -ce & -be & -d & -f & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

with the input matrix, ***B***

$$\boldsymbol{B_I} = \begin{bmatrix} 0 \\ a \\ 0 \\ ae \\ 0 \end{bmatrix}$$

And

$$\boldsymbol{B_R} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

then,

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\boldsymbol{x}}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -c & -b & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -ce & -be & -d & -f & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \boldsymbol{x_i} \end{bmatrix} + \begin{bmatrix} 0 \\ a \\ 0 \\ ae \\ 0 \end{bmatrix} \boldsymbol{u} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} r$$

The system output vector, $\boldsymbol{y}$, including the Rotor Angle, ϕ, and Pendulum Angle, θ, is defined,

$$\boldsymbol{y} = \boldsymbol{Cx} + \boldsymbol{Du}$$

with the output matrix, ***C***

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and the feedforward matrix, ***D***

$$\boldsymbol{D} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

then

$$\boldsymbol{y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \boldsymbol{x_i} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \boldsymbol{u}$$

The state feedback system is defined as

$$\boldsymbol{u} = [\boldsymbol{K} \quad \boldsymbol{K_I}]\boldsymbol{x}$$

The Linear Quadratic Regulator (LQR) design method finds the optimal gain matrix, $\boldsymbol{K}$, that minimizes the cost function of output $\boldsymbol{u}$*,*

$$J(\boldsymbol{u}) = \int_{\boldsymbol{0}}^{\infty} (\boldsymbol{x^T Q x} + \boldsymbol{u^T R u})dt$$

The matrix, determines the relative weights of state value contributions to the cost function. For the purpose of this design, the weights of Pendulum Angle, θ, and Rotor Angle, ϕ, are equal thus resulting in a control law that minimizes both with equal weight. The value of $\boldsymbol{R}$, determines the ratio of weight of control effort to control error. Increasing value of $\boldsymbol{R}$ increases the weight of control error at the expense of control effort.

The weight matrix, $\boldsymbol{Q}$, is selected with unit diagonal values, and $\boldsymbol{R}$ is set to 1. Thus,

$$\boldsymbol{Q} = \boldsymbol{C} * \boldsymbol{C^T}$$

and

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Before proceeding, the controllability of the Edukit Rotary Inverted Pendulum system plant may be evaluated.

$\boldsymbol{A_I}$ is a 5 x 5 matrix while $\boldsymbol{B_I}$ is a 1 x 5 matrix. Then the controllability matrix, $\boldsymbol{M_C}$, is

$$\boldsymbol{M_C} = [\boldsymbol{B}\ \ \boldsymbol{AB}\ \ \boldsymbol{A^2B}\ \ \boldsymbol{A^3B}\ \boldsymbol{A^4B}]$$

Applying the numerical values of a, b, e, and d, the controllability matrix, $\boldsymbol{M_C}$, is

$$\boldsymbol{M_C} = [\boldsymbol{B}\ \ \boldsymbol{AB}\ \ \boldsymbol{A^2B}\ \ \boldsymbol{A^3B}] = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0.803 & -0.3003 & -44.7352 & 33.5040 \\ 0.803 & -0.3003 & -44.7352 & 33.5040 & 2485.9 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Computation of $\boldsymbol{M_C}$ above shows that the full row rank of $\boldsymbol{M_C}$ is 5 and that the system is controllable.

The computation of the gain vector, ***K****,* by LQR methods produces

$$K_\theta = 1.52; \quad K_{\dot{\theta}} = 0.61;\ \ K_\phi = 2.42;\ K_{\dot{\phi}} = 2.44;\ K_I = 1.00$$

Step response of the Edukit system was characterized by application of a step command of amplitude Rotor Angle Command input. The simulated response and experimental response of Rotor Angle to this step command is shown in *Figure 68*. The simulated response and experimental response of Pendulum Angle is shown in *Figure 69*.

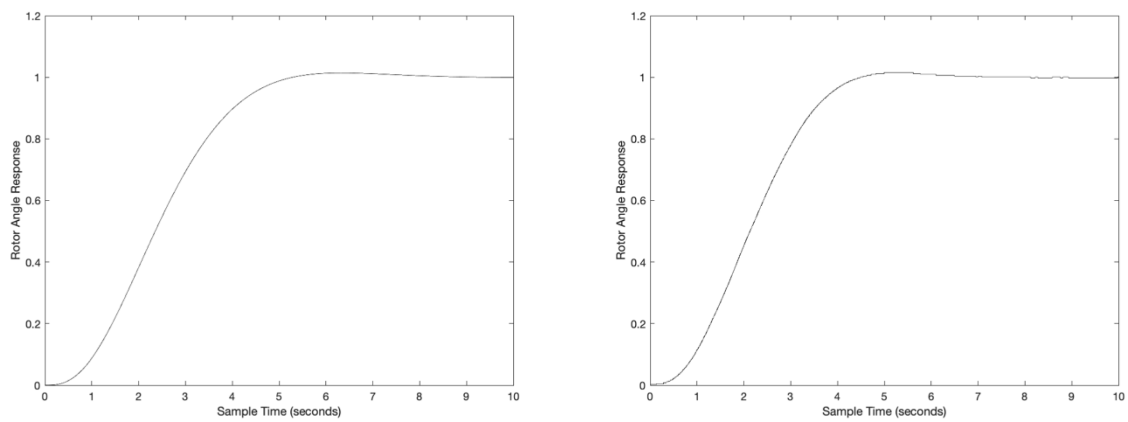


Figure 68. Suspended Pendulum Mode Simulated step response of Rotor Angle (left) and experimentally measured step response (right) for the Edukit Rotary Inverted Pendulum with Full State Feedback Controller for a unit amplitude step in Rotor Angle Control step signal applied at t = 0s.

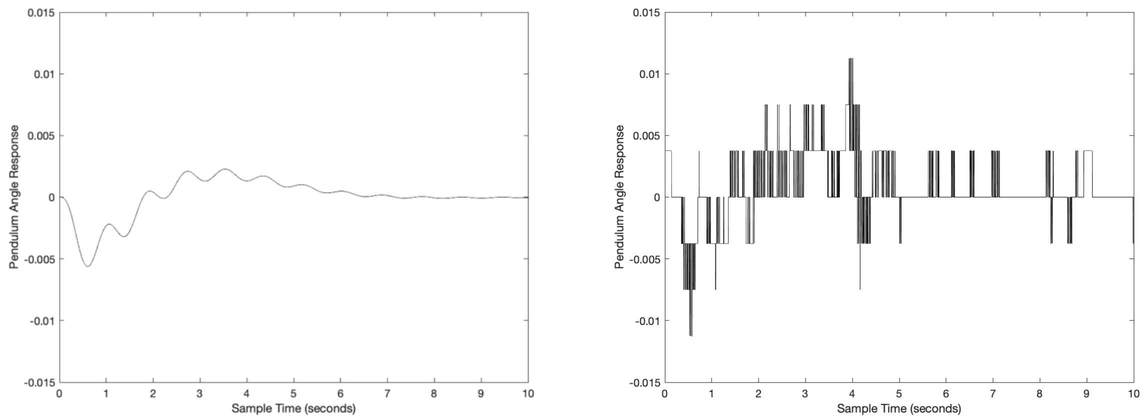


Figure 69. Suspended Pendulum Mode Simulated response of Pendulum Angle (left) and experimentally measured response of Pendulum Angle (right) for the Edukit Rotary Inverted Pendulum with Full State Feedback Controller for a unit amplitude step in Rotor Angle Control step signal applied at t = 0s. Note that the quantization of angle steps 0f 0.16 degrees is visible.

16. Edukit Programmable Rotor Plant Transfer Function

The Edukit system rotor actuator is based on the precise Stepper Motor Controller Interface. [ST_IHM01A1_2019] This, including the Rotor Actuator Plant with Acceleration Control described in Appendix E: Rotor Actuator Plant with Acceleration Control enables and accurate specification of rotor response. The digiral control algorithm ensures that Rotor Angle response to Rotor Control Input is that of a double integrator with unity gain. This is verified in the system identification desscribd in Appendix F: Rotor Actuator Plant Transfer Function.

The Rotor Actuator transfer function, $G_{Rotor}(s)$ is a double integrator of the form

$$G_{Rotor}(s) = \frac{1}{s^2}$$

This system permits a wide range of high performance control system designs and design challenges for instruction.

This section below, describes the implementation of the Programmable Rotor Plant and examples of its operation.

The Edukit digital Rotor Actuator system enables *programmability* of the Rotor Actuator Plant. The Rotor actuator indicated in *Figure 70 (a)* may be augmented with a compensator H_{Rotor} as shown in *Figure 70 (b)* to produce a new Rotor Actuator Plant, $P_{Rotor}(s)$.

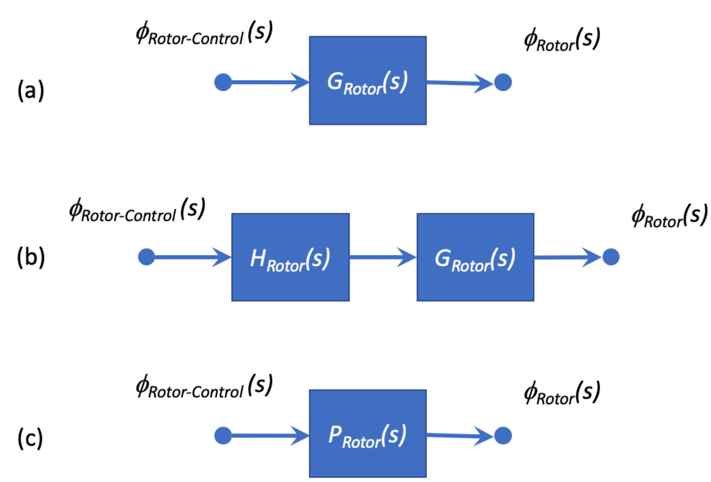


Figure 70. The Rotor Control Transfer function, $G_{Rotor}(s)$, indicated in (a), may be augmented with the compensator, H_{Rotor}, shown in (b) to produce the Rotor Plant, $P_{Rotor}(s)$, shown in (c).

The Edukit system provides a user interface for the optional specification of this plant. Without specification, the double integrator with unity gain is selected.

The user can also specify one of two Rotor Plant designs.

Rotor Plant Design for All Control Modes

A specification of $P_{Rotor}(s)$ may be made that is applicable to all operating modes. This introduces the H_{Rotor} compensator in the form of an IIR high pass filter. Thus, $P_{Rotor}(s)$ takes the form

$$P_{Rotor}(s) = H_{Rotor}(s)G_{Rotor}(s) = \frac{1}{s^2 + \omega_N s}$$

The compensator introduced has the form

$$H_{Rotor}(s) = \frac{s}{s + \omega_N}$$

The Edukit digital control system applies $H_{Rotor}(s)$ in series with $G_{Rotor}(s)$ to yield $P_{Rotor}(s)$.

The Edukit control system introduces a digital IIR filter constructed to create $H_{Rotor}(s)$. Operating limits are shown in *Table 11*.

Inverted and Suspended Mode Operation		
Control Mode	All modes	
Natural Frequency	Lower Limit: 0.0 rad/sec	Upper Limit: 100 rad/sec

Table 11. Programmable Rotor Plant configuration limits for $P_{Rotor}(s) = \frac{1}{s^2+\omega_N s}$. *The Programmable Plant supports operation in Full State Feedback for both Suspended Mode and Inverted Mode operation.*

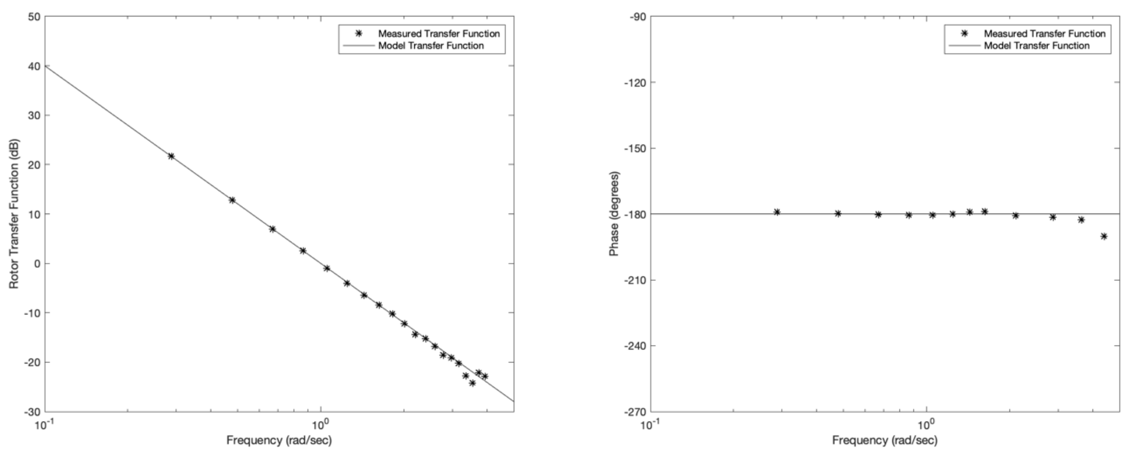


Figure 71. Comparison of model-predicted Rotor Angle Transfer Function Magnitude (left) and Phase (right) with measured Rotor Angle Transfer Function under Full State feedback control with Rotor Transfer Function computed by the Rotor System Identification system. This Transfer Function corresponds to the Rotor Plant without Rotor Plant Design applied and thus with a Transfer Function of $G_{Rotor}(s) = 1/s^2$.

Examples of operation of the Rotor Plant Design system are shown in *Figure 71*, *Figure 72*, *Figure 73* and *Figure 74* for $\omega_N = 0$, $\omega_N = 0{,}1$, $\omega_N = 1$, and $\omega_N = 4$, respectively.

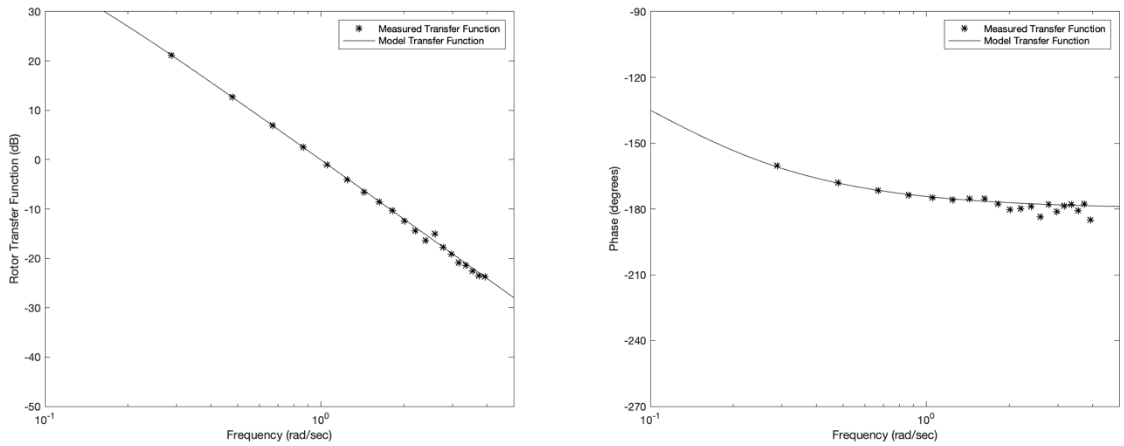


Figure 72. Comparison of model-predicted Rotor Angle Transfer Function Magnitude (left) and Phase (right) with measured Rotor Angle Transfer Function under Full State feedback control with Rotor Transfer Function computed by the Rotor System Identification system. This Transfer Function corresponds to the Rotor Plant without Rotor Plant Design applied and thus with a Transfer Function of $G_{Rotor}(s) = 1/(s^2 + 0.1 * s)$.

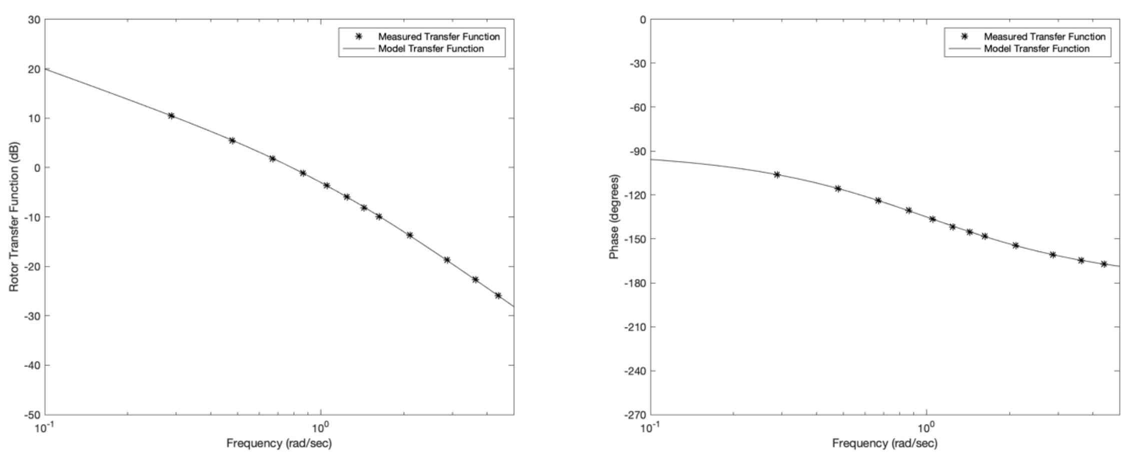


Figure 73. Comparison of model-predicted Rotor Angle Transfer Function Magnitude (left) and Phase (right) with measured Rotor Angle Transfer Function under Full State feedback control with Rotor Transfer Function computed by the Rotor System Identification system. This Transfer Function corresponds to the Rotor Plant without Rotor Plant Design applied and thus with a Transfer Function of $G_{Rotor}(s) = 1/(s^2 + s)$.

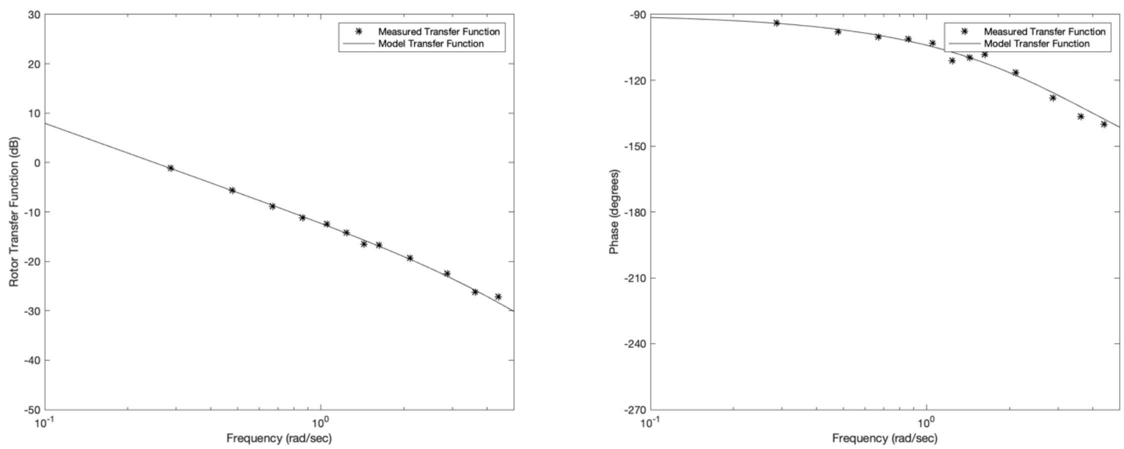


Figure 74. Comparison of model-predicted Rotor Angle Transfer Function Magnitude (left) and Phase (right) with measured Rotor Angle Transfer Function under Full State feedback control with Rotor Transfer Function computed by the Rotor System Identification system. This Transfer Function corresponds to the Rotor Plant without Rotor Plant Design applied and thus with a Transfer Function of $G_{Rotor}(s) = 1/(s^2 + 4s)$.

The effect of the Pendulum Rotor Plant Poles for Suspended Mode are shown in for each of the values of ω_N for Suspended Mode in *Table 12* and for Inverted Mode in *Table 13*.

Rotor Plant Transfer Function	$1/(s^2)$	$1/(s^2 + 0.1s)$	$1/(s^2 + s)$	$1/(s^2 + 4s)$
Pendulum Rotor - Plant Poles	0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i -0.1384 + 7.4720i -0.1384 - 7.4720i	0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i -0.1384 + 7.4720i -0.1384 - 7.4720i -0.1000 + 0.0000i	0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i -0.1384 + 7.4720i -0.1384 - 7.4720i -1.0000 + 0.0000i	0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i -0.1384 + 7.4720i -0.1384 - 7.4720i -4.0000 + 0.0000i

Table 12. Poles for the Pendumum - Rotor Plant for Suspended Mode for values of ω_N.

Rotor Plant Transfer Function	$1/(s^2)$	$1/(s^2 + 0.1s)$	$1/(s^2 + s)$	$1/(s^2 + 4s)$
Pendulum Rotor - Plant Zeros	0 0 0 0 -7.6130 7.3362	0 0 0 -7.6130 7.3362 -0.1000	0 0 0 7.3362 -7.6130 -1.0000	0 0 0 7.3362 -7.6130 -4.0000

Table 13. Poles for the Pendumum - Rotor Plant for Inverted Mode for values of ω_N.

Rotor Plant Design for Full State Feedback with Integral Action

However, the user may select to specify the Rotor Actuator Plant as a second order transfer function with entry of Natural Frequency, ω_N, and Damping Coefficient, ζ.

$$P_{Rotor}(s) = H_{Rotor}(s)G_{Rotor}(s) = \frac{{\omega_N}^2}{s^2 + 2\zeta\omega_N s + {\omega_N}^2}$$

The compensator introduced has the form

$$H_{Rotor}(s) = \frac{{\omega_N}^2 s^2}{s^2 + 2\zeta\omega_N s + {\omega_N}^2}$$

The Edukit digital control system applies $H_{Rotor}(s)$ in series with $G_{Rotor}(s)$ to yield $P_{Rotor}(s)$.

The Edukit control system introduces a digital IIR filter constructed to create $H_{Rotor}(s)$.

It is required that designs that apply this Programmable Rotor Plant, include integral gain for PID Output Feedback or Integral Action for Full State Feedback. This is to ensure that small drift in operation resulting from finite numerical computation precision of the IIR filter does not lead to drift in position of the controller.

The user specified value for natural frequency, ω_N and for damping coefficient, ζ, computes and

This system supports operation in Full State Feedback operation with Integral Action with the operational limit parameters shown in *Table 14*.

Suspended Mode Operation		
Supported Control Modes	Dual PID and Full State Feedback Designs	
Natural Frequency	Lower Limit: 0.5 rad/sec	Upper Limit: 2 rad/sec
Damping Coefficient	Lower Limit: 0.0	Upper Limit: 5
Inverted Mode Operation		
Supported Control Mode	Full State Feedback with Integral Action	
Natural Frequency	Lower Limit: 0.5 rad/sec	Upper Limit: 2
Damping Coefficient	Lower Limit: 0.0	Upper Limit: 5

Table 14. Programmable Rotor Plant configuration limits for $P_{Rotor}(s) = \frac{{\omega_N}^2}{s^2+2\zeta\omega_N s+{\omega_N}^2}$. *The Programmable Plant supports operation in Full State Feedback for both Suspended Mode and Inverted Mode operation.*

Demonstration of the wide range of Programmable Plant configurations and accurate agreement between simulation and measured response is shown in *Figure 75* and *Figure 76*. Each system was designed applying the Rotor Plant specification and with the LQR design method of Section 15.

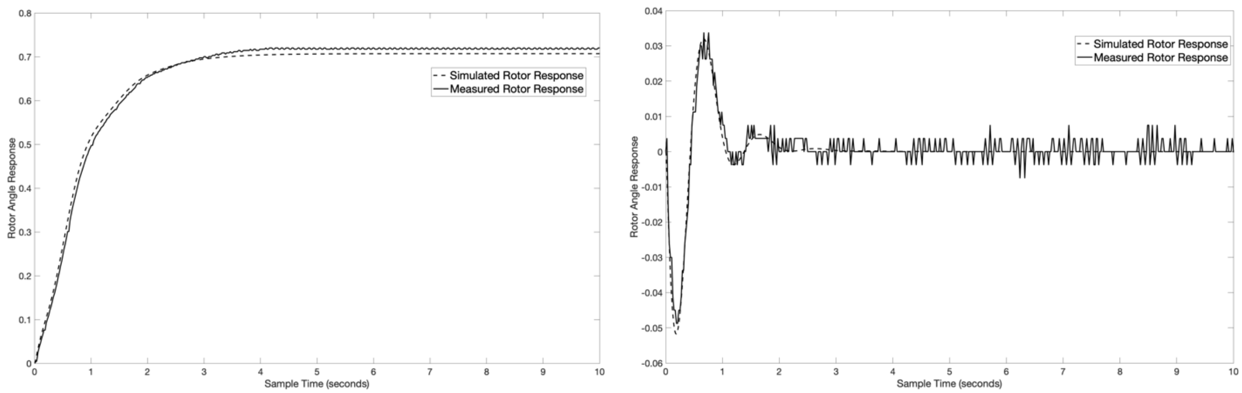


Figure 75. Comparison of simulated and measured response for the Edukit System in Suspended Mode Control with Full State Feedback. The Programmable Plant has been configured with Natural Frequency of 8 rad/sec and Damping Coefficient of . The Rotor Response is shown at left and Pendulum Response at right. Both show accurate agreement between simulated and measured response. Note that the discrete steps of 0.16 degrees in Pendulum Angle are observed in the measured response.

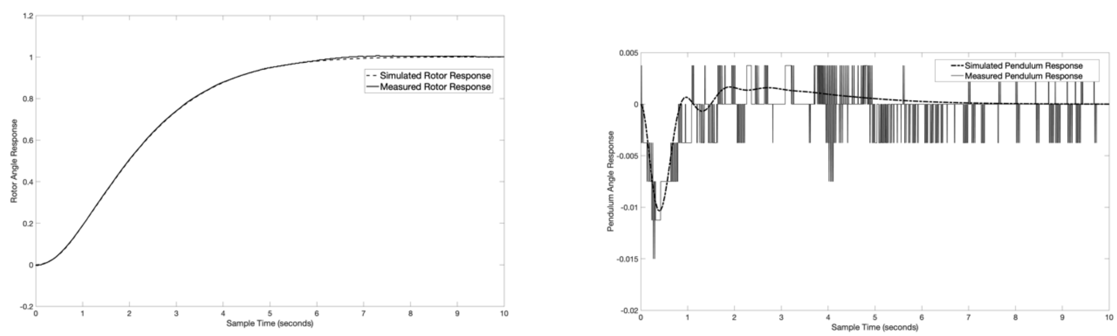


Figure 76. Comparison of simulated and measured response for the Edukit System in Suspended Mode Control with Full State Feedback with Integral Action. The Programmable Plant has been configured with Natural Frequency of 10 rad/sec and Damping Coefficient of 5. The Rotor Response is shown at left and Pendulum Response at right. Both show accurate agreement between simulated and measured response. Note that the discrete steps of 0.16 degrees in Pendulum Angle are observed in the measured response.

Further demonstration of the wide range of Programmable Plant configurations and accurate agreement between simulation and measured response is shown in *Figure 77*. This compares simulated and measured response for the Edukit System in Inverted Mode Control with Full State Feedback and with Integral Action included. This system was designed applying the Rotor Plant specification and with the LQR design method of Section 14.

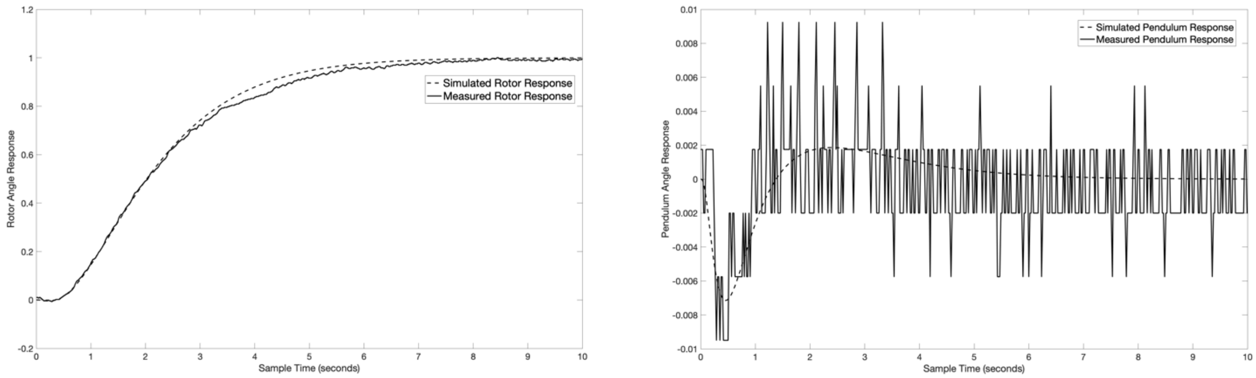
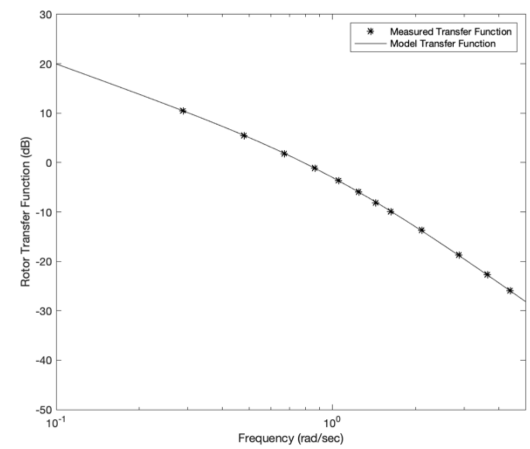


Figure 77. Comparison of simulated and measured response for the Edukit System in Inverted Mode Control with Full State Feedback and with Integral Action. The Programmable Plant has been configured with Natural Frequency of 5 rad/sec and Damping Coefficient of . The Rotor Response is shown at left and Pendulum Response at right. Both show accurate agreement between simulated and measured response. Note that the discrete steps of 0.16 degrees in Pendulum Angle are observed in the measured response.

The Rotor Angle Step Response was also measured by application of a step signal of amplitude 40 degrees to the Rotor Angle Reference at $t = 0$ and then sampling Rotor Angle over time. This was conducted with the Rotor Control System operating under full state feedback with gain values determined by LQR design. This applies the data logging feature of the Real Time Control Systems Workbench. Then, using the methods developed for deriving Sensitivity Function Transfer Function Frequency Response, Rotor Plant Transfer Function was computed. The comparison between simulated response of the Rotor Plant model without application of Rotor Plant Design and with $H_{Rotor}(s) = 1$ and the measured response is shown in *Figure 78*. The comparison between simulated response and measured response of the Rotor Plant model with Rotor Plant Design with $\omega_N = 1$ and



with $\zeta = 1$, $\zeta = 0.5$, and $\zeta = 0.1$ is shown in

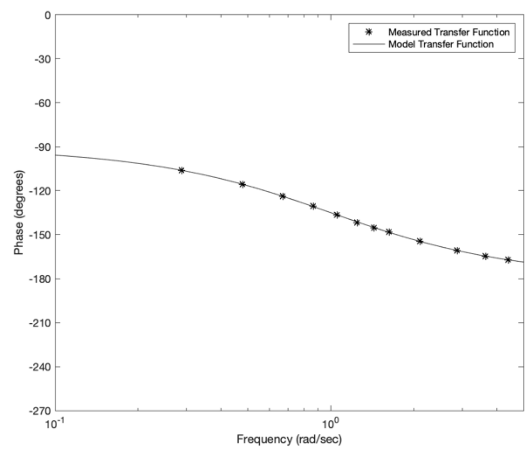


Figure 79, Figure 80 and Figure 81, respectively.

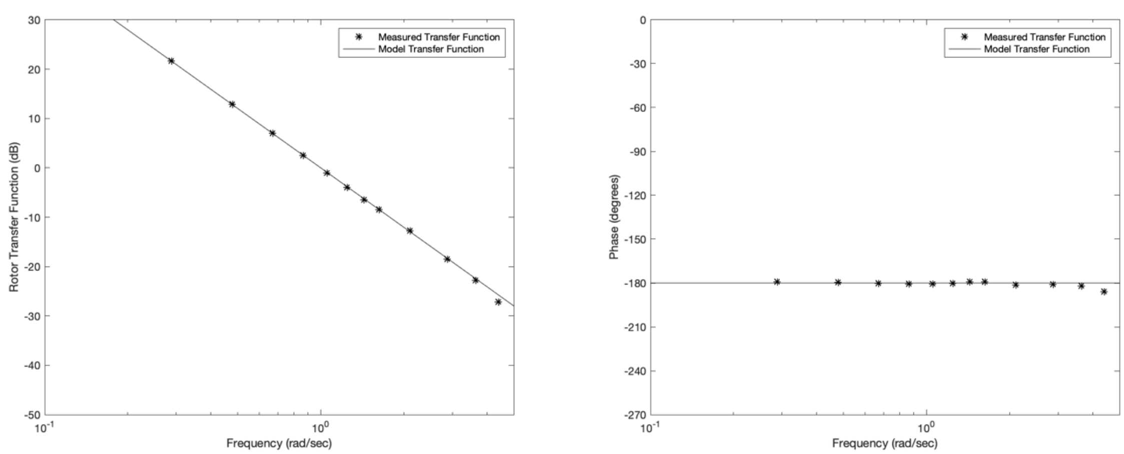


Figure 78. Comparison of model-predicted Rotor Angle Transfer Function and measured Rotor Angle Transfer Function Magnitude (left) and Phase (right) under Full State feedback control with Rotor Transfer Function computed by the Rotor System Identification system. This Transfer Function

corresponds to the Rotor Plant without Rotor Plant Design applied and thus with a Transfer Function of $G_{Rotor}(s) = 1/s^2$.

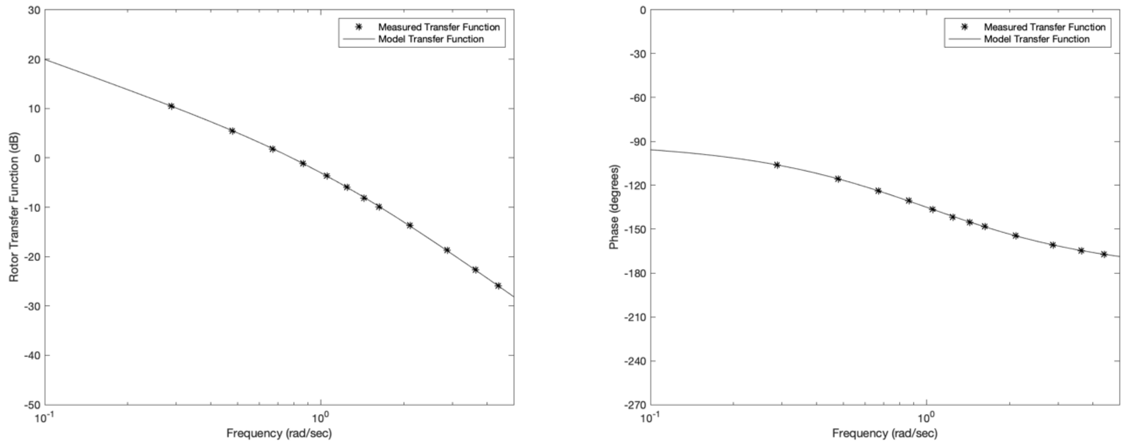


Figure 79.Comparison of model-predicted Rotor Angle Transfer Function and measured Rotor Angle Transfer Function Magnitude (left) and Phase (right) under Full State feedback control with Rotor Transfer Function computed by the Rotor System Identification system. This Transfer Function corresponds to the Rotor Plant with Rotor Plant Design applied and Transfer Function with $\zeta = 1$ *and* $\omega_N = 1$.

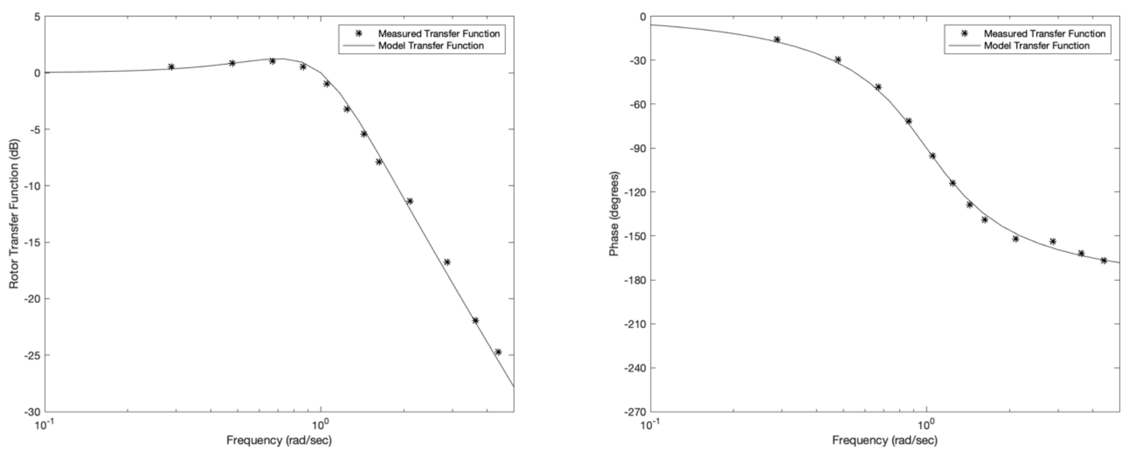
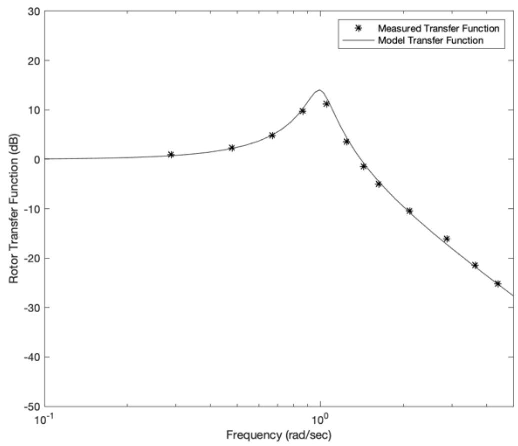


Figure 80.Comparison of model-predicted Rotor Angle Transfer Function and measured Rotor Angle Transfer Function Magnitude (left) and Phase (right) under Full State feedback control with Rotor Transfer Function computed by the Rotor System Identification system. This Transfer Function corresponds to the Rotor Plant with Rotor Plant Design applied and Transfer Function with $\zeta = 0.5$ *and* $\omega_N = 1$.



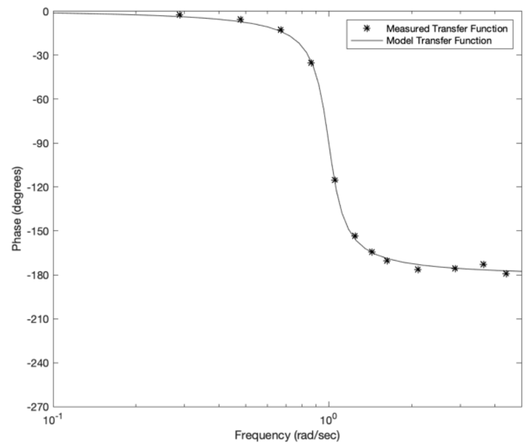


Figure 81.Comparison of model-predicted Rotor Angle Transfer Function and measured Rotor Angle Transfer Function Magnitude (left) and Phase (right) under Full State feedback control with Rotor Transfer Function computed by the Rotor System Identification system. This Transfer Function corresponds to the Rotor Plant with Rotor Plant Design applied and Transfer Function with $\zeta = 0.1$ *and* $\omega_N = 1$.

17. References

[Astrom_2019] Karl Johan Astrom and Richard M. Murray. 2008. Feedback Systems: An Introduction for Scientists and Engineers. Available here provided by the authors: http://www.cds.caltech.edu/~murray/books/AM08/pdf/fbs-public_18Aug2019.pdf

[Furuta_1992] K. Furuta, M. Yamakita, and S. Kobayashi, "Swing-up control of inverted pendulum using pseudo-state feedback," Journal of Systems and Control Engineering, vol. 206, no. 6, pp. 263-269, 1992.

[Hast_2013] M. Hast, K. J. Åström, B. Bernhardsson and S. Boyd, "PID design by convex-concave optimization," 2013 European Control Conference (ECC), Zurich, 2013, pp. 4460-4465,

[Nise_2015] Norman S. Nise, 2015, Control Systems Engineering (7th ed.). John Wiley & Sons, Inc., New York, NY, USA.

[Seborg_1989] Seborg, D.E., T.F. Edgar and D.A. Mellichamp (1989). Process Dynamics and Control. John Wiley & Sons.

[Speyer_2010] J. L. Speyer and D. H. Jacobson. Primer on Optimal Control Theory. SIAM, 2010.

[ST_IHM01A1_2019] https://www.st.com/en/ecosystems/x-nucleo-ihm01a1.html

[ST_L6474_2019] https://www.st.com/b/en/motor-drivers/l6474.html

[ST_Nucleo_2019] https://www.st.com/b/en/evaluation-tools/nucleo-f401re.html

[Yaniv_2004] O. Yaniv and M. Nagurka. 2004. Design of PID controllers satisfying gain margin and sensitivity constraints on a set of plants. Automatica 40, 1 (January, 2004), 111–116.

18. Appendix A: Edukit Quick Start Guide

The Edukit system operates entirely on its real time control system based on the STM32 processor and the IHM01A1 Stepper Motor interface.

The Edukit system operation is, therefore, independent of other computing platforms that may be connected. However, data transport for display of control system operation along with configuration commands may be provided over a serial interface. These systems are described in the next sections.

This Quick Start Guide describes first demonstration procedures for the Edukit.

1. System Assembly, Firmware Installation and Startup

1) The Edukit system is first assembled.
   a. Please see the Assembly Guides here:
      i. This Assembly Guide includes a soldering step
      ii. This Assembly Guide does not require soldering, but includes two additional low cost components.

2) The Edukit system, as first assembled, requires Firmware installation.
   a. Please see the Installation Guides here:
      i. Use this guide if your computing platform is an Apple Mac
      ii. Use this guide if your computing platform operates on the Windows OS

2. System Default Mode Operation

System Requirements

3) The Edukit system requires a standard USB 2.0 interface provided by a USB 5.0V power source or a standard Windows or Mac OSX computer.

4) Your computer requires a USB interface:

   Some Mac laptops of the 2015 and previous versions, include USB ports. More recent versions require and adapter from their ports to USB. This is the Apple USB-C to USB Adapter.

System Power and Data Interface

1) Power is first applied to the IHM01A1 Stepper Motor Controller by attaching the Edukit DC power supply unit to a 110V or 220V AC mains connection.

2) The Edukit USB cable is then attached to either a computer or to a USB 5V power supply adapter.

3. System Start

When power is applied to the Edukit system by its USB cable, its processor will boot and launch a process to provide stable inverted pendulum operation. These steps will occur:

1) The Edukit system will transmit a prompt to a computer that is connected to the USB cable. The prompt requests entry of a mode selection.

2) If no response is returned from the computer after a waiting time period of 10 seconts, a default demonstration mode operation starts.

3) The Edukit system will first determine if the pendulum is moving or is motionless in a stable position.

4) If the pendulum is moving, the Edukit system will wait until the pendulum is motionless and vertical down.

5) If stable, the Edukit system will measure the angle of the pendulum relative to vertical as required for control action to follow.

6) Then, when complete, the Edukit system will signal the user by displacing the rotor in a sequence of small motions as an alert signal.

7) The user should then raise the pendulum to a vertical position.

8) When the pendulum angle is within three degrees of vertical, the control system will start and stable operation will commence.

9) After a 10 second period of operation, an amplitude-modulated sinusoidal reference tracking signal will start and inverted pendulum operation will display rotor angle control as well as pendulum control.

10) If the pendulum motion vertical positioning does not occur, the Edukit system will wait, perform the angle calibration and signal the user again.

4. System Reset and Restart

The Edukit system may also be restarted by application of the Reset button. This is the black button on the Nucleo board shown in *Figure 82*.

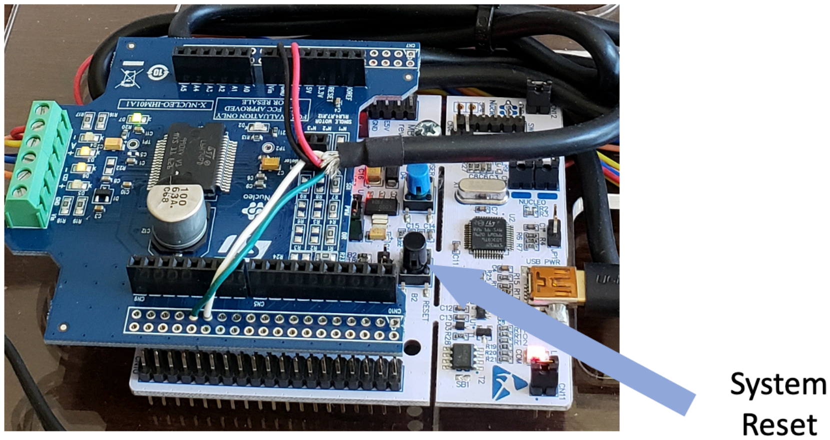


Figure 82. The Edukit system may be reset by depressing this reset button on the Nucleo board.

19. Appendix B: Edukit Rotary Inverted Pendulum: Real Time Control System Workbench

1. Introduction

The Edukit Rotary Inverted Pendulum may operate independently of any computing platform. Its control system implementation is supported in an open source high speed, real time C code implementation operating on the STM32 processor.

A Real Time Control System Workbench is available that communicates commands and receives data for real time display. This enables:

1) Real time configuration of Stepper Motor Interface parameters
2) Real time configuration of Control System parameters
3) Real time Display of Motor Interface and Control System parameters
4) Real time display of user-selected for Sensitivity Function Step Response Measurements
5) Selection of Data Logging operation with measured data archived in a user-selected directory

The Real Time Control System Workbench is designed to support both Windows and Mac OSX platforms. This system samples and displays data at a rate of 50 Hz (a rate limit set serial data transfer rate and by the graphics bandwidth of Windows and Mac OSX platforms).

An additional high speed sampling system interface operating on both Windows and Mac OSX is available, for high speed sampling and storage of data at the full 500 Hz rate of the Edukit system. This is accompanied by data processing provided by direct computation and display of system response and each of the four Sensitivity Function frequency response spectra.

Figure 83 shows the user interface of the Real Time Control System Workbench. The user has selected the options of Sine Off (removing a sinusoidal tracking reference command signal) and adding Rotor Step On (providing a step signal occurring each 16,384 control steps and 32 seconds.

2. The Real Time Control System Workbench software

The Real Time Control System Workbench for Matlab and Octave is available at this link

https://sites.google.com/view/ucla-st-motor-control/home

Installation guides for Octave are also available at this link above

1) Matlab operation
    a. Download the Matlab version: This will be file
       Edukit_Real_Time_Control_System_Workbench_Matlab.m

b. Then, open this with Matlab. Select the **Editor** tab of Matlab.

c. Review the Edukit Quick Start guides in Appendix A: Edukit Quick Start Guide

d. Connect the Edukit power supply to a mains outlet

e. Connect the Edukit USB cable to your computer

f. The Edukit system will start automatically

g. Select **Run** in the Matlab **Editor** tool bar

h. The Real Time view will begin.

1) Octave operation

a. Create unique directory on your machine for the Real Time Control System Workbench

b. Download the Octave version into this directory: This will include two files

Edukit_Real_Time_Control_System_Workbench_Octave.m

animatedline.m

c. These files should be **both** stored in the same directory

d. Then, open **both** with Octave

e. Within the **Editor** tab, select the tab of the Edukit_Real_Time_Control_System_Workbench_Octave.m

i. Review the Edukit Quick Start guides in Appendix A: Edukit Quick Start Guide

f. Connect the Edukit power supply to a mains outlet

g. Connect the Edukit USB cable to your computer

h. The Edukit system will start automatically

i. Select **Run** in the Octave **Editor** tool bar

j. The Real Time view will begin.

3. Starting the Real Time Control System Workbench

The Real Time Control System Workbench may be started from Matlab or Octave at anytime.

However, if another resource is using the serial interface to Edukit (for example, the screen utility of Mac OSX or putty interface of Windows) then these must be terminated first. These systems are described in the next section.

4. Graphical Data Windows at Left Column

First, the data displayed and user controls are summarized here.

1) Display of selected Sensitivity Function including Complimentary Sensitivity Function, Sensitivity Function, Load Disturbance Sensitivity Function and Noise Rejection Sensitivity Function.
2) Rotor Step signal indicator (On or Off)
3) Pendulum Controller Proportional Gain
4) Pendulum Controller Integral Gain
5) Pendulum Controller Derivative Gain
6) Rotor Controller Proportional Gain
7) Rotor Controller Integral Gain
8) Rotor Controller Derivative Gain
9) Stepper Motor Torque Current
10) Stepper Motor Maximum Acceleration (applicable during Motor Test operation)
11) Stepper Motor Maximum Deceleration (applicable during Motor Test operation)
12) Stepper Motor Maximum Speed (applicable during Motor Test operation)
13) Stepper Motor Maximum Speed (applicable during Motor Test operation)

14) Step Size of parameter value adjustment steps
    a. This size ranges from values of 0.5, 2, 20, 50, 1 and 100
    b. This is increased or decreased by clicking on Inc Step or Dec Step buttons

5. Control System Configuration Control Buttons Lower Row of Interface

1) Exit: Exit Real Time Workbench
2) Stop Control: Terminate control and exit Real Time Workbench
3) Sine On: Enable Modulated Sine Waveform Rotor Tracking Reference
4) Sine Off: Disable Modulated Sine Waveform Rotor Tracking Reference

5) Inc Step: Increase Step Size of user parameter value input
6) Dec Step: Decrease Step Size of user parameter value input

Each of the following controls adjust parameter values by an amount equal to Step Size for each button click

7) Inc Pend P: Increase Pendulum Controller Proportional Gain
8) Inc Pend P: Decrease Pendulum Controller Proportional Gain
9) Inc Pend I: Increase Pendulum Controller Proportional Gain
10) Inc Pend I: Decrease Pendulum Controller Proportional Gain
11) Inc Pend D: Increase Pendulum Controller Proportional Gain
12) Inc Pend D: Decrease Rotor Controller Proportional Gain
13) Inc Rotor P: Increase Rotor Controller Proportional Gain
14) Inc Rotor P: Decrease Rotor Controller Proportional Gain
15) Inc Rotor I: Increase Rotor Controller Proportional Gain
16) Inc Rotor I: Decrease Rotor Controller Proportional Gain
17) Inc Rotor D: Increase Rotor Controller Proportional Gain
18) Inc Rotor D: Decrease Rotor Controller Proportional Gain

6. Control System Configuration Control Buttons Right Column of Interface

1) Log Data: Start data logging of three 40 sec sweeps starting at next sweep

2) Rotor Step On: Enables Rotor Step reference track signal
3) Rotor Step Off: Disables Rotor Step reference track signal

Note also:
4) Rotor Step On: Enables Complimentary Sensitivity Function measuremen
5) Rotor Step Off: Disables Complimentary Sensitivity Function measurement

6) Pendulum Impulse On: Enables Pendulum Impulse reference track signal
7) Pendulum Impulse Off: Disables Pendulum Impulse reference track signal

8) Load Dist On: Enables Load Disturbance Rejection Sensitivity Function measurement
9) Load Dist On: Disables Load Disturbance Rejection Sensitivity Function measurement
Note that Load Disturbance response is small for effective design. Thus, this signal is displayed with a 10 times magnification.

10) Noise Sens On: Enables Noise Rejection Sensitivity Function measurement
11) Noise Sens Off: Disables Noise Rejection Sensitivity Function measurement

12) Sensitivity On: Enables Sensitivity Function measurement
13) Sensitivity Off: Disables Sensitivity Function measurement

The following controls apply when the system is operating in Motor Control Characterization Mode as selected by the Edukit Serial Interface. These controls are not applicable or active in normal closed loop operation.

14) Dec Min Spd: Decreases minimum speed in by amount equal to selected Step Size.
15) Inc Min Spd: Increases minimum speed in by amount equal to selected Step Size.

16) Dec Max Spd: Decreases maximum speed in by amount equal to selected Step Size.
17) Inc Max Spd: Increases maximum speed in by amount equal to selected Step Size.

18) Dec Torq: Decreases Motor Torque Current by amount equal to selected Step Size.
19) Inc Torq: Increases Motor Torque Current by amount equal to selected Step Size.

Examples of Real Time Workbench views for multiple modes include these:

1) *Figure 83*: Rotor Response and Complimentary Sensitivity Function step response
2) *Figure 84*: Sensitivity Function step response
3) *Figure 85*: Load Disturbance Sensitivity Function step response
4) *Figure 86*: Noise Rejection Sensitivity Function step response

Note that the Real Time Workbench provides effective measurement of the response of each of the four Sensitivity functions.

In addition, as noted, an additional high speed sampling system, the Sensitivity Function Measurement System operating on both Windows and Mac OSX is available, for high speed sampling and storage of data at the full 500 Hz rate of the Edukit system. This is accompanied by data processing provided by direct computation and display of system response and each of the four Sensitivity Function frequency response spectra.

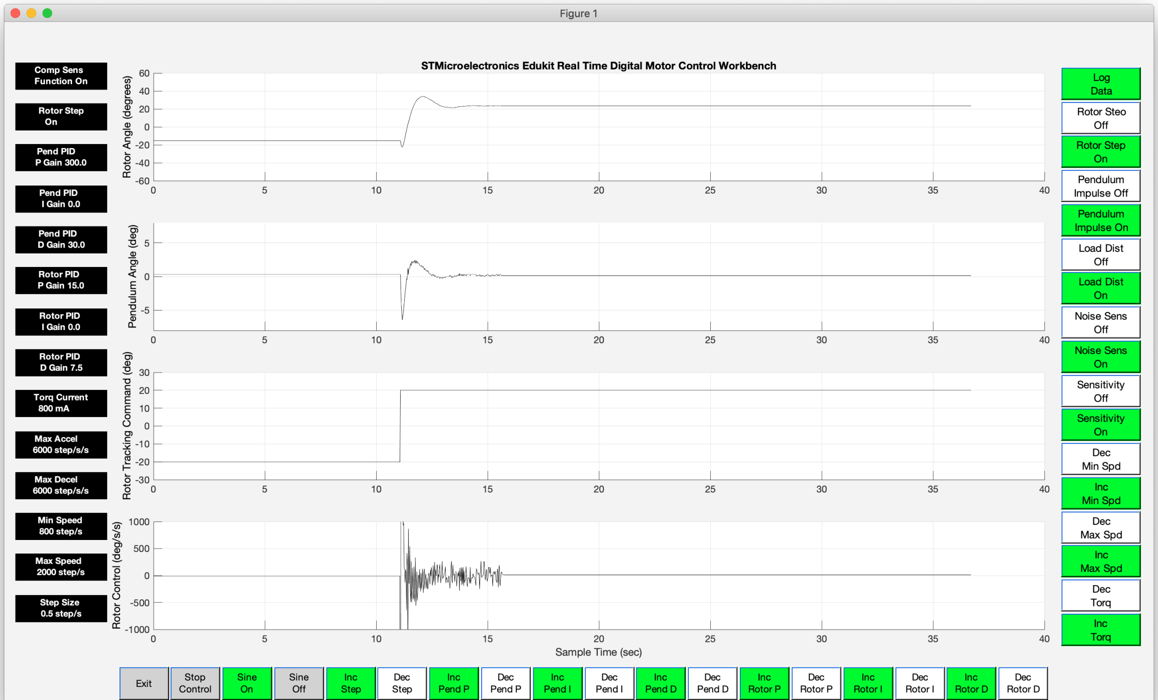
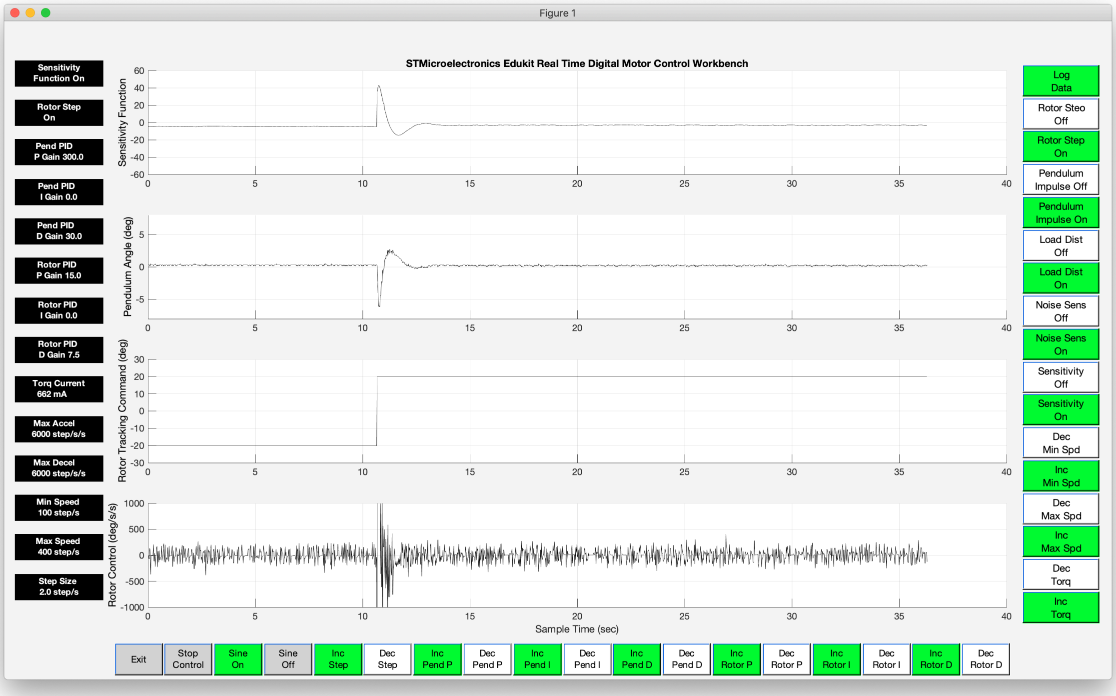


Figure 83. The Edukit Rotary Inverted Pendulum Real Time Control System Workbench in Matlab. This shows part of a trace of real time data corresponding to operation with system response to a 40 degree step in the Rotor Angle Tracking Command. This also provides measurement of the Complimentary Sensitivity Function step response.



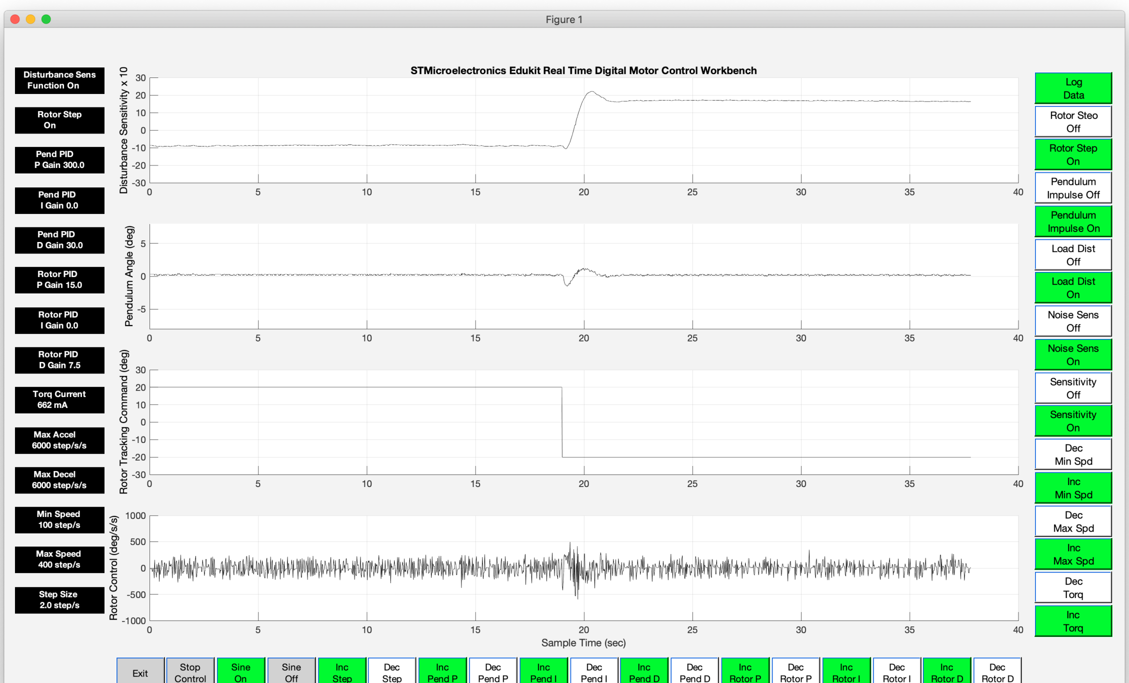


Figure 84. The Edukit Rotary Inverted Pendulum Real Time Control System Workbench. This shows measurement of the Sensitivity Function with part of a trace of real time data.

Figure 85. The Edukit Rotary Inverted Pendulum Real Time Control System Workbench. This shows measurement of the Load Disturbance Rejection Sensitivity Function with part of a trace of real time data. Note that Load Disturbance response is small for effective design. Thus, this signal is displayed with a 10 times magnification.

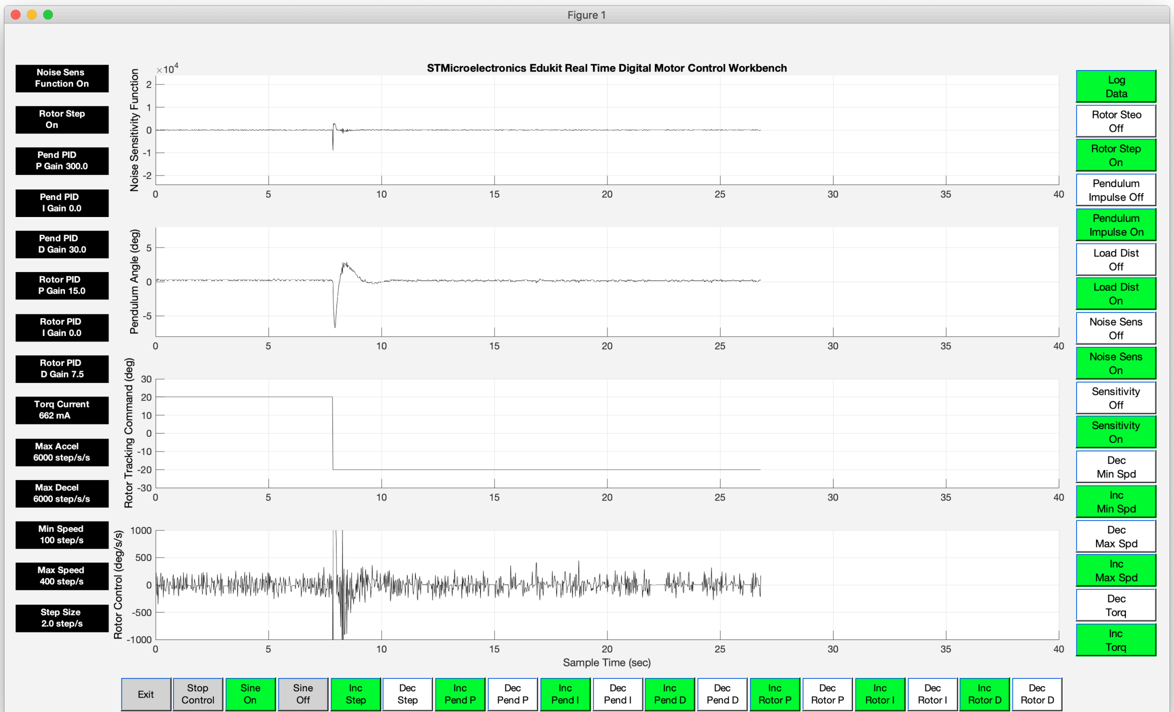


Figure 86. The Edukit Rotary Inverted Pendulum Real Time Control System Workbench. This shows measurement of the Noise Rejection Sensitivity Function with part of a trace of real time data.

20. Appendix C: Edukit Sensitivity Function Frequency Response Computation System

1. Introduction

The four Sensitivity Functions that enable performance characterization of control systems are each measured by the Edukit system. These are applied and described with examples in Sections 8, 9, 11, 12.

The Sensitivity Function measurements may be selected by the Control System Workbench using the control buttons of

1. Sensitivity On/Off: Selects Sensitivity Function
2. Rotor Step On/Off: Selects Complimentary Sensitivity Function
3. Load Dist On/Off: Selects Load Disturbance Rejection Sensitivity Function
4. Noise Sens On/Off: Selects Noise Sensitvity Function

2. Operation

To acquire Sensitivity Function data, follow this procedures

1. First select one of the Sensitivity Function options.
2. Then, select Log Data
3. At this time, select a number of measurement Sweeps. Each sweep requires 40 seconds.
4. The default and minimum for Frequency Response Computation is 3 sweeps. However, this can be increased to improve signal to noise ratio in measurements.

Output data will be stored in *.mat files in the directory selected at the start of Workbench operation. The data files will be of the format:

1. Sensitivity Function
   a. Edukit_Sensitivity_Function__<Date>_<Time>.mat
2. Complimentary Sensitivity
   a. Edukit_Step_Respone_Rotor_Angle_(degrees)__<Date>_<Time>.mat
3. Load Disturbance Rejection Sensitivity Function
   a. Edukit_Disturbance_Sensitivity_Function__<Date>_<Time>.mat
4. Selects Noise Sensitvity Function
   a. Edukit_ Noise_Sensitivity_Function__<Date>_<Time>.mat

3. Frequency Response Computation

The Edukit system includes the capability for introduction of a step signal stimulus for probing of the control system response for each of the four Sensitivity Functions.

Frequency response for each Sensitivity Function may then be determined by computing the ratio of the Fourier Transform of both the system response and the system stimulus for each frequency component.

A Matlab script, Edukit_Sensitivity_Function_Freq_Response_Computation.m is provided for this computation. This operates on a selected Sensitivity Function data file and returns the computed frequency response spectrum.

This system will operate on a minimum of three data sweeps. However, as desired, this system will also signal average a greater number of sweeps. This may be useful in computation of frequency response over a very large dynamic range and for small magnitude in the frequency response spectrum.

21. Appendix D: Integrated Rotary Inverted Pendulum: Serial Control Interface

1. Introduction

Interactive control and data acquisition with the Edukit system may be performed over any USB serial terminal connection between a computer (laptop or workstation) and the Edukit platform.

The USB connection provides both data access and power for the Nucleo processor. When the USB connection is made, the system will start and create a communication session. If no computer is connected or if there is no response, the system will start in a default mode.

2. System Default Mode Operation

System Requirements

1) Your computer requires a USB interface:

   Some Mac laptops of the 2015 and previous versions, include USB ports. More recent versions require and adapter from their ports to USB. This is the Apple USB-C to USB Adapter.

2) Your computer also requires a serial communication interface.

   Methods for using the native communication interface on Mac OSX or application of an open source serial interface on Windows is described later in this section.

System Power and Data Interface

3) Power is first applied to the IHM01A1 Stepper Motor Controller by attaching the Edukit DC power supply unit to a 110V or 220V AC mains connection.
4) The Edukit USB cable is then attached to either a computer or to a USB 5V power supply adapter.

3. System Start

When power is applied to the Edukit system by its USB cable, its processor will boot and launch a process to provide stable inverted pendulum operation. These steps will occur:

When power is applied to the Edukit system by its USB cable, its processor will boot and launch a process to provide stable inverted pendulum operation. These steps will occur:

1) The Edukit system will transmit a prompt to a computer that is connected to the USB cable. The prompt requests entry of a mode selection.

2) If no response is returned from the computer after a waiting time period of 10 seconts, a default demonstration mode operation starts.

3) The Edukit system will first determine if the pendulum is moving or is motionless in a stable position.

4) If the pendulum is moving, the Edukit system will wait until the pendulum is motionless and vertical down.

5) If stable, the Edukit system will measure the angle of the pendulum relative to vertical as required for control action to follow.

6) Then, when complete, the Edukit system will signal the user by displacing the rotor in a sequence of small motions as an alert signal.

7) The user should then raise the pendulum to a vertical position.

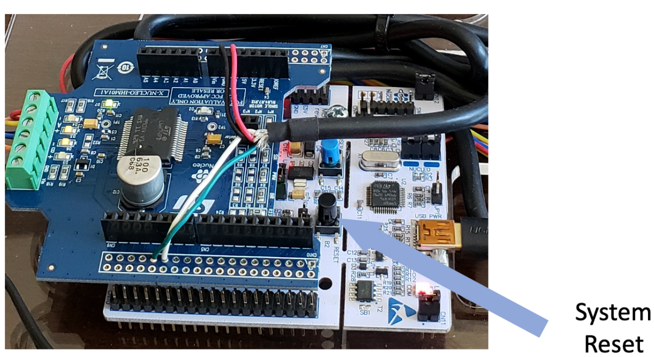
8) When the pendulum angle is within three degrees of vertical, the control system will start and stable operation will commence.

9) After a 10 second period of operation, an amplitude-modulated sinusoidal reference tracking signal will start and inverted pendulum operation will display rotor angle control as well as pendulum control.

10) If the pendulum motion vertical positioning does not occur, the Edukit system will wait, perform the angle calibration and signal the user again.

5. System Reset and Restart

The Edukit system may also be restarted by application of the Reset button. This is the black button on the Nucleo board shown below.



The Edukit system enables selection of operating modes via the Serial Interface. These are selected by entering a Mode Command Entry in response to the Edukit prompts these Command Entries are summarized below.

Input Mode Command Entry	Mode Description
1	Inverted Pendulum Control. After entering 1, the user is prompted for: a) Selection of a Rotor Chirp tracking reference signal b) Selection of a Step Rotor tracking reference signal c) Selection of a modulated sine Rotor tracking reference signal d) Selection of a Rotor tracking “Comb” reference signal composed of the sum of a series of sinusoidal signals of log-spaced frequency.
2	Suspended Pendulum Control. After entering 2, the user is prompted for a) Selection of a Rotor Chirp tracking reference signal b) Selection of a Step Rotor tracking reference signal c) Selection of a modulated sine Rotor tracking reference signal d) Selection of a Rotor tracking “Comb” reference signal composed of the sum of a series of sinusoidal signals of log-spaced frequency.
s	Single PID mode requiring entry only of Pendulum PID Control gains with model Rotor Transfer Function as described in Section 13. After entering **s**, the user is prompted for a) Selection of a Rotor Chirp tracking reference signal b) Selection of a Step Rotor tracking reference signal c) Selection of a modulated sine Rotor tracking reference signal d) Selection of a Pendulum Impulse signal
g	General mode enabling entry of PID gains for both Pendulum PID and Rotor PID as well as motor configuration parameters. After entering **g**, the user is prompted for a series of design option to be described below.
t	Test mode for testing of Rotor Actuator and Pendulum Angle Encoder. This system will move the rotor through a series of angles and also prompt the user to rotate the pendulum by 360 degrees in clockwise and counterclockwise rotation. This verifies operation of the actuator and sensor.
r	Rotor control mode offering direct control over the mator as well as selection of motor system configuration parameters.
c	Motor control characterization mode enabling a sinusoidal chirp signal or square wave chirp signal.
q	During operation, entering the character, q, will cause the control system to exit and rotor motion will stop.

6. General Mode Operation

If the user selects the General Mode "g" selection, the following prompts are available:

1) Dual PID or State Feedback
    a. This enables the user to select design parameters for Dual PID or Full State Feedback as described in this Manual. The parameters described in these section may be entered as examples accordingly.
    b. If Dual PID is selected the following options are available
        i. Gains for the Pendulum and Rotor PID controllers
        ii. Selection of Inverted or Suspended Mode
        iii. Selection or Rotor Chirp Drive
        iv. Selection of Rotor Step Drive
        v. Selection of Rotor Sine Drive
        vi. Selection of Rotor Comb Drive
        vii. Selection of Pendulum Impulse Drive
        viii. Selection of one of the four Sensitivity Functions
    c. If State Feedback is selected the following options are available
        i. Gains for the Pendulum Angle, Pendulum Angle Derivative, Rotor Angle, and Rotor Angle Derivative gains
        ii. Integrator Compensator Gain if this is included in design.
            1. The parameters described in the sections on Full State control may be entered as examples accordingly.
        iii. Selection of Inverted or Suspended Mode
        iv. Selection or Rotor Chirp Drive
        v. Selection of Rotor Step Drive
        vi. Selection of Rotor Sine Drive
        vii. Selection of Rotor Comb Drive
        viii. Selection of Pendulum Impulse Drive
        ix. Selection of one of the four Sensitivity Functions
        x. Option to select Rotor Plant Design as described in Section 16. If this option is selected the user is prompted for selection of:
            1. Rotor Plant Natural Frequency
            2. Rotor Plant Damping Coefficient
            3. Rotor Plant Gain

4. System Operation Viewing and Control Over Serial Terminal

Mac OSX users should review Section 8 in this Appendix in order to use the Apple **screen** utility.

Windows users should review Section 9 in this Appendix order to install and use the **putty** serial terminal utility.

Upon application of USB power source or the user pressing the Nucleo system reset button, the system will start and present a prompt.

The Edukit system will transmit these prompts over the serial link and wait for responses from a serial terminal.

This is the sequence that occurs after system boot and selecting Mode 1

The user entries are indicated in red font.

```
System Starting Prepare to Enter Mode Selection...
********  System Start Mode Selections  ********
Enter 1 at prompt for Inverted Pendulum Control
Enter 2 at prompt for Suspended Pendulum Control
Enter 's' at prompt for Single PID: With Prompts for Pendulum Controller Gains
Enter 'g' at prompt for General Mode: With Prompts for Full State Feedback and PID Controller Gains
Enter 't' at prompt for Test Mode: Test of Rotor Actuator and Pendulum Angle Encoder
Enter 'r' at prompt for Rotor Control Mode: Direct Control of Rotor Actuator

System Starting Prepare to Enter Mode Selection...
System Starting Prepare to Enter Mode Selection...
System Starting Prepare to Enter Mode Selection...

********  System Start Mode Selections  ********
Enter Mode Selection Now or System Will Start in Default Mode in 5 Seconds: 1
Mode 1 Configured
Enter 1 to Enable Rotor Chirp Drive; 0 to Disable: 0
Enter 1 to Enable Step Drive; 0 to Disable: 0
Enter 1 to Enable Sine Drive; 0 to Disable: 0
Enter 1 to Enable Rotor Tracking Comb Signal; 0 to Disable: 0
Motor Profile Speeds Set at Min 800 Max 2000 Steps per Second and Suspended Mode 0
Motor Torque Current Set at 800.000000
Prepare for Control Start - Initial Rotor Position: 0
Test for Pendulum at Rest - Stabilize Pendulum Now
Pendulum Now at Rest and Measuring Pendulum Down Angle
Adjust Pendulum Upright By Turning CCW Control Will Start When Vertical
```

The reported Edukit data is best viewed using the Edukit Real Time Control System Workbench.

5. Example Serial Interface Session with General Design and Dual PID Feedback in Suspended Mode

This session follows the system design of Sections 8 and 9

User entries are shown in red font for this session:

```
System Starting Prepare to Enter Mode Selection...
********  System Start Mode Selections  ********
Enter 1 at prompt for Inverted Pendulum Control
Enter 2 at prompt for Suspended Pendulum Control
Enter 's' at prompt for Single PID: With Prompts for Pendulum Controller Gains
Enter 'g' at prompt for General Mode: With Prompts for Full State Feedback and PID Controller Gains
Enter 't' at prompt for Test Mode: Test of Rotor Actuator and Pendulum Angle Encoder
Enter 'r' at prompt for Rotor Control Mode: Direct Control of Rotor Actuator

Enter Mode Selection Now or System Will Start in Default Mode in 5 Seconds: g
Enter 0 for Dual PID - Enter 1 for State Feedback: 0
Enter Pendulum PID Proportional Gain: 10.00
Enter Pendulum PID Integral Gain: 0.00
Enter Pendulum PID Differential Gain: 5.00
Enter Rotor PID Proportional Gain: 2.00
Enter Rotor PID Integral Gain: 0.00
Enter Rotor PID Differential Gain: 2.00
Enter 0 for Inverted Mode - Enter 1 for Suspended Mode: 1
Enter 1 to Enable Rotor Chirp Drive; 0 to Disable: 0
Enter 1 to Enable Step Drive; 0 to Disable: 0
Enter 1 to Enable Sine Drive; 0 to Disable: 0
Enter 1 to Enable Rotor Tracking Comb Signal; 0 to Disable: 0
Enter 1 to Enable Pendulum Impulse Signal; 0 to Disable: 0
MEnter 1 to Enable Disturbance Rejection Sensitivity Function Analysis; 0 to Disable: 0
Enter 1 to Enable Noise Rejection Sensitivity Function Analysis; 0 to Disable: 0
Enter 1 to Enable Sensitivity Function Analysis; 0 to Disable: 0
Enter 1 for Rotor Plant Design Grotor = Wn2/(s2 + 2D*s + Wn2)
Pendulum PID Gains:   P: -10.00; I: -0.00; D: -5.00
Rotor PID Gains:      P: -2.00; I: -0.00; D: -2.00
Suspended Mode gain values must be negative or zero
Motor Profile Speeds Set at Min 800 Max 2000 Steps per Second and Suspended Mode 1
Motor Torque Current Set at 800.000000
MSuspended Mode selected...
Prepare for Control Start - Initial Rotor Position: 0
Test for Pendulum at Rest - Stabilize Pendulum Now
Pendulum Now at Rest and Measuring Pendulum Down Angle
Suspended Mode Control Will Start in 3 Seconds
2       2       -11     0       -15     0.0     0       -16     0
2       2       -7      0       -3      0.0     0       -4      0
2       2       -3      0       4       0.0     0       4       0
2       2       0       0       5       0.0     0       6       0
2       2       2       0       5       0.0     0       5       0
2       2       6       0       10      0.0     0       10      0
2       2       9       1       12      0.0     0       11      -1
2       2       18      2       17      0.0     0       17      0
.
.
.
```

Now, the screen or putty terminal session may be closed, releasing the serial port.

Then, the Real Time System Workbench may be started in Matlab.

6. Example Serial Interface Session with General Design and Dual PID Feedback in Inverted Mode

This session follows the system design of Sections 11 and 12

User entries are shown in red font for this session:

```
System Starting Prepare to Enter Mode Selection...
********  System Start Mode Selections  ********
Enter 1 at prompt for Inverted Pendulum Control
Enter 2 at prompt for Suspended Pendulum Control
Enter 's' at prompt for Single PID: With Prompts for Pendulum Controller Gains
Enter 'g' at prompt for General Mode: With Prompts for Full State Feedback and PID Controller Gains
Enter 't' at prompt for Test Mode: Test of Rotor Actuator and Pendulum Angle Encoder
Enter 'r' at prompt for Rotor Control Mode: Direct Control of Rotor Actuator

Enter Mode Selection Now or System Will Start in Default Mode in 5 Seconds: g
Enter 0 for Dual PID - Enter 1 for State Feedback: 0
Enter Pendulum PID Proportional Gain: 300.00
Enter Pendulum PID Integral Gain: 0.00
Enter Pendulum PID Differential Gain: 30.00
Enter Rotor PID Proportional Gain: 15.00
Enter Rotor PID Integral Gain: 0.00
Enter Rotor PID Differential Gain: 7.50
Enter 0 for Inverted Mode - Enter 1 for Suspended Mode: 0
Enter 1 to Enable Rotor Chirp Drive; 0 to Disable: 0
Enter 1 to Enable Step Drive; 0 to Disable: 0
Enter 1 to Enable Sine Drive; 0 to Disable: 0
Enter 1 to Enable Rotor Tracking Comb Signal; 0 to Disable: 0
Enter 1 to Enable Pendulum Impulse Signal; 0 to Disable: 0
MEnter 1 to Enable Disturbance Rejection Sensitivity Function Analysis; 0 to Disable: 0
Enter 1 to Enable Noise Rejection Sensitivity Function Analysis; 0 to Disable: 0
Enter 1 to Enable Sensitivity Function Analysis; 0 to Disable: 0
Enter 1 for Rotor Plant Design Grotor = Wn2/(s2 + 2D*s + Wn2)
Pendulum PID Gains:    P: -10.00; I: -0.00; D: -5.00
Rotor PID Gains:       P: -2.00; I: -0.00; D: -2.00
Suspended Mode gain values must be negative or zero
Motor Profile Speeds Set at Min 800 Max 2000 Steps per Second and Suspended Mode 1
Motor Torque Current Set at 800.000000
MSuspended Mode selected...
Prepare for Control Start - Initial Rotor Position: 0
Test for Pendulum at Rest - Stabilize Pendulum Now
Pendulum Now at Rest and Measuring Pendulum Down Angle
Suspended Mode Control Will Start in 3 Seconds
2       2       -11     0       -15     0.0     0       -16     0
2       2       -7      0       -3      0.0     0       -4      0
2       2       -3      0       4       0.0     0       4       0
2       2       0       0       5       0.0     0       6       0
2       2       2       0       5       0.0     0       5       0
2       2       6       0       10      0.0     0       10      0
2       2       9       1       12      0.0     0       11      -1
2       2       18      2       17      0.0     0       17      0
.
.
.
```

Now, the screen or putty terminal session may be closed, releasing the serial port.

Then, the Real Time System Workbench may be started in Matlab.

7. Example Serial Interface Session with General Design and Full State Feedback Mode

User entries are shown in red font for this session:

```
System Starting Prepare to Enter Mode Selection...
********  System Start Mode Selections  ********
Enter 1 at prompt for Inverted Pendulum Control
Enter 2 at prompt for Suspended Pendulum Control
Enter 's' at prompt for Single PID: With Prompts for Pendulum Controller Gains
Enter 'g' at prompt for General Mode: With Prompts for Full State Feedback and PID Controller Gains
Enter 't' at prompt for Test Mode: Test of Rotor Actuator and Pendulum Angle Encoder
Enter 'r' at prompt for Rotor Control Mode: Direct Control of Rotor Actuator

Enter Mode Selection Now or System Will Start in Default Mode in 5 Seconds: g
Enter 0 for Dual PID - Enter 1 for State Feedback: 1
Enter Pendulum Angle Proportional Gain: 5.55
Enter Pendulum Angle Derivative Gain: 0.26
Enter Rotor Angle Proportional Gain: 2.54
Enter Rotor Angle Derivative Gain: 0.28
Enter Integral Compensator Gain (zero to disable): 1.00
Enter 0 for Inverted Mode - Enter 1 for Suspended Mode: 1
Enter 1 to Enable Rotor Chirp Drive; 0 to Disable: 0
Enter 1 to Enable Step Drive; 0 to Disable: 0
Enter 1 to Enable Sine Drive; 0 to Disable: 0
Enter 1 to Enable Rotor Tracking Comb Signal; 0 to Disable: 0
Enter 1 to Enable Pendulum Impulse Signal; 0 to Disable: 0
Enter 1 to Enable Disturbance Rejection Sensitivity Function Analysis; 0 to Disable: 0
Enter 1 to Enable Noise Rejection Sensitivity Function Analysis; 0 to Disable: 0
Enter 1 to Enable Sensitivity Function Analysis; 0 to Disable: 0
Enter 1 to Enter Rotor Plant Design:
Enter Natural Frequency (rad/sec) of Minimum 1 and Maximum 10 for Suspended Mode: 2.00
Enter Rotor Damping Coefficient of Minimum 1 and Maximum 5 : 5.00
Enter Rotor Plant Gain of Minimum 0.1 and Maximum 1 : 1.00
Inverted Mode Rotor Plant Gain: 1.00
Pendulum PID Gains:    P: -5.55; I: -0.00; D: -0.26
Rotor PID Gains:       P: -2.54; I: -0.00; D: -0.28
Suspended Mode gain values must be negative or zero
Motor Profile Speeds Set at Min 800 Max 2000 Steps per Second and Suspended Mode 1
Motor Torque Current Set at 800.000000
Suspended Mode selected...
Prepare for Control Start - Initial Rotor Position: 0
Test for Pendulum at Rest - Stabilize Pendulum Now
Pendulum Now at Rest and Measuring Pendulum Down Angle
Suspended Mode Control Will Start in 3 Seconds
Initial Rotor Position: 0
2       2       -11     0       -15     0.0     0       -16     0
2       2       -7      0       -3      0.0     0       -4      0
2       2       -3      0       4       0.0     0       4       0
2       2       0       0       5       0.0     0       6       0
2       2       2       0       5       0.0     0       5       0
2       2       6       0       10      0.0     0       10      0
2       2       9       1       12      0.0     0       11      -1
198     2       18      2       17      0.0     0       17      0
.
.
.
```

Now, the screen or putty terminal session may be closed, releasing the serial port.

Then, the Real Time System Workbench may be started in Matlab.

8. Example Serial Interface Session with Single PID Mode

The Edukit Single PID mode of Section 13 provides a Rotor Control System and requests a user design for Pendulum Control.

```
Enter Mode Selection Now or System Will Start in Default Mode in 5 Seconds: g
System Starting Prepare to Enter Mode Selection...
********  System Start Mode Selections  ********
Enter 1 at prompt for Inverted Pendulum Control
Enter 2 at prompt for Suspended Pendulum Control
Enter 's' at prompt for Single PID: With Prompts for Pendulum Controller Gains
Enter 'g' at prompt for General Mode: With Prompts for Full State Feedback and PID Controller Gains
Enter 't' at prompt for Test Mode: Test of Rotor Actuator and Pendulum Angle Encoder
Enter 'r' at prompt for Rotor Control Mode: Direct Control of Rotor Actuator

Enter Mode Selection Now or System Will Start in Default Mode in 5 Seconds: s
 *** Starting Single PID Configuration Mode ***

Enter Pendulum PID Proportional Gain: 300.00
Enter Pendulum PID Integral Gain: 0.00
Enter Pendulum PID Differential Gain: 30.00
Enter 1 to Enable Rotor Chirp Drive; 0 to Disable: 0
Enter 1 to Enable Step Drive; 0 to Disable: 0
Enter 1 to Enable Sine Drive; 0 to Disable: 0
Enter 1 to Enable Pendulum Impulse; 0 to Disable: 0
Pendulum PID Gains:     P: 300.00; I: 0.00; D: 30.00
Rotor PID Gains:        P: 15.00; I: 0.00; D: 15.00
Motor Profile Speeds Set at Min 800 Max 2000 Steps per Second and Suspended Mode 0
Motor Torque Current Set at 800.000000
Prepare for Control Start - Initial Rotor Position: 0
Test for Pendulum at Rest - Stabilize Pendulum Now
```

9. Example Serial Interface Session with Test Mode Operation

The Edukit Test Mode permits testing or Rotor Operaton and Pendulum Angle Encoder operation. An example session follows below.

```
System Starting Prepare to Enter Mode Selection...
********  System Start Mode Selections  ********
Enter 1 at prompt for Inverted Pendulum Control
Enter 2 at prompt for Suspended Pendulum Control
Enter 's' at prompt for Single PID: With Prompts for Pendulum Controller Gains
Enter 'g' at prompt for General Mode: With Prompts for Full State Feedback and PID Controller Gains
Enter 't' at prompt for Test Mode: Test of Rotor Actuator and Pendulum Angle Encoder
Enter 'r' at prompt for Rotor Control Mode: Direct Control of Rotor Actuator

Enter Mode Selection Now or System Will Start in Default Mode in 5 Seconds: t
Test Mode Configured
Motor Profile Speeds Set at Min 800 Max 2000 Steps per Second and Suspended Mode 0
Motor Torque Current Set at 800.000000
Motor Profile Speeds Min 0 Max 0
Motor Profile Acceleration Max 6000 Deceleration Max 6000

********  Starting Rotor Motor Control Test  ********
Motor Position at Zero Angle: 0.00
Next Test in 3s
```

```
Motor Position Test to -45 Degree Angle: -45.00
Correct motion shows rotor rotating to left
Next Test in 3s

Motor Position Test to Zero Angle: 0.00
Correct motion shows rotor returning to zero angle
Next Test in 3s

Motor Position at 90 Degree Angle: 90.00
Correct motion shows rotor rotating to right
Next Test in 3s

Motor Position at Zero Angle: 0.00
Correct motion shows rotor rotating to zero angle
Rotor Actuator Test Cycle Complete, Next Test in 3s

*************  Starting Encoder Test  ***************

Permit Pendulum to Stabilize in Vertical Down
Angle will be measured in 3 seconds
Encoder Angle is: -0.15
(Correct value should lie between -0.5 and 0.5 degrees))

Manually Rotate Pendulum in Clock Wise Direction One Full 360 Degree Turn and Stabilize Down
Angle will be measured in 10 seconds
Encoder Angle is: -360.00
(Correct value should lie between -359.5 and -360.5 degrees)

Manually Rotate Pendulum in Counter Clock Wise Direction One Full 360 Degree Turn and Stabilize Down
Angle will be measured in 10 seconds
Encoder Angle is: 0.15
(Correct value should lie between -0.5 and 0.5 degrees)

Test Operation Complete, System in Standby, Press Reset Button to Restart
Test Operation Complete, System in Standby, Press Reset Button to Restart
```

10. Mac OSX Operation with screen utility

The native Mac OSX `screen` utility is ideal for interaction with the Edukit system.

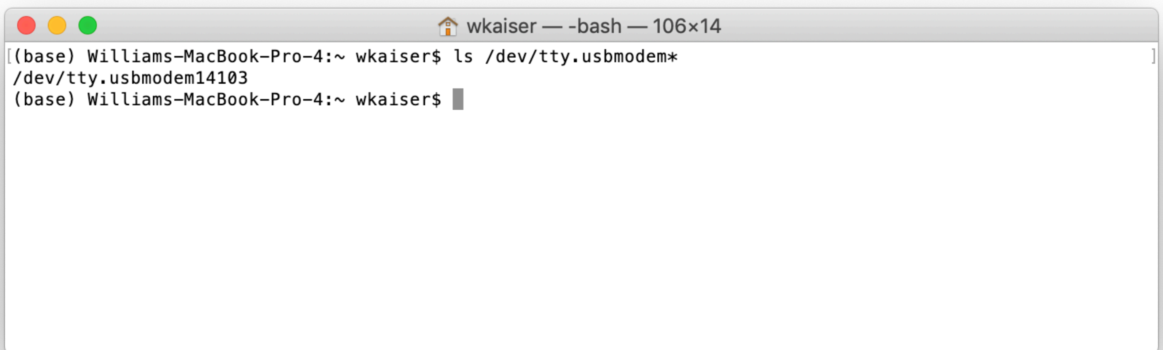
When the Edukit is connected to the Apple Mac platform, a device file interface is automatically assigned by the Max operating system to create a USB serial data interface.

This must be discovered in order to access the platform.

To determine the interface, first connect the USB cable to the Apple platform.

Then execute:

```
ls /dev/tty.usbmodem*
```



The device file interface corresponding to the serial port will appear as shown above.

This device file is needed for serial access. This will be a 4 digit value in early Mac OSX versions, a 5 digit value in Mac OSX Mojave, and a often a 16 hexadecimal character value in Mac OSX Catalina.

Now, start the `screen` utility using this device file.

The `screen` system requires two arguments, the device file and the serial baud rate. This is 230400.

Enter

```
screen /dev/tty.usbmodemXXXXX 230400
```

Where `XXXXX` is the device file number you found in the previous step.

The result will be similar to the output below. This will depend on when **screen** had been started and the current operating cycle of the Edukit.

In this example, the Edukit system has been waiting for a user action to either orient the pendulum in a vertical position to start control, or to enter a command.

Here, the user has entered a command, “1”, and the system will start in operating Mode 1.

Exiting screen

To exit **screen** , first enter Ctrl-a, then enter Ctrl-\

11. Windows Operation with putty utility

PuTTY is a serial terminal for Windows platforms. This is an open source and well-maintained software system.

Installing Putty:

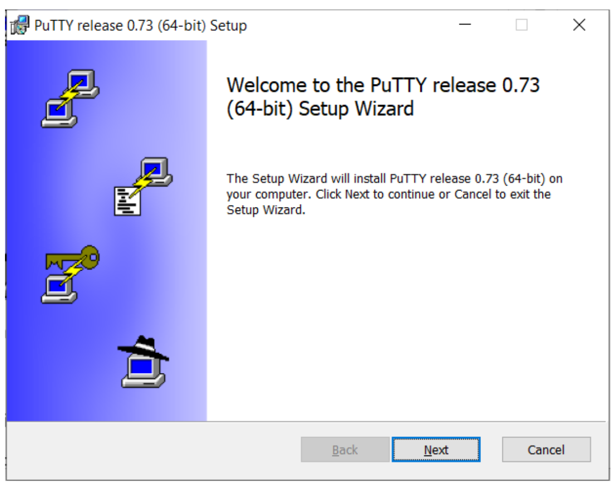
1. Download the PuTTY terminal emulator:

https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

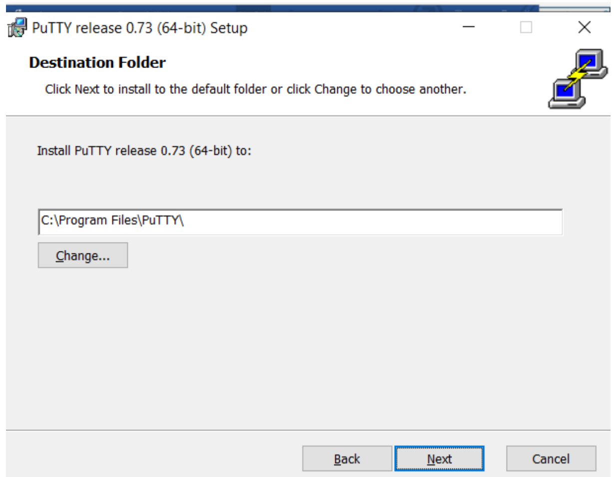
2. For most Windows machines, the proper download is:

putty-64bit-0.73-installer.msi

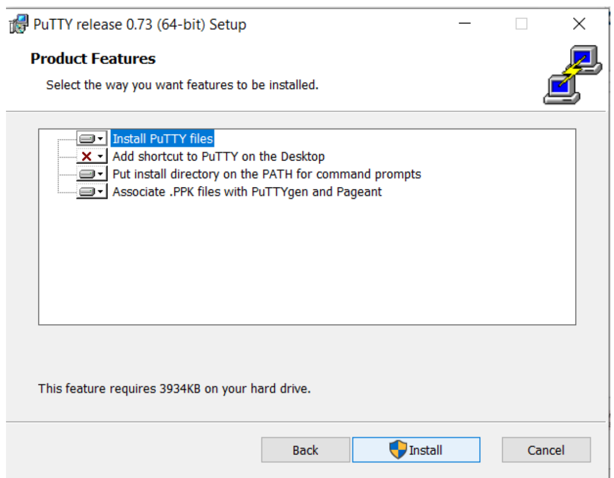
3. Click on the installer the following installation screens will appear:



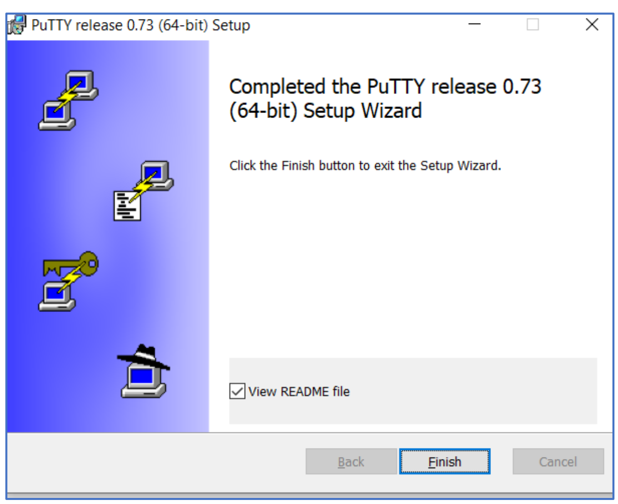
Select Next



Select the default option and Next



Select the default options above



Now, after installation, start the putty program. You may click on a Desktop icon or enter "putty" in the search box of the Windows taskbar to launch the application

Now, Windows automatically assigns a aerial communication port to the Nucleo processor USB link.

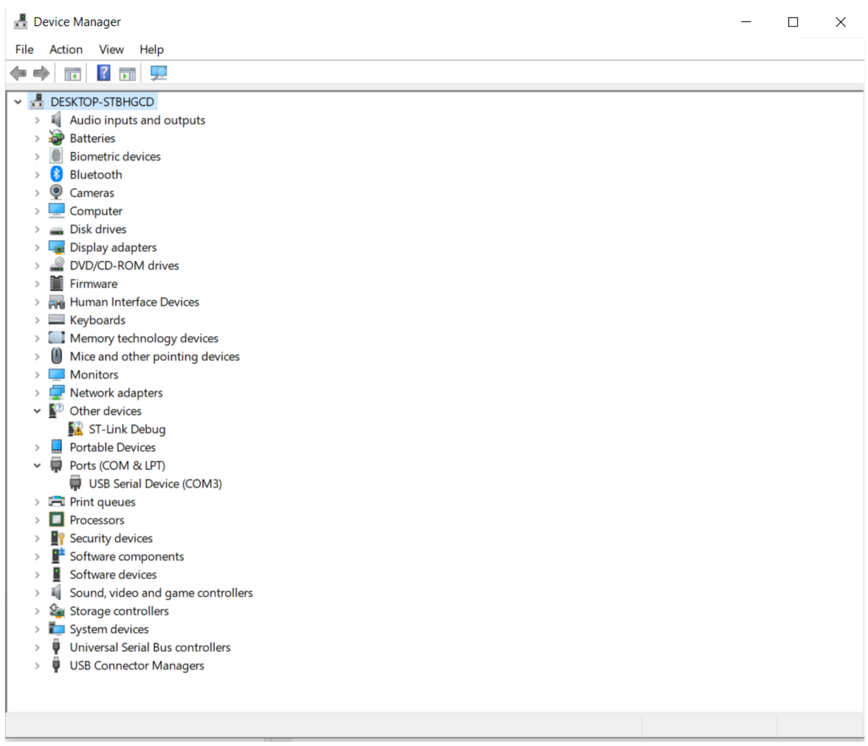
Connect the Edukit platform to your computer by the USB cable. Your computer may require an adapter between its ports and USB interfaces.

When the Edukit is connected to a Windows platform, a device file interface is automatically assigned by the Max operating system to create a USB serial data interface.

This must be discovered in order to access the platform.

To determine the interface, first connect the USB cable to the Windows platform.

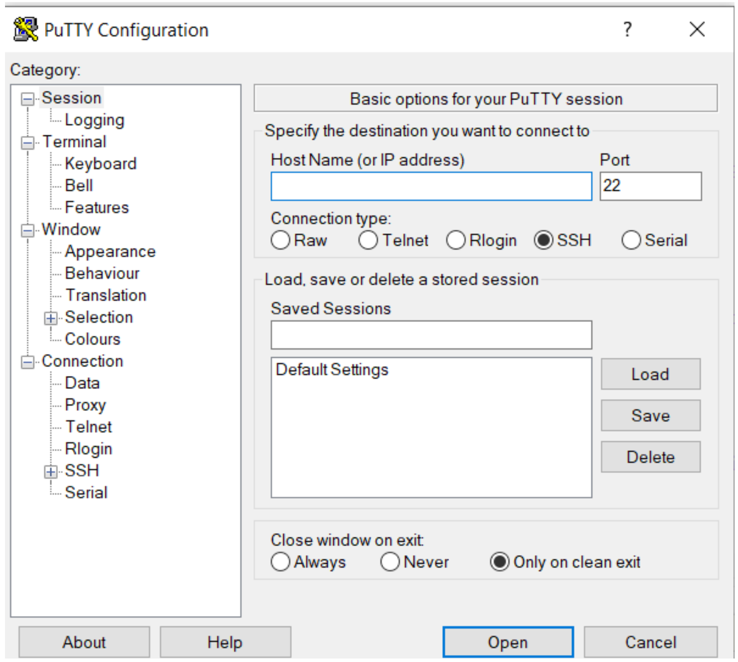
Then, navigate to the Device Manager. (You may enter Device Manager in the Windows search bar).



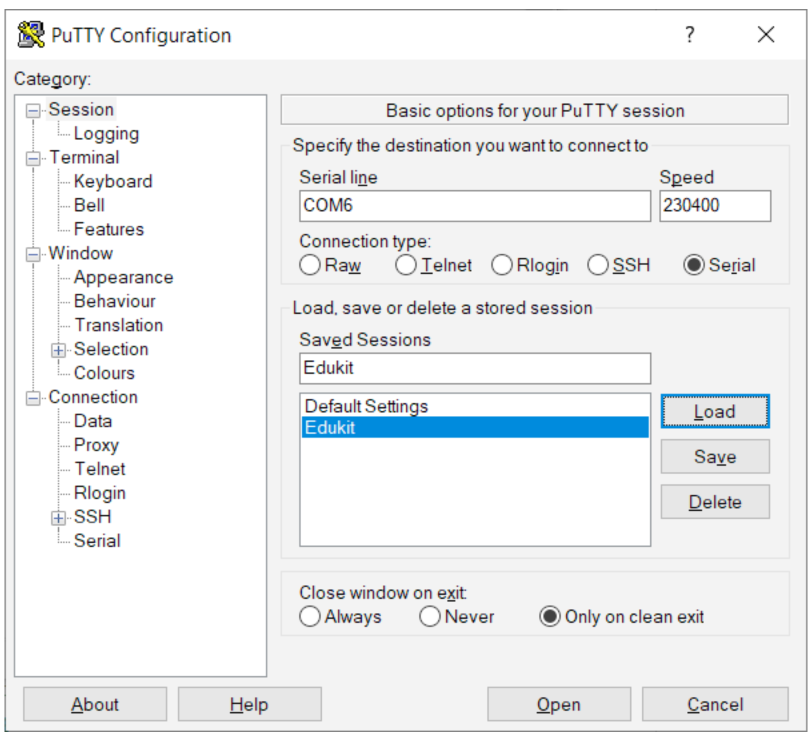
Expand the selection **Ports (COM and LPT)**

The device file interface corresponding to the serial port will appear as shown above.

This device file is needed for serial access by Putty.

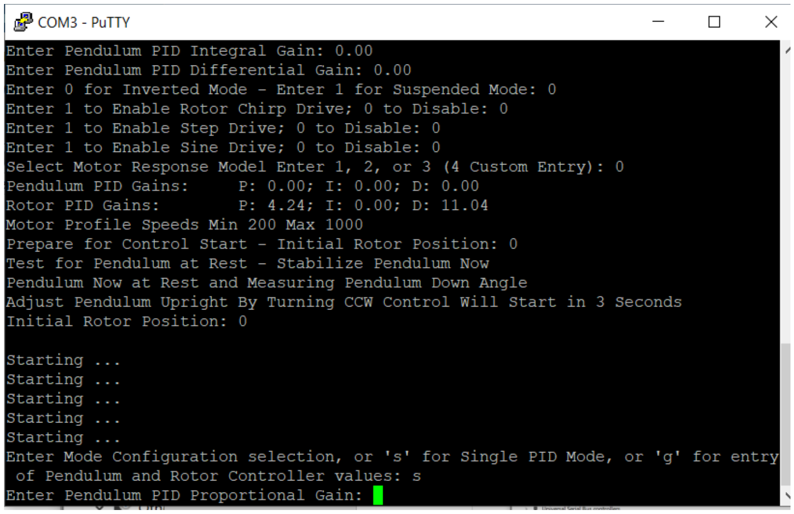


This is the default start window



*Enter the **Serial Port** identification and the **Speed** of 230400*
*This setting can also be provided with a name in the **Saved Sessions** box and then saved with the **Save** button. Later, this can be selected and loaded with **Load** – ready to run*

*Now, Select **Open***



The terminal session will appear as in this window.
Command selections and values may be entered at the keyboard.

*To exit from the session, simply click on the **X** in the upper right corner of the window. This dialog box will appear to enable*

22. Appendix D: Integrated Rotary Inverted Pendulum: Pendulum Dynamics

The transfer function between Rotor Angle and Pendulum Angle may be computed by determining the equations of motion governing Pendulum Angle response to Rotor Angle. The Inverted Pendulum mode is considered first.

Inverted Pendulum Configuration

The Edukit Rotary Inverted Pendulum, shown in *Figure 87*, includes a Pendulum rod that is uniform and has total mass, m.

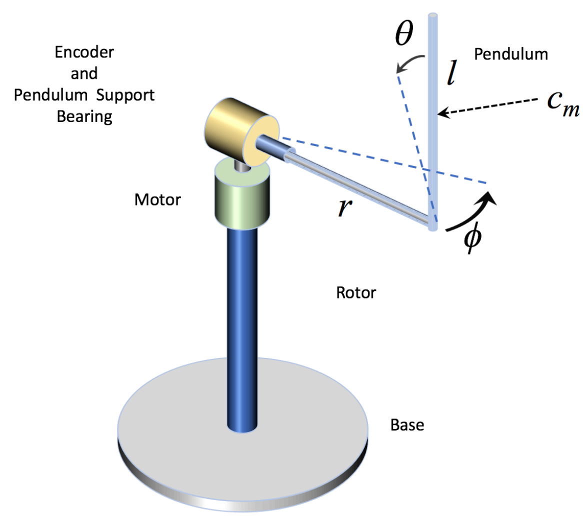


Figure 87. Edukit Rotary Inverted Pendulum with Pendulum of mass, m, composed of a uniform rod.

Now, the motion of the Rotor, ϕ,combined with rotation of the Pendulum about the Rotor, θ, induces a displacement of the Pendulum and a resulting force acting on the Pendulum center of mass, c_m due to the acceleration of the center of mass.

First, the displacement of the center of mass in the direction tangential to Rotor motion and in the horizontal plane, is

$$d_R = r\sin\phi\cos\phi - \frac{l}{2}\sin\theta$$

In the limit of small angles, this is

$$d_R = r\phi - \frac{l}{2}\theta$$

The force acting on center of mass in the direction tangential to Rotor motion and in the horizontal plane, is

$$F_R = m\ddot{d}_R = mr\ddot{\phi} - m\frac{l}{2}\ddot{\theta}$$

The force acting on the center of mass in the vertical direction results from the acceleration due to gravity and is

$$F_V = mg\sin\theta$$

In the limit of small angles, this is

$$F_V = mg\theta$$

This force presents a torque contribution equal to the product of the half-length, $l/2$, and the force acting on the center of mass at $l/2$.

A second torque contribution acting on the pendulum at its point of support is due to friction. This is in a direction opposite to angular velocity and proportional to the product of angular velocity and a damping coefficient, γ.

The torque acting on the Pendulum is the combination of all forces acting on the Pendulum at the location of $l/2$. Considering the torque acting in the direction of positive θ, including the damping force acting on rotation of the pendulum,

$$\tau_{Pend} = \frac{l}{2}F_V + \frac{l}{2}F_R - \frac{l}{2}\gamma\dot{\theta} = m\frac{l}{2}g\theta + m\frac{l}{2}r\ddot{\phi} - m\frac{l^2}{4}\ddot{\theta} - \gamma\dot{\theta}$$

Also, the Pendulum torque, including the damping force on Pendulum rotation as well as the inertial force due to angular acceleration about the Pendulum angle, is

$$\tau_{Pend} = (I_{Pend} + I_{Rotor})\ddot{\theta}$$

Where, I_{Pend} is the moment of inertia of the Pendulum, and I_{Rotor} is the moment of inertia of that part of the rotor shaft attached to the encoder and associated with its rotation about its coaxial axis. This moment of inertia, of course, is small compared to I_{Pend}.

Finally, the equation of motion is

$$(I_{Pend} + I_{Rotor})\ddot{\theta} = m\frac{l}{2}g\theta + m\frac{l}{2}r\ddot{\phi} - m\frac{l^2}{4}\ddot{\theta} - \frac{l}{2}\gamma\dot{\theta}$$

The moment of inertia of the Pendulum is that of a uniform rod rotating about its end,

$$I_{Pend} = m\frac{1}{3}l^2$$

and rearranging

$$\left(\frac{2}{3} + I_{Rotor}\right)\ddot{\theta} = \frac{g}{l}\theta + \frac{r}{l}\ddot{\phi} - \frac{\gamma}{ml}\dot{\theta}$$

The Laplace transform is

$$\left(\frac{2}{3} + I_{Rotor}\right)\theta s^2 = \frac{g}{l}\theta + \frac{r}{l}\phi s^2 - \frac{\gamma}{ml}\theta s$$

The resulting transfer function for Rotor Angle to Pendulum Angle is

$$G_{pend}(s) = \frac{\theta(s)}{\phi(s)} = \frac{{\omega_r}^2 s^2}{s^2 + {}^{\omega_g}/_{Q}\, s - {\omega_g}^2}$$

with

$$\omega_r = \sqrt{\frac{r}{l\left(\frac{2}{3} + \frac{2I_{Rotor}}{ml^2}\right)}}; \qquad \omega_g = \sqrt{\frac{g}{l\left(\frac{2}{3} + \frac{2I_{Rotor}}{ml^2}\right)}}$$

and

$$Q = \omega_g\left({}^{ml}/_{\gamma}\right)$$

Suspended Pendulum Configuration

The computation of the transfer function between Rotor Angle and Pendulum Angle follows, similarly to that of the Inverted Mode.

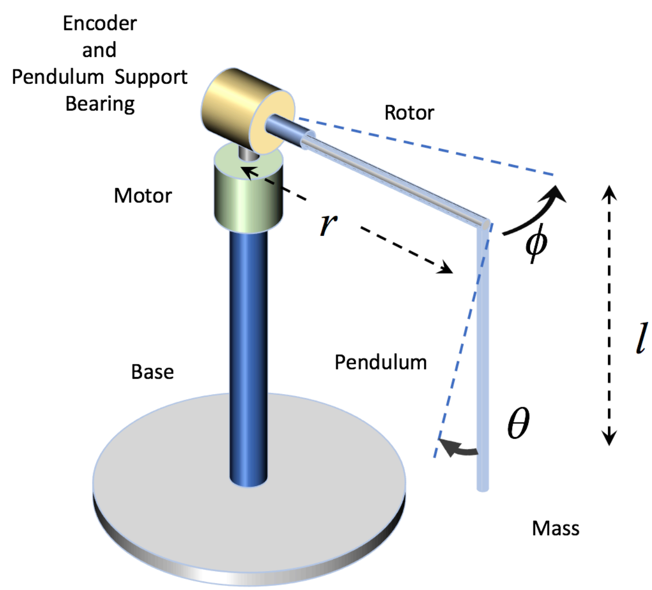


Figure 88. Edukit Rotary Inverted Pendulum in Suspended Mode with Pendulum of mass, m, composed of a uniform rod.

First, the displacement of the center of mass in the direction tangential to Rotor motion and in the horizontal plane, is

$$d_R = r\sin\phi\cos\phi - \frac{l}{2}\sin\theta$$

In the limit of small angles, this is

$$d_R = r\phi - \frac{l}{2}\theta$$

The force acting on center of mass in the direction tangential to Rotor motion and in the horizontal plane, is

$$F_R = m\ddot{d}_R = mr\ddot{\phi} - m\frac{l}{2}\ddot{\theta}$$

The force acting on the center of mass in the vertical direction results from the acceleration due to gravity, is

$$F_V = mg\sin\theta$$

In the limit of small angles, this is

$$F_V = mg\theta$$

This force presents a torque contribution equal to the product of the half-length, $l/2$, and the force acting on the center of mass at $l/2$.

A second torque contribution acting on the pendulum at its point of support is due to friction. This is in a direction opposite to angular velocity and proportional to the product of angular velocity and a damping coefficient, γ.

The torque acting on the Pendulum is the combination of all forces acting on the Pendulum at the location of $l/2$. Considering the torque acting in the direction of positive θ, including the damping force acting on rotation of the pendulum,

$$\tau_{Pend} = -\frac{l}{2}F_V + \frac{l}{2}F_R - \frac{l}{2}\gamma\dot{\theta} = -m\frac{l}{2}g\theta + m\frac{l}{2}r\ddot{\phi} - m\frac{l^2}{4}\ddot{\theta} - \frac{l}{2}\gamma\dot{\theta}$$

Also, the Pendulum torque induces angular acceleration about the Pendulum angle of

$$\tau_{Pend} = (I_{Pend} + I_{Rotor})\ddot{\theta}$$

Finally, the equation of motion is

$$(I_{Pend} + I_{Rotor})\ddot{\theta} = -m\frac{l}{2}g\theta + m\frac{l}{2}r\ddot{\phi} - m\frac{l^2}{4}\ddot{\theta} - \frac{l}{2}\gamma\dot{\theta}$$

The moment of inertia of the Pendulum is that of a uniform rod rotating about its end,

$$I_{Pend} = m\frac{1}{3}l^2$$

and rearranging

$$\left(\frac{2}{3} + I_{Rotor}\right)\ddot{\theta} = -\frac{g}{l}\theta - \frac{\gamma}{ml}\dot{\theta} + \frac{r}{l}\ddot{\phi}$$

The Laplace transform is

$$\left(\frac{2}{3} + I_{Rotor}\right)\theta s^2 = -\frac{g}{l}\theta - \frac{\gamma}{ml}\theta s + \frac{r}{l}\phi s^2$$

The resulting transfer function for Rotor Angle to Pendulum Angle is

$$G_{pend}(s) = \frac{\theta(s)}{\phi(s)} = \frac{{\omega_r}^2 s^2}{s^2 + {}^{\omega_g}/_{Q}\, s + {\omega_g}^2}$$

with

$$\omega_r = \sqrt{\frac{r}{l\left(\frac{2}{3} + \frac{2I_{Rotor}}{ml^2}\right)}}; \qquad \omega_g = \sqrt{\frac{g}{l\left(\frac{2}{3} + \frac{2I_{Rotor}}{ml^2}\right)}}$$

and

$$Q = \omega_g \left({}^{ml}/_{\gamma} \right)$$

The moment of inertia, I_{rotor}, may be found by system identification. Specifically, the value of ω_g may be found by measuring the oscillation frequency of the Pendulum in the Suspended Mode. Also, quality factor, Q, may be estimated. With this value of ω_g and the known dimensions of the Rotor and Pendulum, ω_r may be computed.

This provides the transfer function, $G_{pend}(s)$ including all contributions to the moment of inertia.

$$L = 0.258m$$
$$r = 0.141m$$
$$g = 9.806\ m/s^2$$

$$\left(\frac{2I_{rotor}}{mL^2}\right) = \frac{g}{L(1.19Hz\ \times 2\pi\ rad/sec)^2} - \frac{2}{3} = 0.0139$$

Finally,

$${\omega_r}^2 = \frac{r}{L\left(\frac{2}{3} + \frac{2I_{rotor}}{mL^2}\right)} = \frac{r}{L(0.6805)} = 0.803(rad/sec)^2$$

$${\omega_g}^2 = \frac{g}{L\left(\frac{2}{3} + \frac{2I_{rotor}}{mL^2}\right)} = \frac{g}{L(0.6805)} = 55.85(rad/sec)^2$$

This may be compared with values computed by neglecting the rotor moment of inertia

$$\omega_r{}^2 = \frac{r}{L\left(\frac{2}{3}\right)} = 0.803(rad/sec)^2$$

$$\omega_g{}^2 = \frac{g}{L\left(\frac{2}{3}\right)} = 57.01(rad/sec)^2$$

The resulting transfer function is

$$G_{pend}(s) = \frac{\theta(s)}{\phi(s)} = \frac{\omega_r{}^2 s^2}{s^2 + {}^{\omega_g}/_{Q}\, s + \omega_g{}^2}$$

The Pendulum Angle Step Response was also measured by application of a step signal of amplitude 40 degrees to the Rotor Angle at $t = 0$ and then sampling Pendulum Angle over time. This applies the data logging feature of the Real Time Control Systems Workbench. Then, using the methods developed for deriving Sensitivity Function Transfer Function Frequency Response, Pendulum Plant Transfer Function was computed. The comparison between simulated response of the Pendulum Plant model defined here and the measured response is shown in Figure 89.

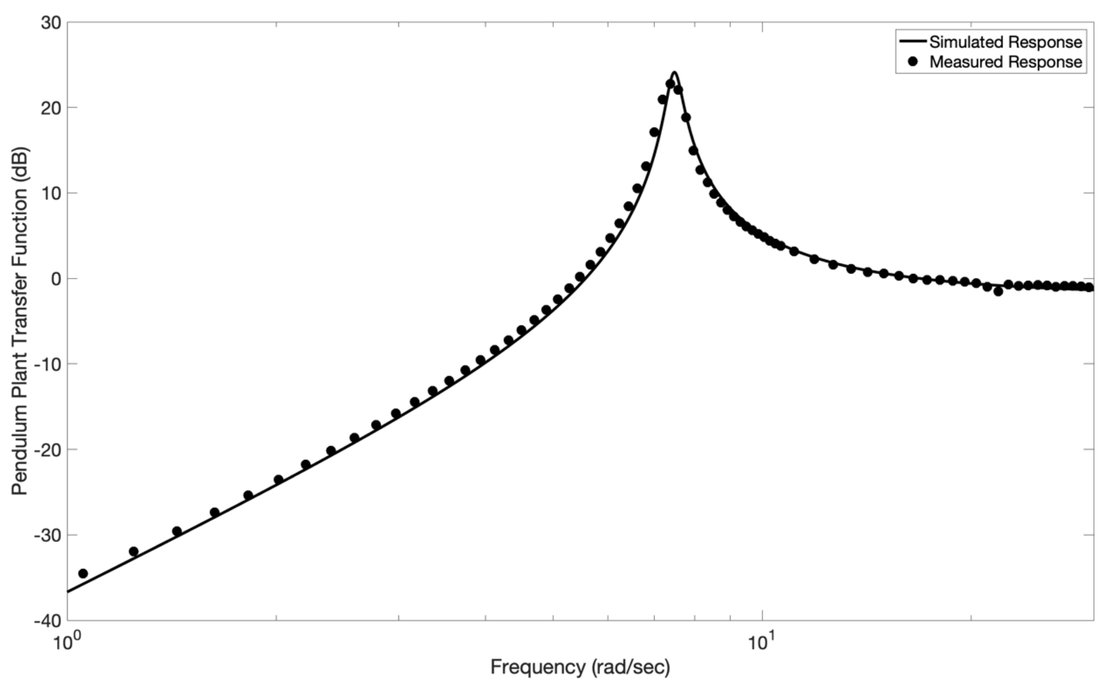


Figure 89. The Edukit Rotary Inverted Pendulum Plant Transfer Function is shown with comparison of Simulated and Measured Response. The agreement between simulation and measured response provides verification of the accuracy of the Pendulum Plant model..

The Pendulum Angle Step Response was measured by application of a step signal of amplitude 40 degrees to the Rotor Angle tracking command and then sampling Pendulum Angle over time.

The computed transfer function is shown for a value of Q = 20 in Figure 89. This permits us to estimate that the Q value of the Edukit Edukit Rotary Inverted Pendulum System is also near 20.

This large value permits most design approaches to ignore the effect of damping on Pendulum Angle motion.

23. Appendix E: Rotor Actuator Plant with Acceleration Control

Introduction

This section describes Stepper Motor operation for both conventional systems as well as the new control interface included in the Edukit System. This new interface is based on a precise digital control interface based on the IHM01A1 Motor Controller integrated circuit included in Edukit. This digital control algorithm produces a high performance motor actuator with a transfer function that is a double integrator with unity gain. Verification of this response is described in the next section, Appendix F: Rotor Actuator Plant Transfer Function.

Stepper Motor Operation in High Speed Control Systems

Conventional robotic control systems requiring broad band frequency response were developed with electric motor systems based on analog actuation and providing torque control. Control system design tools were also developed with reliance on linear models describing plants including these electric motor actuators. Complete systems required integration of rotary angle encoders along with the motor actuator. Motor speed limitations determined by electromagnetic response and inertial forces were accurately described by linear multi-order transfer functions.

Digital motor control systems enable a wide range of advantages including precision, reproducibility, configurability, and integration of actuation and rotary angle sensing. However, in high performance control systems, the digital motor actuator presents a maximum speed limit exhibiting as a maximum digital position step rate. This maximum step rate creates a slew rate limit in motor control. Slew rate limiting, in turn, introduces nonlinear response. Specifically, frequency response characteristics of motor operation in following a rotor command signal is dependent on command signal amplitude due to slew rate limiting. The rotor position follows the rotor command signal for low amplitude (and low speed) while lagging the rotor command signal amplitude for high amplitude.

The slew rate limitation of digital motor actuators presents a challenge to control system design. Specifically, a selection must be made for the digital motor actuator minimum speed, maximum speed, and acceleration characteristics. For each choice system performance is determined along with nonlinear response characteristics that may complicate design.

As an illustration, a design consideration is the selection of stepper motor minimum speed for a closed-loop robotic control system relying on stepper motor actuation. Selection of a high minimum speed value enhances the response of the control system to a tracking command by enabling high slew rate. However, for steady state operation where a variation in a tracking command does not appear, the high minimum speed also induces noise contributed by the actuator into the control system. Specifically, for a steady state, constant control system tracking command, the control system may not provide low speed motion and therefore desired low noise operation. Rather, small disturbances in the control loop result in large amplitude motion since minimum speed is large. As an alternative choice, a stable, low noise system may be implemented with a low stepper motor minimum speed. However, this reduces the performance of the closed loop system to a rapidly changing tracking command signal. Finally, the conventional approach for speed control does not permit sufficiently rapid changes in motor speed.

While limitations appear, successful and high-performance controllers may be implemented with slew rate limited systems. The fundamental nonlinear response of the step rate limited stepper motor system has been addressed by a system identification method producing a model of stepper motor response for amplitudes characteristic of a closed loop system.

These challenges have inspired a new stepper motor actuation system. This system provides all of the advantages of high and low speed operation while also directly reducing nonlinearity in response. This also introduces a control system plant characteristic familiar to control systems engineers.

This introduces a new motor actuation plant that accepts a control input as an acceleration command, as opposed to a position command. The acceleration command is integrated with respect to time to produce a velocity. This is conveniently introduced as simply a new plant model, now including an integrator term, in control system design. The integrator operates at the rate of the robotic control system. Ultimately, the time integral of acceleration values results in position change.

This new integrating stepper motor controller is well-supported by the STMicroelectronics L6474 Stepper Motor Controller. A new method and set of new functions are hosted on the STM32 microprocessor that supports the closed loop control system and interfaces to the L6474.

The new integrating stepper motor controller seeks to enable highly agile motor speed change from a minimum of 2 steps per second to over 65,000 steps per second with also the most rapid acceleration of over 65,000 steps per second per second. This provides not only nearly the lowest possible speed, but also the highest.

A critical component of this development is the introduction of a method enabling rapid change in stepper motor step rate and therefore, velocity. Velocity is adjusted by directly controlling the period of the pulse width modulator (PWM) that controls stepper motor microstepping. The PWM period may be changed with a response time that is short compared to the control system operating cycle and is, therefore, effectively instantaneous relative to the control loop cycle rate. This Appendix describes the new interface.

Stepper Motor PWM Signal Generation Background

The L6474 Motor Driver applies Pulse Width Modulation (PWM) methods for generation of sinusoidal motor coil current waveforms enabling microstepping control as shown in **Figure 90.** This is enabled with auto-adjusted decay of current switching for accurate and low noise motion. [STMicroelectroics_1_2020]

As described in the L6474 Datasheet, the motor movement is defined by the step clock signal applied to the STCK pin. As shown in **Figure 91,** at each step clock rising edge, the motor is moved by one microstep in the direction selected by DIR input (high for forward direction and low for reverse direction) and absolute position is consequently updated.[STMicroelectroics_2_2020]

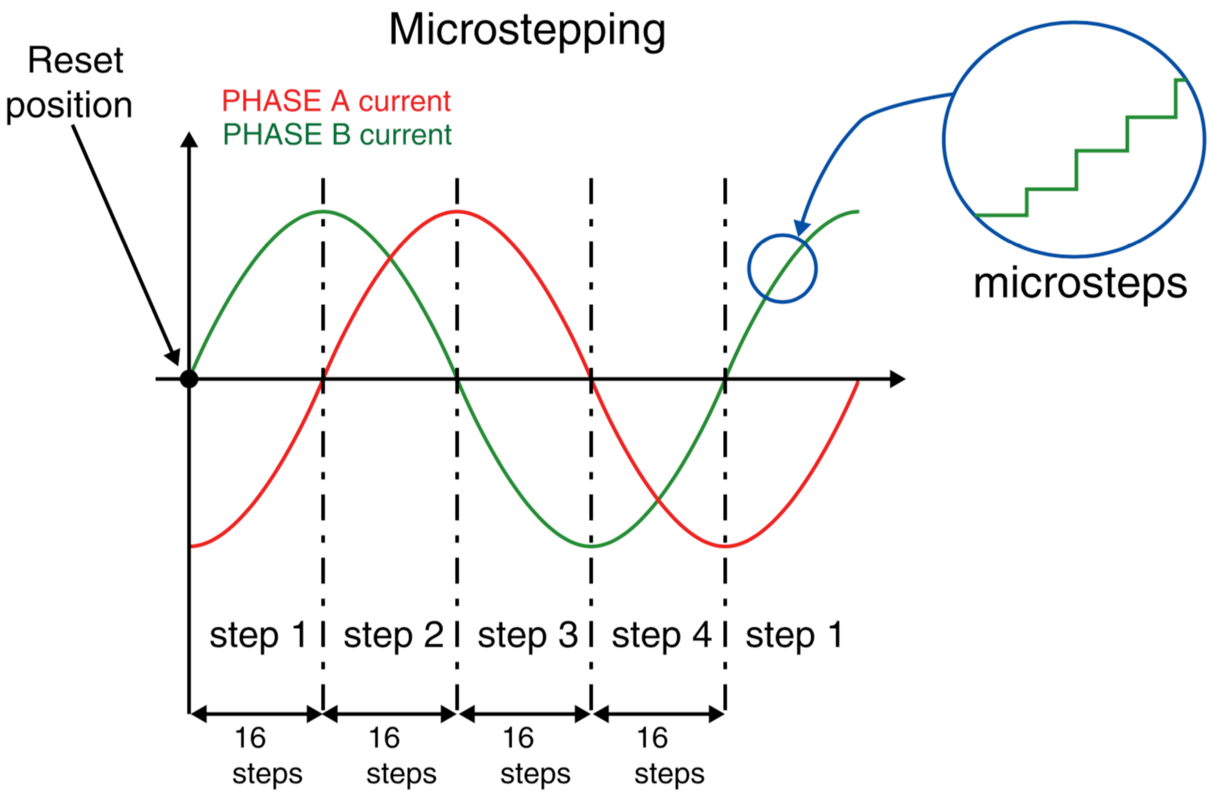


Figure 90: *The L6474 Motor Driver Microstepping current profile [STMicroelectroics_1_2020].*

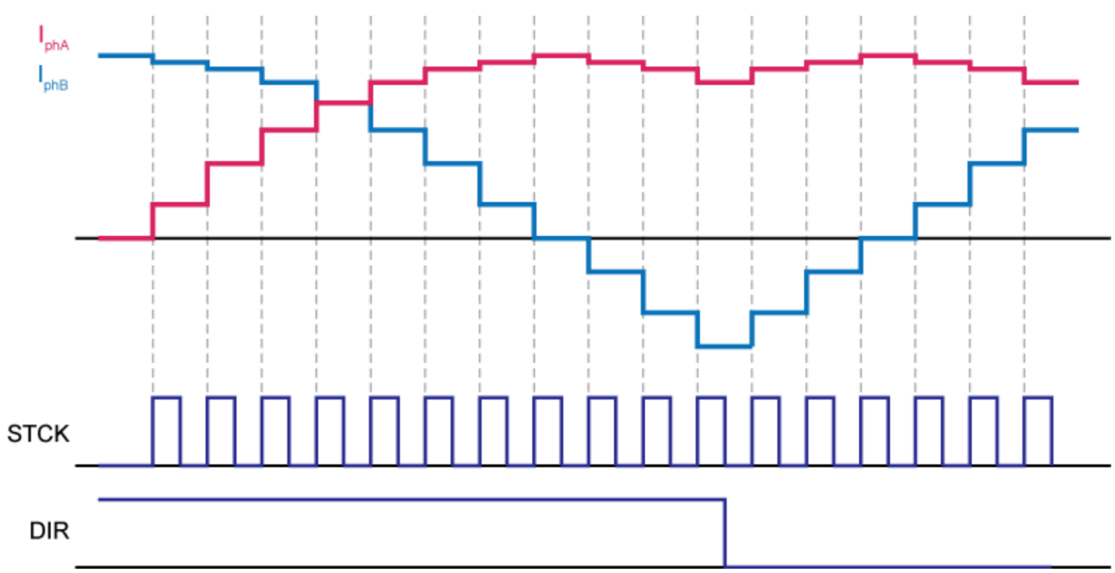


Figure 91: *The L6474 Motor Driver Microstepping schedule [STMicroelectroics_2_2020].*

The L6474, IHM01A1 Interface, and the STM32 Nucleo system provide microstepping capability through the STMicroelectronics PWM system. This determines cycle rate through a process that establishes counter values, referred to as Compare Value 1 and Compare Value 2 along with a threshold value as shown in **Figure 92**. The counter clock rate is determined by the internal clock frequency of 84MHz, and a prescaler divider of 1024 to generate an 82kHz counter input signal.

Figure 92 describes the method by which the PWM signal period is generated in the SPN1 system. Here, the counter counts up until a threshold is reached, and then resets to 0. At this reset, the PWM signal rising edge is triggered. The PWM falling edge and the PWM duty cycle are determined by the event at the time when the counter value exceeds Compare Value 1, and for a second PWM signal, Compare Value 2. The value of the PWM duty cycle determines the motor position within a microstep cycle.

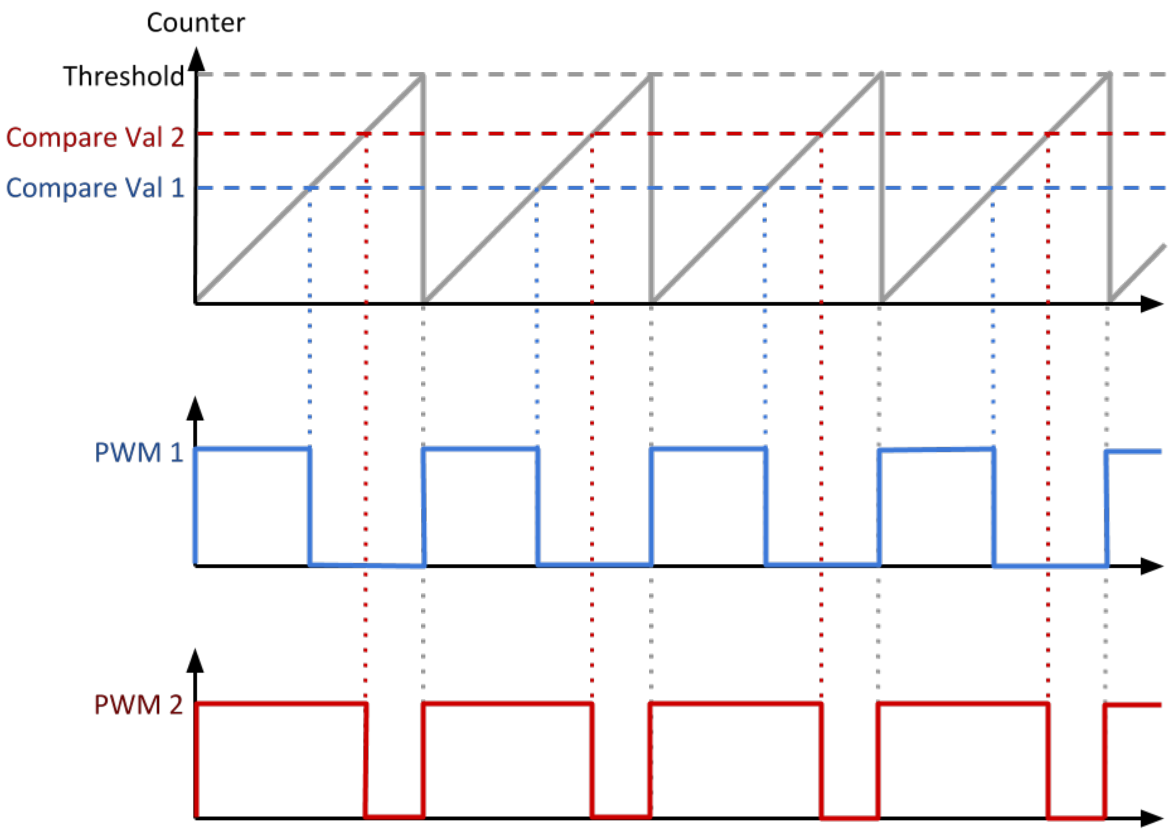


***Figure 92**: PWM signal generation using a counter. Two compare values are shown, resulting in two PWM signals with the same period but different duty cycles.*

Stepper Motor Microstep Rapid Speed Control

Stepper motor step rate, by the standard method, is configured with a minimum speed and an acceleration rate. A new method is introduced here to enable wide dynamic range control of speed along with rapid change in speed.

This takes advantage of the STM32 processor and timer function for generation of interrupt events. Specifically, this configures an interrupt to be triggered when the counter reaches its compare value. At this value, the GPIO STCK pin is toggled, generating a step command to the L6474. This toggle action has the effect of a clock divider: the output frequency is half the frequency of the internally generated PWM signal, and it is always at 50 percent duty cycle. This is described in **Figure 93**.

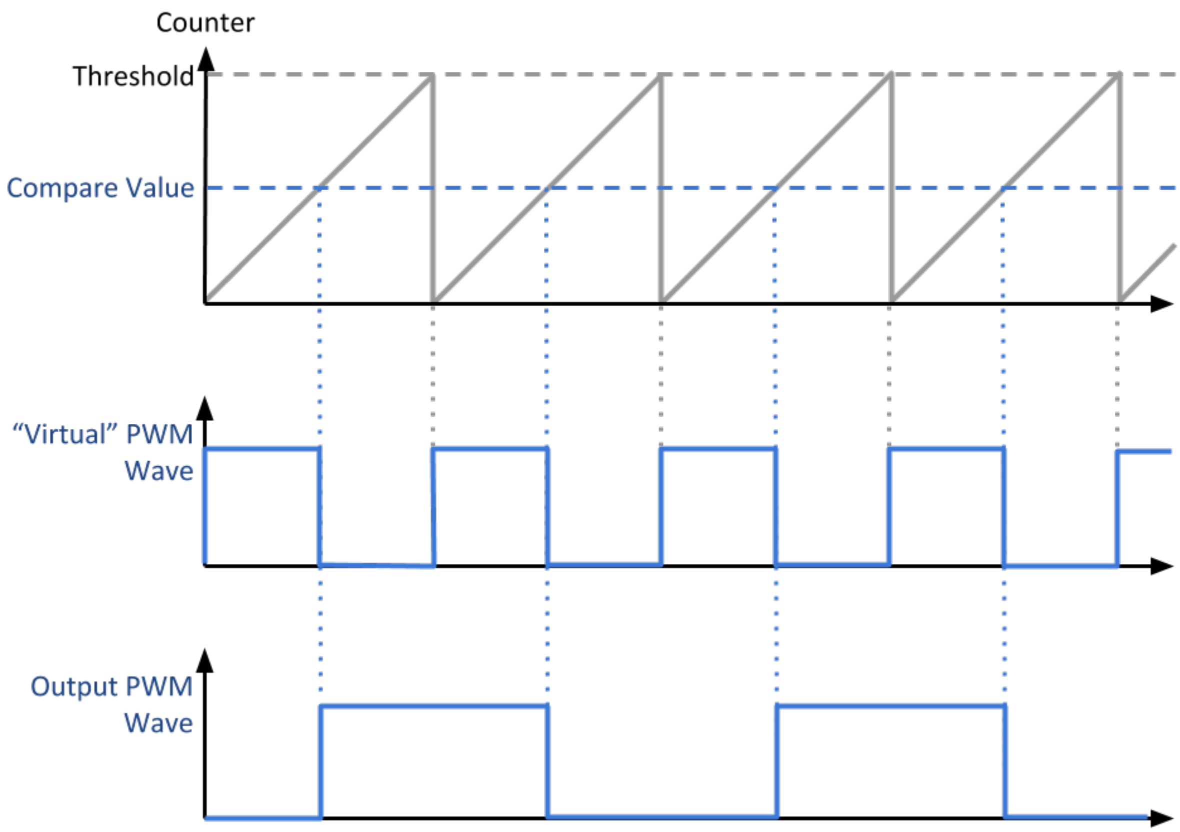


Figure 93. *PWM step signal generation using the toggling method shown for the example of a 50 percent duty cycle waveform. The output PWM wave is simply generated by toggling the output each time the counter reaches the compare value, giving it half the frequency of the "virtual" PWM wave and 50 percent duty cycle.*

Stepper Motor Microstep Rapid Speed Control: Limitations

The requirement for control of motor acceleration requires, in turn, that the PWM cycle rate be changed as often as feasible to yield minimal motion noise. One solution to this is to update the PWM cycle rate at the end of each PWM pulse, or more specifically, in the interrupt service routine that is triggered when the counter reaches the compare value. This approach is effective for incremental change in speed.

However, a limitation exists for this approach under the condition of a large change in speed, as demonstrated in **Figure 94**. When the PWM period is scaled by a factor that is less than the duty cycle percentage, the counter threshold fails to take effect. For example, at a duty cycle of 50%, reducing the period by more than half will cause the threshold to drop below the current count value. The counter will continue to increment out of bounds.

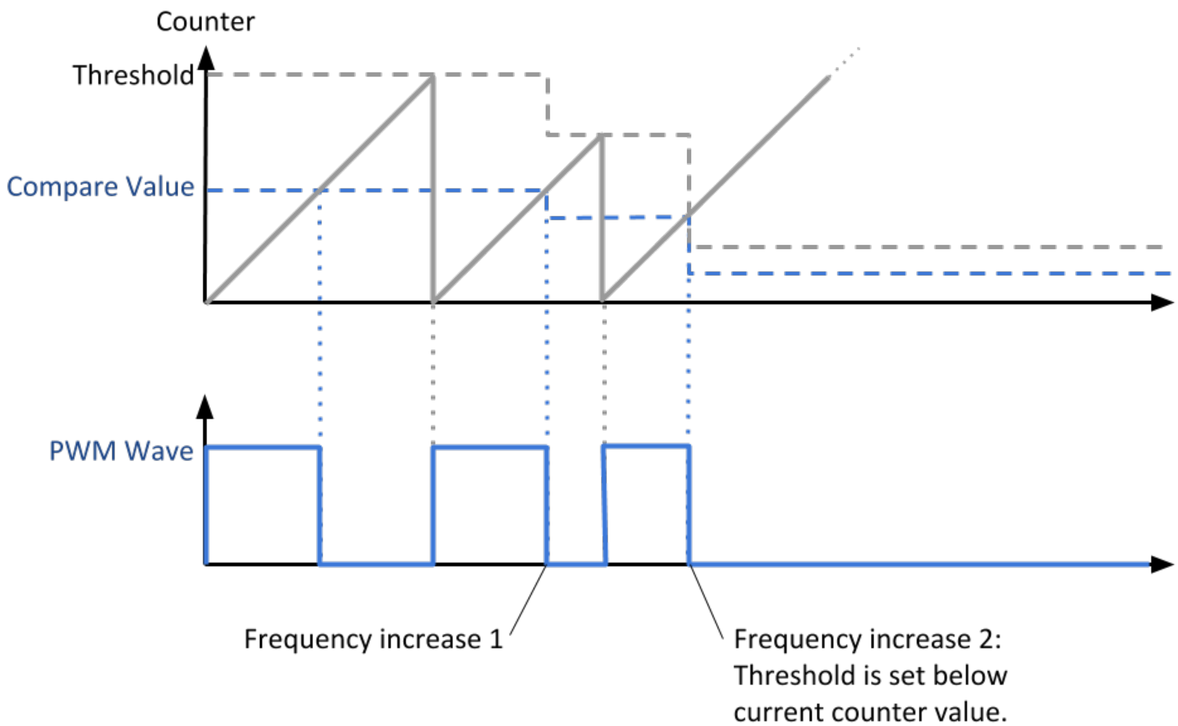


Figure 94. *This example demonstrates modulation of PWM cycle period under the condition of small and large speed change controlled by frequency change. This shows two frequency increase events that are intended to be made by changing period at the PWM pulse interrupt. The first change executes as desired. The frequency increase in the second example attempts to increase frequency to the extent that the required new counter threshold is set below the current counter value, thus preventing detection fof counter threshold crossing and creation of an operation fault*

This limitation may not appear for small acceleration rates corresponding to increase in speed from sufficiently high initial speed values. However, this limitation appears, for fine control over acceleration even at very low speeds. To illustrate, consider an example where the initial (minimum) speed, v_1, is 5 steps/sec, and we wish to accelerate at a rate, acceleration, a, of 50 steps/sec^2 over a time interval, Δt. In the interrupt for the PWM pulse commanding the first step, it is known that 1/5th of a second has passed since the initial speed is 5 steps/sec. Then, following the computation of a new speed, v_2 , is $v_2 = v_1 + a\Delta t$ to obtain a speed of $5 + 50 \times 1/5 = 15$. An increase in speed to a desired value of 15 steps/sec is more than twice as fast as the current speed, so the PWM period, and therefore threshold, will be reduced by greater than one-half. This creates the fault situation similar to that of Frequency Increase 2 in **Figure 94**.

Stepper Motor Microstep Rapid Speed Control: Solution

The new method for Stepper Acceleration Rate Control is based on the methods supported by the X-CUBE-SPN1 stepper motor software. In this method, acceleration can be adjusted at any time. Acceleration and corresponding new speed values take effect at the time when a PWM pulse interrupt is triggered. In the interrupt service routine, a new speed is computed, $v_2 = v_1 + a\Delta t$ for acceleration, a, and time interval, Δt. Now, it is noted that PWM frequency and stepper motor speed are equal. Thus, the frequency values for the initial speed, f_1, and final speed, f_2, may be computed as $f_2 = f_1 + a\Delta t$. Finally, since the time elapsed since the last pulse equal to the PWM period and where the period for the PWM frequency $1/f$, the frequency may be computed as $f_2 = f_1 + a/f_1$. Finally, the new counter threshold can then be calculated as $f_{counter}/f_2$, where $f_{counter}$ is $84MHz/1024 \cong 82kHz$. (It is noted also that in this toggling method, the frequency division caused by the toggling must be included in computation.)

There is yet one more limitation to consider, specifically at very low speed. This is simply because at low speeds, PWM pulses do not occur at high rate and thus there will be a significant delay between the time a desired acceleration is set by the control loop and when this acceleration takes effect on the next PWM pulse. This is most apparent where abrupt change from low speed to high speed at high acceleration rate is required. Thus, a solution to this is valuable for high speed control systems that benefit from wide dynamic range in speed combined with highly agile speed control.

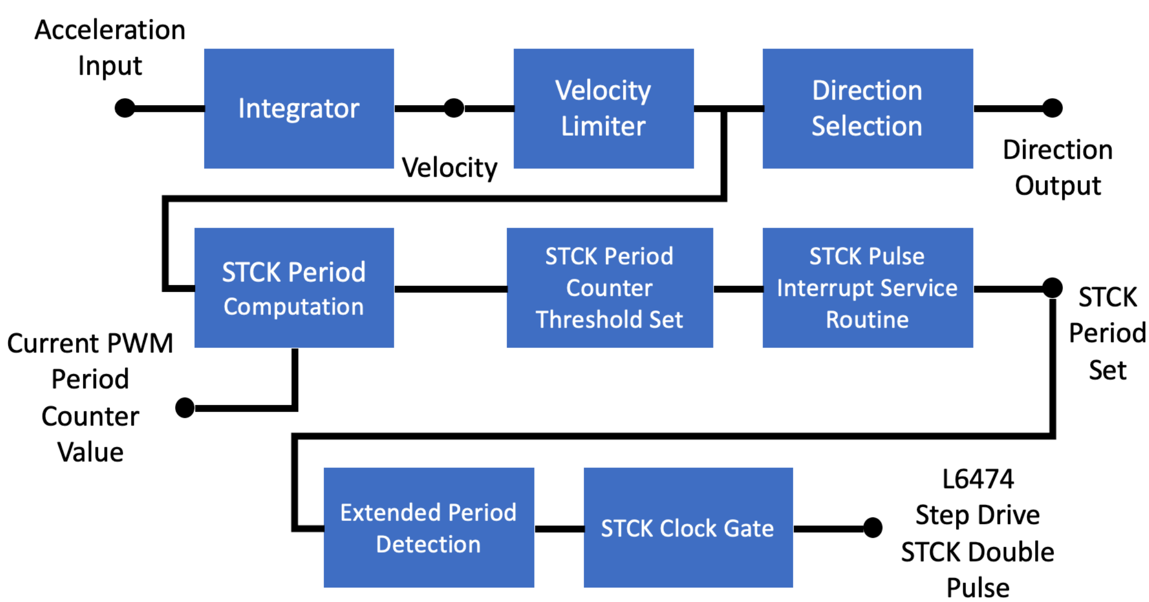


Figure 95. The block diagram of the Integrator Mode Microstep Speed Control is shown. A control input is provided as the Acceleration Input. This is integrated to compute a velocity that is compared against maximum limits by the Velocity Limiter. Velocity direction is determined and configures the L6474 direction selection. The velocity is provided to the STCK Period Computation that acquires the current PWM counter values. A new STCK Period Counter Threshold is set. When the STCK period is completed, a an STCK Pulse Interrupt Service Routine executes. In the event of a signal change less than a threshold, the STCK Clock Period is extended and an STCK Clock Gate is closed, halting Stepper Advance. If Extended Period Detection is not satisfied, the STCK Clock Gate is open and a double pulse is provided to the L6474 to advance the Stepper Motor.

Integrator Mode Stepper Motor Microstep Speed Control

The Integrator Mode Stepper Motor Speed Control method also relies on a constant-frequency control loop that will be present in control systems integrating the Stepper Motor as a digital actuator. Specifically, during each cycle of the control loop response to control signals will require update of acceleration values. Ultimately, the time integral of acceleration values results in position change.

A block diagram of this new system is shown in **Figure 95**. Acontrol input is provided as the Acceleration Input. This is integrated to compute a velocity that is compared against maximum limits by the Velocity Limiter. Velocity direction is determined and configures the L6474 direction selection. The velocity is provided to the STCK Period Computation that acquires the current PWM counter values. A new STCK Period Counter Threshold is set. When the STCK period is completed, a an STCK Pulse Interrupt Service Routine executes. In the event of a signal change less than a threshold, the STCK Clock Period is extended and an STCK Clock Gate is closed, halting Stepper Advance. If Extended Period Detection is not satisfied, the STCK Clock Gate is open and a double pulse is provided to the L6474 to advance the Stepper Motor.

In a method similar to the system described above, the next motor speed is calculated as $v_2 = v_1 + a\Delta t$, where Δt is now a constant, which we will label as T_S (sampling period). To avoid the limitation associated with large change in counter threshold due to large speed increase, the new, commanded, motor speed is not supplied instantly as received, rather this is set in the PWM pulse interrupt service routine as before. So obviously, this does not yet solve the problem of accelerating at low speeds.

However, the current value of the PWM counter *may be read* inside the loop, allowing avoidance of the limitation and producing motion that is as accurate as possible at the same time. This new method is based on an approach where the time remaining until the start of the next pulse, τ, is measured. While executing the loop, accurate motion control may be obtained by proactively updating τ. The following relations are used:

$$v_2 = v_1 + aT_S$$

$$\tau_2 = \tau_1 \left(\frac{v_1}{v_2}\right)$$

where, τ_1 is the time remaining within the PWM period *prior* to the next interrupt, and τ_2 is the time remaining *corresponding to the new speed.*

To illustrate, if a time remaining until the next pulse starts is τ_1, and $v_2$is twice the speed of v_1, then the time remaining should be decreased according to $\tau_2 = \tau_1/2$. This generalizes for any ratio between speeds.

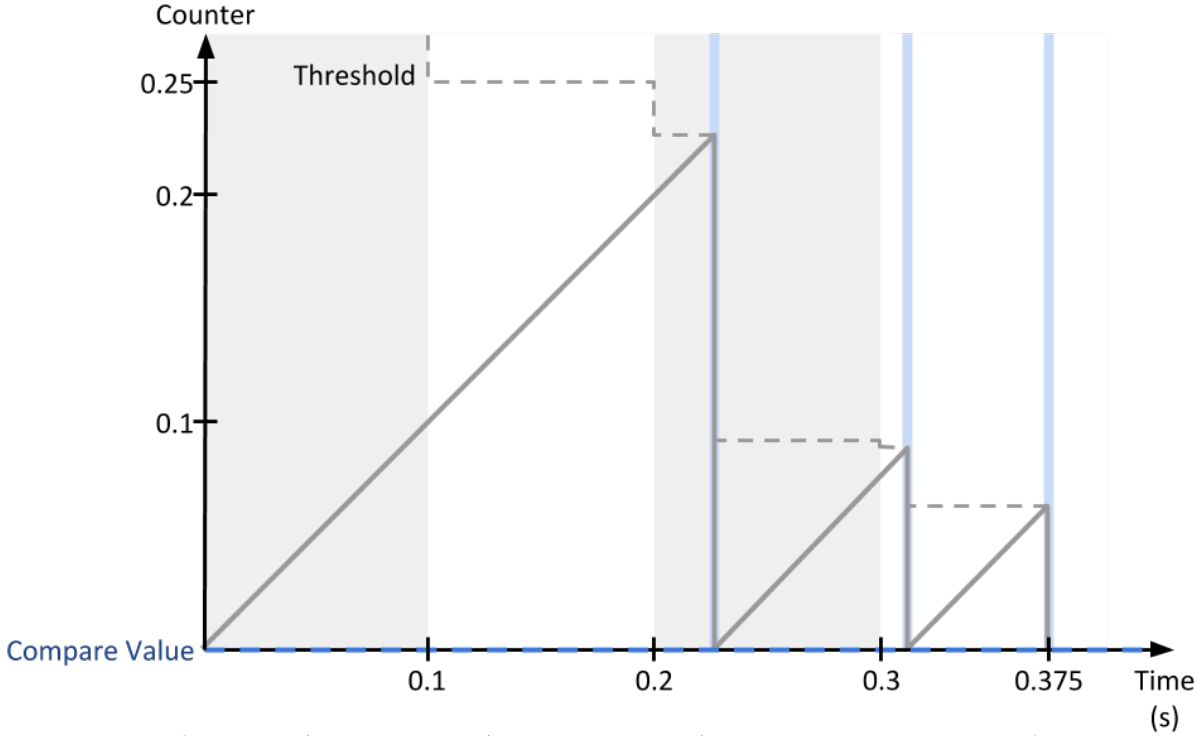
Since time remaining is directly proportional to the difference between the threshold values for each of the speeds, TH_1 and TH_2, respectively, and the current counter value, CNT, then

$$TH_2 - CNT = (TH_1 - CNT)\left(\frac{v_1}{v_2}\right)$$

And finally,

$$TH_2 = CNT + (TH_1 - CNT)\left(\frac{v_1}{v_2}\right)$$

This proactive adjustment of the counter threshold ensures motion that accurately reflects the desired acceleration, and with a sufficiently small sampling period T_S, it is very precise. Figure 5 shows an illustration of this operation for a starting speed of 1 step/sec, an acceleration of 50 steps/sec^2, and $T_S = 0.1s$. The Appendix shows pseudocode that also describes this algorithm.



Time (s)	Count	Speed (steps/s)	Threshold
0.1	0.1	1 → 6	1 → 0.25
0.2	0.2	6 → 11	0.25 → 0.227
0.227	0.227 → 0	11	0.227 → 1/11 = 0.0909
0.3	0.0727	11 → 16	0.0909 → 0.0852
0.3125	0.0852 → 0	16	0.0852 → 1/16 = 0.0625
0.375	0.0625 → 0	16	0.0625

Sanity check: (0.1s · 1 steps/s) + (0.1s · 6 steps/s) + (0.1s · 11 steps/s) + (0.075s · 16 steps/s) = 3 steps

Figure 96. *Example of precise acceleration control with an initial speed of 1 step/s, an acceleration of 50 steps/s², and a sampling time T_S of 0.1s. For simplicity, the counter value has arbitrary resolution and increases with a slope of 1 count/s. The table shows how different values evolve over time: the clear shaded rows represent loop execution and the blue shaded rows represent the PWM pulse interrupt. Note that 3 pulses (steps) occur in 0.375 seconds, which is confirmed to be correct by the calculation below the table. Note that this does not consider frequency division by toggling, but taking this into account is straightforward. Please also see the pseudocode in the Appendix for clarification on how values are updated.*

Integrator Mode Stepper Motor Microstep Speed Control Pseudocode

Below is the pseudocode for Integrator Mode Stepper Motor Speed control. In practice, safe use of volatile variables and numerical precision and overflow need to be considered. Also note that:

- Velocity v can be positive or negative and its sign determines motor direction, which is configured via a separate GPIO pin.
- The actual applied threshold is 1 less than the value of TH because the counter starts at 0. The counter should end at $(TH - 1)$ so that it cycles through TH values.

Define $f_{counter} = f_{clock}/(\text{prescaler})$
Define T_s to be time the interval between loop executions

Let $v_{curr} =$ the minimum possible positive velocity
Let $TH_{curr} = f_{counter}/|v_{curr}|$ (PWM counter threshold)

Configure and initialize PWM signal
Start loop

Function loop(input: a (acceleration)):

> Let $v_1 = v_{curr}$
> Let $TH_1 = TH_{curr}$
>
> Let $v_2 = v_1 + a \cdot T_s$
> Limit the magnitude of v_2 so it is between the min and max allowed values
> If v_1 and v_2 have different signs, toggle the direction GPIO pin for the stepper driver
> Let $CNT =$ the current PWM counter value
> Let $TH_2 = CNT + (TH_1 - CNT) \cdot |v_1/v_2|$
>
> Set $v_{curr} = v_2$
> Set $TH_{curr} = TH_2$
> applyPwmThreshold($TH_{curr} - 1$)

Function pwmPulseInterrupt:

> Set $TH_{curr} = f_{counter}/|v_{curr}|$
> applyPwmThreshold($TH_{curr} - 1$)

Integrator Mode Stepper Source Code Implementation

apply_acceleration() Function

a. Configurable Parameters of apply_acceleration()

First, a series of parameters are defined to support `apply_acceleration()`

```
#define PWM_COUNT_SAFETY_MARGIN 4
#define MAXIMUM_ACCELERATION 65535
#define MAXIMUM_DECELERATION 65535
#define MIN_POSSIBLE_SPEED 2
#define MAXIMUM_SPEED 65535
#define SAMPLE_FREQUENCY 500
```

`PWM_COUNT_SAFETY_MARGIN` sets the minimum number of count values that must remain in the PWM period counter to permit a change in the PWM counter value. This is present to ensure that sufficient processing time is available to update the PWM counter value prior to reaching the end of the PWM period. At the frequency of the PWM period measurement clock, 82 kHz, this is a time of 48 microseconds.

`MAXIMUM_ACCELERATION` sets the maximum acceleration rate input value
`MAXIMUM_DECELERATION` sets the maximum deceleration rate input value

`MIN_POSSIBLE_SPEED` is the minimum velocity of 2 steps per second provided by **`apply_acceleration()`**
`MAXIMUM_SPEED` is the maximum velocity of 65535 steps per second provided by **`apply_acceleration()`**

`SAMPLE_FREQUENCY` is the control loop cycle rate. The **`apply_acceleration()`** function will be executed during each cycle of the closed loop system.

b. Variables of apply_acceleration()

Second, variables are defined:

```
int32_t target_velocity_prescaled;
```

`target_velocity_prescaled` is updated by `apply_acceleration()` in each control cycle.

```
volatile uint32_t desired_pwm_period = 0;
volatile uint32_t current_pwm_period = 0;
```

`desired_pwm_period` is the PWM period value computed by `apply_acceleration()` to be applied in the current control loop cycle and based on an incremental change in PWM period relative to `current_pwm_period`

`current_pwm_period` is the PWM period value in the current control loop cycle

c. Description of apply_acceleration()

`apply_acceleration()` computes the values of PWM period to enable a specified velocity corresponding to a step rate. The L6474 system employs a timer system operating as a count-up counter initialized at value zero and counting up to a threshold value. The threshold value is set as the PWM period according to the timer clock rate set to 82 kHz. `apply_acceleration()` enables change in PWM counter value for adjustment of speed over a wide dynamic range within each cycle of a control loop.

A timer interrupt is generated when the timer counter reaches the threshold value.

A timer interrupt service routine, `Main_StepClockHandler()`, then executes to update the PWM period to the current desired value.

`apply_acceleration()` includes arguments of the requested acceleration supplied by the control loop, acc, the current target_velocity, and the loop sample frequency.

```
void apply_acceleration(int32_t acc, int32_t* target_velocity_prescaled, uint16_t
sample_freq_hz) {
	uint32_t current_pwm_period_local = current_pwm_period;
	uint32_t desired_pwm_period_local = desired_pwm_period;
```

The direction of rotor motion during the previous cycle and entering the function is determined.

```
	motorDir_t old_dir = *target_velocity_prescaled > 0 ? FORWARD : BACKWARD;
```

The requested acceleration values are compared against limits and the value is saturated between the limits.

```
	if (old_dir == FORWARD) {
		if (acc > MAXIMUM_ACCELERATION) {
			acc = MAXIMUM_ACCELERATION;
		} else if (acc < -MAXIMUM_DECELERATION) {
			acc = -MAXIMUM_DECELERATION;
		}
	} else {
		if (acc < -MAXIMUM_ACCELERATION) {
			acc = -MAXIMUM_ACCELERATION;
		} else if (acc > MAXIMUM_DECELERATION) {
			acc = MAXIMUM_DECELERATION;
		}
	}
```

A new velocity is determined according to the time integral of acceleration. The integration time interval is inverse of the sample frequency.

```
*target_velocity_prescaled += L6474_Board_Pwm1PrescaleFreq(acc) / sample_freq_hz;
```

A new motor direction must be computed

```
motorDir_t new_dir = *target_velocity_prescaled > 0 ? FORWARD : BACKWARD;
```

A new speed is computed, `speed_prescaled` that is saturated at lower and upper limits. The limits are computed as the limits determined by determining a speed based on the return value of `L6474_Board_Pwm1PrescaleFreq.` For a value of SPEED. This function returns:

```
*target_velocity_prescaled = TIMER_PRESCALER * BSP_MOTOR_CONTROL_BOARD_PWM1_FREQ_RESCALER *
SPEED
```

`TIMER_PRESCALER` is defined as 1024

`BSP_MOTOR_CONTROL_BOARD_PWM1_FREQ_RESCALER` is defined as 2

```
uint32_t speed_prescaled;
if (new_dir == FORWARD) {
        if (*target_velocity_prescaled < L6474_Board_Pwm1PrescaleFreq(MIN_POSSIBLE_SPEED)) {
                *target_velocity_prescaled = L6474_Board_Pwm1PrescaleFreq(MIN_POSSIBLE_SPEED);
        } else if (*target_velocity_prescaled > L6474_Board_Pwm1PrescaleFreq(MAXIMUM_SPEED)) {
                *target_velocity_prescaled = L6474_Board_Pwm1PrescaleFreq(MAXIMUM_SPEED);
        }
        speed_prescaled = *target_velocity_prescaled;
        } else {
                if (*target_velocity_prescaled > -L6474_Board_Pwm1PrescaleFreq(MIN_POSSIBLE_SPEED)) {
                        *target_velocity_prescaled = -L6474_Board_Pwm1PrescaleFreq(MIN_POSSIBLE_SPEED);
                } else if (*target_velocity_prescaled < -L6474_Board_Pwm1PrescaleFreq(MAXIMUM_SPEED)) {
                        *target_velocity_prescaled = -L6474_Board_Pwm1PrescaleFreq(MAXIMUM_SPEED);
                }
        speed_prescaled = *target_velocity_prescaled * -1;
}
```

Then, the current value of the computed PWM period, `effective_pwm_period`, is set equal to `desired_pwm_period_local`.

The value of `desired_pwm_period_local` will now be computed according to the new velocity.

```
uint32_t effective_pwm_period = desired_pwm_period_local;
```

A new PWM period, desired for the current control cycle, is computed based on speed and on the frequency of the counter that determines PWM waveform period. This is computed as the current frequency of the PWM clock divided by the speed in steps/sec.

```
                desired_pwm_period_local = HAL_RCC_GetSysClockFreq() / speed_prescaled;
```

Then, a new direction may result from the change in velocity. This new direction is computed.

```
        if (old_dir != new_dir) {
                L6474_Board_SetDirectionGpio(0, new_dir);
        }
```

Then, we evaluate whether in this current control loop cycle, that current PWM period is non-zero period.

```
if (current_pwm_period_local != 0)
```

Next, the current value of the PWM counter is computed.

```
pwm_count = L6474_Board_Pwm1GetCounter();
```

Then, the time remaining in the PWM period is computed as pwm_time_left.

```
pwm_time_left = current_pwm_period_local - pwm_count;
```

The time remaining is compared with the safety margin. If the time remaining is greater than the safety margin, a new PWM period is computed.

```
if (pwm_time_left > PWM_COUNT_SAFETY_MARGIN) {
```

A new count remaining value is computed based on the ratio of the desired period value to the previous period value.

```
new_pwm_time_left = pwm_time_left * desired_pwm_period_local / effective_pwm_period;
```

In the event that the desired PWM period is less than that of the previous period, the counts remaining value in the period is reduced. Otherwise, the counts remaining value is increased.

If the new value of PWM counts remaining is different from previous, a comparison is made of the counts remaining value for the desired period, `new_pwm_time_left`, to the safety margin. If this is less, the `new_pwm_time_left` is set to the safety margin value.

```
if (new_pwm_time_left != pwm_time_left) {
                    if (new_pwm_time_left != pwm_time_left) {
                           if (new_pwm_time_left < PWM_COUNT_SAFETY_MARGIN) {
                                  new_pwm_time_left = PWM_COUNT_SAFETY_MARGIN;
                           }
```

Then, a new value for the PWM period is computed based on the counts remaining as the current count value summed with the new count value remaining.

```
current_pwm_period_local = pwm_count + new_pwm_time_left;
```

The new PWM period is set.

```
L6474_Board_Pwm1SetPeriod(current_pwm_period_local);
```

The current value of the PWM period is assigned in preparation for the next cycle of computation.

```
current_pwm_period = current_pwm_period_local;
```

Main_StepClockHandler()

The L6474 system will increment the PWM period counter at its clock rate until the count up action reaches `current_pwm_period`.

This will result in the PWM period counter reaching the value of `current_pwm_period`.

An interrupt will be generated by the timer system.

The timer interrupt will be serviced by the interrupt service routine function, `Main_StepClockHandler()`.

This assigned a new period value according to the global variable `desired_pwm_period`.

This then calls the function, `L6474_Board_Pwm1SetPeriod()` with an argument of `desired_pwm_period_local`.

This function includes the `MACRO__HAL_TIM_SetAutoreload()`. Its arguments are the PWM timer, `&hTimPwm1`, and the desired counter count-up threshold value, `desired_pwm_period_local – 1`.

This compares the `desired_pwm_period` with zero and continues if the value is non-zero.

This then executes to set the timer counter threshold value upon which an interrupt will occur, thus establishing a PWM period.

This also sets the current value of the PWM period in preparation for the next control cycle in the global variable, `current_pwm_period`.

```
void Main_StepClockHandler() {
       uint32_t desired_pwm_period_local = desired_pwm_period;
       if (desired_pwm_period_local != 0) {
              L6474_Board_Pwm1SetPeriod(desired_pwm_period_local);
              current_pwm_period = desired_pwm_period_local;
       }
}
```

24. Appendix F: Rotor Actuator Plant Transfer Function

Motor Controller Response Model

The Rotor Control Plant, G_{Rotor}, accepts a control input, of $\phi_{\text{Rotor-Control}}$, and yields output angle, of ϕ_{Rotor}, as shown in Figure 97.

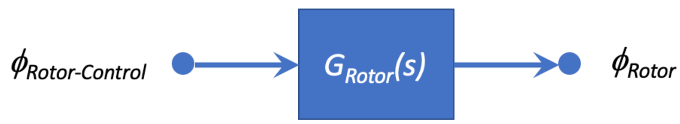


Figure 97. Rotor Control Plant with input control signal of $\phi_{Rotor-Control}$, and output angle, of ϕ_{Rotor}.

During each cycle of control system operation for the Edukit Rotary Inverted Pendulum, new target signals are supplied to the Rotor Plant. The Rotor Control system supported by the Stepper Motor Controller is described in detail in Appendix E: Rotor Actuator Plant.

The Rotor Control, stepper motor control system, accepts a control input setting *acceleration* of the Rotor. Thus, *position* of the Rotor is the double-integral with respect to time of the control input.

The G_{Rotor} plant transfer function was determined directly by system identification. This included these steps:

1) Establishing closed loop operation of the Edukit system for direct control of Rotor only. Here, the control of pendulum position was not included. Control action is applied only to the Roror Control Plant. PID controllers were defined and applied to Rotor control.
2) Control loop cycle rate was greater than 500 Hz. During each control cycle, Measurement of both of $\phi_{\text{Rotor-Control}}$ and ϕ_{Rotor} time series during system operation closed loop operation. were recorded.
3) For each example of a PID controller, application of a control signal step function was applied as a Rotor Angle tracking command.
4) Solving for Rotor Control Plant definition was performed by these steps:
    a. Defining a second order response model for G_{Rotor} of the form:

$$G_{Rotor} = \frac{\phi_{\text{Rotor}}}{\phi_{\text{Rotor-Control}}} = \frac{a}{s^2 + bs + c}$$

    b. Performing an exhaustive and high resolution grid search over the values of the coeffiicents, *a*, *b,* and *c*, for the minimum rms error between model-predicted response, and measured response.

5) The sensitivity of this method for detection of model-errors was confirmed by testing the error resulting from the selection of coefficient values differing from values determined by grid search minimization.
6) Validation of this method was confirmed by direct comparison between model-predicted and measured response of Rotor Control.
7) Validation of this method was confirmed by direct comparison between model-predicted and measured response of Rotor Control for both output feedback and full state feedback systems.

The Rotor Position Control system architecture for this system identification is shown in Figure 98.

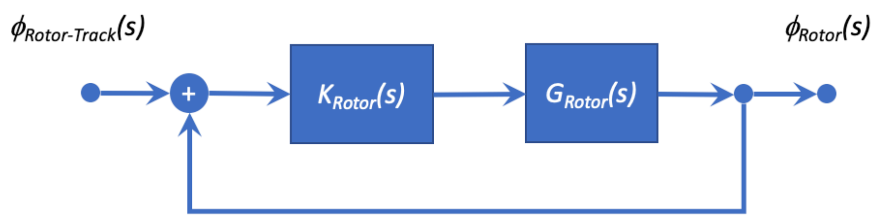


Figure 98. Rotor Position Control loop with input control signal of $\phi_{Rotor-Track}$, and output angle, of ϕ_{Rotor}.

A PID controller was defined of the form:

$$K = K_P + \frac{1}{K_I s} + K_D s$$

PID gain coefficients yielding underdamped response were selected:

$$K_P = 72, K_I = 36, K_D = 3.6$$

For all tests, the values of the coefficients coeffiicents, *a*, *b,* and *c* were found to be

$$1.0 \geq a \geq 0.94; \quad 0.0 \leq b \leq 0.04, \quad 0.0 \leq c \leq 0.04$$

Thus, the Rotor Angle Response Transfer Function, G_{Rotor}, was determined to be consistent with the estimate of *a = 1.0*, *b = 0.0,* and *c = 0.0.* Then,

$$\mathrm{G_{Rotor}} = \frac{\phi_{\mathrm{Rotor}}}{\phi_{\mathrm{Rotor-Control}}} = \frac{a}{s^2 + bs + c}$$

This is consistent with the system operation that determines *position* of the Rotor as the double-integral with respect to time of the control input.

Direct comparison of of model-predicted Rotor Angle response and measured Rotor Angle response under PID output feedback control is shown in Figure 99.

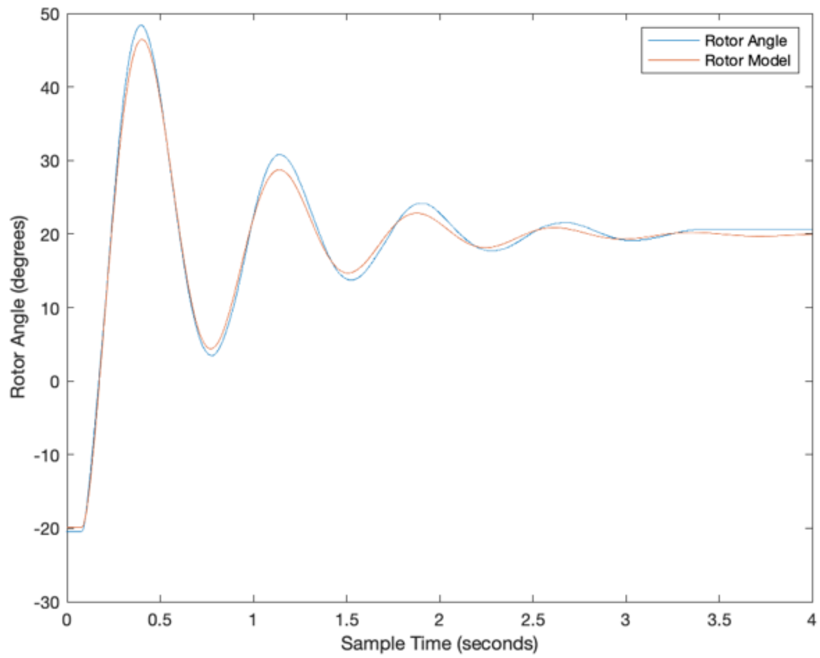


Figure 99. Comparison of model-predicted Rotor Angle response and measured Rotor Angle response under PID output feedback control

The Rotor Angle Step Response was also measured by application of a step signal of amplitude 40 degrees to the Rotor Angle Reference at $t = 0$ and then sampling Rotor Angle over time. This was conducted with the Rotor Control System operating under full state feedback with gain values determined by LQR design. This applies the data logging feature of the Real Time Control Systems Workbench. Then, using the methods developed for deriving Sensitivity Function Transfer Function Frequency Response, Rotor Plant Transfer Function was computed. The comparison between simulated response of the Rotor Plant model defined here and the measured response is shown in Figure 100.

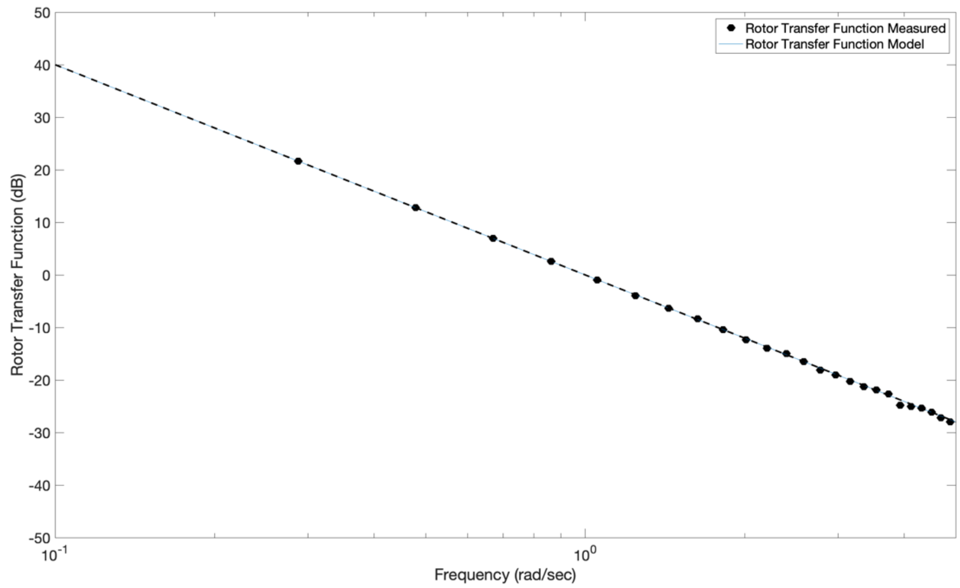


Figure 100. Comparison of model-predicted Rotor Angle Transfer Function and measured Rotor Angle Transfer Function under Full State feedback control with Rotor Transfer Function computed by the Rotor System Identification system.

A second validation step was performed. First, using the G_{Rotor} plant model above, a full state feedback controller was defined. Here, the difference between input tracking reference and output is yet more sensitive to plant definition than for the case of direct output feedback control as in the PID system above.

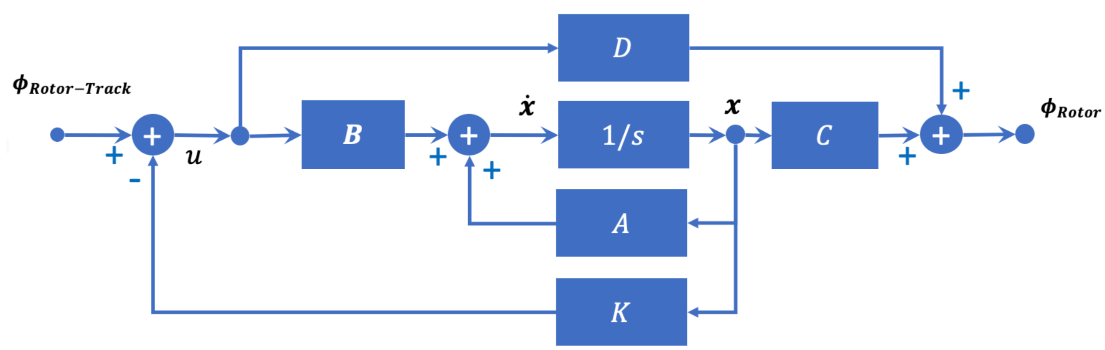


Figure 101. Full State Feedback Rotor Position Control loop with input control signal of $\phi_{Rotor-Track}$, and output angle, of ϕ_{Rotor}.

The full state feedback control loop is shown in Figure 101. Computation of system dynamics follows.

First, the relationship between the control signal, $\boldsymbol{u}$**,** and Rotor Angle, ϕ_{Rotor} is determined by the rotor actuator system that determines *position* of the Rotor is the double-integral with respect to time of the control input. Then,

$$\ddot{\phi}_{\mathrm{Rotor}} = -c\phi - b\dot{\phi} + a\boldsymbol{u}$$

Now, the full state feedback state vector $\boldsymbol{x}$ is defined including Rotor Angle and its time derivative

$$\boldsymbol{x} = \begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix}$$

The state dynamics are defined

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

with

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \end{bmatrix}$$

and the state matrix, $\boldsymbol{A}$

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ -c & -b \end{bmatrix}$$

with the input matrix, $\boldsymbol{B}$

$$\boldsymbol{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

then,

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -c & -b \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ a \end{bmatrix} \boldsymbol{u}$$

The system output vector, $\boldsymbol{y}$**,** including the Rotor Angle, ϕ, and Pendulum Angle, θ**,** is defined,

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u}$$

with the output matrix, $\boldsymbol{C}$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and the feedforward matrix, $\boldsymbol{D}$

$$\boldsymbol{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

then

$$\boldsymbol{y} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \boldsymbol{u}$$

The state feedback control law is defined as

$$\boldsymbol{u} = \boldsymbol{Kx}$$

The Linear Quadratic Regulator (LQR) design method finds the optimal gain matrix, $\boldsymbol{K}$, that minimizes the cost function of output $\boldsymbol{u}$. Application of LQR design using the Matlab **LQR** function results in gain values, of

$$\boldsymbol{K}_{\boldsymbol{\Phi}} = 1.00, \boldsymbol{K}_{\dot{\boldsymbol{\Phi}}} = 1.73$$

.

The direct comparison with model-predicted and experimental response of the Rotor Control system with the Rotor Plant defined as above with its double integrator relationship between control input and rotor position is shown in Figure 102. Agreement between model-predicted and experimental results provides additional verification of the Rotor Plant model.

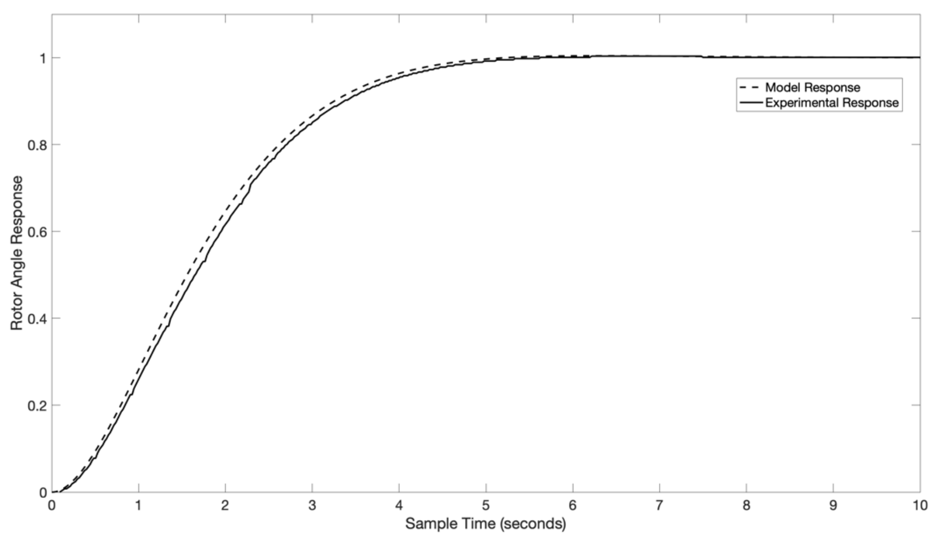


Figure 102. Comparison of model-predicted Rotor Angle response and measured Rotor Angle response under full state feedback control

A third validation step was performed. Using the G_{Rotor} plant model above, a full state feedback controller was defined including integral action applied to the error difference between Rotor Track command and Rotor Angle as shown in Figure 103.

The state vector defined above is augmented with a third state, the integral of the difference between the state output and the reference track command:

$$\dot{\boldsymbol{x}}_{\boldsymbol{I}} = \boldsymbol{r} - \boldsymbol{C}\boldsymbol{x}$$

where $\boldsymbol{r} = \phi_{\text{Rotor-Track}}$

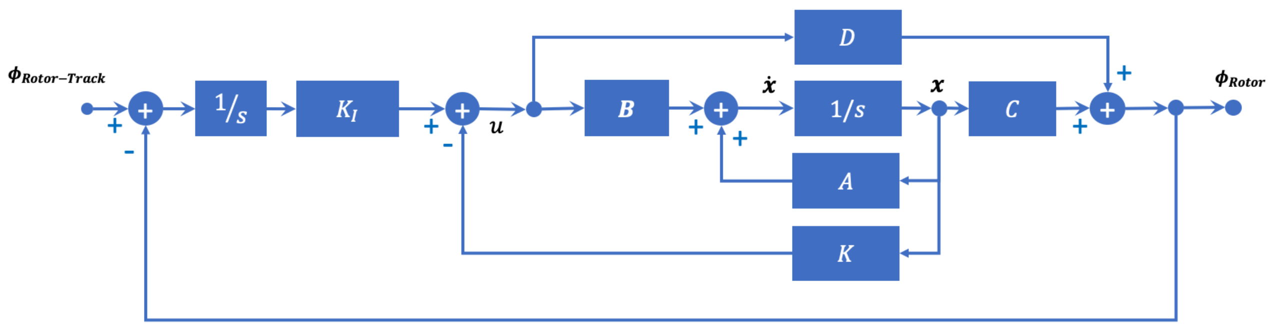


Figure 103. Full State Feedback Rotor Position Control loop including integral action with input control signal of $\phi_{Rotor-Track}$, and output angle, of ϕ_{Rotor}.

Then, the state dynamics are becomes

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{x}}_{\boldsymbol{I}} \end{bmatrix} = \boldsymbol{A}_{\boldsymbol{I}}\boldsymbol{x} + \boldsymbol{B}_{\boldsymbol{I}}\boldsymbol{u} + \boldsymbol{B}_{\boldsymbol{R}}\boldsymbol{r}$$

With

$$\boldsymbol{A}_{\boldsymbol{I}} = \begin{bmatrix} 0 & 1 & 0 \\ -c & -b & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{B}_{\boldsymbol{I}} = \begin{bmatrix} 0 \\ a \\ 0 \end{bmatrix}$$

$$\boldsymbol{B}_{\boldsymbol{R}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The system output vector, $\boldsymbol{y}$**,** including the Rotor Angle, ϕ, and Pendulum Angle, θ,

$$\boldsymbol{y} = \boldsymbol{Cx} + \boldsymbol{Du}$$

The state feedback control law is defined as

$$\boldsymbol{u} = [\boldsymbol{K} \quad \boldsymbol{K_I}]\boldsymbol{x}$$

The Linear Quadratic Regulator (LQR) design method finds the optimal gain matrix, $\boldsymbol{K}$, that minimizes the cost function of output $\boldsymbol{u}$. Application of LQR design using the Matlab **LQR** function results in gain values, of

$$\boldsymbol{K_\Phi} = 1.79, \boldsymbol{K_{\dot{\Phi}}} = 1.10, \boldsymbol{K_I} = 1$$

The direct comparison with model-predicted and experimental response of the Rotor Control system with the Rotor Plant defined as above with its double integrator relationship between control input and rotor position is shown in Figure 104. Agreement between model-predicted and experimental results provides additional verification of the Rotor Plant model.

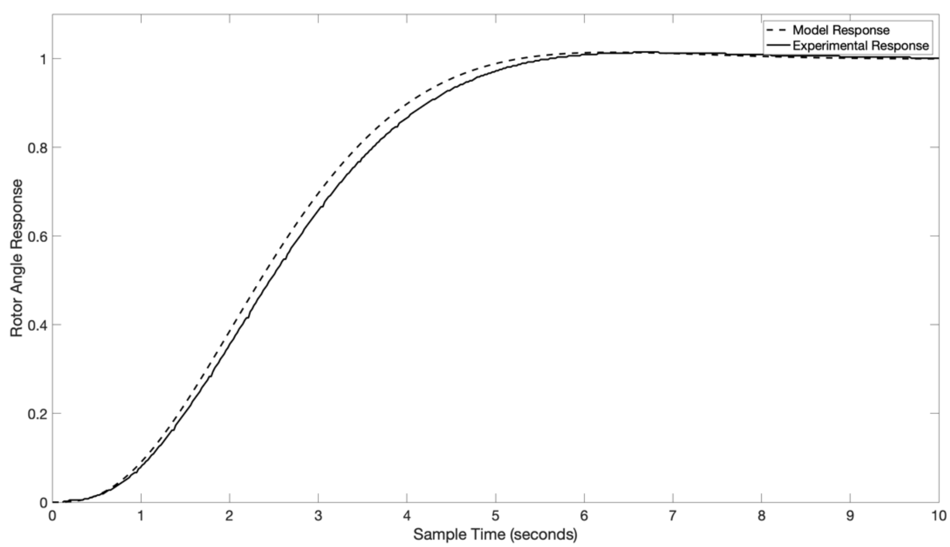


Figure 104. Comparison of model-predicted Rotor Angle response and measured Rotor Angle response under full state feedback control including integral action.

25. Appendix G: Pendulum Angle Error Autocompensation System

The Edukit Rotary Inverted Pendulum and other similar systems may be subject to long term drift errors resulting from at least two sources. First, the Pendulum controller requires accurate measurement of Pendulum Angle relative to the gravity vector. An initial offset of this value will lead to drift in the controller system output with a rate dependent on magnitude of the offset error. Pendulum Angle calibration is performed by a process where Pendulum Angle is measured for the Pendulum operating prior to control system operation such that the Pendulum Angle corresponds to the Pendulum operating in a suspended, vertical mode. Prior to initiation of control operation, the Pendulum is manually rotated into a near vertical orientation. At this time, the control system measures Pendulum Angle and subtracts the angle of 180 degrees from all subsequent measurement, yielding an angle measured relative to the gravity vector.

One error source results from the finite resolution of the optical encoder. This finite resolution implies that at initiation of the Pendulum Angle control operation, the recording of vertical down position and subsequent vertical upwards position will include a small offset error of the order of one optical encoder step or, 0.16 degrees. During control operation, this small bias error will induce a drift Rotor Position as the control system seeks to eliminate this error.

A second error source results from tilt in the Rotary Inverted Pendulum structure relative to the gravity vector. A departure of the structure from vertical orientation (due to a slope in the base support structure relative to earth level) will introduce an error in operation. This results from the small, but finite difference between the normal to the plane of rotation of the Rotary Inverted Pendulum and the gravity vector. This will also lead to an error as the Rotor Control system seeks to ensure the Pendulum Angle is zero.

The error in operation appears as a drift and long term error between ϕ_{rotor} and $\phi_{rotor-control}$. A control system, referred to as Autocompensation compensates for the platform slope and encoder offset errors. Autocompensation introduces an additional PID control operating on the input of the low pass filtered difference between ϕ_{rotor} and $\phi_{rotor-control}$ and providing an output summed with the measured pendulum angle, $\theta(s)$. A first order low pass filter, $H_{as}(s)$, at corner frequency of f_{as} is applied to the difference between ϕ_{rotor} and $\phi_{rotor-control}$.

The Autocompensation introduces, therefore, a filter and PID control block into the Dual PID controller. The block diagram for the Autocompensation control is shown in *Figure 105*.

It is important to note that the implementation of the Edukit Rotary Inverted Pendulum offers three operating modes for Autocompensation:

1) Autocompensation Operation Disabled
2) Autocompensation Operation Enabled for all operating time
3) Autocompensation Operation Enabled for an initial period, T_{as} , and with Autocompensation control suspended for later time. This option enables errors to be removed during a short initial operating period and then suspension of Autocompensation control for all later time.

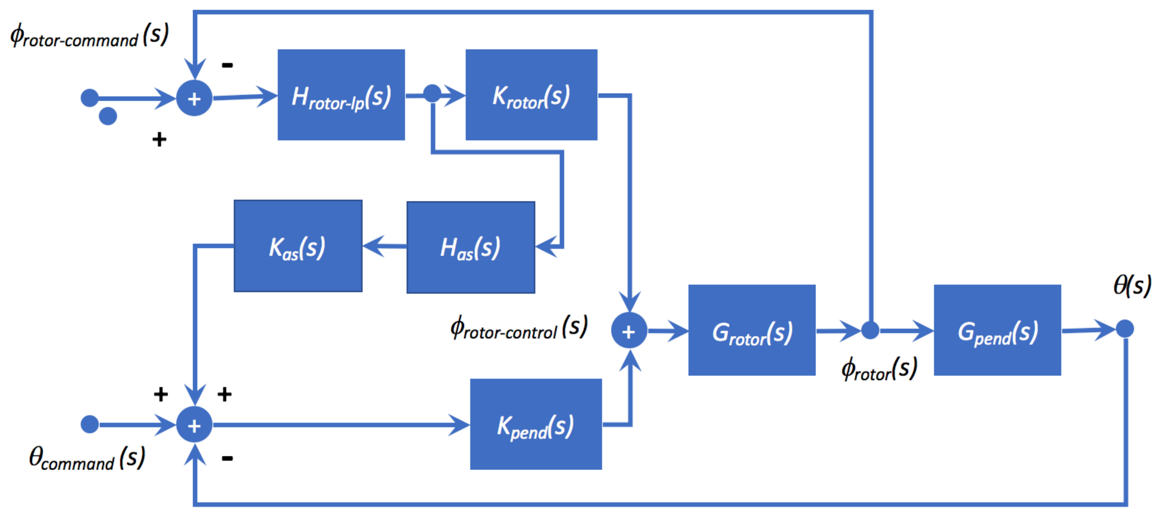


Figure 105. The Autocompensation PID Control System

The relationship between ϕ_{rotor} and $\phi_{rotor-control}$ is below.

$$\phi_{rotor}(s) = \phi_{rotor}(s)G_{pendulum}(s)K_{pend}(s)G_{rotor}(s) + \left(\phi_{rotor-command}(s) - \phi_{rotor}(s)\right)H_{rotor-lp}(s)K_{rotor}(s)G_{rotor}(s) + \left(\phi_{rotor-command}(s) - \phi_{rotor}(s)\right)H_{as}(s)K_{as}(s)K_{pend}(s)G_{rotor}(s)$$

Finally, noting the negative feedback contribution of K_{pend}, the transfer function from rotor command angle to rotor angle is.

$$\frac{\phi_{rotor}(s)}{\phi_{rotor-command}(s)} = \frac{\left(H_{rotor-lp}(s)K_{rotor}(s)G_{rotor}(s) - H_{as}(s)K_{as}(s)K_{pend}(s)G_{rotor}(s)\right)}{\left(1 + H_{rotor-lp}(s)K_{rotor}(s)G_{rotor}(s) - H_{as}(s)K_{as}(s)K_{pend}(s)G_{rotor}(s) + G_{pendulum}(s)K_{pend}(s)G_{rotor}(s)\right)}$$

The contribution of K_{as} to the input of K_{pend} includes the gain relating measured rotor angle position (Rotor Angle Measurement Gain) with encoder angle position of (Encoder Angle Measurement Gain). Further, the output of the controller, is scaled by the gain factor between measured angle and motor control angle gain (Rotor Angle Control Gain). This control system, operating at low relative frequency, is stable and requires only proportional control.

The Edukit Rotary Inverted Pendulum control system for rotor angle applies these typical values:

$$K_p = 0.005;\ K_i = 0; K_d = 0$$
$$f_{as} = 0.005\ Hz$$