

# SELF-ERECTING, ROTARY MOTION INVERTED PENDULUM

*Quanser Consulting Inc.*

## 1.0 SYSTEM DESCRIPTION

The self-erecting rotary motion inverted pendulum consists of a rotary servo motor system (SRV-02) which drives an independent output gear which also meshes with a potentiometer. The rotary pendulum arm is mounted to output gear. At the end of the pendulum arm is a hinge instrumented with a quadrature encoder. The pendulum attaches to the hinge.

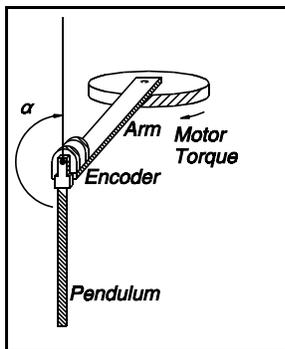


Figure 1.2: Schematic of self erecting experiment

The potentiometer measures the rotation of the arm while the encoder measures the rotation of the pendulum. The potentiometer does not have physical stops but the measurement does have a discontinuity. The maximum range that can be measured with the potentiometer is 352 degrees. The pendulum angle on the other hand is measured via a quadrature encoder and thus the number of turns that can be measured are virtually unlimited.

The purpose of the experiment is to design a controller that starts with the pendulum in the “down” position, swings it up and maintains it upright.

## 1.1 ASSEMBLY

This experiment is performed with the SRV-02 in the high gear ratio configuration shown in Figure 1.3. Use the supplied gears to achieve the configuration shown. Attach the pendulum arm to the central gear of the SRV-02 using the two brass thumb screws.

### 1.1.1 Calibration

Loosen the set-screw on the potentiometer gear of the SRV-02 (See figure 1.3). Attach a 6 Pin Mini-Din Cable from the SRV-02 to the Quick connect box labelled S1&S2 ( See Figure 7.1). Power up the power module and measure the voltage at the binding post labelled S1 with respect to the ground terminal. This is the voltage of the potentiometer. Rotate the pendulum arm such that it points to the “zero” measurement of the SRV-02.



Figure 1.1 : SRV-02 shown in low gear ratio configuration and the gears required for high gear ratio configuration

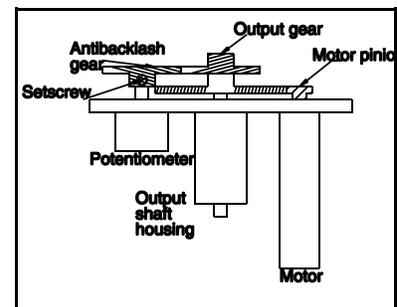


Figure 1.3: High gear ratio configuration

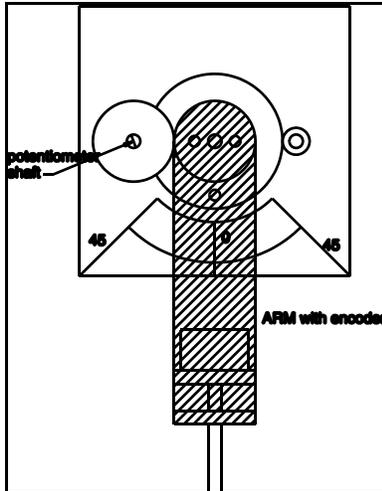


Figure 1.4: Calibration position

Using a small screwdriver, rotate the potentiometer shaft while the arm remains pointing to the zero position and observe the voltage at S1. Adjust the position of potentiometer shaft such that when the arm is at “zero” the voltage measured is around zero volts ( $\pm 0.1$  V). NOTE that this can happen at two positions. **One of them is wrong:** You can measure zero volts either when the wiper of the potentiometer is in the middle of the range, which is correct, or when the wiper is not in contact with anything! To make sure you have the correct zero volts, turn slightly left and right and ensure that the measurement is **continuous**. It should not jump to +5 volts or -5 volts suddenly. At the zero volts tighten the set screw. You have now calibrated the system.

### 1.1.2 Safe mounting

Mount the system on a table such that the pendulum swings in the front and ensure that the pendulum will not collide with any objects while it swings. Clamp the SRV-02 to the table using a “C” clamp that can be obtained from any hardware store.

## 2.0 MATHEMATICAL MODELLING

### 2.1 NONLINEAR MODEL

Consider the simplified model in Figure 2.1. Note that  $l_p$  is half the actual length of the pendulum ( $l_p = 0.5 L_p$ ).

The nonlinear differential equations are derived to be:  
where

$$(m_p r^2 + J_b) \ddot{\theta} + m_p r \ddot{\alpha} l_p \cos(\alpha) - m_p r \dot{\alpha}^2 l_p \sin(\alpha) = T$$

$$m_p l_p \cos(\alpha) \ddot{\theta} r - m_p l_p \sin(\alpha) \dot{\alpha} \dot{\theta} r + m_p \ddot{\alpha} l_p^2 - m_p g l_p \sin(\alpha) = 0$$

- T : input torque from motor ( Nm )
- $m_p$ : mass of rod (Kg)
- $l_p$ : centre of gravity of rod (m) (half of full length)
- $J_b$ : Inertia of Arm and gears(Kgm<sup>2</sup>)
- $\theta$ : Deflection of arm from zero position(Rad)
- $\alpha$ : Deflection of pendulum from vertical UP position(Rad)

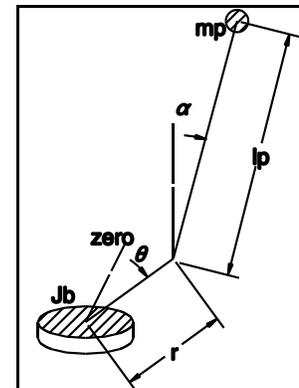


Figure 2.1: Simplified model

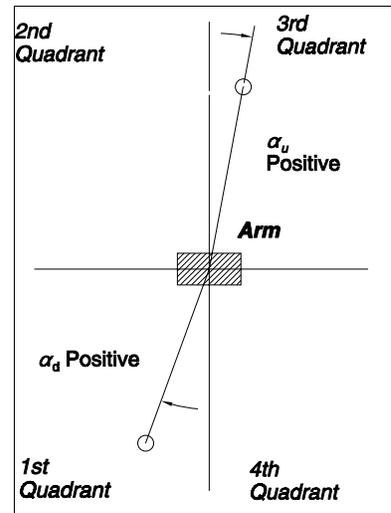


## 2.2 LINEAR MODEL

The linear equations resulting from the above are:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha}_u \\ \ddot{\theta} \\ \ddot{\alpha}_u \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{m_p r g}{J_b} & 0 & 0 \\ 0 & \frac{g (m_p r^2 + J_b)}{I_p J_b} & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \alpha_u \\ \dot{\theta} \\ \dot{\alpha}_u \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{J_b} \\ -\frac{r}{J_b I_p} \end{bmatrix} F \quad (2.2)$$

Note that the zero position for all the above equations is defined as the pendulum being vertical “up”. A positive angle in the pendulum is defined for a fall to the right when looking at the pendulum from the motor shaft. A positive rotation of the arm is defined as clockwise when looking at the SRV-02 from above. (See Figure 2.2)



**Figure 2.2** Definitions of  $\alpha_u$  and  $\alpha_d$ . (VIEW IS FROM MOTOR SHAFT LOOKING OUT TOWARDS THE PENDULUM)

For the DOWN position, the following equation holds:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha}_d \\ \ddot{\theta} \\ \ddot{\alpha}_d \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{m_p r g}{J_b} & 0 & 0 \\ 0 & \frac{-g (m_p r^2 + J_b)}{I_p J_b} & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \alpha_d \\ \dot{\theta} \\ \dot{\alpha}_d \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{J_b} \\ \frac{r}{J_b I_p} \end{bmatrix} F \quad (2.3)$$

where  $\alpha_d=0$  is defined for the pendulum suspended. Its motion is relative to the “down” axis and is positive for clockwise rotations observed from the motor shaft.

## 3.0 CONTROL SYSTEM DESIGN

The controller will consist of two main parts. One will be the “swing up” controller while the second is the “balance” controller. The swing up controller will oscillate the arm until it has built up enough energy in the pendulum that it is almost upright at which point the “balance” controller is turned on and is used to maintain the pendulum vertical.

### 3.1 Swing up control

Swing up control will essentially control the position of the arm in order to destabilize the “down” position. Assume the arm position can be commanded via  $\theta_d$ . Then the feedback:

$$\theta_d = P \alpha_d + D \dot{\alpha}_d$$

can be made to destabilize the system with the proper choice of the gains P & D. This means we want to command the arm based on the position and rate of the pendulum. This approach is intuitive: it simply makes sense that by moving the arm back and forth one can eventually bring up the pendulum.

For the servo arm to track the desired position, we design a PD controller:

$$V_d = K_p (\theta_d - \theta) + K_d \dot{\theta}$$

This is a position control loop that controls the voltage applied to the motor so that  $\theta$  tracks  $\theta_d$ .

Designing a swing-up controller in this fashion ensures that the arm is not commanded to move beyond the potentiometer limits. By limiting  $\theta_d$ , we ensure that the arm does not reach positions that cause a collision of the pendulum with the table upon which the SRV-02 rests.

The motor equations are:

$$V = I_m R_m + K_m K_g \dot{\theta}$$

where

V (volts) :	Voltage applied to motor
$I_m$ (Amp):	Current in motor
$K_m$ (V/ (rad sec <sup>-1</sup> ):	Back EMF Constant
$K_g$ :	Gear ratio in motor gearbox and external gears
$\theta$ (Rad):	Arm angular position

The torque generated by the motor is:

$$T = K_m K_g I_m = J_b \ddot{\theta}$$

$$\frac{\theta}{V} = \frac{1}{s \left( \frac{J_b R_m}{K_m K_g} s + K_m K_g \right)}$$

Substituting system parameters we obtain:

$$\ddot{\theta} = -25 s \theta + 47.2 V$$

and closing the loop we have the arm position control transfer function:

$$\frac{\theta}{\theta_d} = \frac{47K_p}{s^2 - (-25+47K_p)s + 47K_p}$$

which has a characteristic polynomial of the type:

$$s^2 + 2 \zeta \omega_o s + \omega_o^2 \quad \text{and} \quad t_p = \frac{\pi}{\omega_o \sqrt{(1-\zeta^2)}}$$

where  $\zeta$  is the damping ratio and  $t_p$  is the peak time.

Now we need to select the damping ratio and peak time to obtain  $K_p$  &  $K_d$ .

Consider the pendulum in the system. Its parameters are:

$l_p(\text{m})$  : 0.20 m (centre of gravity is located at half of full length)  
 $m_p(\text{Kg})$ : 0.15 Kg

The natural period for small oscillations is given by:

$$T = 2 \pi \sqrt{\frac{l_p}{g}}$$

We want the arm to react to these movements. Therefore the closed loop response of the arm should be considerably faster than the natural frequency of the pendulum. The natural period of the pendulum is 0.97 seconds. The natural frequency is  $\omega_p = 2 \pi/T = 6.45$  rad/sec. It would then be reasonable to design a closed loop controller for the arm position which has the following specifications:

$\omega_o = 6 \omega_p = 38.75$  rad/sec  
 zeta = 0.707

The factor of 6 is rather arbitrary and could have been 5 or 7.

Using this approach (see **servpos.m**) , we obtain the gains:

$$K_p = .55 \text{ V/}^\circ$$

$$K_d = -0.01 \text{ V / (}^\circ \text{ sec}^{-1})$$

A SIMULINK block diagram that simulates (**s\_down.m**) the swing up controller is shown in Figure 3.1. The "open loop" system consists of the state space system given in equation **2.3**. the outputs are converted to degrees within the block. The input to the system is a voltage which should be saturated to +/- 5 Volts in order to obtain a realistic simulation. To run the simulation you must run the m file **down.m** first which loads the model parameters into the MATLAB workspace.

The results of the simulation are shown in Figure 3.2. The simulation indicates that this method of swing up control can be used to bring up the pendulum. This is not exact analysis however since the simulation is based on a linear model and once the angle exceeds a few degrees, the model is not valid. Conceptually however, the simulation shows that it can be done.

Note that when the pendulum amplitude gets larger, it starts interfering with the motion of the arm to the extent that the arm does not track the desired position at all. This effect is dealt with in the implementation section by changing the limit on  $\theta_d$  dynamically (see section 4.3).

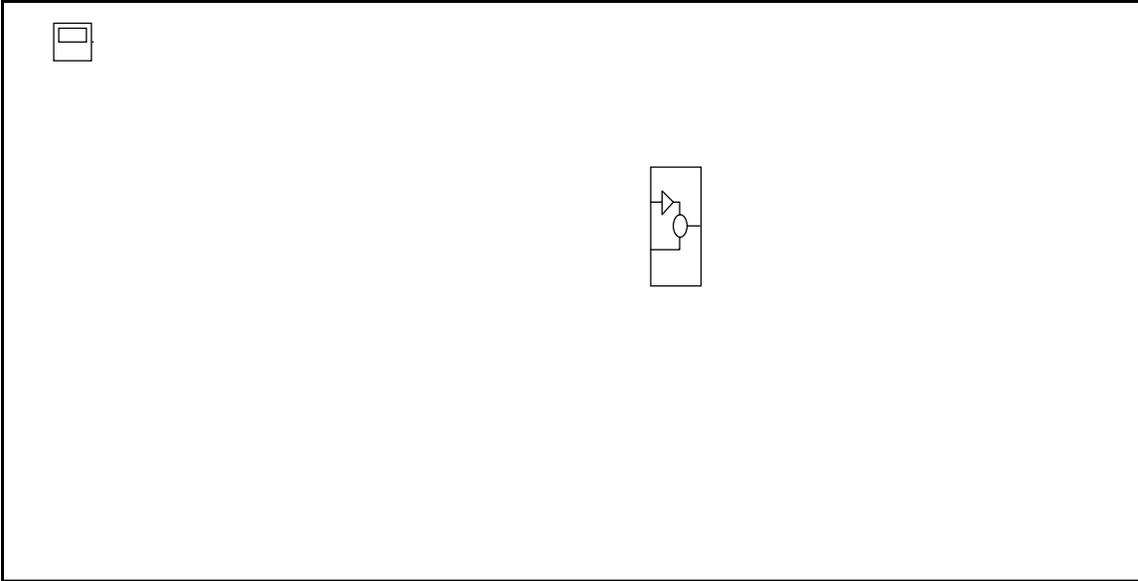


Figure 3.1: SIMULINK simulation of swing up controller (s\_down.m)

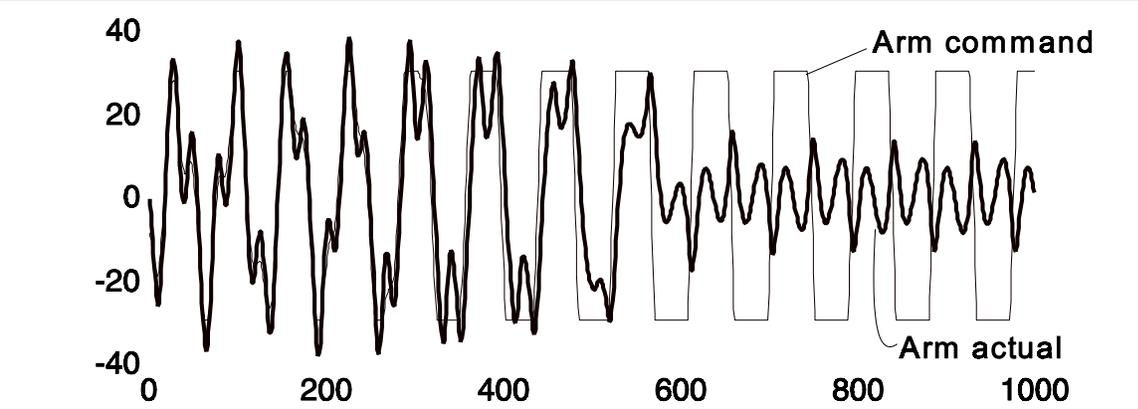


Figure 3.2: Simulation results showing arm command and actual arm angle . Note how the actual does not track the desired when the pendulum angle is large.

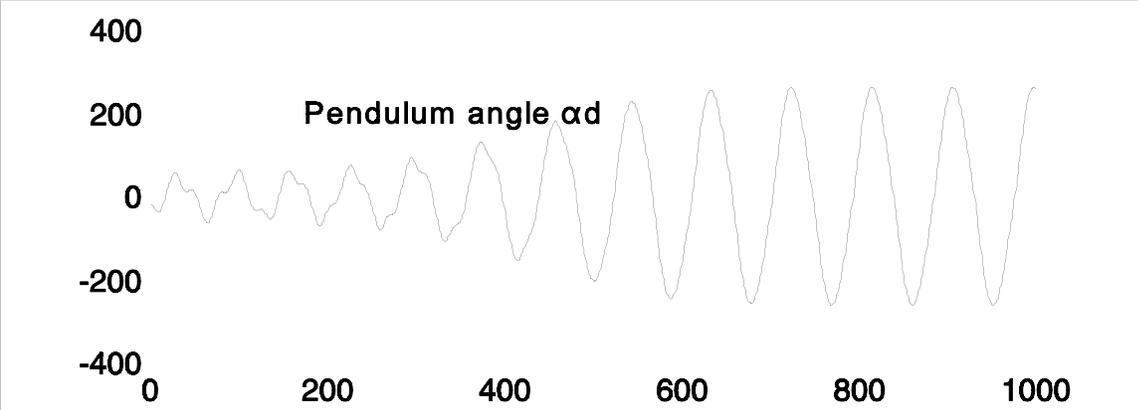


Figure 3.3: Simulation results showing that the pendulum can be brought up

### 3.2 Balance control

Assuming the pendulum is almost upright, a state feedback controller can be implemented that would maintain it upright (and handle disturbances up to a certain point). The state feedback controller is designed using the linear quadratic regulator and the linear model of the system.

The linear model that was developed is based on a torque  $T$  applied to the arm. The actual system however is voltage driven. From the motor equations derived above we obtain:

$$T = V \frac{K_m K_g}{R} - \frac{K_m^2 K_g^2}{R} \dot{\theta}$$

substituting this into the matrix equation and inserting the parameter values results in

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha}_u \\ \ddot{\theta} \\ \ddot{\alpha}_u \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -67.4 & -25.6 & 0 \\ 0 & 112.8 & 25.6 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \alpha_u \\ \dot{\theta} \\ \dot{\alpha}_u \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 47.4 \\ -47.4 \end{bmatrix} V$$

which is the desired representation.

Defining the LQR weighting matrices:

$$Q = \text{diag} ( [ .25 \ 4 \ 0 \ 1 ] )$$

$$r = .05$$

results in the feedback gains:

$$K = [ -.039 \ -6 \ -.042 \ -.11]$$

( Note that gains have been converted to V/deg and V / (deg/sec). (See *rotpen.m* )

### 3.3 Mode Control

The MODE controller determines when to switch between the two controllers. This requires some intuition (and a few tricks!).

Experimentally, we note that if you maintain the destabilizing motion to a fixed saturation amplitude (as in the simulation), there will come a time when the arm is moving too much and it actually starts reducing the amplitude of oscillations (*beating*).

So, to start from rest, you apply a few large movements until the amplitude of oscillation is large enough and then you switch to a smaller command amplitude that will slowly bring the pendulum higher.

Next, once you have attained an acceptable amplitude of pendulum oscillations, you want to start the stabilizing controller at the right time. Given that the stabilizing controller is linear, it has a small region in which it can stabilize the moving pendulum. Experimentally, we determine that it can stabilize the pendulum when it is about 15 degrees ( $\alpha_k$ ) from the vertical and not moving faster than 200 deg/sec. So, the “swing-up” controller should ensure that when the pendulum is almost upright, it is not moving too quickly. This is controlled by adjusting the gain P & D in the swing up controller.

Once the pendulum is up, it is easy to keep it up. But what if someone disturbs it? What should the system do? With the above scheme, if the disturbance causes the pendulum to swing beyond the 15 degrees then the mode switches to 0 and the swing up controller kicks in.... but this is not very good because the disturbance may be too large and the arm would move too far off from the zero position. It would be better to switch the mode to 0 when the angle is greater than maybe one or two degrees. Once then pendulum stabilizes, we should make the criterion a little more strict. This is done by changing the value of  $\theta_k$  dynamically. To start, we make  $\alpha_k = 15$  degrees but once the system has stabilized we change  $\theta_k$  to 1 degree. We should also ensure the value of MODE does not bounce. It is a switch after all and switches need “de-bouncing”. This is done by rate limiting the output of the AND gate and feeding it through a “deadband”. The parameters of the rate limiters and the deadband are obtained experimentally.

## 4.0 CONTROL SYSTEM IMPLEMENTATION

The system is implemented in SIMULINK. The controller is then run in WinCon which runs the SIMULINK diagram in realtime under Windows.

Figure 4.1 shows the SIMULINK controller *q\_rp\_se.m* supplied with the system. You can run this controller directly from WinCon by simply loading *q\_rp\_se.wcl* ( or .prj) into WinCon. If you want to perform changes to the parameters you need MATLAB, SIMULINK and Realtime Workshop. If you want to design other controller you also need Watcom C++.

The subsystems of the controller are now described in detail:

### 4.1 Calibration and differentiation

This block simply converts the value measured from the board to the appropriate units. The arm angle is measured in volts with a calibration constant of 35 Deg/volt. while the encoder measures 0.08789 degrees per count. The values are also fed through high pass filters which essentially differentiate the signals. Note also that the SRV-02

potentiometer voltage is passed through a “bias removal block” which measures the initial voltage for a short duration and then holds it in memory. This initial voltage is subsequently subtracted from all following samples.

## 4.2 Pendulum Up/Down measurements

This block converts the measurement of the pendulum angle from the “Full” position to two other frames of reference ( see figure 2.2).

**4.2.1  $\alpha_f$  (Full)** : The full position measures values which start at zero (hanging down) and increases in the positive direction when the pendulum rotates clockwise and decreases when the pendulum rotates counterclockwise. The value of the angle measured can exceed 360 degrees many times over in both directions. This value is obtained directly from the encoder counter.

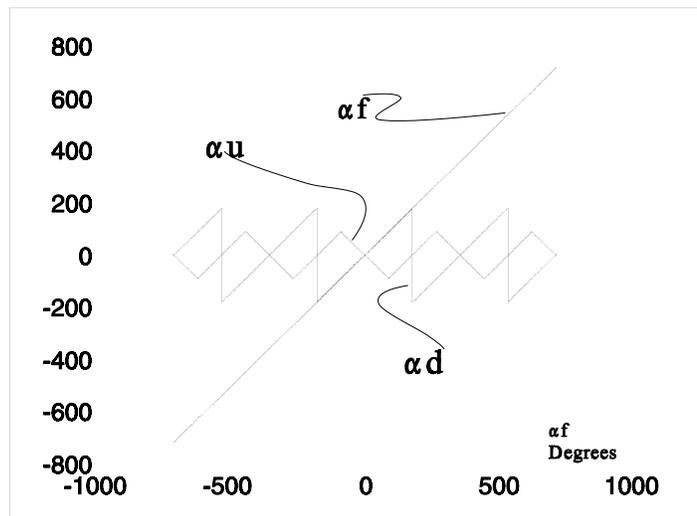
**4.2.2  $\alpha_u$  (UP)** : The UP position is relative to the vertical axis in the up position. It gives a zero value when the pendulum is straight up and a positive value when the pendulum is tilting to the right of the vertical and a negative value when it is to the left of the vertical. It is obtained by performing the following operation:

$$\alpha_u = -\sin^{-1} ( \sin(\alpha_f) ) \quad ( \text{note } \sin^{-1} \text{ returns values between } \pm\pi/2 )$$

**4.2.3  $\alpha_d$  (DOWN)** : The DOWN position is essentially the same as the FULL position expect that it is limited to values between  $\pm 180$  degrees (*NOT SATURATED*: Wrapped around!). This is the value used in the “swing-up” controller since the feedback is relative to the down position. If we feed back the FULL value and the pendulum does a full swing then swing up control will be ineffective! The value  $\alpha_d$  is obtained using the following equation:

$$\alpha_d = \tan^{-1} ( \sin(\alpha_f) , \cos(\alpha_f) )$$

( note  $\tan^{-1}$  returns values between  $\pm\pi$  )



Up / Down and Full measurements

Note that all trigonometric operations are performed on the angles converted to radians. The results are then converted back to degrees.

## 4.3 Swing up control

This is the “positive feedback loop” used to swing the pendulum up. It consists of two loops. The inner loop performs position control of the arm :

$$V_d = K_p (\theta - \theta_d) + K_d \dot{\theta}$$

The outer loop of the system generates the command  $\theta_d$  via the following equation:

$$\theta_d = P \alpha_d + D \dot{\alpha}_d$$

with  $P = 0.5$  deg/deg and  $D = 0.0$  deg / (deg/sec). These two values are crucial in bringing up the pendulum smoothly. You can tune the value of D to adjust the “damping” in the system.

Note also that  $\theta_d$  is limited to  $\pm 8$  Degrees. For the first 3 ( $T_{high}$ ) seconds of control however,  $\theta_d$  is multiplied by  $(1+Kh)$  resulting in a “high gain” system for a short time ( $Kh = 1.5$ ). This effectively speeds up the initial response to get maximum swing in the shortest possible time. Ideally, you want to move the arm back and forth a minimum number of times and bring the pendulum up quickly. If you do not have a high gain period, the system will still come up but it may take up to 30 oscillations of the arm to bring the pendulum up. Note that you may have to tap the pendulum slightly to start the feedback loop.

#### 4.4 Mode control

This block determines which of the two voltages should be fed to the motor. Initially, at startup, we want to feed the voltage computed by the swing up controller. When all of the conditions listed below are met, we want to switch to the “stabilizing” controller output voltage.

##### ***Stabilizing conditions:***

- $|\alpha_u| < \alpha_k$  (ie angle small enough)
- $|\theta| < \theta_k$  (ie arm position small enough)
- $\cos(\alpha_f) < 0$  (ie pendulum is up)
- $|\dot{\alpha}_f| < \dot{\alpha}_k$  (ie pendulum moving slowly enough)

When all of the above conditions are true, the output of the AND gate is 1. We limit the rate of change of the output and pass it through a backlash block in order to “de-bounce” it. This output is the signal **MODE**. When MODE is 0, the voltage fed to the motor is obtained from the “swing-up” controller, When MODE is 1, the voltage to the motor is obtained from the stabilizing controller.

Furthermore, MODE is fed to a delay block ( $0.2 / (s+0.2)$ ) and its output is fed to a comparator. When the output of the comparator is 1, the value of  $\alpha_k$  is reduced by 14 degrees to 1 degree. This now changes the stabilizing conditions to the following “**let**

**go**” conditions. When any of the following conditions is NOT TRUE, the system switches MODE to 0 and goes back to swing up mode.

**“Let Go” Conditions:**

- $|\alpha_u| < \alpha_k$  (ie angle disturbed too much )
- $|\theta| < \theta_k$  (ie arm position too far)
- $\cos(\alpha_f) < 0$  (ie pendulum is not up)
- $|\dot{\alpha}_f| < \dot{\alpha}_k$  (ie pendulum moving too quickly)

## 4.5 Balance control

This is the stabilizing state feedback controller. It simply feeds back the voltage:

$$V_s = -(k_1 \theta + k_2 \alpha + k_3 \dot{\theta} + k_4 \dot{\alpha})$$

as obtained in the LQR design.

This voltage maintains the pendulum upright

## 5.0 RESULTS

Typical results are shown in Figure 5.1,5.2 and 5.3. These are all obtained from the same run. Always start the system with the cart in the middle and the pendulum in the DOWN position. The controller begins by resetting the encoder counter to zero and measuring the bias in the potentiometer for subsequent subtraction. Thus the initial position is defined as “zero”.

The controller starts by applying a de-stabilizing arm position command limited to  $\pm 8^*(1+Kh)$  Deg. as shown in Figure 5.1 This results in the pendulum swinging with increasing amplitude. At  $T = 3$  seconds ( as set in Figure 4.4 ) the arm command is limited to  $\pm 8$  Deg. The destabilizing controller continues however to increase the amplitude of the pendulum oscillations.

At some point, all the following conditions are met:

- Pendulum UP angle < 15 degrees
- Pendulum speed < 200 deg/sec
- Arm position < 25 deg
- Pendulum Full angle in 2nd or 3rd quadrant

This sets MODE = 1 and the state feedback controller output  $V_s$  is fed to the motor rather than the destabilizing voltage  $V_d$ . This stabilizes the pendulum and keeps it upright.

The Mode is maintained to the value 1 unless one of the following conditions becomes false:

- Pendulum UP angle less than 2 degrees
- Arm position < 25 deg
- Pendulum velocity < 200 deg/sec
- Pendulum Full angle is in 2nd or 3rd quadrant as defined in Figure 2.2

If you apply a tap to the pendulum, it will remain upright unless the tap was so hard that one of the above conditions is not maintained. If MODE switches to zero, then the pendulum falls and the cart will start oscillating again until the pendulum comes up and MODE switches to 1. Note however that there may be situations when the tap was so hard that the pendulum swinging in full rotations and never stabilizes. It is of course possible to design a controller that avoids this situation. One way would be to implement a controller that detects full rotations and turns off the system momentarily until the pendulum is simply swinging in the down mode and then resume to the swing up controller in low gain. A different method of applying a disturbance is to push against the arm until the controller “gives up” and lets the pendulum fall.

## 6.0 SYSTEM PARAMETERS

Parameter	Symbol	Value	Units
Motor Torque constant (and back emf constant in SI units)	Km	0.00767	Nm/Amp V/ rad sec <sup>-1</sup>
Motor Armature Resistance	Rm	2.6	Ω
Internal gearbox ratio	N/A	14.1	N/A
External gearbox ratio(high gear ratio)	N/A	5	N/A
Total gear ratio	Kg	60.5	N/A
Motor pinion # teeth	NA	24	teeth
Total inertia after gearbox including arm	J_b	0.0044	Kg m <sup>2</sup>
Pendulum true length	Lp	0.43	m
Pendulum mass	mp	0.14	Kg
Servo position potentiometer sensitivity	N/A	35	Deg/Volt
Pendulum angle encoder resolution	N/A	1024 * 4	count/rev
Pendulum angle calibration constant	N/A	.08789	deg/count

## 7.0 WIRING

Wire the system as shown in Figure 7.1. The following cables are supplied with the system:

Wire type	From	To
6 Pin Mini Din - 6 Pin Mini Din	SRV-02 6 Pin Mini Din socket (either one)	Quick Connect S1&S2 socket
2 Pin Din - 2 Banana	Power module RED to OUT BLACK to GND FORK to (-)	Motor on cart
5 Pin Din - 5 Pin Din	Pendulum angle encoder	Enc #0 on <b>MultiQ</b> terminal board
5 Pin Din - 4 x RCA	Quick Connect "To A/D"	<b>Potentiometer</b> S1 to <b>MultiQ</b> A/D #0  <b>Not Used</b> S2 to MultiQ A/D #1 S3 to MultiQ A/D #2 S4 to MultiQ A/D #3
2 Pin Din - RCA	Quick Connect "From D/A"	<b>MultiQ</b> D/A #0 (Analog output)

## 8.0 SOFTWARE SUPPLIED WITH YOUR SYSTEM

File name	Function	Required SW
rotpen.map	Derive differential equations	MAPLE
rotp_dm.m	Linear model for down position	MATLAB
rotpen_m.m	Linear model for up position	MATLAB
down.m	Linear model parameters for “down” system	MATLAB
servpos.m	Design bring up Controller (Kp &Kd)	MATLAB
rotpen.m	Design state feedback controller	MATLAB
s_down.m	Simulate “bring up” controller	SIMULINK etc..
q_rp_se.m	SIMULINK controller	SIMULINK etc..
q_rp_se.wcl	Precompiled WinCon Realtime controller from SIMULINK controller	WinCon etc..

**Questions? Contact us at [help@quanser.com](mailto:help@quanser.com)**

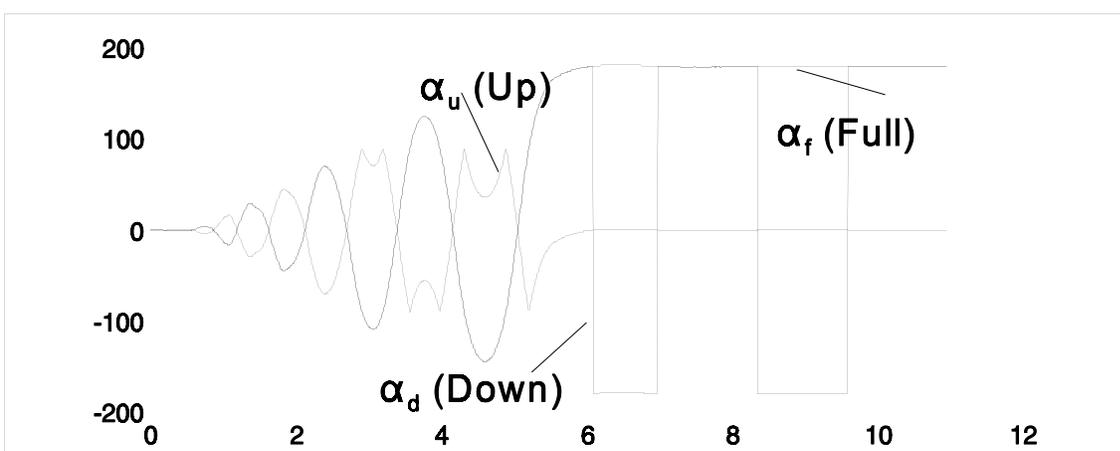
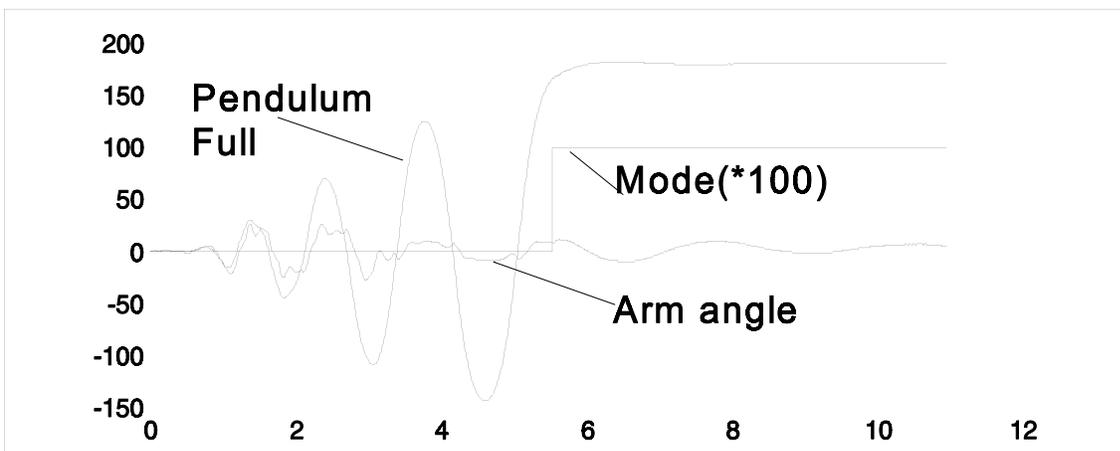
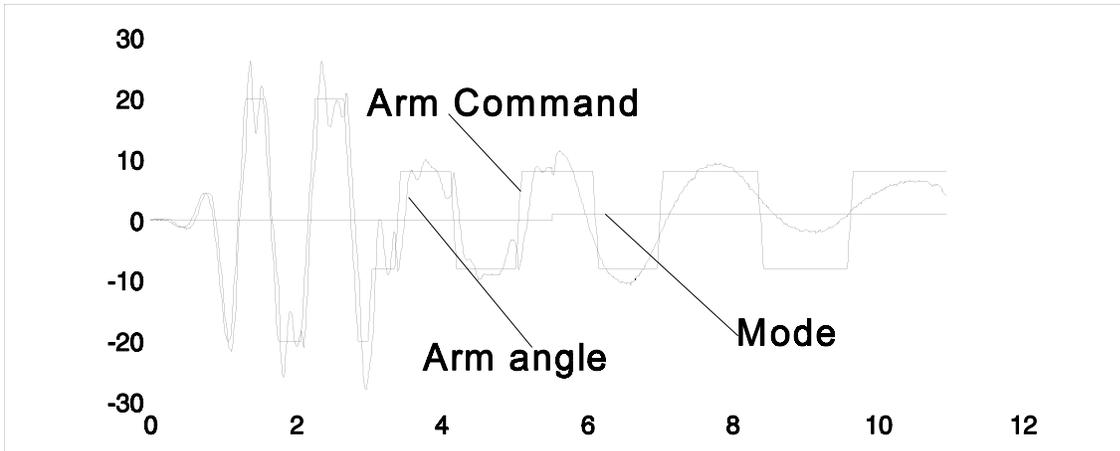


Figure 5.1, 5.2 & 5.3: Actual results obtained from the system. (Note MODE is multiplied by 100 for visibility)