Rotary Inverted Pendulum

Eric Liu

1 Aug 2013
1 State Space Derivations

1.1 Electromechanical Derivation

Consider the given diagram. We note that the voltage across the motor can be described by:

\[ e_b = k_m \omega_m \]  

(1.1)

where \( k_m \) is the back-emf constant of the motor, and \( \omega_m \) is the speed of the motor shaft.

By Kirchhoff’s Laws, the circuit shows that:

\[ V_m - I_m - L_m \frac{dI_m}{dt} - k_m \omega_m = 0. \]  

(1.2)

If we assume that the motor inductance \( L_m \) is insignificant, (1.2) can be reduced to:

\[ V_m - I_m - k_m \omega_m = 0 \]  

(1.3)

Solving for \( I_m \),
\[ I_m = \frac{V_m - k_m \omega_m}{R_m} \quad (1.4) \]

The motor torque is proportional to the voltage applied and is described as:

\[ \tau_m = \eta_m k_t I_m \quad (1.5) \]

where \( \eta_m \) is the motor efficiency and \( k_t \) is the current-torque constant.

If we consider the internal gear system as two gears with negligible friction and moment of inertia, we know that:

\[ \tau_{eq} = \eta_g K_g \tau_m \quad (1.6) \]

where \( \eta_g \) is the gearbox efficiency, and \( K_g \) is the total gear ratio. (Appendix, Figure (1)) \( K_g \) is provided in the manual, or otherwise can be calculated as the product of the exterior and interior gear ratios. That is,

\[ K_g = K_{ge} K_{gi} = \frac{N_4}{N_3} \frac{N_2}{N_1} \]

Additionally, we see that due to rotational alignments, \( \theta_m = K_g \theta_l \), implying that:

\[ \omega_m = K_g \omega_l \quad (1.7) \]

Examining the equivalent torque with respect to the load, we have:

\[ \tau_{eq} = J_{eq} \frac{d\omega_l}{dt} + B_{eq} \omega_l + rF \quad (1.8) \]

where \( F \) is the equivalent force to be applied to our pendulum, \( B_{eq} \) is the viscous damping coefficient, and \( J_{eq} \) is the moment of inertia as seen at the output, which can be also calculated as \( J_{eq} = J_l + \eta_g J_m K_g^2 \). \( (J_m \) is the moment of inertia of the internal gear.)

Substituting in (1.4), (1.5), (1.6), and (1.7), we get:

\[ \eta_m \eta_g k_t K_g \left( \frac{V_m - K_g k_m \omega_l}{R_m} \right) = J_{eq} \frac{d\omega_l}{dt} + B_{eq} \omega_l + rF \quad (1.9) \]
1.2 Pendulum Motion Derivations

Examining the schematic (Appendix, Figure (2)), we note that if the pendulum arm must travel in a fixed circular trajectory with rotational movement solely tangential to its trajectory. Consequently, if we model the pendulum arm as a point mass at it’s center point with distance $L$ from the joint, we can derive the following base equations.

**Case 1: Hanging pendulum**

\[ F_y = mg + m\ddot{y} \quad (2.1) \]
\[ F_x = m\ddot{x} \quad (2.2) \]
\[ \tau_m + F_x y + F_y x = 0 \quad (2.3) \]

Now fixing the coordinate system relative to the rotating arm’s base, we can also establish that:

1. \[ y(t) = -L \cos \alpha \]
2. \[ x(t) = r \theta(t) + L \sin \alpha \]

Where $r$ is the length of the rotating arm.

Substituting, our base equations can now be expressed as:

\[ F_y = mg + m\frac{d^2}{dt^2}(-L \cos \alpha) \quad (2.4) \]
\[ F_x = m\frac{d^2}{dt^2}(r \theta + L \sin \alpha) \quad (2.5) \]
\[ J_p \ddot{\alpha} + F_y L \sin \alpha + F_x L \cos \theta = 0 \quad (2.6) \]

We note that for small angles, $\sin \theta \approx \theta$, $\cos \theta \approx 1$. In addition,

\[ \frac{d^2}{dt^2} \cos \theta = \left( \frac{d\theta}{dt} \right)^2 \cos \theta - \sin \theta \frac{d^2\theta}{dt^2} \]

and
\[
\frac{d^2}{dt^2} \sin \theta = - \left( \frac{d\theta}{dt} \right)^2 \sin \theta + \cos \theta \frac{d^2 \theta}{dt^2}
\]

Consequently our equations are now:

\[
\begin{align*}
F_y &= mg + mL(\dot{\alpha}^2 + \alpha \ddot{\alpha}) \\
F_x &= m(r \dot{\theta} + L(\dot{\alpha}^2 \alpha + \dot{\alpha})) \\
J_p \ddot{\alpha} + F_y \alpha + F_x L &= 0
\end{align*}
\]

(2.7) (2.8) (2.9)

Linearizing around \(\alpha = 0\), we note that \(\dot{\alpha}^2 \approx 0, \alpha \ddot{\alpha} \approx 0\), so:

\[
\begin{align*}
F_y &= mg \\
F_x &= m(r \ddot{\theta} + L \ddot{\alpha}) \\
J_p \ddot{\alpha} + F_y \alpha + F_x L &= 0
\end{align*}
\]

(2.10) (2.11) (2.12)

We also know that \(J_p = \frac{1}{3} mL^2\). Combining, we get:

\[
\frac{1}{3} mL^2 \ddot{\alpha} + mgL\alpha + m(r \ddot{\theta} + L \ddot{\alpha})L = 0
\]

(2.13)

which we can rewrite as:

\[
\frac{4}{3} mL^2 \ddot{\alpha} + mgL\alpha + mrL \ddot{\theta} = 0
\]

(2.14)

Redefining equation (1.9) in terms of \(\theta\) and setting the input force such that \(F = F_x\), we have:

\[
\eta_m \eta_g k_i K_q \left( \frac{V_m - K_y k_m \dot{\theta}}{R_m} \right) = mr^2 \ddot{\theta} + mrL \ddot{\alpha} + J_{eq} \ddot{\theta} + B_{eq} \dot{\theta}
\]

(2.15)

We can solve for \(\ddot{\theta}\) and \(\ddot{\alpha}\) by first rewriting the notation. Let

1. \(a = J_{eq} + mr^2\)
2. \(b = mLr\)
3. \(c = \frac{4}{3} mL^2\)
4. \( d = mgL \)

5. \( E = ac - b^2 \)

6. \( G = \frac{\eta_m \eta_g k_i K_g^2 + B_{eq} R_m}{R_m} \)

Then we have:

\[
\begin{align*}
\ddot{\alpha} + d\alpha + b\dot{\theta} &= 0 \\
\frac{\eta_m \eta_g k_i K_g^2}{R_m} V_m - G\dot{\theta} &= a\dot{\theta} + b\ddot{\alpha}
\end{align*}
\]

solving, we get:

\[
\begin{align*}
\ddot{\alpha} &= \frac{ad}{E} \alpha + \frac{bG}{E} \dot{\alpha} - \frac{b \eta_m \eta_g k_i K_g^2}{R_m E} V_m \\
\ddot{\theta} &= \frac{bd}{E} \alpha - \frac{cG}{E} \dot{\theta} + \frac{c \eta_m \eta_g k_i K_g^2}{R_m E} V_m
\end{align*}
\]

which consequently can be written in state space form:

\[
\begin{pmatrix}
\dot{\theta} \\
\dot{\alpha} \\
\dot{\theta} \\
\ddot{\alpha}
\end{pmatrix} =
\begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & \frac{bd}{E} & -\frac{cG}{E} & 0 \\
0 & \frac{ad}{E} & \frac{bG}{E} & 0
\end{pmatrix}
\begin{pmatrix}
\theta \\
\alpha \\
\dot{\theta} \\
\ddot{\alpha}
\end{pmatrix} +
\begin{pmatrix}
0 \\
0 \\
\frac{c \eta_m \eta_g k_i K_g}{R_m E} \\
-\frac{b \eta_m \eta_g k_i K_g}{R_m E}
\end{pmatrix} V_m
\]

Case 2: Inverted Pendulum

In the case of the inverted pendulum, we start with slightly different base equations:

\[
\begin{align*}
F_y &= mg + m\ddot{y} \\
F_x &= m\ddot{x} \\
\tau_m - F_x y - F_y x &= 0
\end{align*}
\]

Fixing the coordinate system relative to the rotating arm’s base, we establish that:

1. \( y(t) = L \cos \alpha \)
2. \( x(t) = r\theta(t) - L\sin \alpha \)

Following a similar derivation to the hanging pendulum, we solve to get:

\[
\frac{4}{3} mL^2 \ddot{\alpha} - mgL\alpha - mrL\ddot{\theta} = 0 \tag{3.4}
\]

and

\[
\eta_m \eta_g k_t K_g \left( \frac{V_m - K_g k_m \dot{\theta}}{R_m} \right) = mr^2 \ddot{\theta} - mrL\ddot{\alpha} + J_{eq} \ddot{\theta} + B_{eq} \dot{\theta} \tag{3.5}
\]

Doing the same substitutions as mentioned before, we have:

\[
ca \ddot{\alpha} - d\alpha - b\ddot{\theta} = 0 \tag{3.6}
\]

\[
\eta_m \eta_g k_t K_g \frac{V_m}{R_m} - G\dot{\theta} = a\ddot{\theta} - b\dddot{\alpha} \tag{3.7}
\]

solving, we get:

\[
\ddot{\alpha} = \frac{ad}{E} \alpha - \frac{bG}{E} \theta + \frac{b\eta_m \eta_g k_t K_g}{R_mE} V_m \tag{3.8}
\]

\[
\ddot{\theta} = \frac{bd}{E} \alpha - \frac{cG}{E} \dot{\theta} + \frac{c\eta_m \eta_g k_t K_g}{R_mE} V_m \tag{3.9}
\]

which leads to the state space equation:

\[
\begin{pmatrix}
\dot{\theta} \\
\dot{\alpha} \\
\ddot{\alpha}
\end{pmatrix}
= \begin{pmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & \frac{bd}{E} & \frac{cG}{E} & 0
\end{pmatrix}
\begin{pmatrix}
\theta \\
\alpha \\
\dot{\theta} \\
\dot{\alpha}
\end{pmatrix}
+ \begin{pmatrix}
0 \\
0 \\
\frac{c\eta_m \eta_g k_t K_g}{R_mE} \\
\frac{b\eta_m \eta_g k_t K_g}{R_mE}
\end{pmatrix}
V_m \tag{3.10}
\]
2 Model Verification

Using our state space model, we can evaluate the stability of both the hanging pendulum equilibrium and the inverted pendulum equilibrium by using MATLAB (Appendix 2.1, 2.2) to calculate the transfer function.

For the hanging equilibrium, the transfer function is:

\[
\frac{EK_g b \eta g \eta m k_t s}{R_m (-Gdb^2 + E^2s^3 + EGcs^2 + Eads + Gacd)}
\]

which can be rearranged as

\[
\frac{EK_g b \eta g \eta m k_t s}{R_m (E^2s^3 + EGcs^2 + Eads - Gdb^2 + Gacd)}
\]

Looking at the transfer function, we note that all eigenvalues are negative, indicating that the equilibrium is stable, as expected.

For the inverted equilibrium, the transfer function is:

\[
\frac{-EK_g b \eta g \eta m k_t s}{R_m (E(ads - Gcs^2) - E^2s^3 - Gbd + Gacd)}
\]

which can be rearranged as

\[
\frac{K_g b \eta g \eta m k_t s}{R_m (E^2s^3 + Gcs^2 - Eads + Gbd - Gacd)}
\]

Looking at the transfer function, we note that not all eigenvalues are negative, indicating that the equilibrium is unstable, which is also as expected.
Subsequently, we can attempt to confirm the validity of our model by calculating the theoretical pulse response of the hanging pendulum system and comparing it to experimental results, using variable values obtained from the SRV02 User Manual (Appendix, Figure (3)) combined with values obtained from the SRV02 Rotary Pendulum Manual (Appendix, Figure (4)).

Using MATLAB’s Simulink and subsequent data collection methods (Appendix, Figure (5), Appendix 2.3), we plot the experimental response, response using SRV02 user manual’s suggested state space value’s, and our own model’s response.
From these charts, we see that the experimental results indicate far less damping than expected in the manual’s model, and significantly less than our model as well. However, the period length and initial amplitude are reasonably close. If we decrease the damping by changing the motor armature resistance, $R_m$ from 2.6Ω to 0.6Ω in our model, it closely mirror the experimental results.

Unfortunately, we could not confirm the actual motor armature resistance at the time, and thus did not make the change in our systems model. However, it is noted that the electrical control system is sufficient for controlling either scenario.
3 Control

3.1 Hanging Pendulum

We now use a design procedure known as pole placement. If we express the state system model as:

\[ \dot{\vec{x}}(t) + \vec{A}\vec{x}(t) + \vec{B}u(t) \]  
\[ y(t) = \vec{C}\vec{x}(t) \]

and \( u(t) \) is the output and \( r(t) = 0 \) is the control system’s input, we can express \( u(t) \) as a function of the form

\[ u(t) = f[\vec{x}(t)] = \vec{K}\vec{x}(t) \]

where \( \vec{K} \) is a 1 \( \times \) \( n \) vector of constant gains. This function is a control law that allows all poles of the closed-looped system below to be placed in any desirable locations. The rule can be expressed as

\[ u(t) = -K_1x_1(t) - K_2x_2(t) - \ldots - K_nx_n(t) \]
We can use such an regulatory control system to control our hanging pendulum, by setting the desired location and positioning of the pendulum as "zero", and hence our equilibrium point. The control system will force the physical system to settle into its equilibrium at a pace determined by the desired poles. In our system, we picked the poles to be:

\[
P = 5 \times [-1 - 2 - 1/5 - 2.5]
\]  \hspace{1cm} (3.5)

Subsequently we use MATLAB to solve for the corresponding K-values by using the "place" command (Appendix 2.4).

Once we have the corresponding K-values, we can fully implement the control system using simulink (Appendix, Figure (6)), and compare results with a theoretical simulation (Appendix, Figure (7)).

We note that in order to compare the theoretical model to the actual system, there exists formulaic transformation between \(\alpha, \theta\) and the measured voltage output from the equipment sensors. In our test trial, we accordingly adjust the values by:

\[
\theta = (V_\theta + 0.08) \times \frac{\pi}{5.42} \hspace{1cm} (3.6)
\]

\[
\alpha = (V_\alpha - 2.05) \times \frac{\pi}{2} \hspace{1cm} (3.7)
\]
3.1.1 Resulting Scopes

Theoretical $\theta$ and $\alpha$: 

![Graphs showing theoretical values of $\theta$ and $\alpha$.]
Experimental $\theta$ and $\alpha$: 

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure}
\caption{Graph showing the experimental values of \theta and \alpha.}
\end{figure}
We notice that the experimental result has slightly less damping than our theoretical. In practice, the motor requires a minimum amount of voltage in order to move. It must overcome this minimum "requirement" barrier in order to respond. Hence, for small oscillations after the system has essentially reached equilibrium, the actual system is incapable of implementing further control measures. However, this control system achieved the desired effects of moving the pendulum system to equilibrium. In addition, Figure (8) in the appendix is an example of a model used to alternate the equilibrium by adding a pulse input to the output system.
3.2 Inverted Pendulum

In the case of the inverted pendulum, we note that it’s equilibrium is unstable, and is quicker to fall out of the linear region defined by the model derivation. To ensure an appropriate system reaction time, we adjusted the poles to decrease the time constants by half:

\[
P = 10 \times [-1 - 2 - 1/5 - 2.5];
\] (3.8)

Subsequently, we used MATLAB again to solve for the corresponding K-values, and tested a theoretical simulink model to check for expected output (Appendix, Figure (9)). Upon confirmation that the theoretical model exhibited expected behavior, we then implemented the control system using simulink as well (Appendix, Figure (10)).

The experimental simulink model settings were re-optimized to allow for sample time of 0.0025 seconds, an upgrade from the previous 0.01 sample time used for the hanging pendulum system. In addition, restrictions were set to limit current flow to the motor only when the rotary arm was less than 90 degrees from equilibrium and the voltage was less than 8 V.

In addition, several minor precautions were taken prior to experimentation. First, the pendulum connecting pin was re-checked to ensure minimal slipping. Secondly, the voltmeter measuring the equilibrium point was reset to ensure accuracy. Finally, the rotary arm was held initially in place in order to give the system enough time to begin reacting.

Although this model succeeded in attaining equilibrium, it still exhibited some negative effects, primarily initial instability and subsequent overreaction to any type of input. While this may be in part caused by sensory limitation, we would still like to re-evaluate our pole-placement to produce a more optimal response reaction. Hence we turn to LQR (Linear Quadratic Regulator) algorithm.
3.3 LQR control

LQR sets the poles of the system based on a mathematical algorithm that minimizes a cost function with weighting factors determined by the user. Thus for our system:

\[
\dot{x}(t) + A\dot{x}(t) + Bu(t) \\
y(t) = C\dot{x}(t)
\] (3.1)

We consider the weighting factors to be our state variables \( \dot{x}(t) \) and input \( u(t) \).

In general notation, the minimization function can be written as:

\[
\min J = \int_{0}^{\infty} (\dot{x}(t)^T Q \dot{x}(t) + \dot{u}(t)^T R \dot{u}(t)) dt
\] (3.9)

where Q and R are appropriate matrices containing weights used to penalize their corresponding factor. We also note that for practical purposes, Q and R must be chosen such that the terms in the minimization function are positive definitive, or occasionally semi-positive definitive.

In our control system, if we have a feedback response such that \( \dot{u}(t) = K\dot{x}(t) \), we can solve the Riccati Equation:

\[
A^T P + PA - PBR^{-1}B^T P + Q = 0
\] (3.10)

for P, from which we can evaluate K, where \( K = R^{-1}B^T P \). Then our transition matrix \( (A - BK) \) has stable eigenvalues that minimize the cost function \( J \).

For the inverted pendulum, we choose to weight only \( \alpha, \theta \) and our input voltage.

Thus our Q and R matrices are:

\[
Q = \begin{pmatrix}
q_1 & 0 & 0 & 0 \\
0 & q_2 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
\text{and } R = (r_1)
\] (3.11)
The questions remains as to how to pick the values within the Q and R matrices. For our physical model, we would prefer that input voltage to be under 10 v, with priority set for quicker response to $\alpha$ over $\theta$, and an overall adequate settling time. We use LQR as an iterative process, picking initial weight values and adjusting to fit our criterion. Since we know that the total range of both angles is $2\pi$, we picked initial values:

\[ q_1 = \left( \frac{1}{2\pi} \right)^2 \]  
\[ q_2 = \left( \frac{1}{2\pi} \right)^2 \]  
\[ r_1 = \left( \frac{1}{10} \right)^2 \] 

We then solved for the corresponding K-values and examined the theoretical response using a simulink model, (Appendix, Figure (9)). Testing various Q and R matrices, we can produce the following table:

<table>
<thead>
<tr>
<th>Trial</th>
<th>Q1, Q2</th>
<th>R</th>
<th>K-Values</th>
<th>$\theta_{max}$</th>
<th>$\theta_{min}$</th>
<th>$\alpha_{max}$</th>
<th>$\alpha_{min}$</th>
<th>$V_{max}$</th>
<th>$V_{min}$</th>
<th>Settle Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Previous Pole Placement Results: [-11.7 37.2 -3.6 5.3]</td>
<td>2.75</td>
<td>-0.15</td>
<td>0.80</td>
<td>-0.35</td>
<td>20.0</td>
<td>-15</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1/$\pi^2$, 1/$\pi^2$</td>
<td>0.01</td>
<td>[-1.09 10.3 -1.80 2.9]</td>
<td>1.70</td>
<td>0</td>
<td>0.11</td>
<td>-0.03</td>
<td>2.5</td>
<td>-1.50</td>
<td>2.5</td>
</tr>
<tr>
<td>3</td>
<td>1/$\pi^2$, 25/$\pi^2$</td>
<td>0.01</td>
<td>[-1.09 22.3 -1.81 3.0]</td>
<td>1.70</td>
<td>0</td>
<td>0.11</td>
<td>-0.03</td>
<td>2.4</td>
<td>-1.50</td>
<td>2.5</td>
</tr>
<tr>
<td>4</td>
<td>1/$\pi^2$, 100/$\pi^2$</td>
<td>0.01</td>
<td>[-1.09 274.1 -196 12]</td>
<td>1.70</td>
<td>0</td>
<td>0.09</td>
<td>-0.02</td>
<td>2.5</td>
<td>-1.20</td>
<td>2.4</td>
</tr>
<tr>
<td>5</td>
<td>1/$\pi^2$, 1/$\pi^2$</td>
<td>0.005</td>
<td>[-2.20 22.6 -2.03 3.3]</td>
<td>1.80</td>
<td>0</td>
<td>0.14</td>
<td>-0.05</td>
<td>3.5</td>
<td>-2.00</td>
<td>2.0</td>
</tr>
<tr>
<td>6</td>
<td>1/$\pi^2$, 25/$\pi^2$</td>
<td>0.005</td>
<td>[-2.25 26.2 -2.12 3.5]</td>
<td>1.70</td>
<td>0</td>
<td>0.13</td>
<td>-0.04</td>
<td>3.5</td>
<td>-1.40</td>
<td>2.0</td>
</tr>
<tr>
<td>7</td>
<td>1/$\pi^2$, 100/$\pi^2$</td>
<td>0.005</td>
<td>[-2.25 34.5 -2.30 3.8]</td>
<td>1.70</td>
<td>0</td>
<td>0.10</td>
<td>0.03</td>
<td>3.5</td>
<td>-1.30</td>
<td>2.4</td>
</tr>
<tr>
<td>8</td>
<td>1/$\pi^2$, 1/$\pi^2$</td>
<td>0.001</td>
<td>[-5.03 32.1 -2.94 4.7]</td>
<td>1.85</td>
<td>0</td>
<td>0.25</td>
<td>-0.10</td>
<td>7.8</td>
<td>-3.30</td>
<td>1.7</td>
</tr>
<tr>
<td>9</td>
<td>1/$\pi^2$, 25/$\pi^2$</td>
<td>0.001</td>
<td>[-5.03 44.2 -3.35 4.4]</td>
<td>1.75</td>
<td>0</td>
<td>0.18</td>
<td>-0.05</td>
<td>7.8</td>
<td>-2.00</td>
<td>1.8</td>
</tr>
<tr>
<td>10</td>
<td>1/$\pi^2$, 100/$\pi^2$</td>
<td>0.001</td>
<td>[-5.03 66.4 -3.96 5.5]</td>
<td>1.75</td>
<td>0</td>
<td>0.12</td>
<td>-0.03</td>
<td>7.8</td>
<td>-1.20</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Trial 8 fit our specifications best, so we proceeded to implement it’s corresponding K-values into our sinusoidal equilibrium system using the same simulink model (Appendix, Figure (10)).

Thus we complete our optimal control system for the Inverted Pendulum, which can be observed in application.
4 Appendix

4.1 Figures

Figure 1: Gear motors

Figure 2: Pendulum Derivation Diagram
Figure 3: SRV02 Motor Specifications

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Matlab Variable</th>
<th>Value</th>
<th>Variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>V&lt;sub&gt;nom&lt;/sub&gt;</td>
<td>Motor nominal input voltage</td>
<td>Rm</td>
<td>6.0 V</td>
<td>± 12%</td>
</tr>
<tr>
<td>R&lt;sub&gt;m&lt;/sub&gt;</td>
<td>Motor armature resistance</td>
<td>Lm</td>
<td>2.6 Ω</td>
<td>± 12%</td>
</tr>
<tr>
<td>L&lt;sub&gt;m&lt;/sub&gt;</td>
<td>Motor armature inductance</td>
<td></td>
<td>0.18 mH</td>
<td>± 12%</td>
</tr>
<tr>
<td>k&lt;sub&gt;t&lt;/sub&gt;</td>
<td>Motor current-torque constant</td>
<td>kt</td>
<td>7.68 x 10&lt;sup&gt;-3&lt;/sup&gt; N m/A</td>
<td>± 12%</td>
</tr>
<tr>
<td>k&lt;sub&gt;m&lt;/sub&gt;</td>
<td>Motor back-emf constant</td>
<td>km</td>
<td>7.68 x 10&lt;sup&gt;-3&lt;/sup&gt; V/(rad/s)</td>
<td>± 12%</td>
</tr>
<tr>
<td>K&lt;sub&gt;y&lt;/sub&gt;</td>
<td>High-gear total gear ratio</td>
<td>Kg</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>K&lt;sub&gt;y&lt;/sub&gt;</td>
<td>Low-gear total gear ratio</td>
<td>Kg</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>η&lt;sub&gt;m&lt;/sub&gt;</td>
<td>Motor efficiency</td>
<td>eta.m</td>
<td>0.69</td>
<td>± 5%</td>
</tr>
<tr>
<td>η&lt;sub&gt;p&lt;/sub&gt;</td>
<td>Gearbox efficiency</td>
<td>eta.g</td>
<td>0.90</td>
<td>± 10%</td>
</tr>
<tr>
<td>J&lt;sub&gt;m,rotor&lt;/sub&gt;</td>
<td>Rotor moment of inertia</td>
<td>Jm_rotor</td>
<td>3.90 x 10&lt;sup&gt;-7&lt;/sup&gt; kg · m&lt;sup&gt;2&lt;/sup&gt;</td>
<td>± 10%</td>
</tr>
<tr>
<td>J&lt;sub&gt;tech&lt;/sub&gt;</td>
<td>Tachometer moment of inertia</td>
<td>Jtech</td>
<td>7.06 x 10&lt;sup&gt;-8&lt;/sup&gt; kg · m&lt;sup&gt;2&lt;/sup&gt;</td>
<td>± 10%</td>
</tr>
<tr>
<td>J&lt;sub&gt;eq&lt;/sub&gt;</td>
<td>High-gear equivalent moment of inertia</td>
<td>Jeq</td>
<td>9.76 x 10&lt;sup&gt;-5&lt;/sup&gt; kg · m&lt;sup&gt;2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>B&lt;sub&gt;eq&lt;/sub&gt;</td>
<td>High-gear equivalent viscous damping coeff.</td>
<td>Beq</td>
<td>0.015 N · m · (rad/s)</td>
<td></td>
</tr>
<tr>
<td>m&lt;sub&gt;b&lt;/sub&gt;</td>
<td>Mass of bar load</td>
<td>m_b</td>
<td>0.038 kg</td>
<td></td>
</tr>
<tr>
<td>L&lt;sub&gt;b&lt;/sub&gt;</td>
<td>Length of bar load</td>
<td>L_b</td>
<td>0.1525 m</td>
<td></td>
</tr>
</tbody>
</table>

Note: length of bar load can vary between 0.1525 m and 0.175 m depending on placement of pendulum arm, but has little impact on overall results.

Figure 4: SRV02 Pendulum Specifications

6.1 System Specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coupled Arm Length</td>
<td>20</td>
<td>cm</td>
</tr>
<tr>
<td>Long Pendulum Length</td>
<td>65</td>
<td>cm</td>
</tr>
<tr>
<td>Short Pendulum Length</td>
<td>35</td>
<td>cm</td>
</tr>
<tr>
<td>Long Pendulum mass</td>
<td>0.231</td>
<td>kg</td>
</tr>
<tr>
<td>Short Pendulum mass</td>
<td>0.128</td>
<td>kg</td>
</tr>
<tr>
<td>ROTPEN mass <em>without pendulum</em></td>
<td>0.278</td>
<td>kg</td>
</tr>
<tr>
<td>Potentiometer Bias Power</td>
<td>±12</td>
<td>Volts</td>
</tr>
<tr>
<td>Potentiometer Measurement Range</td>
<td>±5</td>
<td>Volts</td>
</tr>
<tr>
<td>Encoder Resolution</td>
<td>4096</td>
<td>Counts/Rev.</td>
</tr>
</tbody>
</table>

Table 6 - ROTPEN Specifications
Figure 5: Model Checking Simulink Diagram

Figure 6: Simulink for Hanging Equilibrium
Figure 7: Simulink for Hanging Theoretical Comparison

Figure 8: Example of Alternating Equilibrium
Figure 9: Simulink for Inverted Theoretical Comparison

Figure 10: Simulink for Inverted Equilibrium
4.2 MATLAB code

2.1 Transfer function (Downwards)

```matlab
% Ruisen (Eric) Liu
% Calculate the transfer function for pendulum’s 
downwards equilibrium

clear all

% Downwards system
syms s a b c d E F G eta_m eta_g k_t K_g R_m
sIA = [ s 0 -1 0 ; 0 s 0 -1 ; 0 -b*d/E s+c*G/E 0 ; 0 a
       *d/E -b*G/E s ];
B = [ 0 ; 0 ; c*eta_m*eta_g*k_t*K_g/(R_m*E) ;
     -b*eta_m*eta_g*k_t*K_g/(R_m*E) ];
phi = inv(sIA);
pretty(phi)

% Want alpha value
C = [0 1 0 0];
D = 0;
% Transfer Function - no D matrix
Trans = C*phi*B;
pretty(simple(Trans))
```
2.2 Transfer function (Upright)

% Ruisen (Eric) Liu
% Calculate the transfer function for pendulum's upright equilibrium.

clc

% Upright system
syms t s a b c d E F G eta_m eta_g k_t K_g R_m
A = [s 0 -1 0; 0 s 0 -1; 0 -b*d/E s+c*G/E 0; 0 -a*d/E b*G/E s];
B = [0; 0; c*eta_m*eta_g*k_t*K_g/(R_m*E); b*eta_m*eta_g*k_t*K_g/(R_m*E)];
phi = inv(A);
pretty(phi);

% Want alpha value
C = [0 1 0 0];
D = 0;

% Transfer Function - no D matrix
Trans = C*phi*B;
pretty(simple(Trans))
2.3 Model Check

% Ruisen (Eric) Liu

% Confirmation of Model Accuracy

% Simulink Experimental Plotting

% Run Inverted_Pendulum.mdl Simulink file and use:

\[
t_{\text{sim}} = \text{PendulumAngle.time};
\]
\[
z = \text{PendulumAngle.signals.values};
\]

\begin{verbatim}
figure(1)
plot(t_sim, z)
xlabel('Time(S)');
ylabel('Voltage (V)');
title('Experimental Results');
axis([0, 10, 1.75, 2.5]);
grid on
\end{verbatim}

%Check Step Response from Model

%variables t s a b c d E F G eta_m eta_g k_t K_g

% Given values

\[
R_m = 0.6; \quad \% \text{Motor Resistance}
\]
\[
k_t = 7.68 \times 10^{-3}; \quad \% \text{motor current-torque constant}
\]
\[
K_g = 70; \quad \% \text{High-gear total gear ratio}
\]
\[
eta_m = 0.69; \quad \% \text{Motor efficiency}
\]
\[
eta_g = 0.90; \quad \% \text{Gearbox efficiency}
\]
\[
J_{eq} = 9.76 \times 10^{-5}; \quad \% \text{High-gear equivalent moment w/o load}
\]
\[
B_{eq} = 0.015; \quad \% \text{High-gear equivalent viscous damping coefficient}
\]
\[
m = 0.128; \quad \% \text{mass of short pendulum}
\]
\[
r = .1525; \quad \% \text{radius (when pendulum fixed in shortest position)}
\]
\[
L = .175; \quad \% \text{half length of short pendulum}
\]
\[
g = 9.81; \quad \% \text{gravitational constant}
\]
35  %Combined values
36  a = J_eq + m*r^2;
37  b = m*L*r;
38  c = 4/3*m*L^2;
39  d = m*g*L;
40  E = a*c - b^2;
41  F = a*c + b^2;
42  G = (eta_m*eta_g*k_t*k_g^2*K_g^2 + B_eq*R_m)/R_m;

44  %Evaluation of State Space
47  A = [0 0 1 0; 0 0 0 1; 0 b*d/E -c*G/E 0; 0 -a*d/E b*G/E 0];
48  B = [0; 0; c*eta_m*eta_g*k_t*K_g/(R_m*E); -b*eta_m*eta_g*k_t*K_g/(R_m*E)];
49  C = [0 1 0 0]; % want alpha value
50  D = 0;

52  % Calculated and Plot Step Response
53  sys = ss(A,B,C,D);
54  t=linspace(0,10,500);
55  v_in = 2*((t>1) .* (t<1.1)); % define input with gain of 5
56  y = lsim(sys,v_in,t); % obtain y values step (pulse) response
57  y_converted = -y*2/pi + 2.05; % convert 0 to 2.05
58  % Conversion from radians to voltage
59  % Need to confirm voltage setting (2.05) individually
60  % Negative sign for switched interpretation of alpha/x positioning
61  figure(2)
62  plot(t, y_converted)
63  xlabel('Time(S)');
64  ylabel('Voltage (V)');
65  title('Model Step Response');
66  axis([0, 10, 1.75 2.5]);
67  grid on

27
% Compare to Manual Model Values

% Downwards system test values
A_2 = [0 0 1 0; 0 0 0 1; 0 39.32 -14.52 0; 0 -81.78 13.98 0];
B_2 = [0; 0; 25.54; -24.59];
C_2 = [0 1 0 0];
D_2 = 0;

% Manual's Step Response
sys_2 = ss(A_2,B_2,C_2,D_2);
v_in_manual= 5*((t>1).*(t<1.1));
y_manual = lsim(sys_2,v_in,t);
y_new = (-y_manual)*2/ pi + 2.05; % same conversion
figure(3)
plot(t, y_new)
xlabel('Time(S)');
ylabel('Voltage (V)');
title('Manual Step Response');
axis([0, 10, 1.75 2.5]);
grid on
2.4 K-value Finder for Hanging Pendulum

%Ruisen (Eric) Liu
% K–value Finder

%Define system

R_m = 2.6;
k_t = 7.68 * 10^(-3);
K_g = 70;
eta_m = 0.69;
etag = 0.90;
J_eq = 9.76 * 10^(-5);
B_eq = 0.015;
m = 0.128; % mass of short pendulum
r = .1525; % radius (when pendulum fixed in shortest position)
L = .175; % half length of short pendulum
g = 9.81; % gravitational constant

%Combined values

a = J_eq + m*r^2;
b = m*L*r;
c = 4/3*m*L^2;
d = m*g*L;
E = a*c - b^2;
F = a*c + b^2;
G = (eta_m*etag*k_t^2*K_g^2 + B_eq*R_m)/R_m;

%Evaluation of State Space

A = [0 0 1 0; 0 0 0 1; 0 b*d/E -c*G/E 0 ; 0 -a*d/E b*G/E 0];
B = [0 0 ; c*eta_m*etag*k_t*K_g/(R_m*E) ; -b*eta_m*etag*k_t*K_g/(R_m*E)];
C = [0 1 0 0; 1 0 0 0]; % want alpha value
D = 0;
sys = ss(A,B,C,D);
%Evaluate K values

\( P = 5 \times \begin{bmatrix} -1 & -2 & -1.5 & -2.5 \end{bmatrix} \);

\( K = \text{place}(A, B, P) \)

\( A_f = A - B \times K; \)

\( \text{sys}_f = \text{ss}(A_f, B, C, D); \)

\( \text{initial}(\text{sys}, \text{sys}_f, [0 1 0 0]^\prime) \)
2.5 K-value Finder for Inverted Pendulum

```matlab
%Define system
R_m = 2.6;
k_t = 7.68 * 10^(-3);
K_g = 70;
eta_m = 0.69;
eta_g = 0.90;
J_eq = 9.76 * 10^(-5);
B_eq = 0.015;
m = 0.128; % mass of short pendulum
r = .1525; % radius (when pendulum fixed in shortest position)
L = .175; % half length of short pendulum
g = 9.81; % gravitational constant

%Combined values
a = J_eq + m*r^2;
b = m*L*r;
c = 4/3*m*L^2;
d = m*g*L;
E = a*c - b^2;
F = a*c + b^2;
G = (eta_m*eta_g*k_t*2*K_g^2 + B_eq*R_m)/R_m;

%Evaluation of State Space
A = [0 0 1 0; 0 0 0 1; 0 b*d/E -c*G/E 0 ; 0 a*d/E - b*G/E 0];
B = [ 0; 0 ; c*eta_m*eta_g*k_t*K_g/(R_m*E) ; b*eta_m*eta_g*k_t*K_g/(R_m*E)];
C = [0 1 0 0; 1 0 0 0]; % want alpha value
D = 0;
sys = ss(A,B,C,D);

%Evaluate K values
P = 10*[-1 -1.5 -2 -2.5]; % works with multiplicity
```
of 10

K = place(A,B,P)

%LQR

% Penalize alpha much more than theta
Q = [(1/(2*pi))^2 0 0 0 ; 0 (10/(2*pi))^2 0 0; 0 0 0 0];
R = 0.001;

% Theoretical Predictions:
% Current QR settings allow for V < 7.8, theta < 0.28, alpha < 0.25.
% non-penalized/penalized settle times are 1.7/1.8/2.5

[K_LQR, S, E] = lqr(sys, Q, R);

K_LQR