QEA Final Project: Rocky

Lauren Anfenson & Corey Cochran-Lepiz

November 2018

1 Athlete Demographics

1.1 Survivor

1.1.1 Block Diagram

In the Survivor event, our Rocky has to stand still and upright for as long as possible. To do this we implemented several control loops that controlled for the angle, position, and velocity of Rocky.

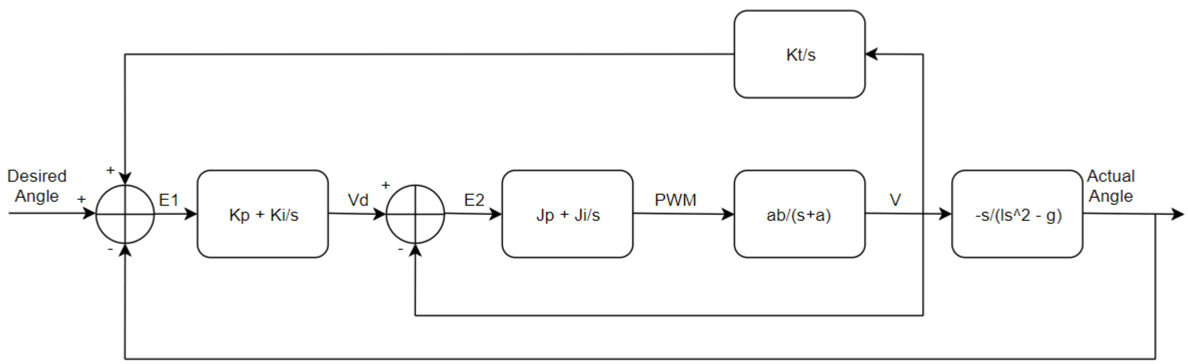

Figure 1: A block diagram of our controls for the Survivor event.

1.1.2 Transfer Functions

The desired angle (in our case 0) is our input to the control loop. The error between the desired and actual angles is then found. The distance between the current and starting positions of the Rocky is then found by integrating the velocity with a transfer function of $\frac{K_t}{s}$. The distance error is then summed with the angle error, effectively modulating the desired angle of the Rocky as it drifts away from its starting position and encouraging it to drift back.

The desired velocity of the Rocky in response to this first error is found with a PI control loop with a transfer function of $K_p + \frac{K_i}{s}$.

The error between the desired velocity and the actual velocity is then found. This error in velocity is related to the PWM signal sent to the motors with another PI control loop with a similar transfer function but different constants: $J_p + \frac{J_i}{s}$.

The motors have a response to the PWM signal that causes the wheels to spin at some velocity. This transfer function was found experimentally to be $\frac{ab}{s=a}$, where $a = 14$ and $b = \frac{1}{400}$.

Finally, the velocity of the Rocky corresponds to some change in its angle. We've modelled Rocky as an inverted pendulum with all of its mass concentrated at the center of mass in order to describe this.

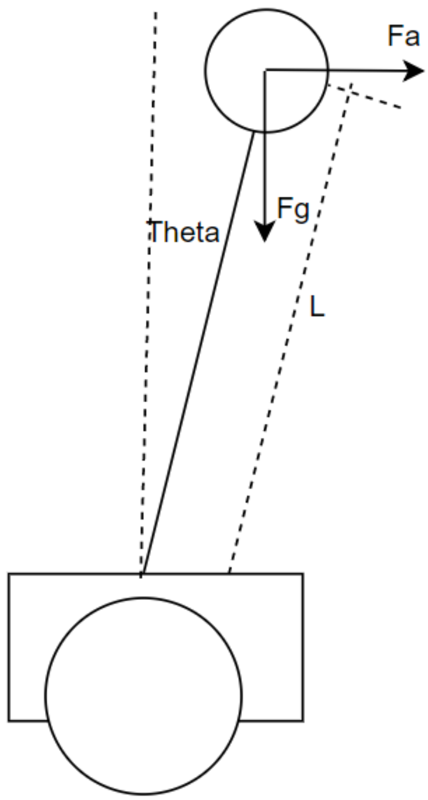

Figure 2: A diagram showing our definitions of theta and the length of the pendulum along with the forces acting on the Rocky.

To find how velocity influences the angle of the Rocky, we can look at the forces acting on an inverted pendulum due to gravity and acceleration when the pendulum is a bit off of its equilibrium. Converting these forces acting at a distance L from the pivot point into torques and using the rotational equivalent of $F = ma$, we get

$$I\alpha = \tau_g - \tau_a \tag{1}$$

where τ_g is the torque due to gravity and τ_a is the torque due to acceleration. This can then be expanded as

$$mL^2\frac{d^2\theta}{dt} - mLgSin(\theta) = -mL\frac{dv}{dt}Cos(\theta) \tag{2}$$

Using the small angle approximation to assume that $Sin(\theta) = \theta$ and $Cos(\theta) = 0$ and then taking the Laplace transform of our ODE we get

$$s^2L\Theta(s) - g\Theta(s) = -sV(s) \tag{3}$$

Rearranging this into a transfer function from V to Θ shows us

$$\frac{\Theta(s)}{V(s)} = \frac{-s}{s^2L - g} \tag{4}$$

which we can then use as the final transfer function in our block diagram.

1.1.3 Poles

Our poles were determined by a Mathematica notebook that created a transfer function for our system from our block diagram. We adjusted our constants using a Manipulate plot in Mathematica, guessing and checking until our poles were as negative as we could get them to ensure we had a stable system.

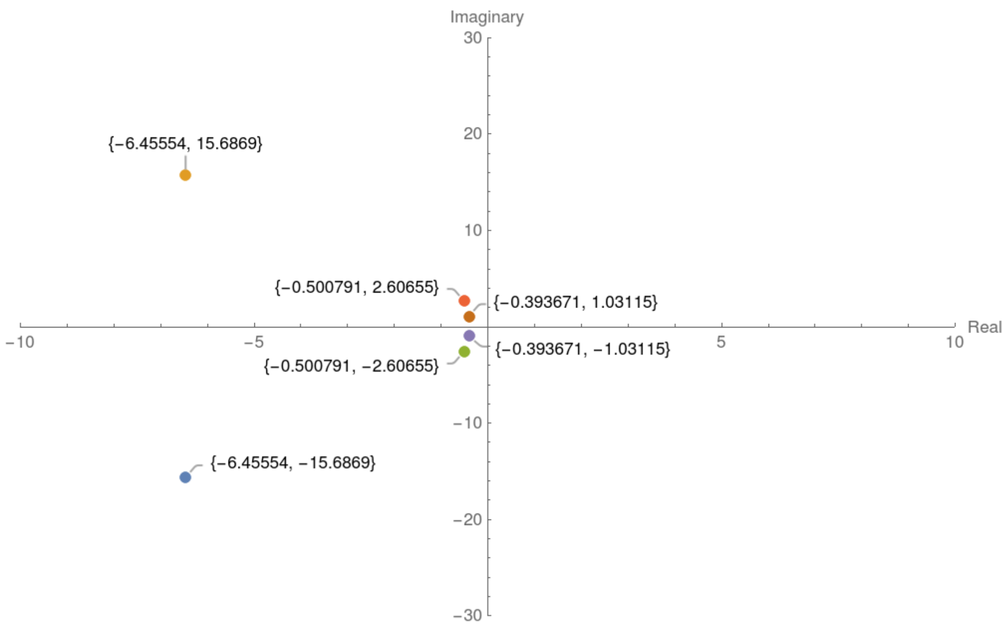

Figure 3: The poles of our transfer function with our chosen constants.

2 Athlete Performance Information

The parameters of our athlete were determined by finding a reasonable range for the constants in our transfer functions and then guessing and checking in Mathematica. We settled on the following values given the current system.

K_p	-60
K_i	-120
J_p	20
J_p	60
K_t	-0.1

3 Mathematica Notebook

In order to run all of the calculations we relied on Mathematica to do the tedious algebra to eliminate a source of error in the project. We defined the transfer functions for each block in the system then solved for the overall transfer function. With that we plugged in our known values (a, b, L, and g) and found the poles, which are where the denominator equals zero. We used Mathematica's Manipulate function to visualize and guess-and-check for different values of K_p, K_i, J_p, J_i and K_t.

QEA Night 10

Exercise

In[1]:=

```
eq1 = θ[s] == e_rr2[s] G_P[s] G_MC[s] G_VC[s];
eq2 = V_d[s] == e_rr1[s] G_PI[s];
eq3 = e_rr1[s] == θ_d[s] - θ[s] + G_DC[s] V[s];
eq4 = V[s] == e_rr2[s] G_P[s] G_MC[s];
eq5 = e_rr2[s] == V_d[s] - V[s];
sol = Solve[{eq1, eq2, eq3, eq4, eq5}, {θ[s], V_d[s], e_rr1[s], V[s], e_rr2[s]}][[1]];
{G_TOTALSYSTEM[s] → θ[s]/θ_d[s] /. sol} (* this is a rule to replace G_TOTALSYSTEM,
you can just extract the value by using the righthand side of the rule *)
trans = θ[s]/θ_d[s] /. sol /. {G_PI[s] → K_p + (K_i / s), G_VC[s] → -s / (L s^2 - g),
   G_MC[s] → (a b) / (s + a), G_P[s] → J_p + (J_i / s), G_DC[s] → K_t / s}
tsumsub = Factor[trans /. {b → 1 / 400, a → 14 , L → .1, g → 9.8}]
```

Out[7]= $\left\{G_{TOTALSYSTEM}[s] \to \frac{G_{MC}[s]\, G_P[s]\, G_{PI}[s]\, G_{VC}[s]}{1 + G_{MC}[s]\, G_P[s] - G_{DC}[s]\, G_{MC}[s]\, G_P[s]\, G_{PI}[s] + G_{MC}[s]\, G_P[s]\, G_{PI}[s]\, G_{VC}[s]}\right\}$

Out[8]= $-\frac{a\, b\, s \left(\frac{J_i}{s} + J_p\right)\left(\frac{K_i}{s} + K_p\right)}{(a+s)\left(-g + L s^2\right)\left(1 + \frac{a b \left(\frac{J_i}{s}+J_p\right)}{a+s} - \frac{a b s \left(\frac{J_i}{s}+J_p\right)\left(\frac{K_i}{s}+K_p\right)}{(a+s)\left(-g+L s^2\right)} - \frac{a b \left(\frac{J_i}{s}+J_p\right)\left(\frac{K_i}{s}+K_p\right) K_t}{s\,(a+s)}\right)}$

Out[9]= $-\left(\left(0.35\, s^2 \left(1.\, J_i K_i + 1.\, s J_p K_i + 1.\, s J_i K_p + 1.\, s^2 J_p K_p\right)\right) / \left(-1372.\, s^3 - 98.\, s^4 + 14.\, s^5 + 1.\, s^6 - 3.43\, s^2 J_i + 0.035\, s^4 J_i - 3.43\, s^3 J_p + 0.035\, s^5 J_p - 0.35\, s^2 J_i K_i - 0.35\, s^3 J_p K_i - 0.35\, s^3 J_i K_p - 0.35\, s^4 J_p K_p + 3.43\, J_i K_i K_t - 0.035\, s^2 J_i K_i K_t + 3.43\, s J_p K_i K_t - 0.035\, s^3 J_p K_i K_t + 3.43\, s J_i K_p K_t - 0.035\, s^3 J_i K_p K_t + 3.43\, s^2 J_p K_p K_t - 0.035\, s^4 J_p K_p K_t\right)\right)$

Out[•]= $-\left(\left(0.249\, s^2 \left(1.\, J_i K_i + 1.\, s J_p K_i + 1.\, s J_i K_p + 1.\, s^2 J_p K_p\right)\right) / \left(-813.4\, s^3 - 98.\, s^4 + 8.3\, s^5 + 1.\, s^6 - 2.4402\, s^2 J_i + 0.0249\, s^4 J_i - 2.4402\, s^3 J_p + 0.0249\, s^5 J_p - 0.249\, s^2 J_i K_i - 0.249\, s^3 J_p K_i - 0.249\, s^3 J_i K_p - 0.249\, s^4 J_p K_p + 2.4402\, J_i K_i K_t - 0.0249\, s^2 J_i K_i K_t + 2.4402\, s J_p K_i K_t - 0.0249\, s^3 J_p K_i K_t + 2.4402\, s J_i K_p K_t - 0.0249\, s^3 J_i K_p K_t + 2.4402\, s^2 J_p K_p K_t - 0.0249\, s^4 J_p K_p K_t\right)\right)$

In[∘]:=
```
poles = ReIm[Values[Solve[Denominator[tsumsub] == 0, s]]];
ListPlot[poles /. {Kp → -88, Ki → -100, Jp → 10, Ji → 56, Kt → -0.1}];
```

... Solve: Solve was unable to solve the system with inexact coefficients. The answer was obtained by solving a corresponding exact system and numericizing the result.

In[10]:=
```
(*Sweep values of kp and ki*)
f[Kp_, Ki_, Jp_, Ji_, Kt_] = ReIm[N[Values[Solve[
      Denominator[tsumsub /. {Kp → Kp, Ki → Ki, Jp → Jp, Ji → Ji, Kt → Kt}] == 0, s]]]];
(*returns list as s→[[values]]*)
Manipulate[
 ListPlot[f[Kp, Ki, Jp, Ji, Kt] /. {Kp → kp, Ki → ki, Jp → jp, Ji → ji, Kt → kt},
  AxesLabel → {"Real", "Imaginary"}, PlotStyle → PointSize[Large],
  PlotRange → {{-10, 10}, {-30, 30}}], {kp, -100, 100},
 {ki, -200, 200}, {jp, -50, 50}, {ji, -60, 60}, {kt, -1, 1}]
```

... Solve: Solve was unable to solve the system with inexact coefficients. The answer was obtained by solving a corresponding exact system and numericizing the result.

Out[11]=