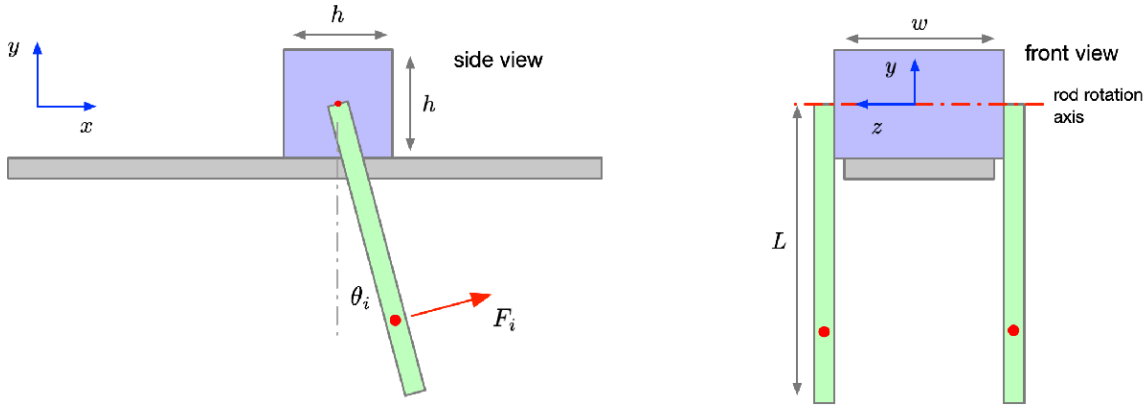
Project 3

Ross Smyth

Table of Contents

This report goes over the development, implementation, and verification of models for Project 3 in the class MEEM 4730. This project is a model of a block and rod system. This system is modeled both "directly" with differential equations derived from Lagrange's equation, and "indirectly" with Simulink's Simscape enviroment. Assumtions in this model include: all joints are frictionless, air resistance is negligible, no mass is consumed by the thrusters, the thrusters are firing from the center of mass of the rods, and that all objects are rigid bodies. For this project the full non-linear dynamics are explored, and two input profiles are implmented. A bang-coat-bang controller, and a swing-free profile .The bang-cost-bang is essentially two pulse with a dead time in between. The swing-free manuver attmepts to use the natural dynamics of the rods to reduce the swinging of the rods and the drift of the brick after the manuver is complete.



Model diagram

Differential Equation Model

Intro

This sytem is a non-linear system, and to simulate the dynamics a differential equation model was developed that defines the equations of motion. A each time step the current state of the system is input to this model, from which the derivatives of each of the parameters can be found, which is then integrated to get the next set of states. The full differential equation derivation is shown step-by-step in Appendix A.

This system is defined with a set of coupled, ordinary differential equations. These equations were derived via a reformulation of classical, Newtonian, mechanics called analytical mechanics, specifically Lagrangian mechanics in this case. This reformulation allows one to derive the equations of motion for a complicated system as long as some generlized coordintes of the system can be defined, and the kinetic and potential energy can be defined in terms of these coordinates. Once the energies are defined, the potential energy is subtracted from the kinetic energy to form a scalar expression of this energy difference called the Lagrangian. The Lagrangian can be put through Lagrange's equation for each generlized coordnate, and the equation of motion for that coordniate will be output. Below is Lagrange's equation, where L represents the Lagrangian, and Q represents generlized power, which expresses external forces on the system.

$$\frac{\partial}{\partial t}\left(\frac{\partial}{\partial \dot{q}_i} L\right) - \frac{\partial}{\partial q_i} L = Q_i$$

For the system at-hand, the generlized coordinates used are the following:

$$\begin{aligned} x_b &= \text{Brick position} \\ \theta_1 &= \text{Rod 1 angle} \\ \theta_2 &= \text{Rod 2 angle} \end{aligned}$$

Kinematics and energy

The next step towards obtaining the equations of motion is defining the kinematics and the energies. For kinematics, this essentially means representing any relationships between items in the system. Even though the position is not one of the generalized coordinates, it is still needed to calculate the linear kinetic energy of the rods. For this system it is clear that the absolute position of the rods is dependent on the position of the block, so that relationship is shown below. The brick's position vector is also shown for clarity.

$$\vec{x_b} = \begin{bmatrix} x_b \\ 0 \\ 0 \end{bmatrix}$$

$$\vec{x}_{\text{rod}_j} = \vec{x_b} + \frac{L}{2} \begin{bmatrix} \sin(\theta_j) \\ -\cos(\theta_j) \\ \pm\frac{w}{2} \end{bmatrix}$$

Where L is the length of the rods, and w is the width of the brick. From this set (one for each rod) of definitions the kinematics are done. Next the energy of the system is calculated. For the rods, the energy definition is the same for each of them with both linear kinetic energy, rotational kinetic energy, and graviational potential energy. For the brick, if the zero of the gravitational potential is placed at its center of mass there is only liner kinetic energy. Below shows the kinetic energy expression, T, for the rods and brick.

$$T_{\text{rods}_j} = \frac{1}{2}\left[M_r \left\langle \dot{\vec{x}}_{\text{rod}_j}, \dot{\vec{x}}_{\text{rod}_j} \right\rangle + \frac{M_r(3r^2 + L^2)}{12}\dot{\theta}_j^{\,2}\right]$$

$$T_{\text{brick}} = \frac{1}{2}\left[M_b \left\langle \dot{\vec{x}}_b, \dot{\vec{x}}_b \right\rangle\right]$$

And the potential energies:

$$V_{\text{rod}_j} = -M_r g \frac{L}{2}\cos(\theta_j)$$

Where M_r represents the mass of the rods, M_b represents the mass of the brick, r represents the radius of the rods, and $\left\langle \vec{a}, \vec{b} \right\rangle$ represents the vector dot product. From these the Lagranian expression can be calculated by the sum of all three kinetic energies subtracted by the sum of both of the potential energies.

$$L = \sum T_i - \sum V_i$$

Generalized Forces

For a given system, if there are any non-constraint external forces they must be represented in terms of generalized power. This generalized power is the dot product of the generlized force and the time derivative of generlized coordinates effected. For this case the generalized power is represented with an actual mechanical power equation, as it is a linear force applied to the rods, and the absolute position was calculated above.

$$F_j = \begin{bmatrix} \cos(\theta_j) \\ \sin(\theta_j) \\ 0 \end{bmatrix}$$

$$Q = \left\langle F_1, \dot{\vec{x}}_{\text{rod}_1} \right\rangle + \left\langle F_2, \dot{\vec{x}}_{\text{rod}_2} \right\rangle$$

This results in the generalized power expression, which can then be converted to a matrix equation in terms of the time derivative of the generalized coordnate as the equations are linear in terms of them.

$$Q = \begin{bmatrix} F_1\cos(\theta_1) + F_2\cos(\theta_2) & \frac{F_1 L}{2} & \frac{F_2 L}{2} \end{bmatrix} \begin{bmatrix} \dot{x_b} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

These equations make sense, as the force on the block only matter in the x direction, and the generalized force upon the rods are torques half-way along their lengths.

Equations of Motion

With these defined, and the previous energy definitions, the equation of motion can be calculated. For each of the generalized coordinates, $x_b, \theta_1, \text{and } \theta_2$ Lagrange's equation can be applied to the Lagrangian. This results in a set of time-varying equations that are linear in terms of the second time derivative of the generalized coordinates. Because of this, a matrix equation in the form of $Ax = B$ can be formed. In this case the A would be a 3x3 mass matrix, representing the intertia terms, x would be the second time derivatives of the generalized

coordinates, and B would be the right-hand side of coupled equations that define the motion. These are shown below.

$$\begin{bmatrix} M_b + 2M_r & \frac{L\,M_r}{2}\cos(\theta_1) & \frac{L\,M_r}{2}\cos(\theta_2) \\ \frac{L\,M_r}{2}\cos(\theta_1) & \frac{M_r}{12}\,(4\,L^2 + 3\,r^2) & 0 \\ \frac{L\,M_r}{2}\cos(\theta_2) & 0 & \frac{M_r}{12}\,(4\,L^2 + 3\,r^2) \end{bmatrix} \begin{bmatrix} \ddot{x}_b \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \frac{L\,M_r\,\dot{\theta}_1^2}{2}\sin(\theta_1) + \frac{L\,M_r\,\dot{\theta}_2^2}{2}\sin(\theta_2) + F_1\cos(\theta_1) + F_2\cos(\theta_2) \\ \frac{L}{2}\,(F_1 - M_r\,g\sin(\theta_1)) \\ \frac{L}{2}\,(F_2 - M_r\,g\sin(\theta_2)) \end{bmatrix}$$

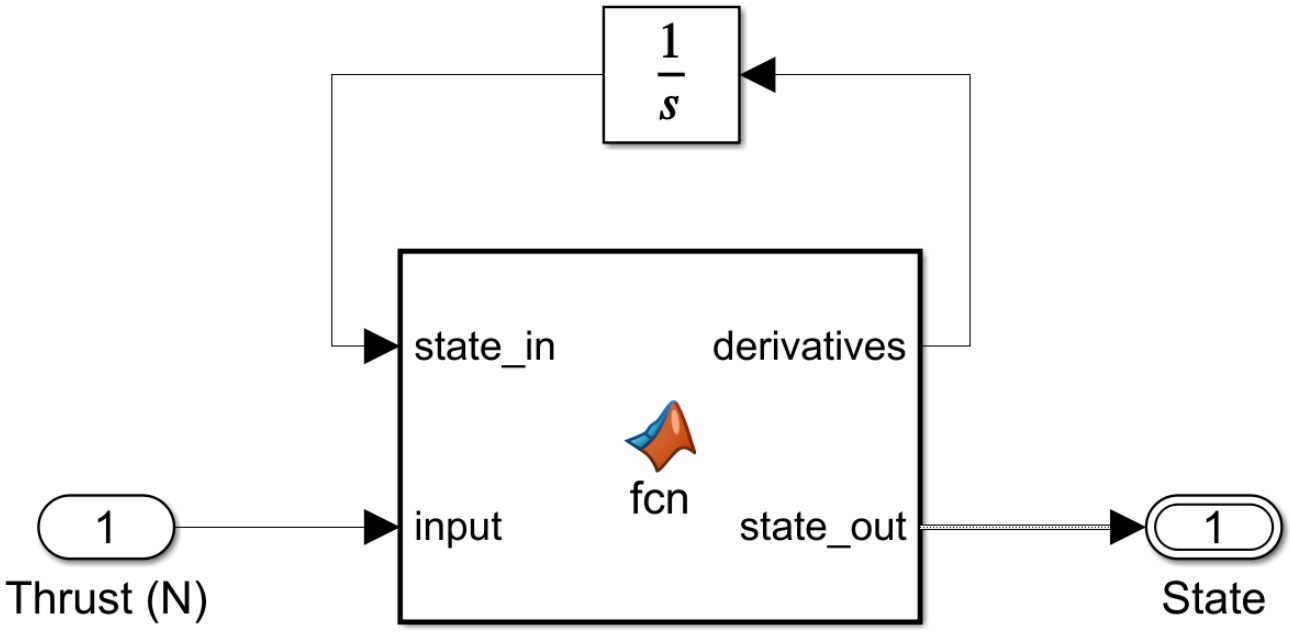
With this equation, the dynamics of the system can be simulated with the following process:

1. Input system states into the mass matrix and right-hand side vector to get a system of linear equations.
2. Solve the system of linear equations $\ddot{\vec{q}} = M^{-1}R$, then concentate with the previous time-step's generalized velocities.
3. Integrate the generlized accelerations and velocities to get the new time step's generalized coordinates and their time derivatives.
4. Go to step 1

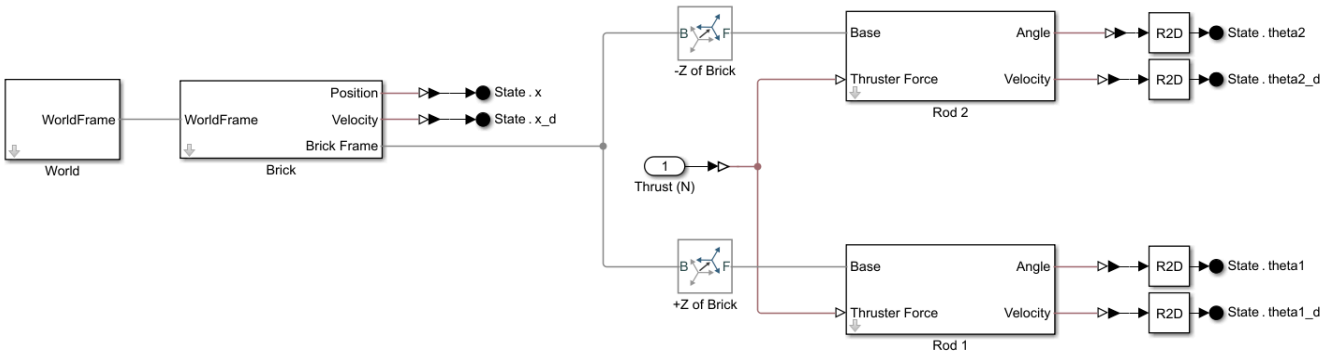
Implementation and Outputs

This ssytem was implemented with two models. Directly with the above differential equation, and also indirectly with Simscape, Simulink's acasual simulation enviroment. Rather than directly building a differential equation model, this enviroment allows for an engineer to put together discrete components such as masses and joints, which allows for greater physical intuition on the system. Both of these models shared the same inputs, either a bang-coast-bang controller, or a swing-free manuver.

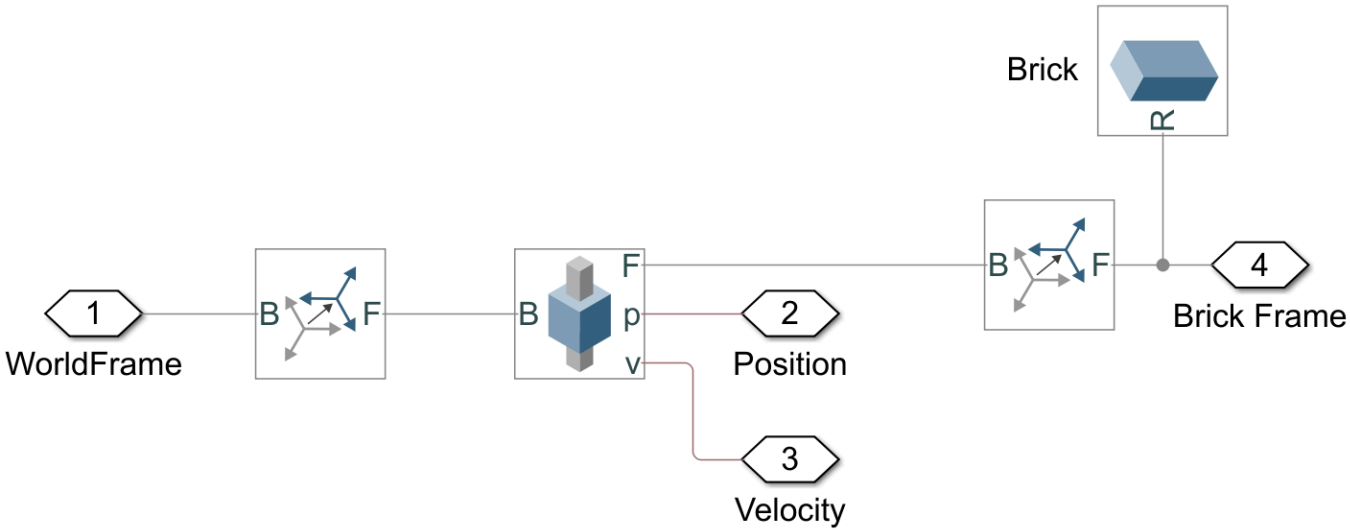
Simulink



Simscape

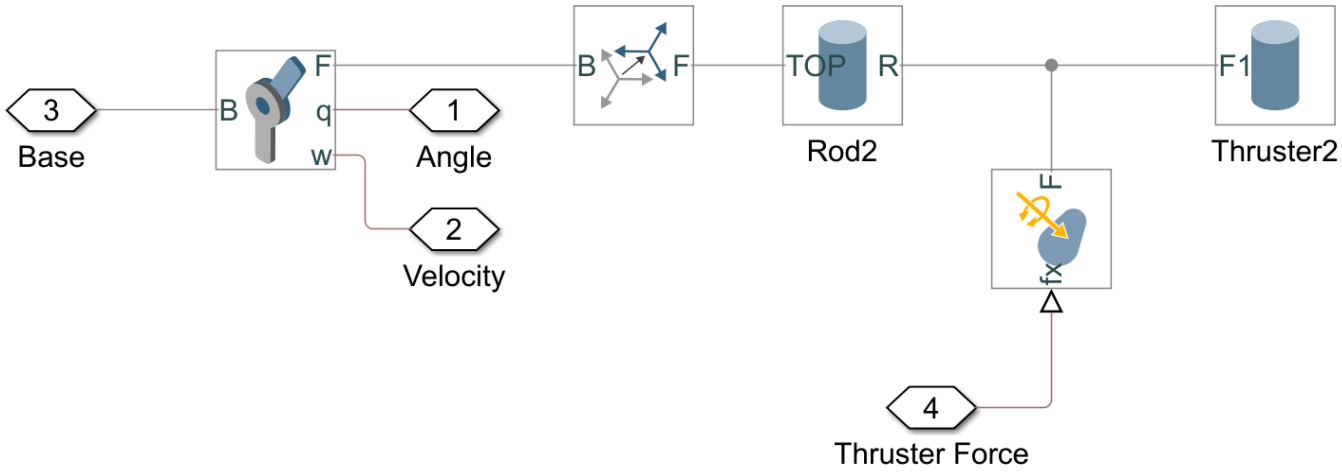


Overall Simscape



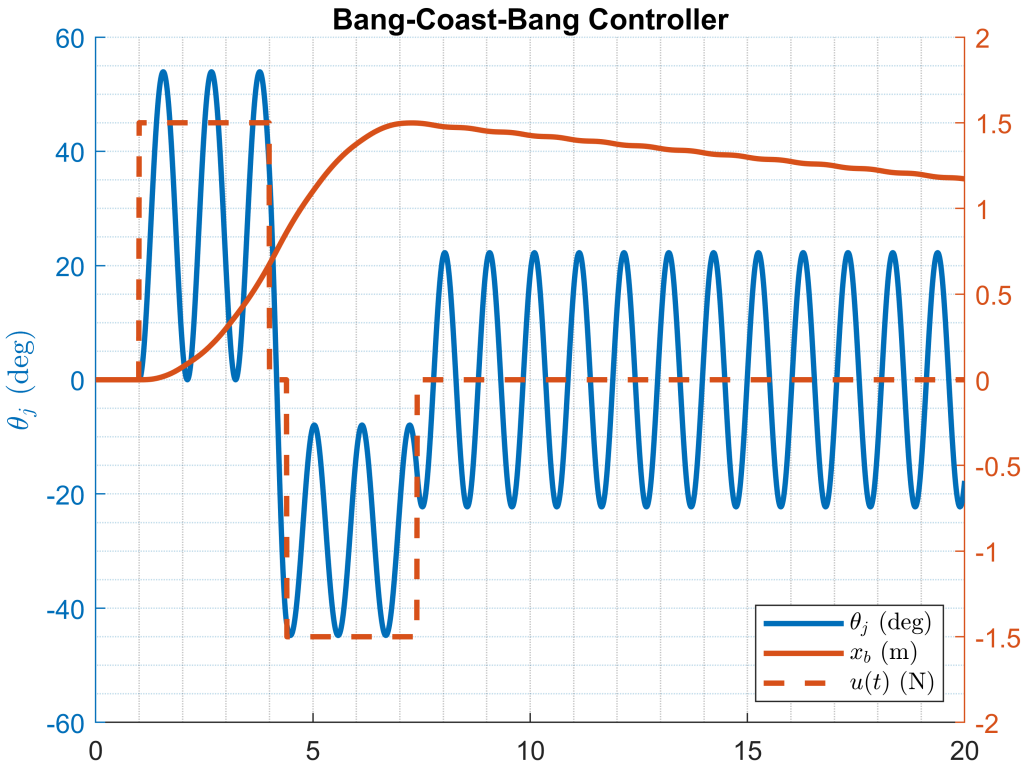
Model

Brick Subsystem

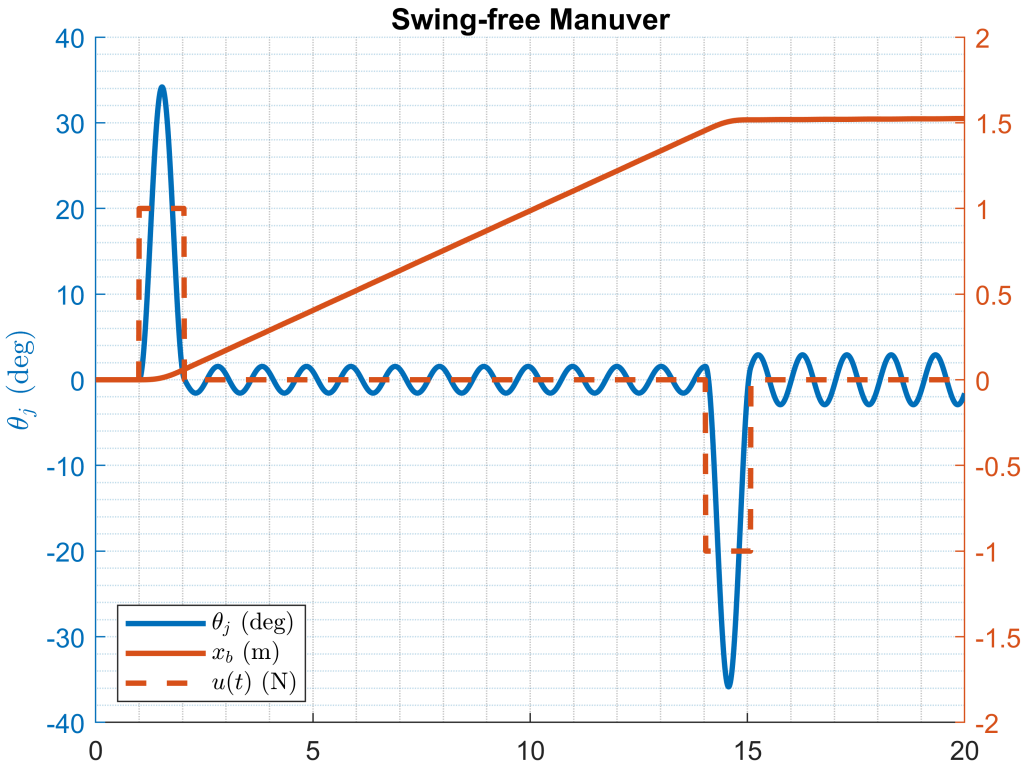


Outputs

Below shows the output of the differential equation model (as the Simscape is nearly identical, see the verification section below), for both the bang-coast-bang controller and the swing-free manuever.



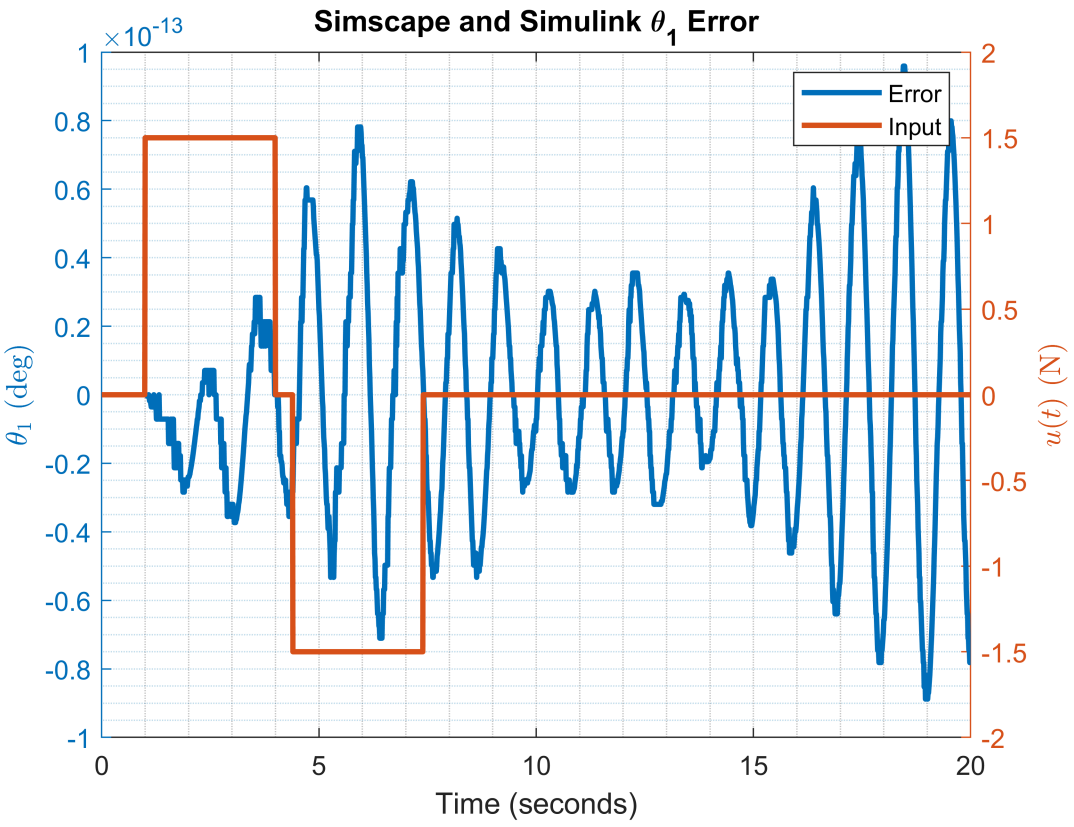
This shows that after end of the manuever, the system drifts away from the end location. This makes sense as the thruster input may not have cancelled out all of the linear momentum of the system if the rod angles' were not at exactly zero degees.



With the swing-free manuver it seems that because the force input time was approximatly the same as the period of the rods' swinging behavior, the momentum of the system was cancelled out extremly well. There is minimal drift from the endpoint.

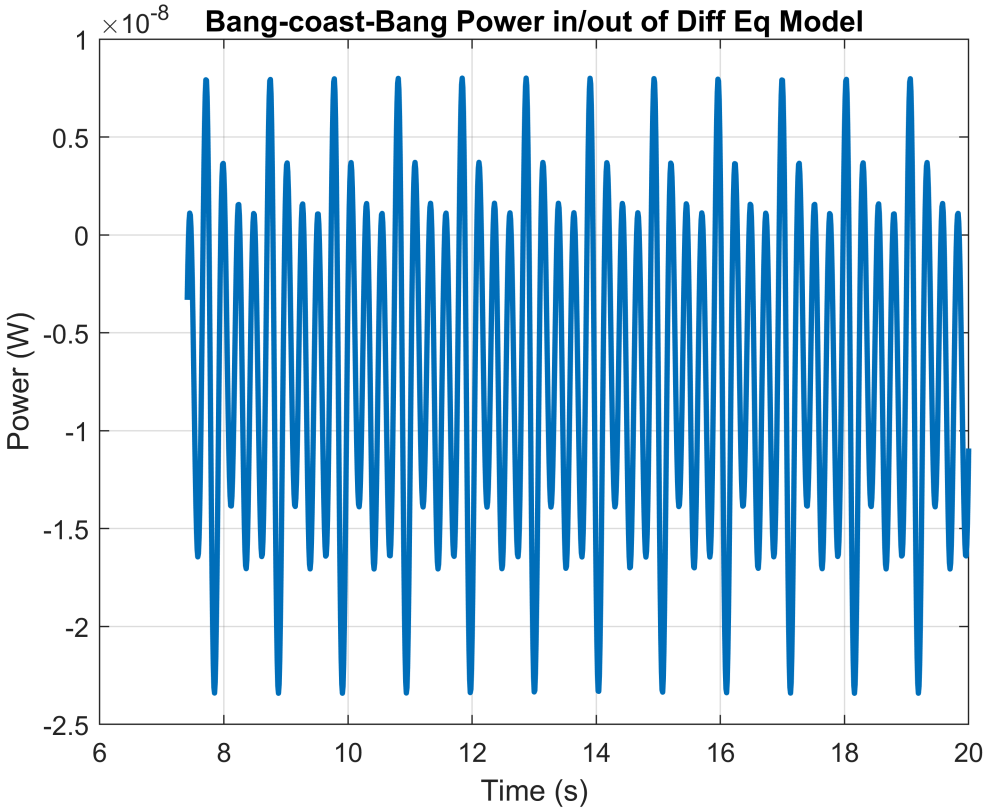
Verification

With the models implemented, verification is the next step. Because there are two models, and there isn't an analytical solution to this system verification must be done between models, and verifying that conservation laws are followed. Below show the error between the two models.



The above plot shows that the error between the two models is on the order of 10^{-13} degrees, which is extremly small (the most accurate spacecraft pointing system in the world, the Hubble Space Telescope's, is only accurate to 10^{-6} degrees) and so it is safe to say that the models are essentially identical. The error increases when the dynamics are faster, and also as time goes on it can be seen that the error increases. If ran for a long time a different integration scheme or a smaller timestep may need to be used.

Another way for verification is checking the conservation laws of the system. In this case energy will be checked, but momentum should be checked as well to ensure full conservation. Only the differential equation models' energy will be shown as above it is shown that the models are nearly identical, so if it conserves energy well, then the Simscape model will as well. Specifically the power will be looked at in detail, during periods were there is no input, the change in energy with time should be zero. The bang-coast-bang controller will be analyzed as the dynamics of that manuever are faster than the dynamics of the swing-free manuver.



The above plot shows the power in/out of the system of the differential equation model after the bang-coast-bang manuever is performed. Theoretically the power should be zero, but since this is a numerical model this will not be the case. The energy in out of the system at each time step is minimal, and it looks to keep stay near zero, and not diverge. This verifies that energy is conserved well over short time-scales, though if simluated for extremly long periods (decades) or if the dynamics of the system remain high for long periods, the leakage of energy may be a concern.

Conclusion

In conclusion, this system model is at least verified between two independent models, one derived using Lagrange's equation, and another put together using Simscape. It is also verfied to conserve energy relatively well. For next steps, building the system in real life and validating the model against real data would be a good next step, or making less assumptions about the system. The assumptions mentioned in the introduction can be reduced with some of the following techniques.

For implementing friction and/or air resistance, Rayleigh's dissipation function would be needed to be used for the velocity dependence. Columb friction would have to use a modified version of this as it does not have a linear dependece with velocity like viscous friction does. For air resistance if the linear resistance model is used then Rayleigh's dissipation function can be used, but if the quadratic model is used it would need to be modified.

Implementing varying mass for the thrusters would not be extremly hard, the main difference would that the mass in the energy expression would have a time dependence. Out of all the assumption, if the rigid body assumption did not hold, which would occur if the fundemental frequency of the rod was low relative to the

dynamics of the system, it would be the most difficult to correct. This would require implementing a (hopefully) simplified partial differential equation model of the rods.

With this model, an engineer can experiment with different thrust profiles and observe how the system changes. For the bang-coast-bang profile an engineer can observe how certain thrust profiles behave in steady-state after the manuver is done, or how the transient dynamics are, for example the maximum angle of the rod. With the swing-free profile, with small thrusts the rods can be stablised at the end of the manuver easily, allowing for stable movements of the brick and rod. Engineers can explore the maximum thrusts that can be used, as once the rods reach a high enough angle the natrual frequency can no longer be assumed to be constant. Since this fully simulates the non-linear dynamics the full range can be explored. Then engineers can implement the profile on a real system and be assured that the model is closer than a linaer model would.

Appendix A

```
clear, clc, close all
```

Parameters

```
t  = sym('t', 'real');  % s,      time
g  = sym('g', 'real');  % m/s/s,  gravity

Mb = sym('Mb', 'real'); % kg,     mass of block
w  = sym('w',  'real'); % m,      width of block

L  = sym('L',  'real'); % m, length of rods
r  = sym('r',  'real'); % m, radius of rods
Mr = sym('Mr', 'real'); % kg, mass of rods
```

Variables

```
% Inputs
F1 = sym('F1', 'real'); % N, rod one thruster
F2 = sym('F2', 'real'); % N, rod two thruster

% Outputs
a   = sym('a',   'real');   % m/s/s, block acceleration
al1 = sym('al1', 'real'); % rad/s/s, rod 1 angular acceleration
al2 = sym('al2', 'real'); % rad/s/s, rod 2 angular acceleration

% Purely symbolic States
xs  = sym('xs',  'real'); % meters,  block displacement
t1s = sym('t1s', 'real'); % radians, rod 1 angle
t2s = sym('t2s', 'real'); % radians, rod 2 angle

x_ds  = sym('xDs',  'real'); % meters/s,  block velocity
t1_ds = sym('t1Ds', 'real'); % radians/s, rod 1 anglular velocity
t2_ds = sym('t2Ds', 'real'); % radians/s, rod 2 anglular velocity

x_dds  = sym('xDDs',  'real'); % meters/s,  block accel
t1_dds = sym('t1DDs', 'real'); % radians/s, rod 1 anglular accel
t2_dds = sym('t2DDs', 'real'); % radians/s, rod 2 anglular accel

% Bound Derivative states
syms  x(t); % meters,  block displacement
syms t1(t); % Radians, Rod 1 angle
syms t2(t); % Radians, Rod 1 angle

x_d  = diff(x,  t); % meters/s,  block velocity
t1_d = diff(t1, t); % radians/s, rod 1 anglular velocity
t2_d = diff(t2, t); % radians/s, rod 2 anglular velocity

x_dd  = diff(x_d,  t); % meters/s,  block accel
t1_dd = diff(t1_d, t); % radians/s, rod 1 anglular accel
t2_dd = diff(t2_d, t); % radians/s, rod 2 anglular accel

% Variable swapping arrays
time_var = [x,  t1,  t2,  x_d,  t1_d,  t2_d,  x_dd,  t1_dd,  t2_dd];
```

```
symb_var = [xs, t1s, t2s, x_ds, t1_ds, t2_ds, x_dds, t1_dds, t2_dds];
```

Kinematics

Put the rods' COM velocity in terms of their angular velocity and the block's velocity.

Assumes that the COM is in the midpoint of the rod.

$$\vec{x}_b = \begin{bmatrix} x_b \\ 0 \\ 0 \end{bmatrix}$$

$$\vec{x}_{\text{rod}_j} = \vec{x}_b + \frac{L}{2} \begin{bmatrix} \sin(\theta_j) \\ -\cos(\theta_j) \\ \pm\frac{w}{2} \end{bmatrix}$$

```
% Block COM Vector
r_block = [x; 0; 0];

% Rod COM vectors
r_rod1 = r_block + L/2 * [sin(t1); -cos(t1);  w/2]; % m, vector to rod 1 COM
r_rod2 = r_block + L/2 * [sin(t2); -cos(t2); -w/2]; % m, vector to rod 2 COM

% Vector velocities
vel_block = diff(r_block, t); % m/s, block velocity

angV_rod1 = [0; 0; t1_d]; % rad/s, rod 1 velocity
angV_rod2 = [0; 0; t2_d]; % rad/s, rod 2 velocity

% Rod velocity
vrod1 = diff(r_rod1, t);
vrod2 = diff(r_rod2, t);
```

Kinetic Energy

$$T_{\text{rods}_j} = \frac{1}{2}\left[M_r \left\langle \dot{\vec{x}}_{\text{rod}_j}, \dot{\vec{x}}_{\text{rod}_j} \right\rangle + \frac{M_r(3r^2 + L^2)}{12}\dot{\theta}_j^{\,2}\right]$$

$$T_{\text{brick}} = \frac{1}{2}\left[M_b \left\langle \dot{\vec{x}}_b, \dot{\vec{x}}_b \right\rangle\right]$$

```
% Rod 1
T_rod1  = 1/2 * (Mr * (vrod1' * vrod1) + (angV_rod1' * angV_rod1) * Mr*(3*r^2+L^2)/12);
T_rod1s = subs(T_rod1, time_var, symb_var);
T_rod1s = simplify(T_rod1s)
```

```
T_rod1s(t) =
```

$$\frac{\text{Mr}\ (4\,L^2\,\text{t1Ds}^2 + 12\cos(\text{t1s})\,L\,\text{t1Ds}\,\text{xDs} + 3\,r^2\,\text{t1Ds}^2 + 12\,\text{xDs}^2)}{24}$$

```
% Rod 2
```

```
T_rod2  = 1/2 * (Mr * (vrod2' * vrod2) + (angV_rod2' * angV_rod2) * Mr*(3*r^2+L^2)/12);
T_rod2s = subs(T_rod2, time_var, symb_var);
T_rod2s = simplify(T_rod2s)
```

T_rod2s(t) =

$$\frac{\mathrm{Mr}\ \left(4\,L^2\,\mathrm{t2Ds}^2+12\cos(\mathrm{t2s})\,L\,\mathrm{t2Ds}\,\mathrm{xDs}+3\,r^2\,\mathrm{t2Ds}^2+12\,\mathrm{xDs}^2\right)}{24}$$

```
% Block
T_block  = 1/2 * Mb * (vel_block' * vel_block);
T_blocks = subs(T_block, time_var, symb_var);
T_blocks = simplify(T_blocks)
```

T_blocks(t) =

$$\frac{\mathrm{Mb}\ \mathrm{xDs}^2}{2}$$

```
% Total Kinetic Energy
T_total = simplify(expand(T_rod1s + T_blocks + T_rod2s))
```

T_total(t) =

$$\frac{\mathrm{Mb}\ \mathrm{xDs}^2}{2}+\mathrm{Mr}\ \mathrm{xDs}^2+\frac{L^2\,\mathrm{Mr}\ \mathrm{t1Ds}^2}{6}+\frac{L^2\,\mathrm{Mr}\ \mathrm{t2Ds}^2}{6}+\frac{\mathrm{Mr}\ r^2\,\mathrm{t1Ds}^2}{8}+\frac{\mathrm{Mr}\ r^2\,\mathrm{t2Ds}^2}{8}+\frac{L\ \mathrm{Mr}\ \mathrm{t1Ds}\ \mathrm{xDs}\cos(\mathrm{t1s}}{2}$$

```
matlabFunction(T_total,'File','scripts/KineticEnergy');
```

Potential Energy

The block doesn't ever have any potential energy.

$$V_{\mathrm{rod}_j}=-M_r g\frac{L}{2}\cos(\theta_j)$$

```
% Rod 1 GPE
V_rod1  = Mr * g * L/2 * -cos(t1);
V_rod1s = subs(V_rod1, time_var, symb_var);
V_rod1s = simplify(V_rod1s);

% Rod 2 GPE
V_rod2  = Mr * g * L/2 * -cos(t2);
V_rod2s = subs(V_rod2, time_var, symb_var);
V_rod2s = simplify(V_rod2s);

% Total Potential Energy
V_total = simplify(expand(V_rod2s + V_rod1s))
```

V_total(t) =

$$-\frac{L\ \mathrm{Mr}\ g\ \left(\cos(\mathrm{t1s})+\cos(\mathrm{t2s})\right)}{2}$$

Langrange's Equation

```
lagrangian = simplify(expand(T_total - V_total))
```

lagrangian(t) =

$$\frac{\mathrm{Mb}\ \mathrm{xDs}^2}{2} + \mathrm{Mr}\ \mathrm{xDs}^2 + \frac{L^2\ \mathrm{Mr}\ \mathrm{t1Ds}^2}{6} + \frac{L^2\ \mathrm{Mr}\ \mathrm{t2Ds}^2}{6} + \frac{\mathrm{Mr}\ r^2\ \mathrm{t1Ds}^2}{8} + \frac{\mathrm{Mr}\ r^2\ \mathrm{t2Ds}^2}{8} + \frac{L\ \mathrm{Mr}\ g\ \cos(\mathrm{t1s})}{2} + \frac{L\ \mathrm{M}}{}$$

Left Side

$$\frac{\partial}{\partial t}\left(\frac{\partial}{\partial \dot{q}_i} L\right) - \frac{\partial}{\partial q_i} L = Q_i$$

```
% Block Lagrange's equation

% Velocity derivative
dL_dx_ds = diff(lagrangian, x_ds);
dL_dx_d  = subs(dL_dx_ds, symb_var, time_var); % Sub out for time expressions

% Time derivative
d_dtx = diff(dL_dx_d, t);
d_dtx = subs(d_dtx, time_var, symb_var); % Symbolic version

% Position Derivative
dL_dxs = diff(lagrangian, xs);

% Lagrange's Equation
x_left = d_dtx - dL_dxs
```

x_left(t) =

$$-\frac{L\ \mathrm{Mr}\ \sin(\mathrm{t1s})\ \mathrm{t1Ds}^2}{2} - \frac{L\ \mathrm{Mr}\ \sin(\mathrm{t2s})\ \mathrm{t2Ds}^2}{2} + \mathrm{Mb}\ \mathrm{xDDs} + 2\ \mathrm{Mr}\ \mathrm{xDDs} + \frac{L\ \mathrm{Mr}\ \mathrm{t1DDs}\ \cos(\mathrm{t1s})}{2} + \frac{L\ \mathrm{Mr}\ \mathrm{t2}}{}$$

```
% Rod 1 Lagrange's equation

% Velocity derivative
dL_dt1_ds = diff(lagrangian, t1_ds);
dL_dt1_d  = subs(dL_dt1_ds, symb_var, time_var); % Sub out for time expressions

% Time derivative
d_dtt1  = diff(dL_dt1_d, t);
d_dtt1s = subs(d_dtt1, time_var, symb_var); % Symbolic version

% Position Derivative
dL_dt1s = diff(lagrangian, t1s);

% Lagrange's Equation
t1_left = d_dtt1s - dL_dt1s
```

t1_left(t) =

$$\frac{L^2\,\mathrm{Mr}\,\mathrm{t1DDs}}{3}+\frac{\mathrm{Mr}\,r^2\,\mathrm{t1DDs}}{4}+\frac{L\,\mathrm{Mr}\,g\,\sin(\mathrm{t1s})}{2}+\frac{L\,\mathrm{Mr}\,\mathrm{xDDs}\cos(\mathrm{t1s})}{2}$$

```
% Rod 2 Lagrange's equation

% Velocity derivative
dL_dt2_ds = diff(lagrangian, t2_ds);
dL_dt2_d  = subs(dL_dt2_ds, symb_var, time_var); % Sub out for time expressions

% Time derivative
d_dtt2  = diff(dL_dt2_d, t);
d_dtt2s = subs(d_dtt2, time_var, symb_var); % Symbolic version

% Position Derivative
dL_dt2s = diff(lagrangian, t2s);

% Lagrange's Equation
t2_left = d_dtt2s - dL_dt2s
```

```
t2_left(t) =
```

$$\frac{L^2\,\mathrm{Mr}\,\mathrm{t2DDs}}{3}+\frac{\mathrm{Mr}\,r^2\,\mathrm{t2DDs}}{4}+\frac{L\,\mathrm{Mr}\,g\,\sin(\mathrm{t2s})}{2}+\frac{L\,\mathrm{Mr}\,\mathrm{xDDs}\cos(\mathrm{t2s})}{2}$$

Generalized Power

Assumes the thruster is at the midpoint of the rod, which it is given as.

$$F_j=\begin{bmatrix}\cos(\theta_j)\\ \sin(\theta_j)\\ 0\end{bmatrix}$$

$$Q=\left\langle F_1,\ \dot{\vec{x}}_{\mathrm{rod}_1}\right\rangle+\left\langle F_2,\ \dot{\vec{x}}_{\mathrm{rod}_2}\right\rangle$$

```
% Velocity vectors, symbolic
vrod1_s = subs(vrod1, time_var, symb_var);
vrod2_s = subs(vrod2, time_var, symb_var);

% Force vectors, symbolic
F1_vs = F1 * [cos(t1s); sin(t1s); 0];
F2_vs = F2 * [cos(t2s); sin(t2s); 0];

% Generalized power
Ps = F1_vs' * vrod1_s + F2_vs' * vrod2_s;
Ps = expand(Ps);

% Extract coefficient
Q = (simplify(equationsToMatrix(Ps,[x_ds, t1_ds, t2_ds])))'
```

```
Q =
```

$$\begin{pmatrix} F_1 \cos(\text{t1s}) + F_2 \cos(\text{t2s}) \\ \frac{F_1 L}{2} \\ \frac{F_2 L}{2} \end{pmatrix}$$

Code Generation

```
[massMatrix, RVector] = equationsToMatrix([x_left, t1_left, t2_left], [x_dds, t1_dds, t2_dds]);
massMatrix            = simplify(massMatrix)
```

massMatrix =

$$\begin{pmatrix} \text{Mb} + 2\,\text{Mr} & \frac{L\,\text{Mr}\cos(\text{t1s})}{2} & \frac{L\,\text{Mr}\cos(\text{t2s})}{2} \\ \frac{L\,\text{Mr}\cos(\text{t1s})}{2} & \frac{\text{Mr}\,(4\,L^2 + 3\,r^2)}{12} & 0 \\ \frac{L\,\text{Mr}\cos(\text{t2s})}{2} & 0 & \frac{\text{Mr}\,(4\,L^2 + 3\,r^2)}{12} \end{pmatrix}$$

```
RVector = simplify(RVector + Q)
```

RVector =

$$\begin{pmatrix} \frac{L\,\text{Mr}\sin(\text{t1s})\,\text{t1Ds}^2}{2} + \frac{L\,\text{Mr}\sin(\text{t2s})\,\text{t2Ds}^2}{2} + F_1\cos(\text{t1s}) + F_2\cos(\text{t2s}) \\ \frac{L\,(F_1 - \text{Mr}\,g\sin(\text{t1s}))}{2} \\ \frac{L\,(F_2 - \text{Mr}\,g\sin(\text{t2s}))}{2} \end{pmatrix}$$

```
matlabFunction(RVector,'File','scripts/rVector');
matlabFunction(massMatrix,'File','scripts/massMatrix');
```