Drones and Autonomous Systems Laboratory

Ball balancing on the beam class 2

-Ball and Beam system dynamics-

DONGBIN KIM –
Ph.D. Candidate
Mechanical Engineering,
University of Nevada, Las Vegas
1. Introduction – scheme

(Ball and Beam Model)
2. Equation of Motion

\[
\frac{J}{R^2 + M} \dddot{x} + mg\theta = 0
\]

(State Space Representation)

\[
[\dot{x}] = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} [x] + \begin{bmatrix} mb \cdot \frac{g}{(l_b/r_b^2 + m_b)} \end{bmatrix} [u]
\]
2. Equation of Motion – Lagrangian Method

Lagrangian Method

1. Newton – Euler method is a “force balance” approach to dynamics
2. Lagrangian method is an “energy based” approach to dynamics
3. The Lagrangian $L$ is defined as the followings

$$L = K - P$$

($K = \text{kinetic energy}, P = \text{Potential Energy}$)

4. The dynamics equations, in terms of the coordinates used to express the kinetic and potential energy are obtained as

$$F_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i}$$

Where $q_i$ is the coordinates in which the kinetic and potential energy are expressed, $\dot{q}_i$ is the corresponding velocity, and $F_i$ the corresponding force or torque
2. Equation of Motion - Derivation

(Equation of Motion)

\[
\left( \frac{J}{R^2} + M \right) \ddot{r} + mg \theta = 0
\]

(State Space Representation)

\[
\begin{bmatrix}
\dot{x}
\end{bmatrix} =
\begin{bmatrix}
0 & 1 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
x
\end{bmatrix} +
\begin{bmatrix}
mb \cdot \frac{g}{\left( \frac{L}{r^2} + m_b \right)}
\end{bmatrix}
\begin{bmatrix}
u
\end{bmatrix}
\]

(Ball and Beam Model)

Look at the whiteboard
3. Simulation - MATLAB

\[
\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} mb \cdot \frac{g}{J_b \cdot r_b^2 + m_b} \end{bmatrix} u
\]

Ball and Beam Model

\( mb = 0.004; \) mass of the ball
\( r_b = 0.024; \) radius of the ball
\( J_b = (2 \times mb \times rb^2)/3; \) MOMI of the ball
\( g=9.81; \) Gravitational Acceleration.

MatrixA = [0 1; 0 0];
MatrixB = [0; mb*g/(Jb/(rb^2)+mb)]
MatrixC = [1 0];

(Equation of Motion in State Space Representation)
3. Simulation – Transfer function

\[
\begin{bmatrix}
    \dot{x} \\
    \dot{\theta}
\end{bmatrix} =
\begin{bmatrix}
    0 & 1 \\
    0 & 0
\end{bmatrix}
\begin{bmatrix}
    x \\
    \theta
\end{bmatrix} +
\begin{bmatrix}
    0 \\
    m_b \frac{g}{r_b^2 + m_b}
\end{bmatrix}
u
\]

(Book and Beam Model)

Find Transfer Function

\( m_b = 0.004 \); % mass of the ball yellow : 0.002
\( r_b = 0.02 \); % radius of the ball
\( b = 0.01 \); % moment of inertia ball
\( g = 9.81 \); % gravitational acceleration.

\( \text{MatrixA} = [0 \ 1; \ 0 \ 0] \);
\( \text{MatrixB} = [0; mb * g / (r_b^2 + mb)] \);
\( \text{MatrixC} = [1 \ 0] \);

(Equation of Motion in State Space Representation)
3. Simulation — Transfer Function

Transfer Function, Root Locus Results

1. Divergence response results
2. Two zero Poles
3. Step Response -> Divergence
4. How we are going to make the system stable?
5. PID? Full state-feedback?
3. Simulation – Transfer Function

Transfer Function, Root Locus Results

1. **Divergence** response results
2. Two zero Poles
3. Step Response -> **Divergence**
4. How we are going to make the system **stable**?
5. **PID? Full state-feedback?**
3. Simulation – Transfer Function

Transfer Function, Root Locus Results

1. Divergence response results
2. Two zero Poles
3. Step Response -> Divergence
4. How we are going to make the system stable?
5. PID? Full state-feedback?

Ball Position (m)

(PID Compensated Results)
4. Homework

Homework for next course

1. Using Newton method to derive equation of motions
2. Create your own PID compensator to stabilize your system, show the plots, and Simulink or MATLAB code