AE353: Design Problem 04

Two-Wheel Robotics

April 11, 2019

1 Goal

`DesignProblem04` contains a simulation in which a two-wheeled self balancing robot follows a generated path of 120 meters. The robot has two actuators which allow controllability of the angular torque of the two wheels, τ_L and τ_R. The robot is also equipped with sensors which allow the controller to analyze the angular velocity and radius of the upcoming path. The ultimate goal is to balance the robot, maximize speed, and follow the path.

1.1 Requirements

The robot is required to reach and maintain a forward-oriented velocity of $10\frac{\text{m}}{\text{s}}$ within 1 second of the simulation initiating. The robot must not crash, diverge from path, or exceed a pitch angle of $\phi = 12°$ in order to maintain optimal speed and stability.

1.2 Verification

The simulation will contain a verification test to ensure the requirements are satisfied. This will be done by checking if $v \geq 10\frac{\text{m}}{\text{s}}$. The `processdata.v` vs. `processdata.t` will be plotted to verify whether the robot met the desired velocity within the time requirement. A system is verified if the robot maintains $e_{lateral}$ and $e_{heading} \leq 2$.

2 Model

2.1 Robot Dynamics

The non-linearized differential system may be defined by (1):

$$\begin{bmatrix} \ddot{\phi} \\ \dot{v} \\ \dot{w} \end{bmatrix} = f(\phi, \dot{\phi}, v, w, \tau_R, \tau_L), \quad \text{State} = \begin{bmatrix} \dot{\phi} - \dot{\phi}_E \\ v - v_E \\ w - w_E \\ \phi - \phi_E \\ e_{lateral} - e_{lateral,E} \\ e_{heading} - e_{heading,E} \end{bmatrix}, \quad \text{Input} = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} \text{Output} = \begin{bmatrix} e_{lateral} \\ e_{heading} \end{bmatrix} \tag{1}$$

$\phi \rightarrow$	Chassis Angle	$\dot{\phi} \rightarrow$	Change in Chassis Angle
$v \rightarrow$	Forward Velocity	$w \rightarrow$	Turning rate
$\tau_L \rightarrow$	Left Motor Torque	$\tau_R \rightarrow$	Right Motor Torque

The simulator is designed to accept three equations of motion in reference to the position of the robot, (x, y) and θ, defined by (2):

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= w \end{aligned} \tag{2}$$

In order to let the robot keep track of the centerline of the road, equation (3) is implemented to provide a method of letting the robot mitigate the error between robot position and the centerline:

$$\begin{aligned} w_{road} &= \frac{v_{road}}{r_{road}} \\ \dot{e}_{lateral} &= -v\sin(e_{heading}) \\ \dot{e}_{heading} &= w - (\frac{v\cos(e_{heading})}{v_{road} + w_{road}e_{lateral}})w_{road} \end{aligned} \tag{3}$$

2.2 Finding Equilibrium Points

The next crucial part is determining the equilibrium points to find the desired State Space model.

2.3 State Space Model

For the general State Space Model shown in (4), the matrix in (5) is the result of linearizing the non-linear ODE.

$$\begin{aligned} \dot{\mathbf{x}} &= Ax + Bu \\ \mathbf{y} &= Cx + Du \end{aligned} \tag{4}$$

$$\mathbf{A} = \begin{bmatrix} -0.922 & 0 & -0.376 & 8.54 & 0 & 0 \\ 0.453 & 0 & -0.00699 & 6.45 & 0 & 0 \\ 0.895 & 1.01 & -0.43 & 1.33 & 0 & 0 \\ 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -6.28 \\ 0 & 0 & 1.0 & 0 & 0 & 0 \end{bmatrix} \mathbf{B} = \begin{bmatrix} -2.19 & -2.19 \\ 0.578 & 0.578 \\ 2.32 & -2.32 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

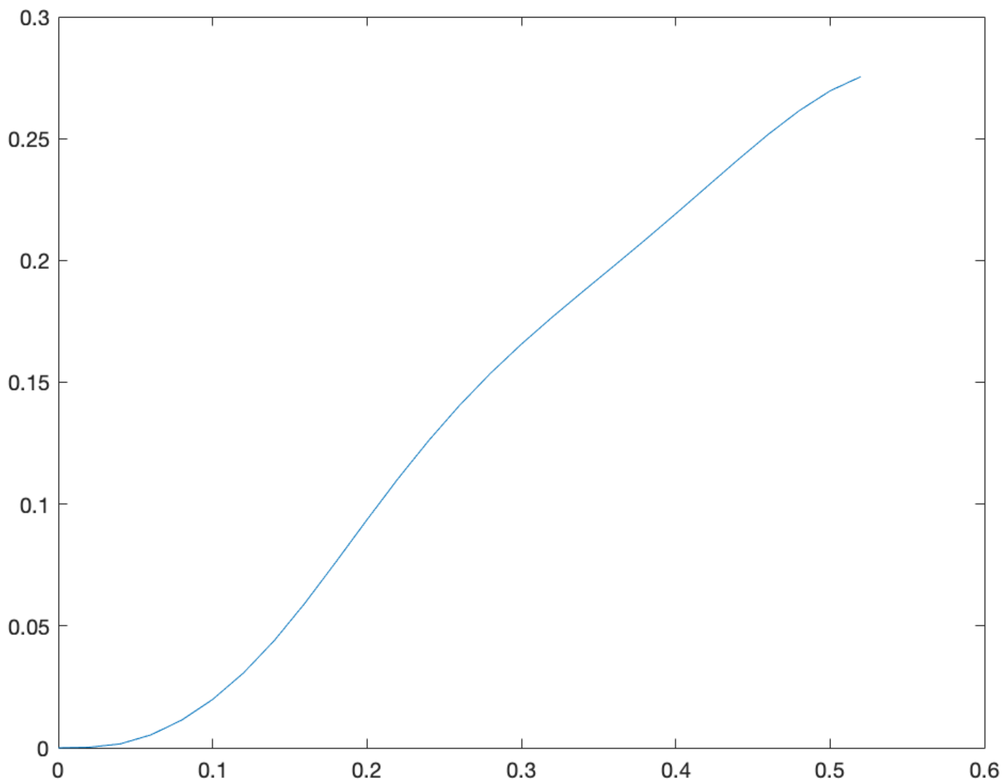
3 Controller Design

3.1 Controller

3.2 Observer

3.3 Completed Controller

4 Conclusions



Position vs time