Modeling and Control of a Two Wheeled Auto-Balancing Robot: a didactic platform for control engineering education

Stott, Edward A

June 2, 2023

1 Introduction

The Two Wheeled Automatic Balancing Robot (TWABR) is an unstable and non-linear electromechanical system consisting of two side wheels that have contact with the floor surface. The wheels are independently driven to balance in the gravity center above the axis of the wheel's rotation. The wheels are driven by two motors coupled to each of them. The motors can be of a DC nature and are controlled by electrical signals by means of a control system based on the inclination reading and the velocity of their gravity center. Its operation is similar to the classic inverted pendulum system. The control objective is to stabilize the TWABR by keeping it in a vertical equilibrium position.The complex dynamics inherent in this platform finds its application in the design and development of control systems for cars, spacecraft, domestic transportation, military transport, among others.

This experiment aims to demonstrate the design of a controllers for a TWABR by means of a linear control tools. In implementing such a control system the following topics will be covered.

1. Modelling the dynamics of an TWABR using the Euler-Lagrange equation
2. Obtaining a linear state-space representation of the system
3. Creating a Simulink model to simulate the non-linear behaviour of the pendulum.
4. Designing a state-feedback control law that improves damping for the pendulum in the downward position and balances it at its vertical upward position
5. Implementing the designed control law on the Simulink model and verifying its performance

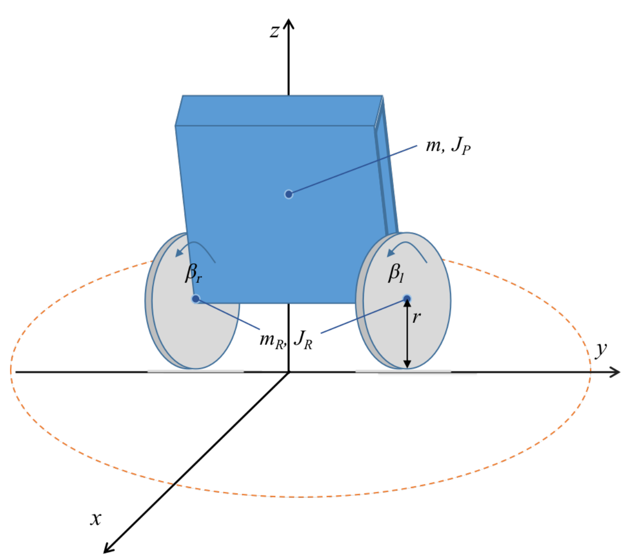

Figure 1: Balanced Free-body diagram of the TWABR system.

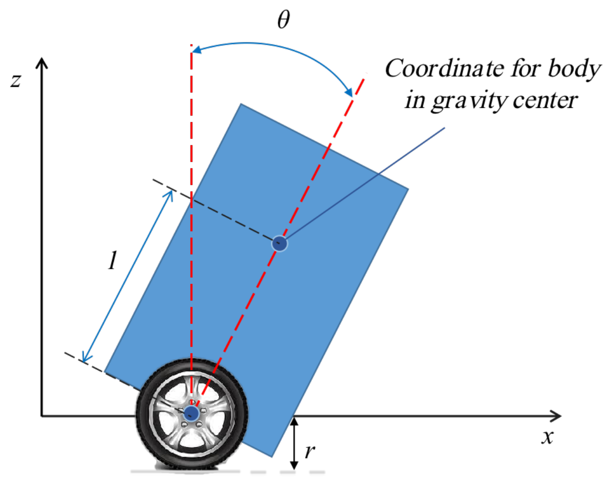

Figure 2: Inclined Free-body diagram of the TWABR system.

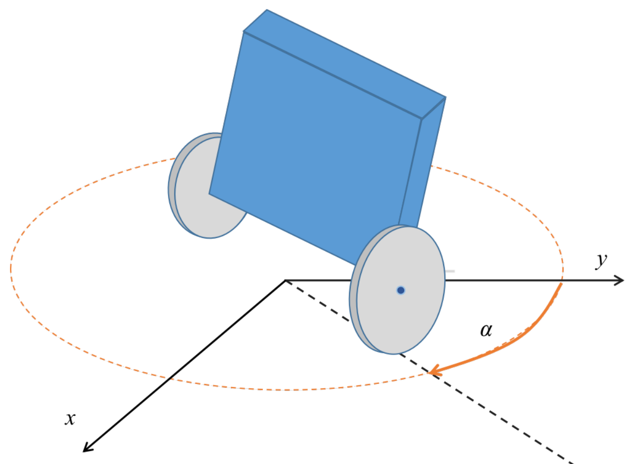

Figure 3: Rotated Free-body diagram of the TWABR system.

2 MATHEMATIC MODEL OF THE TWABR SYSTEM

the following considerations are assumed: 1) the wheels follow a non-slip movement, 2) the wheels of the rotation system are rigid and similar (same radius and masses) and 3) that the weight in the main body is evenly distributed. Frictions in movements are initially considered, but are later omitted.

The position and velocity of the main body are given as:

$$P = \begin{bmatrix} x + l\sin\theta\cos\alpha \\ y + l\sin\theta\sin\alpha \\ r + l\cos\alpha \end{bmatrix}, \quad V = \begin{bmatrix} \dot{x} + l(\dot{\theta}\cos\theta\cos\alpha - \dot{\alpha}\sin\theta\sin\alpha) \\ \dot{y} + l(\dot{\theta}\cos\theta\sin\alpha + \dot{\alpha}\sin\theta\cos\alpha) \\ -l\dot{\alpha}\sin\alpha \end{bmatrix} \tag{1}$$

and the angular velocities of the mobile base and the main body are given as:

$$\omega_m = \begin{bmatrix} 0 \\ 0 \\ \dot{\alpha} \end{bmatrix}, \quad \omega_p = \begin{bmatrix} -\dot{\alpha}\sin\theta \\ \dot{\theta} \\ \dot{\alpha}\cos\theta \end{bmatrix} \tag{2}$$

The motion equations that define the behavior of the TWABR system can be obtained using the Lagrangian dynamics .

The kinetic energy of the main body due to angular displacement can be represented as:

$$KE_p = \frac{1}{2}MV^TV + \frac{1}{2}\omega_p^T J_p \omega_p \tag{3}$$

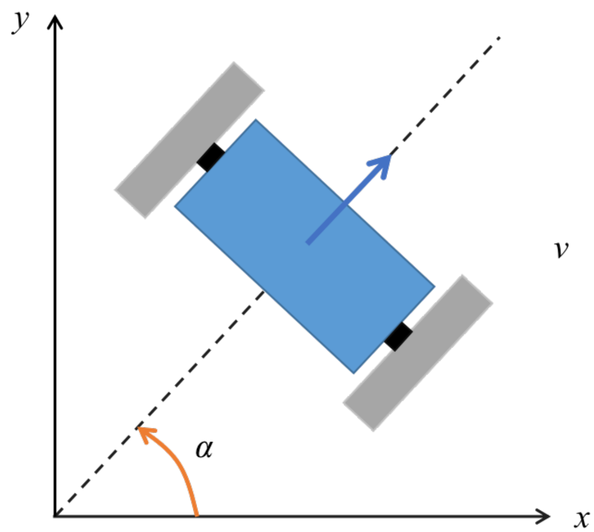

Figure 4: Translational and rotational velocities of a wheeled mobile robot.

The kinetic energy of the mobile base can be represented as:

$$KE_m = \frac{1}{2}mr^2(\dot{\beta_l}^2 + \dot{\beta_r}^2) + \frac{1}{2}(J_{wa} + J_{ra}\gamma^2)(\dot{\beta_l}^2 + \dot{\beta_r}^2) + (J_{wd} + J_{wr})\dot{\alpha}^2 \tag{4}$$

The kinetic energy of the total system is:

$$KE_s = T_p + T_m \tag{5}$$

The potential energy of the system can be represented as:

$$PE_s = Mgl\cos\theta \tag{6}$$

The damping energy of the system can be represented as:

$$D = \frac{1}{2}c_r(\dot{\beta_r} - \dot{\theta})^2 + \frac{1}{2}c_l(\dot{\beta_l} - \dot{\theta})^2 \tag{7}$$

Consequently, the Lagrangian of the system is $L = KE_s - PE_s$. The Lagrangian equation is expressed as:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}_i}) - \frac{\partial L}{\partial q_i} + \frac{\partial D}{\partial \dot{q}_i} = Q_i \quad i = 1, 2, \ldots, n \tag{8}$$

where Q, q and n denote a generalized force, coordinate, and coordinate number, respectively. We define a generalized coordinate as $q = [x, y, \alpha, \theta, \beta_r, \beta_l]^T$, the six Lagrangian equations are:

$$\begin{aligned}
&\textbf{Eq1}: \; M\,\ddot{x} - M\,l\,\cos\alpha\,\sin\theta\,\dot{\alpha}^2 - M\,l\,\cos\alpha\,\sin\theta\,\dot{\theta}^2 + M\,l\,\cos\alpha\,\cos\theta\,\ddot{\theta} - M\,l\,\sin\alpha\,\sin\theta\,\ddot{\alpha}\\
&-2\,M\,l\,\sin\alpha\,\cos\theta\,\dot{\theta}\,\dot{\alpha} = 0\\
&\textbf{Eq2}: \; M\,\ddot{y} - M\,l\,\sin\alpha\,\sin\theta\,\dot{\alpha}^2 - M\,l\,\sin\alpha\,\sin\theta\,\dot{\theta}^2 + M\,l\,\cos\alpha\,\sin\theta\,\ddot{\alpha} + M\,l\,\sin\alpha\,\cos\theta\,\ddot{\theta}\\
&+2\,M\,l\,\cos\alpha\,\cos\theta\,\dot{\theta}\,\dot{\alpha} = 0\\
&\textbf{Eq3}: \; (2\,J_{\mathrm{rd}} + 2\,J_{\mathrm{wd}} + J_{\mathrm{xx}} - J_{\mathrm{xx}}\,\cos^2\theta + J_{\mathrm{zz}}\,\cos^2\theta + M\,l^2 - M\,l^2\,\cos^2\theta)\,\ddot{\alpha} + M\,l\,\cos\alpha\,\sin\theta\,\ddot{y}\\
&+(2\,J_{\mathrm{xx}}\,\cos\theta\,\sin\theta - 2\,J_{\mathrm{zz}}\,\cos\theta\,\sin\theta + 2\,M\,l^2\,\cos\theta\,\sin\theta)\,\dot{\theta}\,\dot{\alpha} - M\,l\,\sin\alpha\,\sin\theta\,\ddot{x} = 0\\
&\textbf{Eq4}: \; (c_l + c_r)\,\dot{\theta} - c_l\,\dot{\beta}_l - c_r\,\dot{\beta}_r + (J_{\mathrm{yy}} + M\,l^2)\,\ddot{\theta} + (J_{\mathrm{zz}}\,\cos\theta\,\sin\theta - J_{\mathrm{xx}}\,\cos\theta\,\sin\theta - M\,l^2\,\cos\theta\,\sin\theta)\,\dot{\alpha}^2\\
&-m\,g\,l\,\sin\theta + M\,l\,\cos\alpha\,\cos\theta\,\ddot{x} + M\,l\,\sin\alpha\,\cos\theta\,\ddot{y} = -T_l - T_r\\
&\textbf{Eq5}: \; (J_{\mathrm{ra}}\,\gamma^2 + m\,r^2 + J_{\mathrm{wa}})\,\ddot{\beta}_r + c_r\,\dot{\beta}_r - c_r\,\dot{\theta} = T_r\\
&\textbf{Eq6}: \; (J_{\mathrm{ra}}\,\gamma^2 + m\,r^2 + J_{\mathrm{wa}})\,\ddot{\beta}_l + c_l\,\dot{\beta}_l - c_l\,\dot{\theta} = T_l
\end{aligned} \tag{9}$$

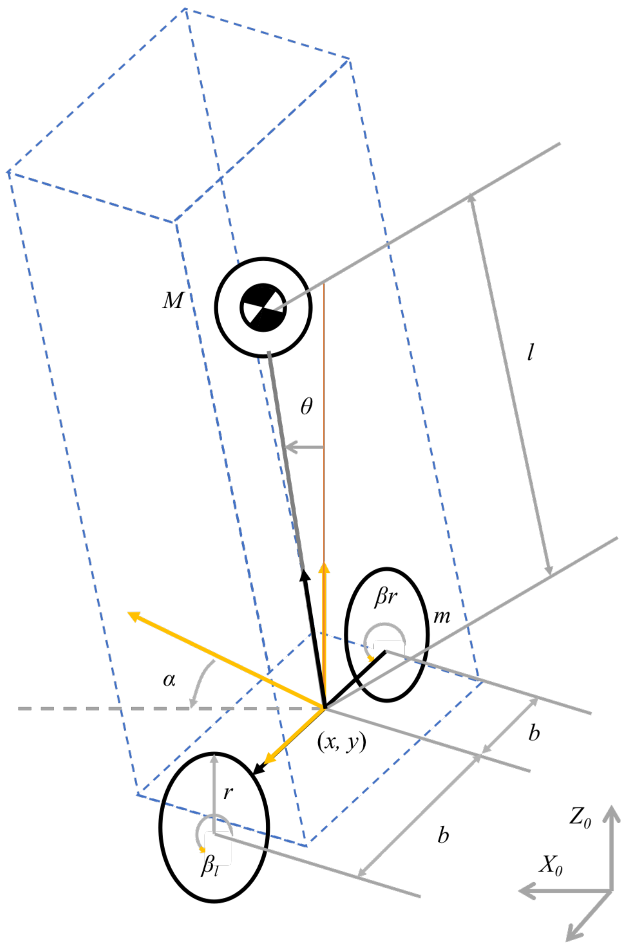

Figure 5: Model of 3DOF wheeled inverted pendulum robot.

which can be rearranged in the matrix form:

$$M(q)\ddot{q} + V\dot{q} + H(q, \dot{q}) = E\tau \tag{10}$$

where

$$E = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \tag{11}$$

The structure of the fourth row of E arises because the motors are mounted on the main body. Notice that there are three constraints on the mobile base by assuming that the wheels do not slip:

$$\dot{x}\sin\alpha - \dot{y}\cos\alpha = 0 \tag{12}$$

$$\dot{x}\cos\alpha + \dot{y}\sin\alpha = \frac{r}{2}(\dot{\beta}_r + \dot{\beta}_l) \tag{13}$$

$$\dot{\alpha} = \frac{r}{2b}(\dot{\beta}_r - \dot{\beta}_l) \tag{14}$$

The motion equation with constraints is expressed as:

$$M(q)\ddot{q} + V\dot{q} + H(q, \dot{q}) = E\tau + A_q^T\lambda \tag{15}$$

where λ is a Lagrangian multiplier and A_q is written as:

$$A_q = \begin{bmatrix} \sin\alpha & -\cos\alpha & 0 & 0 & 0 & 0 \\ \cos\alpha & \sin\alpha & b & 0 & -r & 0 \\ \cos\alpha & \sin\alpha & -b & 0 & 0 & -r \end{bmatrix} \tag{16}$$

Because it is difficult to find λ, we define a matrix S_q composing linear independent vector in the null-space of A_q:

$$S_q = \begin{bmatrix} \cos\alpha & 0 & 0 \\ \sin\alpha & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1/r & b/r & 0 \\ 1/r & -b/r & 0 \end{bmatrix} \tag{17}$$

which shows $A_q S_q = 0$. Next, we follow the standard procedure for the elimination of Lagrange multiplier λ by premultiplication (15) with S_q^T:

$$S_q^T M(q)\ddot{q} + S_q^T V\dot{q} + S_q^T H(q,\dot{q}) = S_q^T E\tau \tag{18}$$

Define $\nu = [v, \dot{\alpha}, \dot{\theta}]^T$, we see $\dot{q} = S_q\nu$ and $\ddot{q} = \dot{S}_q\nu + S_q\dot{\nu}$. Then (18) can be rewritten as:

$$\hat{M}\dot{\nu} + \hat{V}\nu + \hat{H}(\nu,\dot{\nu}) = \hat{E}\tau \tag{19}$$

where $\hat{M} = S_q^T M(q)S_q$, $\hat{H} = S_q^T[M(q)\dot{S}_q\nu + H(q,\dot{q})]$, $\hat{V} = S_q^T V(q)S_q$, $\hat{E} = S_q^T E$.

3 Linear System Analysis

3.1 Linearization

Q1) Define the control variable $x = [x, v, \alpha, \dot{\alpha}, \theta, \dot{\theta}]^T$ and linearize the system around the equilibrium point $[0,0,0,0,0,0]^T$ and express the obtained linear dynamics in the standard state space format. (finding the expression of matrix A and B):

$$\dot{x} = Ax + Bu \tag{20}$$

3.2 Discretization

To implement the control method on the hardware, we need to first discretize the continuous model according to the following relationship:

$$A_d = e^{AT}, \quad B_d = (\int_0^T e^{A\lambda}d\lambda)B \tag{21}$$

Q2) Find the expression of matrix A_d and B_d. (using the function ”c2d” in Matlab)

4 Control Law Design

4.1 Pole placement method

4.1.1 Full state feedback

Pole placement is a method of calculating the optimum gain matrix. Closed-loop pole locations directly impact time response characteristics such as rise time, settling time, and transient oscillations. By placing all poles of the closed-loop discrete-time system in the unit circle, we could obtain a stable system.

Consider our linear discrete-time system $x(k+1) = A_d x(k) + B_d u(k)$ and design the control law as $u(k) = -Kx(k)$, we see:

$$x(k+1) = (A_d - B_d K)x(k) \tag{22}$$

Hence, when $A_d - B_d K$ is stable, the system converges to the origin where the distance and the heading angle are 0, and the pendulum is at the vertical position. However, we also need the BalanceBug to achieve a target destination and heading angle. Hence, the control objective is to modify the control law as $u(k) = -Kx(k) + v(k)$ so that the closed-loop control system is stable and could track a reference set point.

Let $[x, \alpha]^T$ be the output of the system, and let the reference set point be $y_r(k)$, the following equation should be satisfied:

$$x(k+1) = (A_d - B_d K)x(k) + B_d v(k) \tag{23}$$

$$y_r(k) = C_d x(k) \tag{24}$$

Notice that when $y_r(k)$ is achieved, the corresponding state should fulfil $x(k+1) = x(k)$. Hence, we obtain

$$y_r(k) = C_d x(k) = (C_d(I - (A_d - B_d K))^{-1} B_d)v(k) \tag{25}$$

Therefore, the control law is:

$$u(k) = -Kx(k) + [C_d(I - (A_d - B_d K))^{-1} B_d]^{-1} y_r(k) \tag{26}$$

Q3) Find the gain matrix K by placing stable poles. First, arbitrarily select stable poles in the $s-$ domain (locates in the negative half-plane) and according to relationship $z = e^{sT_s}$, where T_s is the sampling time, to obtain stable poles in $z-$ domain. Then, using "place" function in Matlab to obtain gain matrix K.

4.1.2 State observer

The measurable state of the sensor includes distance x, heading angle α and pendulum tilting angle θ. Therefore, we cannot obtain full state feedback. We first introduce a state observer.

A state observer is a filter that allows one to estimate the state of a system from measurements of the input and output signals. Remember the system dynamic is:

$$x(k+1) = A_d x(k) + B_d u(k) \tag{27}$$

$$y(k) = C_d x(k) \tag{28}$$

We are going to construct a state observer to estimate the true state based on the measured signals. Define the estimation state ξ and the estimation error $e_s = x - \xi$ and construct a simple observer for our system:

$$\xi(k+1) = A_d \xi(k) + B_d u(k) + L y_\xi \tag{29}$$

where $y_\xi = C_d \xi - y$ is the difference between estimation output $C_d \xi$ and true output y measured by sensors. The observer contains a copy of the system and also exploits the knowledge of measured signals. Remember that $e_s = x - \xi$, we could also obtain the estimation error dynamic from (29):

$$e_s(k+1) = (A_d - LC_d)e_s(k) \tag{30}$$

when the system $A_d + LC_d$ is stable, the estimation state converges to the true state.

Q4) Similar as (Q3), find the matrix L by placing stable poles.

With a full estimation state from the state observer, we could implement the designed control law on the estimation state $u(k) = -K\xi(k) + [C_d(I - (A_d - B_d K))^{-1} B_d]^{-1} y_r(k)$.

4.2 Model Predictive Control

Model Predictive Control (MPC), also known as Predictive Control, is a control method which has been widely and successfully applied in academic research and industrial applications since it was developed in the 1980s. It has advantages in dealing with large-scale problems with multiple states and control inputs. [1]

The core algorithm of the MPC is that at each sample time instant, a predictive controller will:

1. Take a measurement of the current state and output of the system.
2. Based on an internal nominal model which can forecast the following behaviours of the system, a sequence of optimal control variables that
   (a) minimises specified cost functions
   (b) obeys all constraints

   will be generated;
3. The first part of the generated optimal control sequence will be implemented to the controlled system which can minimise the objective function of the optimisation problem. In this project, only the first element of the optimal sequence is implemented.

4.2.1 Dynamical formulations of a nominal MPC problem

In this section, a nominal model predictive control (Nominal MPC) problem is introduced. Since the Nominal MPC only considers ideal scenarios where mismatches between the model and the system do not exist, neither disturbances nor uncertainties are included. Nominal MPC problems discussed in this work are constrained linear-quadratic problems (CLQ). Besides, dynamic systems investigated in this work are discrete, linear, time-invariant (DLTI) systems, which can be written into the following format:

$$x(t+j+1|t) = A_d x(t+j|t) + B_d u(t+j|t), \tag{31}$$
$$y(t+j|t) = C_d x(t+j|t) + D_d u(t+j|t), \tag{32}$$
$$z(t+j|t) = C_z x(t+j|t) + D_z u(t+j|t), \tag{33}$$

where $x(t+j|t) \in \mathbb{R}^n$, $u(t+j|t) \in \mathbb{R}^m$, $y(t+j|t) \in \mathbb{R}^o$, $z(t+j|t) \in \mathbb{R}^p$ are the system state, control input, measured output and cost signals, respectively. $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{o\times n}$, $D \in \mathbb{R}^{o\times m}$, $C_z \in \mathbb{R}^{p\times n}$, $D_z \in \mathbb{R}^{p\times m}$ are matrix coefficients. j is the prediction step which is subjected to the set $\{0, 1, 2, \ldots, N-1\}$, where N is the length of the prediction horizon.

The state and control input signals are constrained by boundaries which are defined by constrained equations:

$$\underline{x}(t+j|t) \le x(t+j|t) \le \bar{x}(t+j|t), \tag{34}$$
$$\underline{u}(t+j|t) \le u(t+j|t) \le \bar{u}(t+j|t). \tag{35}$$

The quadratic cost function is given in:

$$J = x(t+N|t)^T \hat{C}_z^T \hat{C}_z x(t+N|t) + \sum_{j=0}^{N-1} \left(z(t+j|t) - \bar{z}(t+j|t)\right)^T Q \left(z(t+j|t)\bar{z}(t+j|t)\right), \tag{36}$$

where $\bar{z}(t+j|t)$ is the reference target, superscript T denotes transpose operation, $Q \in \mathbb{R}^{p\times p}$ is a weighting matrix, matrix $\hat{C}_z$ represents the terminal coefficient matrix. The initial state $x(t+0|t)$ is assumed to be given. Therefore, by solving the following quadratic programming problem, one can obtain a sequence of optimal control solutions.

$$\begin{aligned} \min_{u(t+j|t)} \quad & J, \\ s.t.: \quad & x(t+j+1|t) = A_d x(t+j|t) + B_d u(t+j|t), \\ & \underline{x}(t+j|t) \le x(t+j|t) \le \bar{x}(t+j|t), \\ & \underline{u}(t+j|t) \le u(t+j|t) \le \bar{u}(t+j|t), \\ & j = 0, 1, 2 \ldots N-1. \end{aligned} \tag{37}$$

Only the first part of the generalised control solution is implemented to update the system.

[1] To study more about MPC theories, you can choose ELEC70028 Predictive Control module.