ECE MBD: Laboratory session 6

Balanbot – Discretization and Model in-the-Loop

The main objective of this session is to create the control function for the Balanbot and test it by a Model in-the-loop approach. The dynamic behavior of the Balanbot will be retaken from the model already developed in the PED (3^{rd} semester).

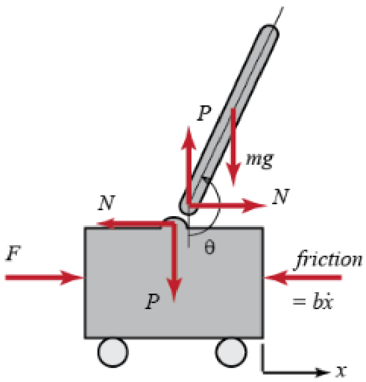
Laboratory work

Plant model

1. As you might recall, the Balanbot is an inverted pendulum, therefore its model can be obtained by considering the method described in:

   http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=SimulinkModeling

   Starting at the free body diagram:



Where:

x : cart's position	b : coefficient of friction for cart
$\dot{x}$: cart's velocity	l : length to pendulum center of mass
$\ddot{x}$: cart's acceleration	J : moment of inertia of the pendulum
θ : pendulum's position (angle)	F : external force applied (by motors)
$\dot{\theta}$: angular velocity	N : interaction force between cart and pendulum in x direction
$\ddot{\theta}$: angular acceleration	P : interaction force between cart and pendulum in y direction
m : mass of pendulum	
M : mass of cart	
g : gravitational constant	

Based on Newton's laws, governing equations of the system can be obtained:

$$\ddot{x} = \frac{1}{M}\sum_{cart} F_x = \frac{1}{M}(F - N - b\dot{x}) \tag{1}$$

$$\ddot{\theta} = \frac{1}{I}\sum_{pend} \tau = \frac{1}{I}(-Nl\cos\theta - Pl\sin\theta) \tag{2}$$

It is necessary, however, to include the interaction forces N and P between the cart and the pendulum in order to fully model the system's dynamics. The inclusion of these forces requires modeling the x- and y-components of the translation of the pendulum's center of mass in addition to its rotational dynamics.

$$m\ddot{x}_p = \sum_{pend} F_x = N \tag{3}$$

$$\Rightarrow N = m\ddot{x}_p \tag{4}$$

$$m\ddot{y}_p = \sum_{pend} F_y = P - mg \tag{5}$$

$$\Rightarrow P = m(\ddot{y}_p + g) \tag{6}$$

However, the position coordinates x_p and y_p are exact functions of θ. Therefore, we can represent their derivatives in terms of the derivatives of θ. First addressing the x-component equations we arrive at the following.

$$x_p = x + l\sin\theta \tag{7}$$

$$\dot{x}_p = \dot{x} + l\dot{\theta}\cos\theta \tag{8}$$

$$\ddot{x}_p = \ddot{x} - l\dot{\theta}^2\sin\theta + l\ddot{\theta}\cos\theta \tag{9}$$

Then addressing the y-component equations gives us the following.

$$y_p = -l\cos\theta \tag{10}$$

$$\dot{y}_p = l\dot{\theta}\sin\theta \tag{11}$$

$$\ddot{y}_p = l\dot{\theta}^2\cos\theta + l\ddot{\theta}\sin\theta \tag{12}$$

These expressions can then be substituted into the expressions for N and P from above as follows.

$$N = m(\ddot{x} - l\dot{\theta}^2\sin\theta + l\ddot{\theta}\cos\theta) \tag{13}$$

$$P = m(l\dot{\theta}^2 \cos\theta + l\ddot{\theta}\sin\theta) + g \qquad (14)$$

Equations (1), (2), (13) and (14) totally described the system in its non-linear model.

Parameter Update

1. The non-linear model can be implemented as above described. Please update the Balanbot parameters as follows:

   m = 0.0715 kg

   M = 0.0179 kg

   g = 9.81 m/s²

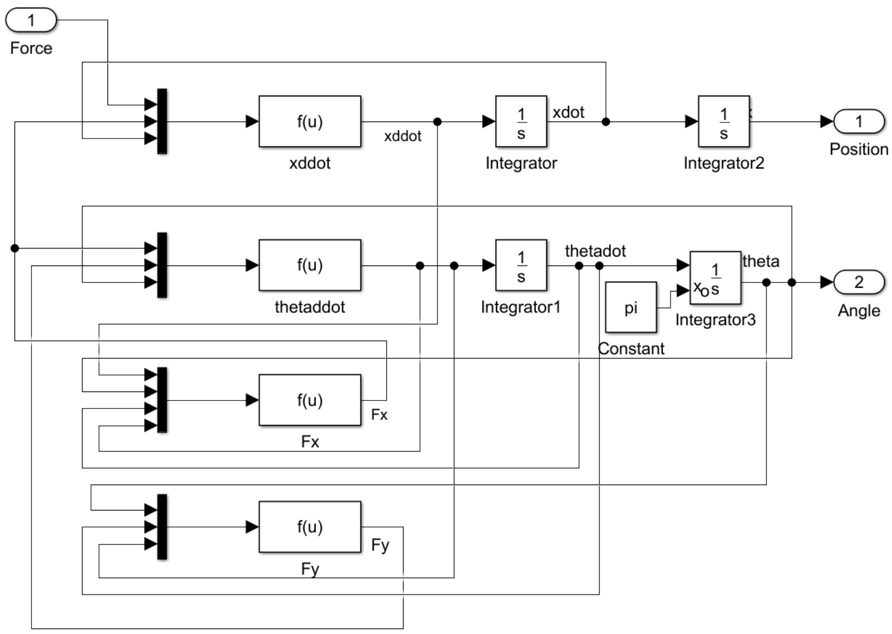
   b = 0.2 Kg/s

   l = 0.11 m

   The moment of inertia J can be calculated from the mass and dimensions of the Balanbot as follows:

   $J = M*W^2/12 + M*H^2/3$; being W = 0.145 m and H = l = 0.11 m

   The model shall look similar to:



Discretization non-linear model

2. Discretize the non-linear Balanbot model considering a sampling time T = 0.001 sec.

   - At which places the discretization should be applied?

- Apply zero force at the input and observe the response. What is the initial pendulum position? Is the system stable? Report your observations.
- Modify the model for having a little initial deviation in the pendulum's position during 0.1 sec, e.g. 5°, but remember that this value shall be converted into radians. Report your findings.

Linearization

3. Now, let us recall the linear model, which transfer function has been obtained accordingly with the following reference:

   http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=SystemModeling

   Here the model is linearized around the up-straight position by considering:

$$\cos\theta = \cos(\pi + \phi) \approx -1$$

$$\sin\theta = \sin(\pi + \phi) \approx -\phi$$

$$\dot{\theta}^2 = \dot{\phi}^2 \approx 0$$

   Please note the change of variable for the pendulum's position ϕ instead of θ.

   Report the derivation of the linear transfer function $\Phi(s)/U(s)$ and $\mathrm{X}(s)/U(s)$. Why is possible to linearize the plant?

Discretization linear model

4. Discretize with the help of MATLAB the linear continuous model (T=0.001 sec)
   - use the three different discretization methods analysed in the preparation session.

System analysis

5. Analyze the allocation of poles and zeros of the continuous as well as discrete transfer functions. Explain your findings.

   Hint: You can use the MATLAB's commands *pole* or *pzmap.*
6. Implement the discrete linear transfer functions in parallel with the discrete non-linear model and test it in open loop. What do you observe?

Control function

7. Please develop two models: one using the continuous plant and solver and other using the discrete plant and solver. The control function in either case shall be discrete (the same in both cases); which shall be implemented according to the requirement listed below. From the exercises, it is expected that you report your development process of your solution: design and testing.

Functional requirements

FR1. The control function for the Balanbot requires a Kalman filter followed by a PID controller.

- The Kalman filter provides a more plausible actual vertical position (given in degrees) using the information coming from the plant.
- Next, the PID controller computes the necessary torque that the motors shall provide in order to keep the 0° vertical position.

FR2. The control function shall operate only in the range between -40° and 40°. In case that the actual position exceeds the operating range then the motors shall stop immediately.

Non-functional requirements

NF1. The entire control function shall be implemented as a triggered sub-system model in MATLAB/Simulink with a sampling time of 0.01 sec.

NF2. The Kalman filter script (provided as an annex) shall be included in a MATLAB function within the control function. It is recommendable to test the Kalman filter in a standalone model as shown as follows:

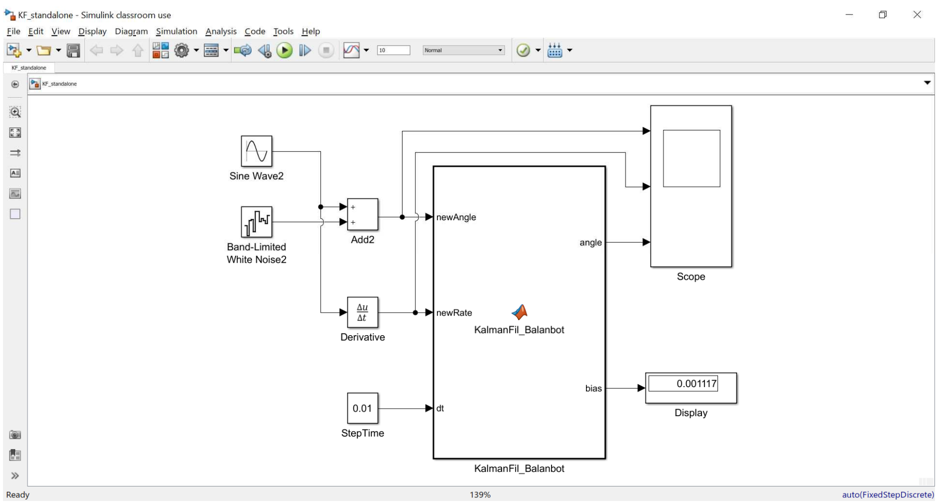


Figure 1: Kalman Filter test

NF3. The PID controller shall be implemented by means of Simulink blocks, i.e. don't use the predefined PID block; according to:

a. The proportional, integral and derivative gains (Kp, Ki and Kd; respectively) shall be adjustable during run

b. Kp shall be multiplied by angle (output of Kalman filter)

c. Ki shall be multiplied by the integral of angle (output of Kalman filter)

d. Kd shall be multiplied directly by newRate (input of the Kalman filter)

   Note: In the hardware implementation this is a reliable signal coming from the gyroscope.

e. The output of the controller (sum of the three previous parts) shall be saturated to -255 and 255

Hint 1: A good starting point for the gains is: Kp=15, Ki=0.35 and Kd=3.5

Hint 2: In order to convert the control output into torque (force) for the plant, a factor of 0.005 is recommended. This factor was empirically found!