Alexandria University
Faculty of engineering

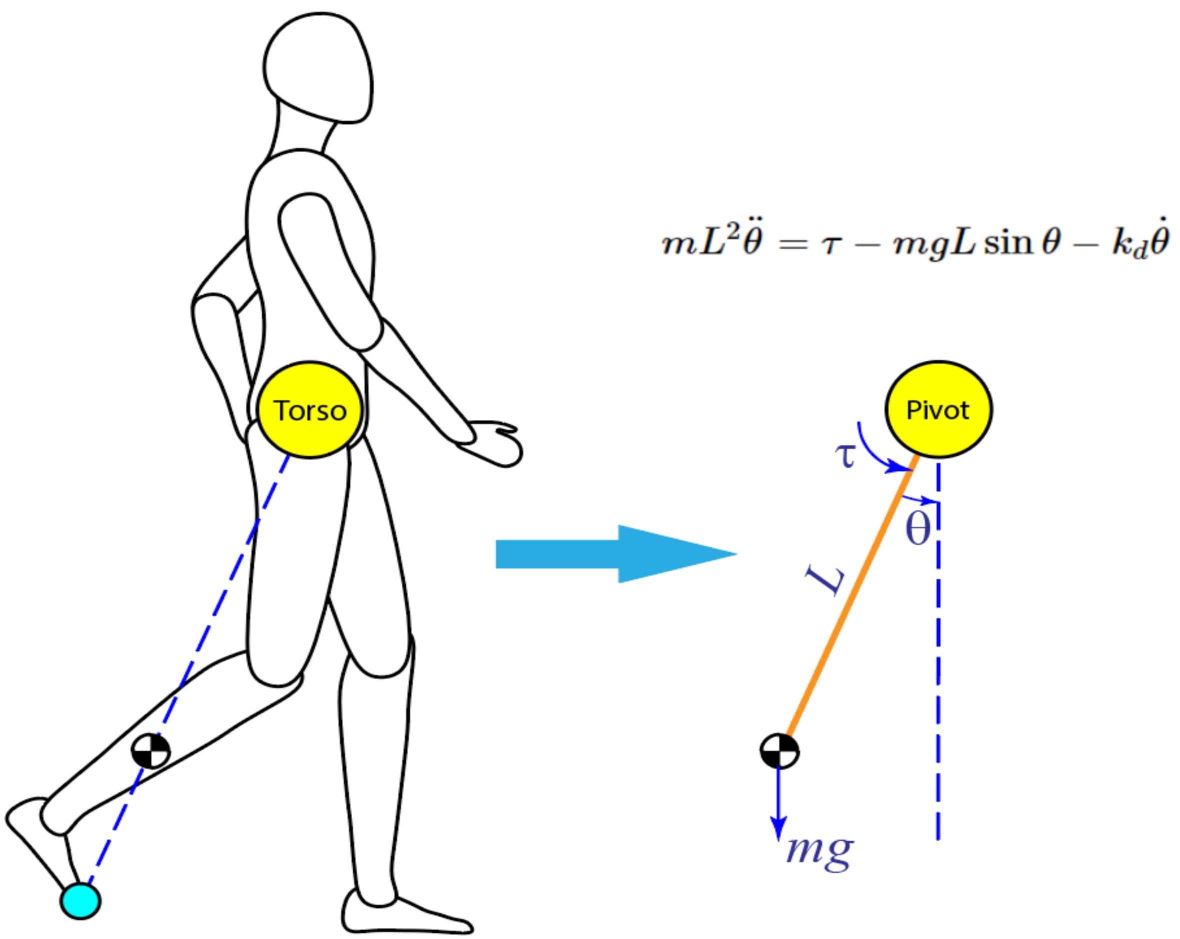
Electrical Engineering Department

Control systems

Task 1
Robotic Leg Modeling and Control

Background

You have just learned how to automatically linearize a Simulink model and tune a PID controller for the linearized plant. In this project, you will obtain an LTI estimation of a robotic leg and design a PID controller to control its motion. As shown below, the robotic leg can be modeled as a simple pendulum, pivoting at the hip joint.



Walking Robot

Simplified Pendulum Model

A motor torque is applied at the hip joint as the control input, and the leg rotation angle, θ, is the control output. There is damping at the hip joint. The equation of motion for the simplified pendulum model is:

$$mL^2\ddot{\theta}=\tau-mgL\sin\theta-k_d\dot{\theta}$$

Required:

you will design a controller for the robotic leg so it has a less than 0.8 second settling time and less than 10% overshoot in the step response. The project includes two parts. In the first part, you will linearize the plant at an operating point and conduct linear analyses to better understand the plant's behavior. In the second part, you will add a PID Controller block to build a feedback control loop and use PID Tuner to tune the PID controller against the design requirements.

Deliverables:

1. The control design onramp using simulink course certificate showing that you have taken the course. (Duplicated certificates will be marked zero for each).

2. Results samples of your work (linearization part, control design part, PID tuning parameters, outputs screen shots, … etc).

3. Use a simulink toolbox to solve the same problem then compare the results.

4. mention another application to emphasize the importance of model linearization and control design.

5. you are required to submit a google drive link of the following:

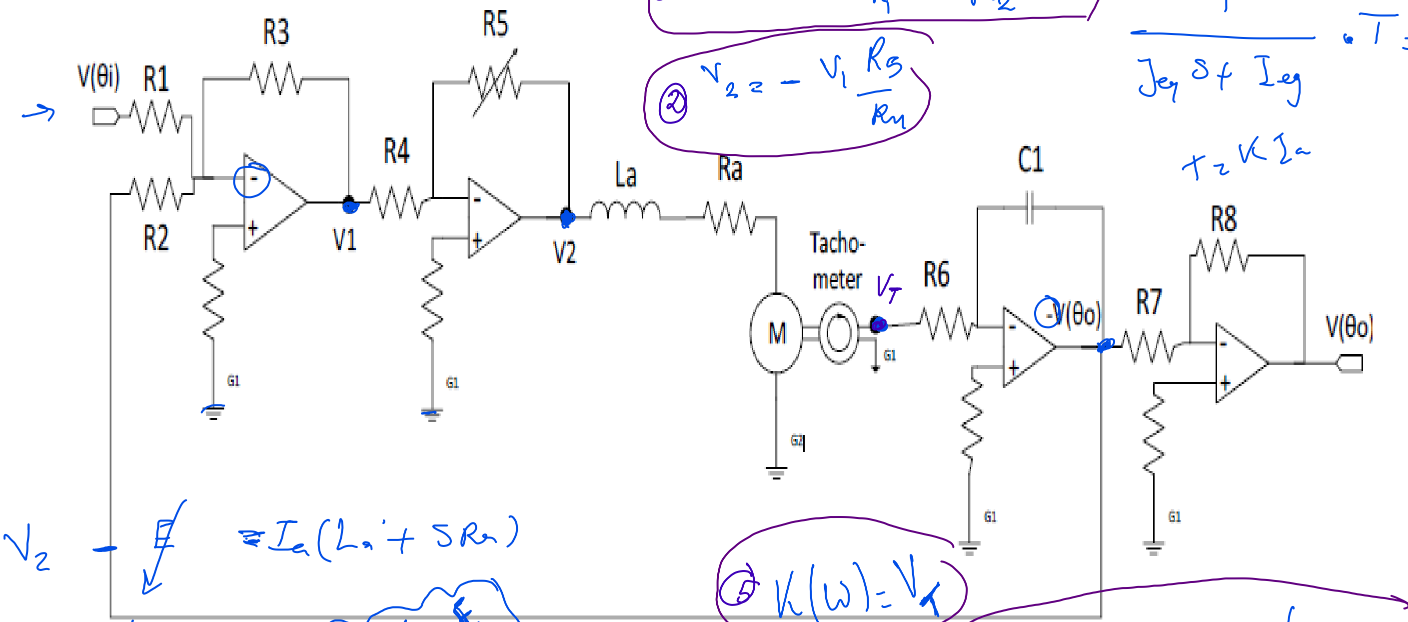
- **a) your certificate.**
- **b) A PDF report of all the deliverable mentioned above.**
- **c) A web view file for your simulink models.**

6.Duplicated work will be marked zero for each.

Alexandria University
Faculty of engineering

Task 2
DC motor control

Part I: Position control



① $V_1 = -R_3\left(\frac{V_{\theta i}}{R_1} + \frac{-V(\theta_o)}{R_2}\right)$

$T = K I_a$

$\frac{1}{J_{eq} s + B_{eq}} \cdot T = \omega$

② $V_2 = -V_1 \frac{R_5}{R_4}$

$T = K I_a$

$V_2 - E = I_a(L_a s + R_a)$

$K\omega$

③ $I_a = \frac{V_2 - E}{sL_a + R_a}$

⑤ $K(\omega) = V_T$

④ $\omega = \frac{T}{B_{eq} + sJ_{eq}}$

⑥ $-V(\theta)_o = -\frac{1}{sR_6C_1} V_T$

- R1=R2=R3=R7=R8=1 kΩ
- R4=1 kΩ
- La=10 mH & Ra=2 Ω. (DC motor equivalent inductance and resistance)
- Km=Kb=1.1 (DC motor constants)
- Jeq=0.1 & Beq=0.001 (equivalent inertia and friction of the system)
- Kt=2 (tacho-meter constant)
- R6=100 kΩ & C1=10 μF
- V(θi)= 10 u(t).

⑦ $V(\theta)_o = -V(\theta)_o \cdot \frac{R_8}{R_7}$

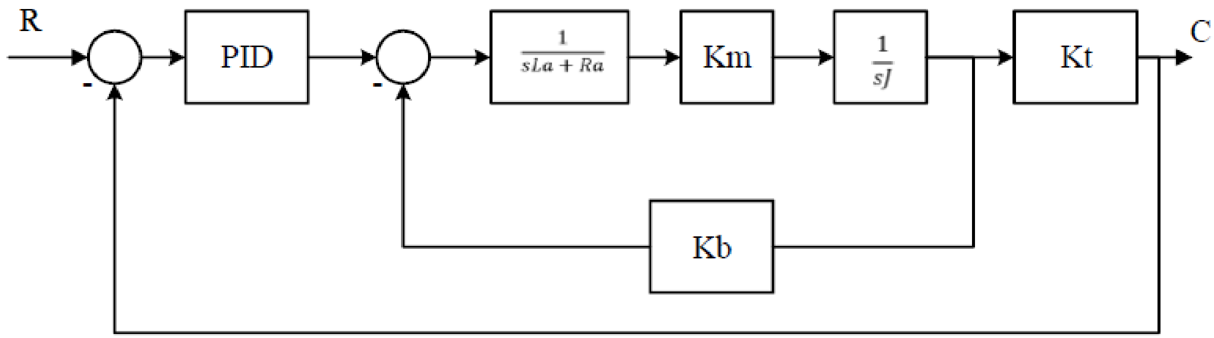
Required:

Draw the complete block diagram for the system using simulink, showing the time response of V(θo) for the following cases

- Case 1: R5=800 Ω Case 4: R5=110 k Ω
- Case 2: R5=1.2 kΩ Case 5: R5=150 k Ω
- Case3: R5=10 kΩ
- Comment for each case.

Alexandria University
Faculty of engineering

Part II: Speed control



- La=100 mH & Ra=1 Ω. (DC motor equivalent inductance and resistance)
- Km=Kb=1.5 (DC motor constants)
- J=0.006 & neglect motor friction (equivalent inertia and friction of the system)
- Kt=0.2 (tacho-meter constant)
- r(t)= u(t).

Required:

Using Simulink, design a PID controller to achieve the following characteristics for the output response.

- Zero steady state error.
- Maximum overshot less than 5%.
- Settling time less than 10 ms.
- Rising time less than 5 ms.

Deliverables:

1. Results samples of your work (position control, speed control, outputs screen shots, … etc).

2. you are required to submit a google drive link of the following:

b) A PDF report of all the deliverable mentioned above.

c) A web view file for your simulink models.

4. Duplicated work will be marked zero for each.

Alexandria University
Faculty of engineering

Task 3

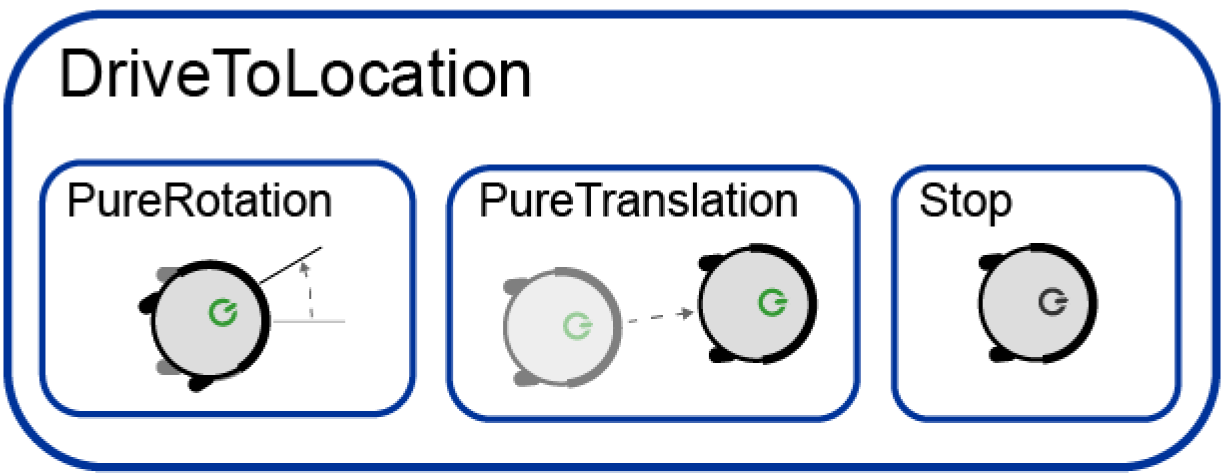
Robot Vacuum Driving Modes

Background:

As you've just learned, using hierarchy in your Stateflow models helps you group states with common functionality. In this project, you will use hierarchy to model something similar to the driving logic of the robotic vacuum that you previously modeled. In that project, the robot could drive in two modes: docking or vacuuming. The robot's instructions for each of these modes were:

SeekDock – drive to the dock and then stop.
Vacuum – drive to the corner of the room and then follow a zig-zag pattern.

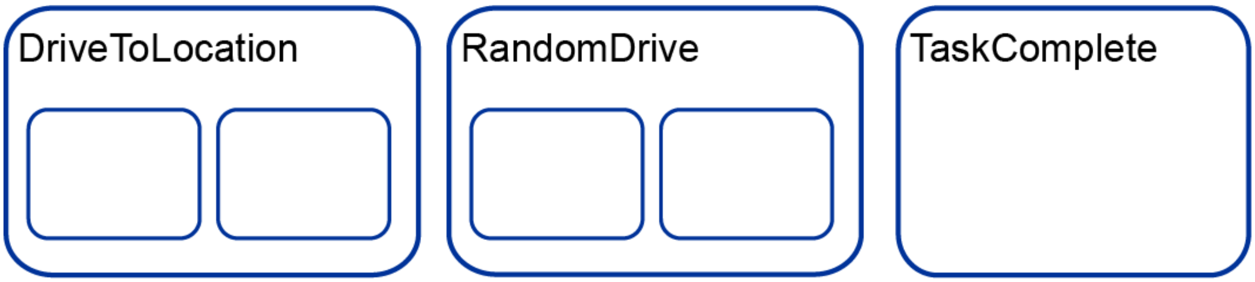
Thus, the robot required two driving behaviors: drive to a location and follow a zig-zag pattern. These can be implemented as the top-level states in a hierarchy. The substates for each are the particular states required to implement each behavior. For example, when driving to a location, the robot's motion can be broken down into: orient using pure rotation, drive using pure translation, and stop.



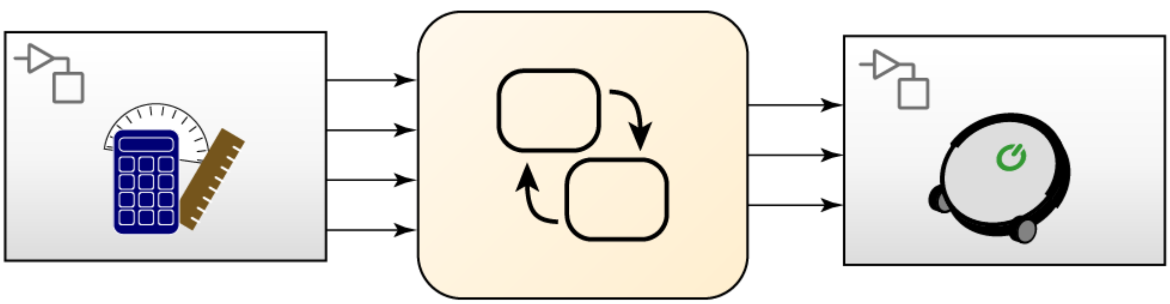
Required:

you will implement a driving pattern in which the robot can either dock or drive a random pattern around the room. (Many early or discount models of robotic vacuums follow a random pattern, which is simple to implement, although less efficient than using a map).

Thus, DriveToLocation and RandomDrive are two superstates. Your Stateflow chart will also have a TaskComplete state in which no further driving commands are sent.



As with the Robot Vacuum Supervisory Control project, the driving dynamics of the robotic vacuum have been built in Simulink. You will connect simulated sensor data and other information from this model to your Stateflow chart in order to set the desired velocity and angular velocity commands to be sent to the robot.



You will build this model in two parts. In Part 1, you will define the high-level structure of the driving mode logic. Then, in Part 2, you will fill in the implementation details and connect the Stateflow chart to Simulink.

Alexandria University
Faculty of engineering

Deliverables:

**1. The state flow onramp certificate showing that you have taken the course.
(Duplicated certificates will be marked zero for each).**

2. Results samples of your work (high level architecture, Implementation details, outputs screen shots, … etc).

3. mention another application to emphasize the importance of stateflow as a modeling tool.

4. you are required to submit a google drive link of the following:

a) your certificate.
b) A PDF report of all the deliverable mentioned above.
c) A web view file for your simulink models.

5. Duplicated work will be marked zero for each.