MCE/EEC 647/747
Homework 6 and Project 1 Part I- Spring 2019
Due 5/2/19

1: Robust Passivity-Based Tracking Control of the WAM Model

The objective is to implement and tune an RPBC tracking controller for joints 1,2 and 4 of the WAM robot by simulation, with joint 3 fixed at zero. Procedure:

1. Use the minimal parameter vector and regressor identified in HW5. A complete set of parameterized matrices, the regressor and the parameter vector are contained in m-file `WAM124Ytheta.m` attached to this homework.
   The inertial parameters of the diagonal elements of M have to be adjusted to include the reflected inertia of the motors and gears. This has been already done in the m-file. Also, since the model of the previous homework did not consider friction, the following columns and corresponding parameters must be appended to the regressor:
   $$Y_f = \begin{bmatrix} \dot{q}_1 & \text{sign}\ (\dot{q}_1) & 0 & 0 \\ 0 & 0 & \dot{q}_2 & 0 \\ 0 & 0 & 0 & \dot{q}_4 \end{bmatrix}$$
2. Use the nominal parameter values declared in the supplied m-file to obtain the numerical value of the nominal Θ (25 entries).
3. Initially tune the controller to track the following reference trajectories:
   $$\begin{aligned} q_1^d(t) &= \sin(wt) \\ q_2^d(t) &= -0.5 + \sin(wt) \\ q_4^d(t) &= 0.75 \sin(wt) \end{aligned}$$
   Use initial conditions that match the desired trajectories at $t = 0$. Once the controller is running, test for convergence for $q(0) \neq q^d(0)$.
4. Apply a random perturbation to the nominal Θ, leaving the value used by the controller the same. Re-tune and observe the increment in chattering levels. Document the results.

2: Optimal Trajectories for Parameter Estimation

The goal is to find reference trajectories that optimize the numerical conditioning of the regressor for identification purposes, as discussed in class. Use the following objective function:
$$J = \lambda_1 \sum_{i=1}^{N} Y(q(t_i), \dot{q}(t_i), \ddot{q}(t_i)) + \frac{\lambda_2}{\sum_{i=1}^{N} \underline{\sigma}(Y(q(t_i), \dot{q}(t_i), \ddot{q}(t_i)))}$$
where N is the number of data points. As constraints, use the maximum and minimum values for the joint positions, velocities and Cartesian velocities shown in Table 1 where $i = 1, 2, 4$, e denotes the end plate position (910 mm along the z_3 axis) and p denotes the

Variable	Min Value	Max Value	Units
q_1	-1	1	rad
q_2	-1	1	rad
q_4	-0.5	2	rad
$\dot{q}_i$	-1	1	rad/s
v_e	-0.5	0.5	m/s
v_p	-0.5	0.5	m/s

Table 1: Position and Velocity Constraints

elbow (origin of frame 3).
Optimize over the coefficients of Fourier expansions for q_i, $i = 1, 2, 4$.

$$q_i(t) = \sum_{j=1}^{N_f} \frac{a_{ij}\sin(wjt)}{wj} - \frac{b_{ij}\cos(wjt)}{wj}$$

Use $N_f = 4$. With this, you have a total of 12 a coefficients and 12 b coefficients.

Procedure:

- Code a function that returns the extended regressor (including the 4 columns associated with friction) given N-by-1 vectors with $q_i(t)$, $\dot{q}_i(t)$ and $\ddot{q}_i(t)$. The returned regressor will have $3N$ rows and as many columns as the parameter vector (it's a stack of regressors evaluated at all time instants).

- Code for the objective function. It receives a candidate vector $[a\ b]$, parameters w, N, λ_1, λ_2 and a time vector. The time vector to be used is [0:0.1:10]. The function evaluates $q_i(t)$, $\dot{q}_i(t)$ and $\ddot{q}_i(t)$ and the 3-row regressor at each time point.
  The condition number to be used is the sum of the condition numbers of every time instance of the 3-row regressor. The same for the minimum singuar value.

- Code for the constraints function. It receives the same arguments as the objective function. It evaluates $q_i(t)$, $\dot{q}_i(t)$ and $\ddot{q}_i(t)$. From there, it calculates v_e and v_p at each time point. With this information, it finds the minimum and maximum values from the data a forms an inequality constraint output (negative when constraints are satisfied).

- Call `fmincon` with the following options:
  ```
  options=optimptions('fmincon','Display','iter','MaxFunctionEvaluations',10000,
  'OptimalityTolerance',1e-11,'StepTolerance',1e-11,'MaxIterations',10000);
  ```
  Use a random guess for $[a\ b]$ and take $\lambda_1 = 1$ and $\lambda_2 = 10$.

- Show the resulting trajectories and verify that they comply with the constraints.

- Tune the controller to follow these trajectories.