AE353: Design Problem 02

Robotic Controllability

March 8, 2019

1 Goal

The code, `DesignProblem02` simulates a "gravity-assisted underactuated robot arm." This robot arm has two joints: one controlled, one free. Optical encoders measure joint angles (q) and velocities (v). The goal is to make the second joint angle track a piecewise-constant reference trajectory.

1.1 Requirements

A *requirement* is a property that the system must have in order to solve the problem.

The second joint angle, q_2 shall follow the piece-wise function henceforth defined as α and have an accuracy of $\pm 2°$ with convergence within 3 seconds.

$$\alpha_i = \begin{cases} q_2 = 5^\circ & t \leq 5 \\ q_2 = 10^\circ & 5 < t \leq 10 \\ q_2 = 15^\circ & 10 < t \leq 15 \\ q_2 = 0^\circ & 15 < t \end{cases} \tag{1}$$

1.2 Verification

A *verification* is a test that is performed to verify the system meets the requirements set in section (1.1).

`DesignProblem02('Controller', 'datafile', 'data.mat')` will run and the data points from `'data.mat'` will calculate the error between the second joint angle, q_2, and reference value, α, at each time-step value of $\Delta t = 0.25[s]$ by the following computation:

$$E(t) = |q_2(t) - \alpha|$$

If $E \leq 2°$ with an accuracy of over 70%, then the requirement is satisfied.

2 State Space Model

The motion of the robot is governed by ordinary differential equations with the form

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + N(q,\dot{q}) = \tau \text{ where,} \tag{2}$$

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \dot{q} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \tau = \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix}$$

q is a matrix of joint angles, $\dot{q}$ is a matrix of joint velocities, τ is a matrix of applied torques, and $M(q), C(q,\dot{q}), N(q,\dot{q})$, are matrix-valued functions of q and/or $\dot{q}$. These functions depend on a number of parameters. Before linearizing, it is important to solve for $\ddot{q}$:

$$\ddot{q} = \frac{\tau - C(q,\dot{q})\dot{q} - N(q,\dot{q})}{M(q)} \tag{3}$$

2.1 Linearization

Now the `Jacobian()` of $\ddot{q}$ and $\dot{q}$ will be utilized for A and $\frac{\tau}{M(q)}$ to find B (for detailed code on calculation of pertinent linearization values, please refer to this [script]):

$$\mathbf{A} = [\ddot{q}]_{(\dot{q})} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 5.5465 & -61.2335 & -1.5762 & -0.8327 \\ -1.9024 & -10.7824 & -0.2776 & -0.3690 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 3.1525 \\ 0.5551 \end{bmatrix}$$

$$u = -K \cdot x$$

2.2 Controllability

The Q matrix was a scaled `5*eye(size(A))` matrix and R was selected to enable the controller to meet the goals in a precise quantity of time. With these values, the controller may now be designed and implemented for the *State Space Model.* To confirm controllability, the following test will be implemented on the linearized system: `rank(W)` $= 4$ in `FindingValues.m`.

2.3 Reference Tracking

In order to enable reference tracking, a component must be added the input:

$$u = -K \cdot x + \mathbf{k_{ref}} \cdot \mathbf{r}, \text{ for } k_{ref} = \texttt{inv(-C*inv(A-B*K)*B)} \text{ and } K = \texttt{lqr(A,B,Q,R)}$$

2.4 Disturbance Rejection

If disturbance is introduced to the simulation, Integral action may be utilized to oppose the effects of unknown disturbance.

$$u = -K \cdot x + k_{ref} \cdot r + d - \mathbf{k_{int}} \cdot \mathbf{v} \text{ for } v = \int_0^t (q_2(\tau) - r(\tau))d\tau$$

3 Control Design

The goal, as stated in requirements, is to make the arm converge asymptotically, tracing points at $q_2 = \alpha$ every 5 seconds and reach convergence within 3 seconds. In order to implement this, the A, B, C, D values must be structured in the following State Space model:

$$\dot{x} = Ax + Bu, \quad y = Cx + Du, \quad u = -Kx + k_{ref}r + d - k_{int}v$$

with

$$x = \begin{bmatrix} q_1 \\ q_2 \\ v_1 \\ v_2 \end{bmatrix}$$

The controller will be implemented by designating the values for r, K, and k_{ref} in order to track and pursue the desired q_2 angle at the desired time, mitigate the settling time, and optimize the efficiency respectively. In section (2.3), the ***L**inear **Q**uadratic **R**egulator* function was utilized to find the optimal value for K. This will enable the controller to reach the desired joint angles in the least amount of time/cost. Two trials will be run, one without disturbance and one accounting for some unknown disturbance. In the codes, colleagues are provided the option to select *1* or *2*:

- 1 utilizes `Controller1.m`, a controller which does not include Integral Action.
- 2 utilizes `Controller2.m`, a controller which attempts to use Integral Action.

Figure 1.$_{(Undisturbed)}$ and **Figure 2.**$_{(Disturbed)}$ display the results of implementing the controller on the swinging arm simulation.

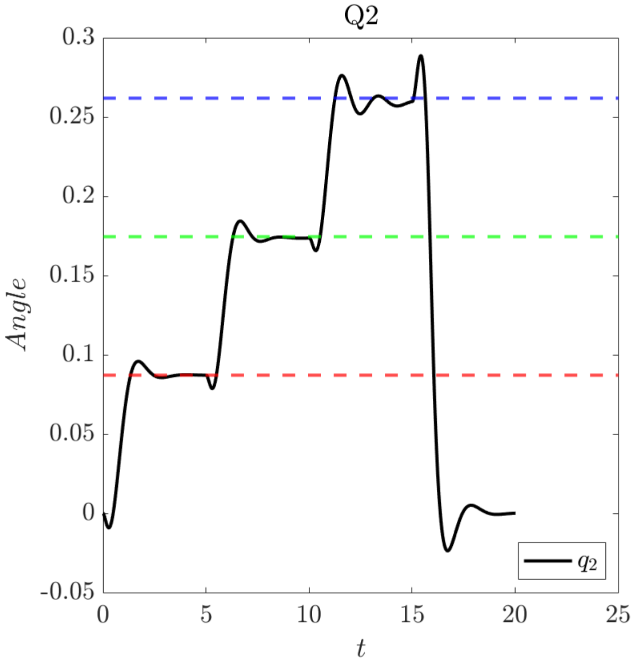

Figure 1

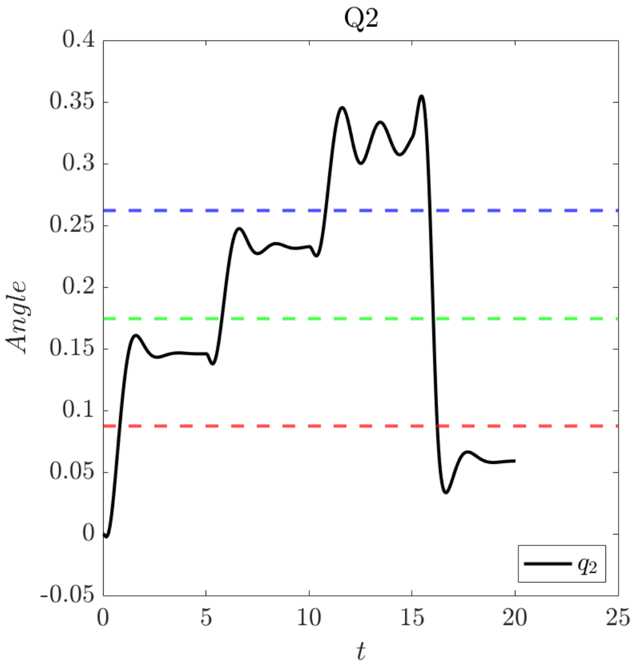

Figure 2

4 Conclusions

This simulation observes one rotational motor and one free joint on a "gravity-assisted under-actuated robot arm." The requirements for this controller test were defined as such from section (1.1): *The second joint angle, q_2 shall follow the piece-wise function as defined as α and maintain an accuracy of* ±2° *and converge within 5 seconds.* To test this requirement, a verification was implemented in the `Test.m` script which observed the absolute value of the difference between $q_2(t)$ and α. If the error term is of the accuracy of 70% or greater, the controller would be declared successful in this requirement. After linearizing the system, various tests were conducted, utilizing `eig(A)` and `rank(W)` to determine asymptotic stability of the open system and controllability. It was interesting to realize that the system was not truly asymptotically stable independently, but maintained controllability. Two trials were conducted, one involving no disturbance and the other simulating some unknown disturbance. In order to attempt to compensate for said disturbance, integral action was utilized in the input, i.e. $u = -Kx + k_{ref}r + d - k_{int}v$ for $v = \int_0^t (q_2\tau - r(\tau))d\tau$. After selecting the desired controller setup in the `Test.m` file, graphs such as those shown as **Figure 1.** and **Figure 2.** were produced. For the case of no disturbance, the controller changed the second joint angle according to α with a settling time of roughly 2 seconds and maintained accuracy of within 0.1°; the verification portion of the script returned an accuracy of approximately 79.7% and $\therefore$ passed the verification process. The disturbance trial, on the other hand, did not have much success and repeatedly contained more than 80% error, indicating the derivation for v was not properly functional. Ultimately, the requirements were satisfied by test of the verification for the undisturbed trial by utilizing optimized state space model, $u = -Kx$, and reference tracking for a piece-wise, α, function, $u = -k_{ref}r$.