

MCE/EEC 647/747: Robot Dynamics and Control
Term Project - Fall 2008
Kinematics, Dynamics, Control and Simulation of RP Planar Arm

Due: Friday, December 12 by 5 PM

Consider a 2-link RP planar manipulator. Assume that the links have lengths a_1 , and a_2 and moments of inertia I_{1z} and I_{2y} for planar rotation about the center of mass. Suppose the center of mass is at the center of each link. It is recommended to use the angle of link 1 and the distance between the centers of mass of link 1 and link 2 as joint variables q_1 and q_2 .

Kinematics and Dynamics

1. Set up appropriate coordinate systems and find the D-H parameter table.
2. Find the T matrices relating each link-attached frame to the world frame, according to your choice of origin points.
3. Find the two velocity Jacobians and the two angular velocity Jacobians relative to each center of mass.
4. Find the $D(q)$, $C(q, \dot{q})$ and $g(q)$ matrices.
5. Set up a Matlab/Simulink simulation implementing the dynamic equation $D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau$. Please set up a parametric simulation system, that is, all parameters (masses, lengths, etc.) should be symbolic. Set up a separate m-file to load values for the parameters.
6. Try out your simulation by setting all parameters to 1 and applying $\tau = [1 \ 1]^T$. Simulate for 1 second and plot all joint positions and velocities against time.
7. Consult with the instructor to see whether you're on the right track and ready for the next stage (control analysis/design).

Linear Parameterization

1. Let the masses of the links be m_1 and m_2 , and let the angle of link be q_1 and the distance between centers of mass be q_2 . Show that the following regressor and parameter vectors satisfy the dynamic equations derived above:

$$Y(q, \dot{q}, \ddot{q}) = \begin{bmatrix} \ddot{q}_1 & q_2^2 \ddot{q}_1 + 2q_2 \dot{q}_1 \dot{q}_2 + g q_2 \cos(q_1) & 2\dot{q}_1 \dot{q}_2 + 2q_2 \ddot{q}_1 + g \cos(q_1) & g \cos(q_1) \\ 0 & \ddot{q}_2 - q_2 \dot{q}_1^2 + g \sin(q_1) & -\dot{q}_1^2 & 0 \end{bmatrix} \quad (1)$$

$$\Theta = \begin{bmatrix} \frac{1}{4}m_1 a_1^2 + I_{1z} + \frac{1}{4}m_2 a_1^2 + I_{2y} \\ m_2 \\ \frac{1}{2}m_2 a_1 \\ \frac{1}{2}m_1 a_1 \end{bmatrix} \quad (2)$$

2. Find the regressor $Y(a, v, q, \dot{q})$ corresponding to the term $\hat{M}a + \hat{C}v + \hat{g}$
3. Take $g = 9.8 \text{ m/s}^2$ and the following nominal values for the parameters:

$$\begin{aligned}
a_1 &= 0.6 \text{ m} \\
a_2 &= 0.4 \text{ m} \\
m_1 &= 10 \text{ kg} \\
m_2 &= 8 \text{ kg} \\
I_{1z} &= 26 \text{ kg-m}^2 \\
I_{2y} &= 12 \text{ kg-m}^2
\end{aligned}$$

Let these parameters generate the nominal parameter vector Θ_0 . Supposing that each basic parameter (m_1 , m_2 , a_1 , a_2 , I_{1z} and I_{2y}) can be off by p percent, determine the maximum error bound for the parameter vector. That is, find $\rho(p)$ so that

$$\|\Theta - \Theta_0\| \leq \rho(p)$$

Do this for $p=10$ percent, $p=25$ percent and $p=80$ percent by taking the worst-case.

Control Design Simulation Setup

The desired joint trajectories are as follows:

$$\begin{aligned}
q_1^d(t) &= 10t \\
q_2^d(t) &= 1 + \cos(30t)
\end{aligned}$$

You must pre-compute the desired velocities and accelerations to be used directly in the control algorithms (avoid online differentiation). Build a simulation using Matlab/Simulink to test each one of the following cases:

1. **Robust passivity-based control:** Generate a “true” set of parameters using $p=10$ percent (random perturbation). Choose a suitable value of deadzone ε and gain matrices Λ and K . Use the true parameters in the plant and the nominal ones in the control law. Plot $\tilde{q}_1(t)$ and $\tilde{q}_2(t)$. Then plot the y position vs. the x position of the end-effector corresponding to $q_1^d(t)$ and $q_2^d(t)$ with a red dashed line. On the same plot, plot the actual y vs. x (using $q_1(t)$ and $q_2(t)$) with a solid line of another color. Finally plot the two components of the control input against time.
2. Repeat the above for $p=25$ percent. Re-adjust the gains and ε if needed.
3. **Adaptive passivity-based control:** Generate a “true” set of parameters using $p=80$ percent (random perturbation). Choose suitable gain matrices Λ , K and Γ . Use the true parameters in the plant and the nominal ones as initial parameter guesses in the control law. Run the adaptive system and tune the gain matrices to obtain good parameter and error convergence. Plot each adapted parameter vs. time. Plot $\tilde{q}_1(t)$ and $\tilde{q}_2(t)$. Then plot the y position vs. the x position of the end-effector corresponding to $q_1^d(t)$ and $q_2^d(t)$ with a red dashed line. On the same plot, plot the actual y vs. x (using $q_1(t)$ and $q_2(t)$) with a solid line of another color. Finally plot the two components of the control input against time. Did the parameters converge to their true values?

How to turn in the exam:

All hand calculations must be explained. Please scan any handwritten sheets. Email the instructor, attaching the hand calculations and m-file(s). Although the use of packages like *Robotica* is allowed for verification purposes, the Jacobians and all required transformation matrices must be shown in relationship to the chosen coordinate systems. Students are encouraged to use symbolic engines (Matlab, Python, Mathematica, Maple, etc.) to assist with symbolic matrix multiplications and simplification, although it is not necessary to use these packages to carry the operations.