Control in Robotics
Part 2: Motion Control

Mohammad Motaharifar

Department of Electrical Engineering
University of Isfahan

Spring 2022

Table of contents

Introduction

- The simplest control approach for robots is to derive a control law for each joint of a manipulator based on a single-input/single-output model.
- Coupling effects among the joints are regarded as disturbances to the individual systems.
- In reality, the dynamic equations of a robot manipulator form a complex, nonlinear, and multivariable system.
- Here, we treat the robot control problem in the context of nonlinear, multivariable control.
- This approach allows us to provide more rigorous analysis of the performance of control systems, and also allows us to design robust and adaptive nonlinear control laws that guarantee stability and tracking of arbitrary trajectories.

Introduction

- Matrix Form of robot dynamics by considering the effect of the actuators

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + B\dot{q} + g(q) = u \tag{1}$$

- B is the damping matrix.
- We will take B $= 0$ for simplicity.
- We leave it as an exercise for the students to show that the properties of passivity, skew-symmetry, bounds on the inertia matrix and linearity in the parameters continue to hold for the system.

PD Control

- An independent joint PD-control scheme for set-point control of rigid robots can be written in vector form as

$$u = K_P \tilde{q} - K_D \dot{q} \tag{2}$$

- $\tilde{q} = q_d - q$ is the error between the desired joint displacements q_d and the actual joint displacements q, and K_P ,K_D are diagonal matrices of (positive) proportional and derivative gains, respectively.
- We first show that, in the absence of gravity, that is, if g is zero, the PD control law achieves asymptotic stability of the desired joint positions.

PD Control

Theorem

The PD control law achieves asymptotic tracking of the desired joint positions, provided that the gravity vector g is zero.

- *P*roof: Consider the Lyapunov function candidate

$$V = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{2}\tilde{q}^T K_P \tilde{q}$$

- The first term in the above equation is the kinetic energy of the robot and the second term accounts for the proportional feedback $K_P\tilde{q}$.
- Note that V represents the total kinetic energy that would result if the joint actuators were to be replaced by springs with stiffnesses represented by K_P and with equilibrium positions at q_d.

PD Control

- Thus V is a positive function except at the "goal" $q = q_d$, $\dot{q} = 0$, at which point V is zero.
- The idea is to show that along any motion of the robot, the function V is decreasing to zero.
- This will imply that the robot is moving toward the desired goal configuration.
- The time derivative of V is given by

$$\dot{V} = \dot{q}^T M(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{M}(q)\dot{q} - \dot{q}^T K_P \tilde{q}$$

PD Control

- Solving for $M(q)\ddot{q}$ in (1) with $g(q) = 0$ using the skew symmetric property of $\dot{M}(q) - 2C(q, \dot{q})$, and using (2):

$$\begin{aligned}\dot{V} &= \dot{q}^T(u - C(q,\dot{q})\dot{q}) + \frac{1}{2}\dot{q}^T\dot{M}(q)\dot{q} - \dot{q}^T K_P\tilde{q}\\ &= \dot{q}^T(u - K_P\tilde{q}) + \frac{1}{2}\dot{q}^T(\dot{M}(q) - 2C(q,\dot{q}))\dot{q}\\ &= \dot{q}^T(u - K_P\tilde{q})\\ &= -\dot{q}^T K_D\dot{q} \le 0\end{aligned}$$

- The above analysis shows that V is decreasing as long as $\dot{q}$ is not zero.
- This, by itself is not enough to prove the desired result since it is conceivable that the manipulator can reach a position where $\dot{q} = 0$ but $q \neq q_d$.

PD Control

- To show that this cannot happen we can use LaSalle's Theorem.
- Suppose $\dot{V} \equiv 0$. Then $\dot{V} = -\dot{q}^T K_D \dot{q} \leq 0$ implies that $\dot{q} \equiv 0$ and hence $\ddot{q} \equiv 0$. From the equations of motion with PD-control

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = K_P \tilde{q} - K_D \dot{q}$$

  must then have $0 = K_P \tilde{q}$ which implies that $\tilde{q} = 0$, $\dot{q} = 0$.
- LaSalles Theorem then implies that the equilibrium is asymptotically stable.
- In case there are gravitational terms present in the dynamic equation (1), we have

$$\dot{V} = \dot{q}^T (u - K_P \tilde{q} - g(q))$$

PD Control

- The presence of the gravitational term in the above equation means that PD control alone cannot guarantee asymptotic tracking.
- In practice there will be a steady state error or offset.
- Assuming that the closed loop system is stable the robot configuration q that is achieved will satisfy

$$K_P \tilde{q} = g(q)$$

- The steady state error can be reduced by increasing the position gain K_P .
- In order to remove this steady state error we can modify the PD control law as (PD+gravity control law)

$$u = K_P \tilde{q} - K_D \dot{q} + g(q)$$

- The term g is canceled and the result is the same as before.

Inverse Dynamics

- We now consider the application of more complex nonlinear control techniques for trajectory tracking of rigid manipulators. Consider again the dynamic equations of an n-link robot in matrix form

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u \tag{3}$$

- The idea of inverse dynamics is to seek a nonlinear feedback control law

$$u = f(q, \dot{q}, t)$$

  which, when substituted into (3), results in a linear closed loop system.
- For general nonlinear systems such a control law may be quite difficult or impossible to find.

Inverse Dynamics

- In the case of the manipulator dynamic equations (3), however, the problem is actually easy.
- By inspecting (3) we see that if we choose the control u according to the equation

$$u = M(q)a_q + C(q, \dot{q})\dot{q} + g(q) \tag{4}$$

- Then, since the inertia matrix M is invertible, the combined system reduces to

$$\ddot{q} = a_q$$

- The term a_q represents a new input to the system which is yet to be chosen.
- The nonlinear control law (4) is called the inverse dynamics control, and results to a linear and decoupled system.

Inverse Dynamics

- By defining diagonal gain matrices K_0 and K_1, we choose

$$a_q = -K_P q - K_D \dot{q} + r$$

- The closed loop system is then the linear system

$$\ddot{q} + K_D \dot{q} + K_P q = r$$

- Now, we choose $r = \ddot{q}_d + K_D \dot{q}_d + K_P q_d$ Then, the tracking error $e = q - q_d$ satisfies

$$\ddot{e} + K_D \dot{e} + K_P e = 0$$

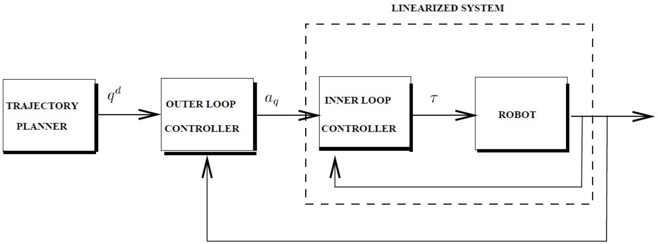
- A simple choice for K_0 and K_1:

$$K_P = \text{diag}(\omega_1^2, \omega_2^2, ..., \omega_n^2)$$
$$K_D = \text{diag}(2\zeta_1\omega_1, 2\zeta_2\omega_2, ..., 2\zeta_n\omega_n)$$

- Then, each joint is a linear second order system with natural frequency ω_i and damping ratio ζ_i.

Inverse Dynamics

- This figure illustrates the notion of inner-loop/outer-loop control.
- By this we mean that the computation of the nonlinear control (4) is performed in an inner loop, with the vectors q, $\dot{q}$, and a_q as its inputs and u as output.
- The outer loop in the system is then the computation of the additional input term a_q.



Robust and Adaptive Inverse Dynamics

- A drawback to the implementation of the inverse dynamics control methodology is the requirement that the parameters of the system be known exactly.
- If the parameters are not known precisely, for example, when the manipulator picks up an unknown load, then the ideal performance of the inverse dynamics controller is no longer guaranteed.
- This section is concerned with robust and adaptive motion control of manipulators.
- The goal of both robust and adaptive control is to maintain performance in terms of stability, tracking error, or other specifications, despite parametric uncertainty, external disturbances, unmodeled dynamics, or other uncertainties.

Robust and Adaptive Inverse Dynamics

- Difference between robust control and adaptive control:
  - A robust controller is a fixed controller, designed to satisfy performance specifications over a given range of uncertainties
  - An adaptive controller incorporates some sort of on-line parameter estimation.
- For example, in a repetitive motion task the tracking errors produced by a fixed robust controller is repetitive, whereas tracking errors produced by an adaptive controller might be expected to decrease over time.
- At the same time, adaptive controllers that perform well in the face of parametric uncertainty may not perform well in the face of unmodeled dynamics.
- An understanding of the tradeoffs involved is therefore important in deciding whether to employ robust or adaptive control design methods in a given situation.

Robust Inverse Dynamics

- The feedback linearization approach relies on exact cancellation of nonlinearities in the robot equations of motion.
- Its practical implementation requires consideration of various sources of uncertainties such as modeling errors, unknown loads, and computation errors.
- Let us return to the Euler-Lagrange equations of motion

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = u$$

- The inverse dynamics control input u is written as

$$u = \hat{M}(q)a_q + \hat{C}(q,\dot{q})\dot{q} + \hat{g}(q) \tag{5}$$

- The notation $\hat{(.)}$ represents the nominal value of $(.)$
- The error $(.) - \hat{(.)}$ indicates our knowledge of the system dynamics.

Robust Inverse Dynamics

- If the control law is substituted into the dynamic equation,

$$\ddot{q} = a_q + \eta(q, \dot{q}, a_q)$$

where

$$\eta = M^{-1}(\tilde{M}a_q + \tilde{C}\dot{q} + \tilde{g})$$

- Next, the outer loop control law is defined as

$$a_q = \ddot{q}_d(t) - K_P\tilde{q} - K_D\dot{\tilde{q}} + \delta a \tag{6}$$

- The tracking error $e = \begin{bmatrix} \tilde{q} & \dot{\tilde{q}} \end{bmatrix}^T = \begin{bmatrix} q - q_d & \dot{q} - \dot{q}_d \end{bmatrix}^T$ has the following dynamics:

$$\dot{e} = Ae + B(\delta a + \eta)$$

where

$$A = \begin{bmatrix} 0 & I \\ -K_p & -K_D \end{bmatrix}, B = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

Robust Inverse Dynamics

- δa is an additional control input that must be designed to overcome the effect of the uncertainty η.
- The basic idea is to compute a timevarying scalar bound, $\rho(e,t)$, on the uncertainty η:

$$\eta \leq \rho(e,t)$$

- Then, δa is selected as

$$\delta a = \begin{cases} -\rho(e,t)\frac{B^T Pe}{\|B^T Pe\|} & : \|B^T Pe\| \neq 0 \\ 0 & : \|B^T Pe\| = 0 \end{cases} \tag{7}$$

- P is a positive definite matrix that satisfies the Lyapunov equation $A^T P + PA = -Q$ with $Q > 0$. Since A is Hurwitz, there alwayse exist such P.

Robust Inverse Dynamics

Theorem

The robust inverse dynamic controller composing the control law (5) with the outer loop (6) and additional term (7) achieves asymptotic stability.

- *Proof*: Consider the Lyapunov function $V = e^T Pe$. Then,

$$\dot{V} = -e^T Qe + 2e^T PB(\delta a + \eta)$$

- Define $w = B^T Pe$ and consider the term $w^T(\delta a + \eta)$. If $w = 0$ this term vanishes and for $w \neq 0$ we have

$$w^T(-\rho \frac{w}{\|w\|} + \eta) \leq -\rho\|w\| + \|w\|\|\eta\| = \|w\|(-\rho + \|\eta\|) \leq 0$$

- Therefore, $\dot{V} \leq -e^T Qe$ which completes the proof.

Robust Inverse Dynamics

- Since the above control term δa is discontinuous on w, solution trajectories on this manifold are not well defined in the usual sense.
- In practice, the discontinuity in the control results in the phenomenon of chattering, as the control switches rapidly across the manifold $w = B^T Pe = 0$.
- One may implement a continuous approximation to the discontinuous control as

$$\delta a = \begin{cases} -\rho(e,t)\frac{B^T Pe}{\|B^T Pe\|} & : \|B^T Pe\| > \epsilon \\ -\frac{\rho(e,t)}{\epsilon} B^T Pe & : \|B^T Pe\| \leq \epsilon \end{cases} \quad (8)$$

- In this case, a solution to the system exists and is uniformly ultimately bounded (u.u.b).

Adaptive Inverse Dynamics

- Consider that the control law (5) but suppose that the parameters are not fixed as in robust approach, but are time varying estimates of the true parameters. Then, set

$$a_q = \ddot{q}_d(t) - K_P\tilde{q} - K_D\dot{\tilde{q}} \tag{9}$$

- It can be shown using the linear paremetrization property that

$$\ddot{\tilde{q}}(t) + K_D\dot{\tilde{q}} + K_P\tilde{q} = \hat{M}^{-1}Y(q,\dot{q},\ddot{q})\tilde{\theta} \tag{10}$$

- Y is the regressor and $\tilde{\theta} = \hat{\theta} - \theta$ and $\hat{\theta}$ is the estimate of θ.
- The state space dynamics of the tracking error

$$\dot{e} = Ae + B\phi\tilde{\theta}$$

where

$$A = \begin{bmatrix} 0 & I \\ -K_p & -K_D \end{bmatrix}, B = \begin{bmatrix} 0 \\ I \end{bmatrix}, \phi = \hat{M}^{-1}Y(q,\dot{q},\ddot{q})$$

Adaptive Inverse Dynamics

- The parameter update law is chosen as

$$\dot{\hat{\theta}} = -\Gamma^{-1}\phi^T B^T P e \tag{11}$$

- Γ is a constant symmetric positive definite gain matrix and P is the symmetric positive definit matrix satisfying the Lyapunove equation $A^T P + PA = -Q$ where Q is a positive definite matrix.
- Notably, the acceleration $\ddot{q}$ is needed in the parameter update law.
- The need for the joint acceleration in the parameter update law presents a serious challenge to its implementation.

Adaptive Inverse Dynamics

Theorem

The adaptive inverse dynamic controller composing the control law (5) with the outer loop (9) and parameter update law (7) achieves asymptotic stability.

- *P*roof: Consider the Lyapunove function candidate

$$V = e^T P e + \frac{1}{2}\tilde{\theta}^T P \tilde{\theta} \tag{12}$$

- Then, $\dot{V}$ is calculated as

$$\dot{V} = -e^T Q e + \tilde{\theta}(\phi^T B^T P e + \Gamma\dot{\hat{\theta}})$$

- Then, using the parameter update law we have $\dot{V} = -e^T Q e$ meaning that the position tracking error is asymptotically stable and the parameter estimation error is bounded.

Passivity-based Control

- The passivity based control is introduced here. In contrast to the inverse dynamics approach, it does not cancel the system nonlinearities and does not lead to a linear closed loop system.
- However, the passivity based method have other advantages with respect to robust and adaptive control.
- Consider the Euler-Lagrange equations as

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = u$$

- The control law is chosen as

$$u = M(q)a + C(q, \dot{q})v + g(q) + Kr \tag{13}$$

where

$$\begin{aligned} v &= \dot{q}_d - \Lambda\tilde{q} \\ a &= \dot{v} = \ddot{q}_d - \Lambda\dot{\tilde{q}} \\ r &= \dot{q} - v = \dot{\tilde{q}}_d + \Lambda\tilde{q} \end{aligned}$$

Passivity-based Control

- K and Λ are constant positive gan matrices.
- After substituting the control law (13) into the system dynamics, we have the following closed loop system:

$$M(q)\dot{r} + C(q,\dot{q})r + Kr = 0 \tag{14}$$

- Note that, in contrast to the inverse dynamics control approach, the above closed loop system is still a coupled nonlinear system. Thus, asymptotic stability is not abvious.

Theorem

The passivity-based controller(13) achieves asymptotic stability.

- *P*roof: Consider the Lyapunove function candidate:

$$V = \frac{1}{2} r^T M(q) r + \tilde{q}^T \Lambda K \tilde{q}$$

Passivity-based Control

- Calculating $\dot{V}$ yields

$$\begin{aligned}\dot{V} &= r^T M\dot{r} + \frac{1}{2}r^T\dot{M}r + 2\tilde{q}\Lambda K\dot{\tilde{q}} \\ &= -r^T Kr + 2\tilde{q}^T\Lambda K\dot{\tilde{q}} + \frac{1}{2}r^T(\dot{M} - 2C)r\end{aligned}$$

- Using the skew symmetry property and the definition of r:

$$\dot{V} = -\tilde{q}^T\Lambda^T K\Lambda\tilde{q} - \dot{\tilde{q}}^T K\dot{\tilde{q}} = -e^T Qe$$

where

$$Q = \begin{bmatrix}\Lambda^T K\Lambda & 0 \\ 0 & K\end{bmatrix}$$

- Therefore, the equilibrium $e = 0$ in error space is globally asymptotically stable.

Passivity-based Robust Control

- The passivity based robust control is easier to implement in terms of uncertainty bounds. Its control law is:

$$u = \hat{M}(q)a + \hat{C}(q, \dot{q})v + \hat{g}(q) + Kr \tag{15}$$

- Using linear parameterization, the above control law becomes

$$u = Y(q, \dot{q}, v, a)\hat{\theta} - Kr \tag{16}$$

- After combination of the above control law with the dynamic equation of the robot, we have

$$M(q)\dot{r} + C(q, \dot{q})r + Kr = Y(\hat{\theta} - \theta)$$

- The term $\hat{\theta}$ is choosen as

$$\hat{\theta} = \theta_0 + \delta\theta$$

- θ_0 is a fixed nominal parameter vector and $\delta\theta$ is an additional control term.

Passivity-based Robust Control

- Then,

$$M(q)\dot{r} + C(q,\dot{q})r + Kr = Y(\tilde{\theta} + \delta\theta)$$

- The vector $\tilde{\theta} = \theta_0 - \theta$ represents parametric uncertainty and it is bounded as

$$\|\tilde{\theta}\| = \|\theta - \theta_0\| \leq \rho$$

- The term $\delta\theta$ is designed as

$$\delta\theta = \begin{cases} -\rho\frac{Y^T r}{\|Y^T r\|} & : \|Y^T r\| > \epsilon \\ -\frac{\rho}{\epsilon} Y^T r & : \|Y^T r\| \leq \epsilon \end{cases}$$

- Using the same Lyapunov function candidate from the stability analysis of the basic passivity based control, the uniform ultimate boundedness of the tracking error is shown.

Passivity-based Adaptive Control

- In the adaptive approach the vector $\hat{\theta}$ in (16) is taken to be a time-varying estimate of the true parameter vector.
- Combining the control law (15) with the dynamic equation of the robot yields

$$M(q)\dot{r} + C(q,\dot{q})r + Kr = Y\tilde{\theta}$$

- $\hat{\theta}$ is computed as

$$\dot{\hat{\theta}} = -\Gamma^{-1}Y^T(q,\dot{q},v,a)r \tag{17}$$

Passivity-based Adaptive Control

Theorem

The passivity-based adaptive control scheme by control law (15) and adaptation law (17) achieves asymptotic stability of the tracking error and boundedness of the parameter estimation error.

- *P*roof: Consider the Lyapunove function candidate:

$$V = \frac{1}{2} r^T M(q) r + \tilde{q}^T \Lambda K \tilde{q} + \frac{1}{2} \tilde{\theta}^T \Gamma \tilde{\theta}$$

- Calculating $\dot{V}$ yields

$$\dot{V} = -\tilde{q}^T \Lambda^T K \Lambda \tilde{q} - \dot{\tilde{q}}^T K \dot{\tilde{q}} + \tilde{\theta}^T (\Gamma \dot{\hat{\theta}} + Y^T r)$$

- Substituting the expression for $\dot{\hat{\theta}}$ yileds

$$\dot{V} = -\tilde{q}^T \Lambda^T K \Lambda \tilde{q} - \dot{\tilde{q}}^T K \dot{\tilde{q}} = -e^T Q e \leq 0$$

Passivity-based Adaptive Control

- Note that we have claimed only that the derivative of the Lyapunov function is negative semi-definite, not negative definite since $\dot{V}$ does not contain any terms that are negative definite in $\tilde{\theta}$.
- Although we conclude only stability in the sense of Lyapunov for the closed loop system resulted from the passivity based adaptive control, further analysis will allow to reach stronger conclusions.
- In fact, we can use the Barbalat's lemma to show the convergence of tracking error to zero.

Comparison with joint-based schemes

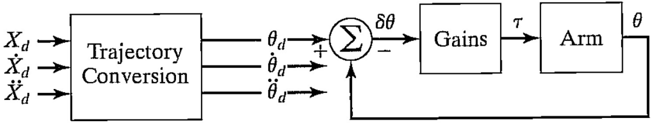
- In all the control schemes for manipulators we have discussed so far, we assumed that the desired trajectory was available in terms of time histories of joint position, velocity, and acceleration.
- Given that these desired inputs were available, we designedjoint-based control schemes, that is, schemes in which we develop trajectory errors by finding the difference between desired and actual quantities expressed in joint space.
- Very often, we wish the manipulator end-effector to follow straight lines or other path shapes described in Cartesian coordinates.

Comparison with joint-based schemes

Trajectory Conversion Method

- In this method, desired trajectories are converted from catesian space to joint space.
- The trajectory-conversion process is quite difficult (in terms of computational expense) if it is to be done analytically. The computations that would be required are

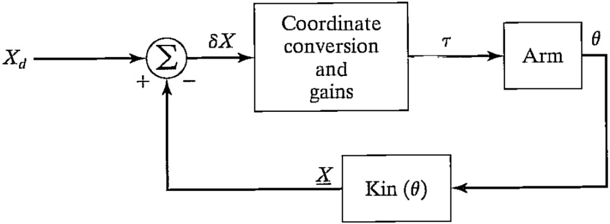
$$\theta_d = INVKIN(X_d)$$
$$\dot{\theta}_d = J^{-1}(\theta)\dot{X}_d$$
$$\ddot{\theta}_d = \dot{J}^{-1}(\theta)\dot{X}_d + J^{-1}(\theta)\ddot{X}_d$$



Comparison with joint-based schemes

The concept of a Cartesian-based control scheme

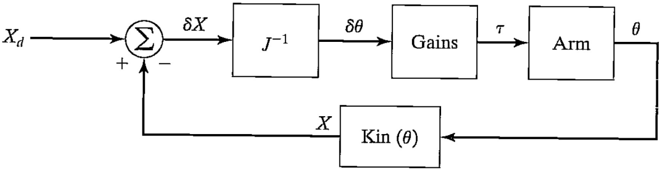
- In the alternative approach, the sensed position of the manipulator is immediately transformed by means of the kinematic equations into a Cartesian description of position.
- This Cartesian description is then compared to the desired Cartesian position in order to form errors in Cartesian space.
- Control schemes based on forming errors in Cartesian space are called Cartesian-based control schemes.



Intuitive schemes of Cartesian control

The inverse-Jacobian Cartesian-control scheme

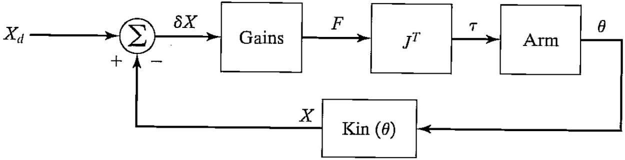
- In this method, Cartesian position is compared to the desired position to form an error, δX, in Cartesian space.
- This error, which may be presumed small if the control system is doing its job, may be mapped into a small displacement in joint space by means of the inverse Jacobian.
- The resulting errors in joint space, $\delta\theta$, are then multiplied by gains to compute torques that will tend to reduce these errors.



Intuitive schemes of Cartesian control

The transpose-Jacobian Cartesian-control scheme

- In this method, the Cartesian error vector is multiplied by a gain to compute a Cartesian force vector.
- This can be thought of as a Cartesian force which, if applied to the end-effector of the robot, would push the end-effector in a direction that would tend to reduce the Cartesian error.
- This Cartesian force vector (actually a force-moment vector) is then mapped through the Jacobian transpose in order to compute the equivalent joint torques that would tend to reduce the observed errors.



Intuitive schemes of Cartesian control

- The inverse-Jacobian controller and the transpose-Jacobian controller have both been arrived at intuitively.
- We cannot be sure that such arrangements would be stable.
- It turns out that both schemes will work (i.e., can be made stable), but not well (i.e., performance is not good over the entire workspace).
- Both can be made stable by appropriate gain selection, including some form of velocity feedback.
- The dynamic response of such controllers will vary with arm configuration.
- Just as we achieved good control with a joint-based controller that was based on a linearizing and decoupling model of the arm, we can do the same for the Cartesian case.