Chapter 4

Robot Dynamics and Control

This chapter presents an introduction to the dynamics and control of robot manipulators. We derive the equations of motion for a general open-chain manipulator and, using the structure present in the dynamics, construct control laws for asymptotic tracking of a desired trajectory. In deriving the dynamics, we will make explicit use of twists for representing the kinematics of the manipulator and explore the role that the kinematics play in the equations of motion. We assume some familiarity with dynamics and control of physical systems.

1 Introduction

The kinematic models of robots that we saw in the last chapter describe how the motion of the joints of a robot is related to the motion of the rigid bodies that make up the robot. We implicitly assumed that we could command arbitrary joint level trajectories and that these trajectories would be faithfully executed by the real-world robot. In this chapter, we look more closely at how to execute a given joint trajectory on a robot manipulator.

Most robot manipulators are driven by electric, hydraulic, or pneumatic actuators, which apply torques (or forces, in the case of linear actuators) at the joints of the robot. The *dynamics* of a robot manipulator describes how the robot moves in response to these actuator forces. For simplicity, we will assume that the actuators do not have dynamics of their own and, hence, we can command arbitrary torques at the joints of the robot. This allows us to study the inherent mechanics of robot manipulators without worrying about the details of how the joints are actuated on a particular robot.

We will describe the dynamics of a robot manipulator using a set of nonlinear, second-order, ordinary differential equations which depend on the kinematic and inertial properties of the robot. Although in principle these equations can be derived by summing all of the forces acting on the coupled rigid bodies which form the robot, we shall rely instead on a Lagrangian derivation of the dynamics. This technique has the advantage of requiring only the kinetic and potential energies of the system to be computed, and hence tends to be less prone to error than summing together the inertial, Coriolis, centrifugal, actuator, and other forces acting on the robot's links. It also allows the structural properties of the dynamics to be determined and exploited.

Once the equations of motion for a manipulator are known, the inverse problem can be treated: the *control* of a robot manipulator entails finding actuator forces which cause the manipulator to move along a given trajectory. If we have a perfect model of the dynamics of the manipulator, we can find the proper joint torques directly from this model. In practice, we must design a feedback control law which updates the applied forces in response to deviations from the desired trajectory. Care is required in designing a feedback control law to insure that the overall system converges to the desired trajectory in the presence of initial condition errors, sensor noise, and modeling errors.

In this chapter, we primarily concentrate on one of the simplest robot control problems, that of regulating the position of the robot. There are two basic ways that this problem can be solved. The first, referred to as *joint space control*, involves converting a given task into a desired path for the joints of the robot. A control law is then used to determine joint torques which cause the manipulator to follow the given trajectory. A different approach is to transform the dynamics and control problem into the task space, so that the control law is written in terms of the end-effector position and orientation. We refer to this approach as *workspace control*.

A much harder control problem is one in which the robot is in contact with its environment. In this case, we must regulate not only the position of the end-effector but also the forces it applies against the environment. We discuss this problem briefly in the last section of this chapter and defer a more complete treatment until Chapter 6, after we have introduced the tools necessary to study constrained systems.

2 Lagrange's Equations

There are many methods for generating the dynamic equations of a mechanical system. All methods generate equivalent sets of equations, but different forms of the equations may be better suited for computation or analysis. We will use a Lagrangian analysis for our derivation, which relies on the energy properties of mechanical systems to compute the equations of motion. The resulting equations can be computed in closed form, allowing detailed analysis of the properties of the system.

2.1 Basic formulation

Consider a system of n particles which obeys Newton's second law—the time rate of change of a particle's momentum is equal to the force applied to a particle. If we let F_i be the applied force on the ith particle, m_i be the particle's mass, and r_i be its position, then Newton's law becomes

$$F_i = m_i \ddot{r}_i \qquad r_i \in \mathbb{R}^3, i = 1, \ldots, n. \tag{4.1}$$

Our interest is not in a set of independent particles, but rather in particles which are attached to one another and have limited degrees of freedom. To describe this interconnection, we introduce *constraints* between the positions of our particles. Each constraint is represented by a function $g_j \colon \mathbb{R}^{3n} \to \mathbb{R}$ such that

$$g_j(r_1, \ldots, r_n) = 0 \qquad j = 1, \ldots, k. \tag{4.2}$$

A constraint which can be written in this form, as an algebraic relationship between the positions of the particles, is called a *holonomic* constraint. More general constraints between rigid bodies—involving $\dot{r}_i$—can also occur, as we shall discover when we study multifingered hands.

A constraint acts on a system of particles through application of *constraint forces*. The constraint forces are determined in such a way that the constraint in equation (4.2) is always satisfied. If we view the constraint as a smooth surface in $\mathbb{R}^n$, the constraint forces are normal to this surface and restrict the velocity of the system to be tangent to the surface at all times. Thus, we can rewrite our system dynamics as a vector equation

$$F = \begin{bmatrix} m_1 I & & 0 \\ & \ddots & \\ 0 & & m_n I \end{bmatrix} \begin{bmatrix} \ddot{r}_1 \\ \vdots \\ \ddot{r}_n \end{bmatrix} + \sum_{j=1}^{k} \Gamma_j \lambda_j, \tag{4.3}$$

where the vectors $\Gamma_1, \ldots, \Gamma_k \in \mathbb{R}^{3n}$ are a basis for the forces of constraint and λ_j is the scale factor for the jth basis element. We do not require that $\Gamma_1, \ldots, \Gamma_k$ be orthonormal. For constraints of the form in equation (4.2), Γ_j can be taken as the gradient of g_j, which is perpendicular to the level set $g_j(r) = 0$.

The scalars $\lambda_1, \ldots, \lambda_k$ are called *Lagrange multipliers*. Formally, we determine the Lagrange multipliers by solving the $3n + k$ equations in equations (4.2) and (4.3) for the $3n + k$ variables $r \in \mathbb{R}^{3n}$ and $\lambda \in \mathbb{R}^k$. The λ_i values only give the relative magnitudes of the constraint forces since the vectors Γ_j are not necessarily orthonormal.

This approach to dealing with constraints is intuitively simple but computationally complex, since we must keep track of the state of all particles in the system even though they are not capable of independent motion. A more appealing approach is to describe the motion of the system in terms of a smaller set of variables that completely describes the configuration of the system. For a system of n particles with k constraints, we seek a set of $m = 3n - k$ variables $q_1, \ldots, q_m$ and smooth functions $f_1, \ldots, f_n$ such that

$$\begin{matrix} r_i = f_i(q_1, \ldots, q_m) \\ i = 1, \ldots, n \end{matrix} \quad \Longleftrightarrow \quad \begin{matrix} g_j(r_1, \ldots, r_n) = 0 \\ j = 1, \ldots, k. \end{matrix} \tag{4.4}$$

We call the q_i's a set of *generalized coordinates* for the system. For a robot manipulator consisting of rigid links, these generalized coordinates are almost always chosen to be the angles of the joints. The specification of these angles uniquely determines the position of all of the particles which make up the robot.

Since the values of the generalized coordinates are sufficient to specify the position of the particles, we can rewrite the equations of motion for the system in terms of the generalized coordinates. To do so, we also express the external forces applied to the system in terms of components along the generalized coordinates. We call these forces *generalized forces* to distinguish them from physical forces, which are always represented as vectors in $\mathbb{R}^3$. For a robot manipulator with joint angles acting as generalized coordinates, the generalized forces are the torques applied about the joint axes.

To write the equations of motion, we define the *Lagrangian*, L, as the difference between the kinetic and potential energy of the system. Thus,

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q),$$

where T is the kinetic energy and V is the potential energy of the system, both written in generalized coordinates.

Theorem 4.1. Lagrange's equations
The equations of motion for a mechanical system with generalized coordinates $q \in \mathbb{R}^m$ and Lagrangian L are given by

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \Upsilon_i \qquad i = 1, \ldots, m, \tag{4.5}$$

where Υ_i is the external force acting on the ith generalized coordinate.

The equations in (4.5) are called *Lagrange's equations*. We will often write them in vector form as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \Upsilon,$$

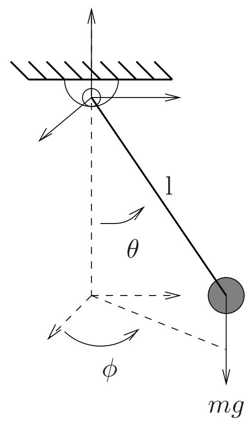


Figure 4.1: Idealized spherical pendulum. The configuration of the system is described by the angles θ and ϕ.

where $\frac{\partial L}{\partial \dot{q}}$, $\frac{\partial L}{\partial q}$, and Υ are to be formally regarded as row vectors, though we often write them as column vectors for notational convenience. A proof of Theorem 4.1 can be found in most books on dynamics of mechanical systems (e.g., [99]).

Lagrange's equations are an elegant formulation of the dynamics of a mechanical system. They reduce the number of equations needed to describe the motion of the system from n, the number of particles in the system, to m, the number of generalized coordinates. Note that if there are *no* constraints, then we can choose q to be the components of r, giving $T = \frac{1}{2}\sum m_i \|\dot{r}_i^2\|$, and equation (4.5) then reduces to equation (4.1). In fact, rearranging equation (4.5) as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial q} + \Upsilon$$

is just a restatement of Newton's law in generalized coordinates:

$$\frac{d}{dt}(\text{momentum}) = \text{applied force}.$$

The motion of the individual particles can be recovered through application of equation (4.4).

Example 4.1. Dynamics of a spherical pendulum
Consider an idealized spherical pendulum as shown in Figure 4.1. The system consists of a point with mass m attached to a spherical joint by a massless rod of length l. We parameterize the configuration of the point mass by two scalars, θ and ϕ, which measure the angular displacement from the z- and x-axes, respectively. We wish to solve for the motion of the mass under the influence of gravity.

We begin by deriving the Lagrangian for the system. The position of the mass, relative to the origin at the base of the pendulum, is given by

$$r(\theta, \phi) = \begin{bmatrix} l \sin\theta \cos\phi \\ l \sin\theta \sin\phi \\ -l \cos\theta \end{bmatrix}. \tag{4.6}$$

The kinetic energy is

$$T = \frac{1}{2} m l^2 \|\dot{r}\|^2 = \frac{1}{2} m l^2 \left(\dot{\theta}^2 + (1 - \cos^2\theta)\dot{\phi}^2 \right)$$

and the potential energy is

$$V = -mgl \cos\theta,$$

where $g \approx 9.8 \text{ m/sec}^2$ is the gravitational constant. Thus, the Lagrangian is given by

$$L(q, \dot{q}) = \frac{1}{2} m l^2 \left(\dot{\theta}^2 + (1 - \cos^2\theta)\dot{\phi}^2 \right) + mgl \cos\theta,$$

where $q = (\theta, \phi)$.

Substituting L into Lagrange's equations gives

$$\begin{aligned}
\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} &= \frac{d}{dt}\left(m l^2 \dot{\theta} \right) = m l^2 \ddot{\theta} \\
\frac{\partial L}{\partial \theta} &= m l^2 \sin\theta \cos\theta\, \dot{\phi}^2 - mgl \sin\theta \\
\frac{d}{dt}\frac{\partial L}{\partial \dot{\phi}} &= \frac{d}{dt}\left(m l^2 \sin^2\theta \dot{\phi} \right) = m l^2 \sin^2\theta\, \ddot{\phi} + 2 m l^2 \sin\theta \cos\theta\, \dot{\theta}\dot{\phi} \\
\frac{\partial L}{\partial \phi} &= 0
\end{aligned}$$

and the overall dynamics satisfy

$$\begin{bmatrix} m l^2 & 0 \\ 0 & m l^2 \sin^2\theta \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} + \begin{bmatrix} -m l^2 \sin\theta \cos\theta\, \dot{\phi}^2 \\ 2 m l^2 \sin\theta \cos\theta\, \dot{\theta}\dot{\phi} \end{bmatrix} + \begin{bmatrix} mgl \sin\theta \\ 0 \end{bmatrix} = 0. \tag{4.7}$$

Given the initial position and velocity of the point mass, equation (4.7) uniquely determines the subsequent motion of the system. The motion of the mass in $\mathbb{R}^3$ can be retrieved from equation (4.6).

2.2 Inertial properties of rigid bodies

To apply Lagrange's equations to a robot, we must calculate the kinetic and potential energy of the robot links as a function of the joint angles

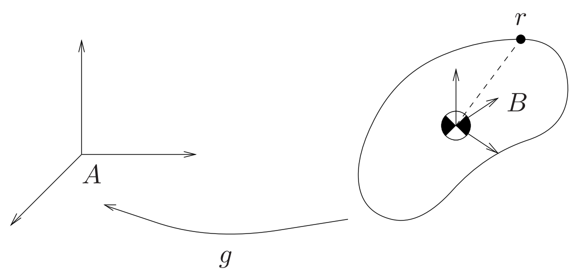


Figure 4.2: Coordinate frames for calculating the kinetic energy of a moving rigid body.

and velocities. This, in turn, requires that we have a model for the mass distribution of the links. Since each link is a rigid body, its kinetic and potential energy can be defined in terms of its total mass and its moments of inertia about the center of mass.

Let $V \subset \mathbb{R}^3$ be the volume occupied by a rigid body, and $\rho(r)$, $r \in V$ be the mass distribution of the body. If the object is made from a homogeneous material, then $\rho(r) = \rho$, a constant. The mass of the body is the volume integral of the mass density:

$$m = \int_V \rho(r)\, dV.$$

The center of mass of the body is the weighted average of the density:

$$\bar{r} = \frac{1}{m} \int_V \rho(r) r\, dV.$$

Consider the rigid object shown in Figure 4.2. We compute the kinetic energy as follows: fix the body frame at the mass center of the object and let (p, R) be a trajectory of the object relative to an inertial frame, where we have dropped all subscripts to simplify notation. Let $r \in \mathbb{R}^3$ be the coordinates of a body point relative to the body frame. The velocity of the point in the inertial frame is given by

$$\dot{p} + \dot{R}\, r$$

and the kinetic energy of the object is given by the following volume integral:

$$T = \frac{1}{2} \int_V \rho(r) \|\dot{p} + \dot{R} r\|^2\, dV. \tag{4.8}$$

Expanding the product in the kinetic energy integral yields

$$T = \frac{1}{2} \int_V \rho(r) \left(\|\dot{p}\|^2 + 2\dot{p}^T \dot{R} r + \|\dot{R} r\|^2 \right) dV.$$

The first term of the above expression gives the translational kinetic energy. The second term vanishes because the body frame is placed at the mass center of the object and

$$\int_V \rho(r)(\dot{p}^T \dot{R}) r \, dV = (\dot{p}^T \dot{R}) \int_V \rho(r) r \, dV = 0.$$

The last term can be simplified using properties of rotation and skew-symmetric matrices:

$$\begin{aligned} \frac{1}{2}\int_V \rho(r)(\dot{R}r)^T(\dot{R}r)\, dV &= \frac{1}{2}\int_V \rho(r)(R\widehat{\omega}r)^T(R\widehat{\omega}r)\, dV \\ &= \frac{1}{2}\int_V \rho(r)(\widehat{r}\omega)^T(\widehat{r}\omega)\, dV \\ &= \frac{1}{2}\omega^T\left(\int_V \rho(r)\widehat{r}^T\widehat{r} dV\right)\omega =: \frac{1}{2}\omega^T\mathcal{I}\omega, \end{aligned}$$

where $\omega \in \mathbb{R}^3$ is the *body* angular velocity. The symmetric matrix $\mathcal{I} \in \mathbb{R}^{\ni\times\ni}$ defined by

$$\mathcal{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = -\int_V \rho(r)\widehat{r}^2 \, dV$$

is called the *inertia tensor* of the object expressed in the body frame. It has entries

$$\begin{aligned} I_{xx} &= \int_V \rho(r)(y^2 + z^2)\, dx\, dy\, dz \\ I_{xy} &= -\int_V \rho(r)(xy)\, dx\, dy\, dz, \end{aligned}$$

and the other entries are defined similarly.

The total kinetic energy of the object can now be written as the sum of a translational component and a rotational component,

$$\begin{aligned} T &= \frac{1}{2}m\|\dot{p}\|^2 + \frac{1}{2}\omega^T\mathcal{I}\omega \\ &= \frac{1}{2}(V^b)^T \begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix} V^b =: \frac{1}{2}(V^b)^T\mathcal{M}V^b, \end{aligned} \tag{4.9}$$

where $\widehat{V}^b = g^{-1}\dot{g} \in se(3)$ is the body velocity, and $\mathcal{M}$ is called the *generalized inertia matrix* of the object, expressed in the body frame. The matrix $\mathcal{M}$ is symmetric and positive definite.

Example 4.2. Generalized inertia matrix for a homogeneous bar Consider a homogeneous rectangular bar with mass m, length l, width w, and height h, as shown in Figure 4.3. The mass density of the bar is

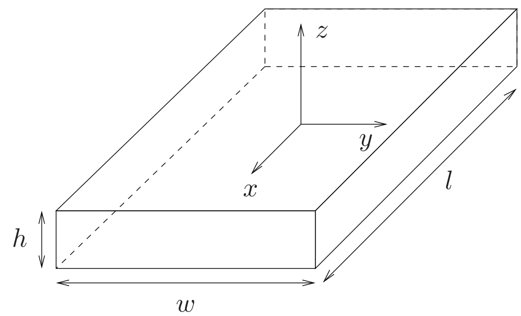


Figure 4.3: A homogeneous rectangular bar.

$\rho = \frac{m}{lwh}$. We attach a coordinate frame at the center of mass of the bar, with the coordinate axes aligned with the principal axes of the bar.

The inertia tensor is evaluated using the previous formula:

$$\begin{aligned}
I_{xx} &= \int_V \frac{m}{lwh}\left(y^2+z^2\right) dV = \frac{m}{lwh}\int_{-h/2}^{h/2}\int_{-w/2}^{w/2}\int_{-l/2}^{l/2}\left(y^2+z^2\right) dx\,dy\,dz \\
&= \frac{m}{lwh}\left(\frac{1}{12}\left(lw^3h + lwh^3\right)\right) = \frac{m}{12}(w^2+h^2), \\
I_{xy} &= -\int_V \frac{m}{lwh}(xy)\,dV = -\frac{m}{lwh}\int_{-h/2}^{h/2}\int_{-w/2}^{w/2}\int_{-l/2}^{l/2}(xy)\,dx\,dy\,dz \\
&= -\frac{m}{lwh}\int_{-h/2}^{h/2}\int_{-w/2}^{w/2}\left(\frac{1}{2}x^2y|_{-l/2}^{l/2}\right) dy\,dz = 0.
\end{aligned}$$

The other entries are calculated in the same manner and we have:

$$\mathcal{I} = \begin{bmatrix} \frac{m}{12}(w^2+h^2) & 0 & 0 \\ 0 & \frac{m}{12}(l^2+h^2) & 0 \\ 0 & 0 & \frac{m}{12}(l^2+w^2) \end{bmatrix}.$$

The inertia tensor is diagonal by virtue of the fact that we aligned the coordinate axes with the principal axes of the box.

The generalized inertia matrix is given by

$$\mathcal{M} = \begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{m}{12}(w^2+h^2) & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{m}{12}(l^2+h^2) & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{m}{12}(l^2+w^2) \end{bmatrix}.$$

The block diagonal structure of this matrix relies on attaching the body coordinate frame at center of mass (see Exercise 3).

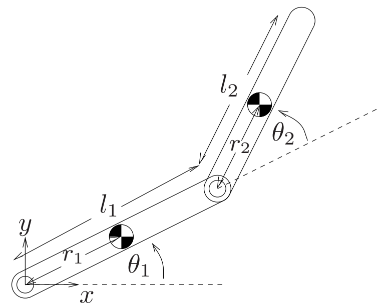


Figure 4.4: Two-link planar manipulator.

2.3 Example: Dynamics of a two-link planar robot

To illustrate how Lagrange's equations apply to a simple robotic system, consider the two-link planar manipulator shown in Figure 4.4. Model each link as a homogeneous rectangular bar with mass m_i and moment of inertia tensor

$$\mathcal{I}_\rangle = \begin{bmatrix} I_{xi} & 0 & 0 \\ 0 & I_{yi} & 0 \\ 0 & 0 & I_{zi} \end{bmatrix}$$

relative to a frame attached at the center of mass of the link and aligned with the principle axes of the bar. Letting $v_i \in \mathbb{R}^3$ be the translational velocity of the center of mass for the ith link and $\omega_i \in \mathbb{R}^3$ be the angular velocity, the kinetic energy of the manipulator is

$$T(\theta, \dot{\theta}) = \frac{1}{2} m_1 \|v_1\|^2 + \frac{1}{2} \omega_1^T \mathcal{I}_\infty \omega_\infty + \frac{\infty}{\in} \Updownarrow_\in \|\sqsubseteq_\in\|^\in + \frac{\infty}{\in} \omega_\in^\mathcal{T} \mathcal{I}_\in \omega_\in .$$

Since the motion of the manipulator is restricted to the xy plane, $\|v_i\|$ is the magnitude of the xy velocity of the center of mass and ω_i is a vector in the direction of the z-axis, with $\|\omega_1\| = \dot{\theta}_1$ and $\|\omega_2\| = \dot{\theta}_1 + \dot{\theta}_2$.

We solve for the kinetic energy in terms of the generalized coordinates by using the kinematics of the mechanism. Let $p_i = (x_i, y_i, 0)$ denote the position of the ith center of mass. Letting r_1 and r_2 be the distance from the joints to the center of mass for each link, as shown in the figure, we have

$$\begin{aligned} \bar{x}_1 &= r_1 c_1 & \dot{\bar{x}}_1 &= -r_1 s_1 \dot{\theta}_1 \\ \bar{y}_1 &= r_1 s_1 & \dot{\bar{y}}_1 &= r_1 c_1 \dot{\theta}_1 \\ \bar{x}_2 &= l_1 c_1 + r_2 c_{12} & \dot{\bar{x}}_2 &= -(l_1 s_1 + r_2 s_{12}) \dot{\theta}_1 - r_2 s_{12} \dot{\theta}_2 \\ \bar{y}_2 &= l_1 s_1 + r_2 s_{12} & \dot{\bar{y}}_2 &= (l_1 c_1 + r_2 c_{12}) \dot{\theta}_1 + r_2 c_{12} \dot{\theta}_2, \end{aligned}$$

where $s_i = \sin\theta_i$, $s_{ij} = \sin(\theta_i + \theta_j)$, and similarly for c_i and c_{ij}. The kinetic energy becomes

$$\begin{aligned} T(\theta,\dot\theta) &= \frac{1}{2}m_1(\dot x_1^2+\dot y_1^2)+\frac{1}{2}\mathcal{I}_{z1}\dot\theta_1^2+\frac{1}{2}m_2(\dot x_2^2+\dot y_2^2)+\frac{1}{2}\mathcal{I}_{z2}(\dot\theta_1+\dot\theta_2)^2 \\ &= \frac{1}{2}\begin{bmatrix}\dot\theta_1\\ \dot\theta_2\end{bmatrix}^T \begin{bmatrix}\alpha+2\beta c_2 & \delta+\beta c_2\\ \delta+\beta c_2 & \delta\end{bmatrix}\begin{bmatrix}\dot\theta_1\\ \dot\theta_2\end{bmatrix}, \end{aligned} \tag{4.10}$$

where

$$\begin{aligned} \alpha &= \mathcal{I}_{z1}+\mathcal{I}_{z2}+m_1r_1^2+m_2(l_1^2+r_2^2)\\ \beta &= m_2l_1r_2\\ \delta &= \mathcal{I}_{z2}+m_2r_2^2. \end{aligned}$$

Finally, we can substitute the Lagrangian $L = T$ into Lagrange's equations to obtain (after some calculation)

$$\begin{bmatrix}\alpha+2\beta c_2 & \delta+\beta c_2\\ \delta+\beta c_2 & \delta\end{bmatrix}\begin{bmatrix}\ddot\theta_1\\ \ddot\theta_2\end{bmatrix}+\begin{bmatrix}-\beta s_2\dot\theta_2 & -\beta s_2(\dot\theta_1+\dot\theta_2)\\ \beta s_2\dot\theta_1 & 0\end{bmatrix}\begin{bmatrix}\dot\theta_1\\ \dot\theta_2\end{bmatrix}=\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix}. \tag{4.11}$$

The first term in this equation represents the inertial forces due to acceleration of the joints, the second represents the Coriolis and centrifugal forces, and the right-hand side is the applied torques.

2.4 Newton-Euler equations for a rigid body

Lagrange's equations provide a very general method for deriving the equations of motion for a mechanical system. However, implicit in the derivation of Lagrange's equations is the assumption that the configuration space of the system can be parameterized by a subset of $\mathbb{R}^n$, where n is the number of degrees of freedom of the system. For a rigid body with configuration $g \in SE(3)$, Lagrange's equations cannot be directly used to determine the equations of motion unless we choose a local parameterization for the configuration space (for example, using Euler angles to parameterize the orientation of the rigid body). Since all parameterizations of $SE(3)$ are singular at some configuration, such a derivation can only hold locally.

In this section, we give a global characterization of the dynamics of a rigid body subject to external forces and torques. We begin by reviewing the standard derivation of the equations of rigid body motion and then examine the dynamics in terms of twists and wrenches.

Let $g = (p, R) \in SE(3)$ be the configuration of a coordinate frame attached to the center of mass of a rigid body, relative to an inertial frame. Let f represent a force applied at the center of mass, with the coordinates of f specified relative to the inertial frame. The translational equations of motion are given by Newton's law, which can written in terms of the linear momentum $m\dot{p}$ as

$$f = \frac{d}{dt}(m\dot{p}).$$

Since the mass of the rigid body is constant, the translational motion of the center of mass becomes

$$f = m\ddot{p}. \tag{4.12}$$

These equations are independent of the angular motion of the rigid body because we have used the center of mass to represent the position of the body.

Similarly, the equations describing angular motion can be derived independently of the linear motion of the system. Consider the rotational motion of a rigid body about a point, subject to an externally applied torque τ. To derive the equations of motion, we equate the change in angular momentum to the applied torque. The angular momentum relative to an inertial frame is given by $\mathcal{I}'\omega^{\int}$, where

$$\mathcal{I}' = \mathcal{R}\mathcal{I}\mathcal{R}^{\mathcal{T}}$$

is the instantaneous inertia tensor relative to the inertial frame and ω^s is the spatial angular velocity. The angular equations of motion become

$$\tau = \frac{d}{dt}(\mathcal{I}'\omega^{\int}) = \frac{\lceil}{\lceil\sqcup}(\mathcal{R}\mathcal{I}\mathcal{R}^{\mathcal{T}}\omega^{\int}),$$

where $\tau \in \mathbb{R}^3$ is specified relative to the inertial frame. Expanding the right-hand side of this equation, we have

$$\begin{aligned}
\tau &= R\mathcal{I}R^T\dot{\omega}^s + \dot{\mathcal{R}}\mathcal{I}\mathcal{R}^{\mathcal{T}}\omega^{\int} + \mathcal{R}\mathcal{I}\dot{\mathcal{R}}^{\mathcal{T}}\omega^{\int} \\
&= \mathcal{I}'\dot{\omega}^{\int} + \dot{\mathcal{R}}\mathcal{R}^{\mathcal{T}}\mathcal{I}'\omega^{\int} + \mathcal{I}'\mathcal{R}\dot{\mathcal{R}}^{\mathcal{T}}\omega^{\int} \\
&= \mathcal{I}'\dot{\omega}^{\int} + \omega^{\int} \times \mathcal{I}'\omega^{\int} - \mathcal{I}'\omega^{\int} \times \omega^{\int},
\end{aligned}$$

where the last equation follows by differentiating the identity $RR^T = I$ and using the definition of ω^s. The last term of this equation is zero, and hence the dynamics are given by

$$\mathcal{I}'\dot{\omega}^{\int} + \omega^{\int} \times \mathcal{I}'\omega^{\int} = \tau. \tag{4.13}$$

Equation (4.13) is called *Euler's equation*.

Equations (4.12) and (4.13) describe the dynamics of a rigid body in terms of a force and torque applied at the center of mass of the object. However, the coordinates of the force and torque vectors are not written relative to a body-fixed frame attached at the center of mass, but rather with respect to an inertial frame. Thus the pair $(f, \tau) \in \mathbb{R}^6$ is not the wrench applied to the rigid body, as defined in Chapter 2, since the point of application is not the origin of the inertial coordinate frame. Similarly, the velocity pair $(\dot{p}, \omega^s)$ does not correspond to the spatial or body velocity, since $\dot{p}$ is not the correct expression for the linear velocity term in either body or spatial coordinates.

In order to express the dynamics in terms of twists and wrenches, we rewrite Newton's equation using the body velocity $v^b = R^T\dot{p}$ and body force $f^b = R^T f$. Expanding the right-hand side of equation (4.12),

$$\frac{d}{dt}(m\dot{p}) = \frac{d}{dt}(mRv^b) = Rm\dot{v}^b + \dot{R}mv^b,$$

and pre-multiplying by R^T, the translational dynamics become

$$m\dot{v}^b + \omega^b \times mv^b = f^b. \tag{4.14}$$

Equation (4.14) is Newton's law written in body coordinates.

Similarly, we can write Euler's equation in terms of the body angular velocity $\omega^b = R^T\omega^s$ and the body torque $\tau^b = R^T\tau$. A straightforward computation shows that

$$\mathcal{I}\dot{\omega}^{\llcorner} + \omega^{\llcorner} \times \mathcal{I}\omega^{\llcorner} = \tau^{\llcorner}. \tag{4.15}$$

Equation (4.15) is Euler's equation, written in body coordinates. Note that in body coordinates the inertia tensor is constant and hence we use $\mathcal{I}$ instead of $\mathcal{I}' = \mathcal{R}\mathcal{I}\mathcal{R}^{\mathcal{T}}$.

Combining equations (4.14) and (4.15) gives the equations of motion for a rigid body subject to an external wrench F applied at the center of mass and specified with respect to the body coordinate frame:

$$\boxed{\begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix} \begin{bmatrix} \dot{v}^b \\ \dot{\omega}^b \end{bmatrix} + \begin{bmatrix} \omega^b \times mv^b \\ \omega^b \times \mathcal{I}\omega^b \end{bmatrix} = F^b} \tag{4.16}$$

This equation is called the *Newton-Euler equation* in body coordinates. It gives a global description of the equations of motion for a rigid body subject to an external wrench. Note that the linear and angular motions are coupled since the linear velocity in body coordinates depends on the current orientation.

It is also possible to write the Newton-Euler equations relative to a spatial coordinate frame. This version is explored in Exercises 4 and 5. Once again the equations for linear and angular motion are coupled, so that the translational motion still depends on the rotational motion. In this book we shall always write the Newton-Euler equations in body coordinates, as in equation (4.16).

3 Dynamics of Open-Chain Manipulators

We now derive the equations of motion for an open-chain robot manipulator. We shall use the kinematics formulation presented in the previous chapter to write the Lagrangian for the robot in terms of the joint angles and joint velocities. Using this form of the dynamics, we explore several fundamental properties of robot manipulators which are of importance when proving the stability of robot control laws.

3.1 The Lagrangian for an open-chain robot

To calculate the kinetic energy of an open-chain robot manipulator with n joints, we sum the kinetic energy of each link. For this we define a coordinate frame, L_i, attached to the center of mass of the ith link. Let

$$g_{sl_i}(\theta) = e^{\widehat{\xi}_1\theta_1} \cdots e^{\widehat{\xi}_i\theta_i} g_{sl_i}(0)$$

represent the configuration of the frame L_i relative to the base frame of the robot, S. The body velocity of the center of mass of the ith link is given by

$$V^b_{sl_i} = J^b_{sl_i}(\theta)\dot{\theta},$$

where $J^b_{sl_i}$ is the body Jacobian corresponding to g_{sl_i}. $J^b_{sl_i}$ has the form

$$J^b_{sl_i}(\theta) = \begin{bmatrix} \xi_1^\dagger & \cdots & \xi_i^\dagger & 0 & \cdots & 0 \end{bmatrix},$$

where

$$\xi_j^\dagger = \mathrm{Ad}^{-1}_{\left(e^{\widehat{\xi}_j\theta_j} \cdots e^{\widehat{\xi}_i\theta_i} g_{sl_i}(0)\right)} \xi_j \qquad j \leq i$$

is the jth instantaneous joint twist relative to the ith link frame. To streamline notation, we write $J^b_{sl_i}$ as J_i for the remainder of this section.

The kinetic energy of the ith link is

$$T_i(\theta, \dot{\theta}) = \frac{1}{2}(V^b_{sl_i})^T \mathcal{M}_i V^b_{sl_i} = \frac{1}{2}\dot{\theta}^T J_i^T(\theta)\mathcal{M}_i J_i(\theta)\dot{\theta}, \tag{4.17}$$

where $\mathcal{M}_i$ is the generalized inertia matrix for the ith link. Now the total kinetic energy can be written as

$$T(\theta, \dot{\theta}) = \sum_{i=1}^{n} T_i(\theta, \dot{\theta}) =: \frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta}. \tag{4.18}$$

The matrix $M(\theta) \in \mathbb{R}^{n \times n}$ is the *manipulator inertia matrix*. In terms of the link Jacobians, J_i, the manipulator inertia matrix is defined as

$$M(\theta) = \sum_{i=1}^{n} J_i^T(\theta)\mathcal{M}_i J_i(\theta). \tag{4.19}$$

To complete our derivation of the Lagrangian, we must calculate the potential energy of the manipulator. Let $h_i(\theta)$ be the height of the center of mass of the ith link (height is the component of the position of the center of mass opposite the direction of gravity). The potential energy for the ith link is

$$V_i(\theta) = m_i g h_i(\theta),$$

where m_i is the mass of the ith link and g is the gravitational constant. The total potential energy is given by the sum of the contributions from each link:

$$V(\theta) = \sum_{i=1}^{n} V_i(\theta) = \sum_{i=1}^{n} m_i g h_i(\theta).$$

Combining this with the kinetic energy, we have

$$L(\theta, \dot\theta) = \sum_{i=1}^{n} \Big(T_i(\theta, \dot\theta) - V_i(\theta) \Big) = \frac{1}{2} \dot\theta^T M(\theta) \dot\theta - V(\theta).$$

3.2 Equations of motion for an open-chain manipulator

Let $\theta \in \mathbb{R}^n$ be the joint angles for an open-chain manipulator. The Lagrangian is of the form

$$L(\theta, \dot\theta) = \frac{1}{2} \dot\theta^T M(\theta) \dot\theta - V(\theta),$$

where $M(\theta)$ is the manipulator inertia matrix and $V(\theta)$ is the potential energy due to gravity. It will be convenient to express the kinetic energy as a sum,

$$L(\theta, \dot\theta) = \frac{1}{2} \sum_{i,j=1}^{n} M_{ij}(\theta) \dot\theta_i \dot\theta_j - V(\theta). \tag{4.20}$$

The equations of motion are given by substituting into Lagrange's equations,

$$\frac{d}{dt} \frac{\partial L}{\partial \dot\theta_i} - \frac{\partial L}{\partial \theta_i} = \Upsilon_i,$$

where we let Υ_i represent the actuator torque and other nonconservative, generalized forces acting on the ith joint. Using equation (4.20), we have

$$\begin{aligned}
\frac{d}{dt} \frac{\partial L}{\partial \dot\theta_i} &= \frac{d}{dt} (\sum_{j=1}^{n} M_{ij} \dot\theta_j) = \sum_{j=1}^{n} \Big(M_{ij} \ddot\theta_j + \dot M_{ij} \dot\theta_j \Big) \\
\frac{\partial L}{\partial \theta_i} &= \frac{1}{2} \sum_{j,k=1}^{n} \frac{\partial M_{kj}}{\partial \theta_i} \dot\theta_k \dot\theta_j - \frac{\partial V}{\partial \theta_i}.
\end{aligned}$$

The $\dot{M}_{ij}$ term can now be expanded in terms of partial derivatives to yield

$$\sum_{j=1}^{n} M_{ij}(\theta)\ddot{\theta}_j + \sum_{j,k=1}^{n} \left(\frac{\partial M_{ij}}{\partial \theta_k}\dot{\theta}_j\dot{\theta}_k - \frac{1}{2}\frac{\partial M_{kj}}{\partial \theta_i}\dot{\theta}_k\dot{\theta}_j \right) + \frac{\partial V}{\partial \theta_i}(\theta) = \Upsilon_i \qquad i = 1, \ldots, n.$$

Rearranging terms, we can write

$$\sum_{j=1}^{n} M_{ij}(\theta)\ddot{\theta}_j + \sum_{j,k=1}^{n} \Gamma_{ijk}\dot{\theta}_j\dot{\theta}_k + \frac{\partial V}{\partial \theta_i}(\theta) = \Upsilon_i \qquad i = 1, \ldots, n, \tag{4.21}$$

where Γ_{ijk} is given by

$$\Gamma_{ijk} = \frac{1}{2}\left(\frac{\partial M_{ij}(\theta)}{\partial \theta_k} + \frac{\partial M_{ik}(\theta)}{\partial \theta_j} - \frac{\partial M_{kj}(\theta)}{\partial \theta_i} \right). \tag{4.22}$$

Equation (4.21) is a second-order differential equation in terms of the manipulator joint variables. It consists of four pieces: inertial forces, which depend on the acceleration of the joints; centrifugal and Coriolis forces, which are quadratic in the joint velocities; potential forces, of the form $\frac{\partial V}{\partial \theta_i}$; and external forces, Υ_i.

The centrifugal and Coriolis terms arise because of the non-inertial frames which are implicit in the use of generalized coordinates. In the classical mechanics literature, one identifies terms of the form $\dot{\theta}_i\dot{\theta}_j$, $i \neq j$ as Coriolis forces and terms of the form $\dot{\theta}_i^2$ as centrifugal forces. The functions Γ_{ijk} are called the *Christoffel symbols* corresponding to the inertia matrix $M(\theta)$.

The external forces can be divided into two components. Let τ_i represent the applied torque at the joint and define $-N_i(\theta, \dot{\theta})$ to be any other forces which act on the ith generalized coordinate, including conservative forces arising from a potential as well as frictional forces. (The reason for the negative sign in the definition of N_i will become apparent in a moment.) As an example, if the manipulator has viscous friction at the joints, then N_i would be defined as

$$-N_i(\theta, \dot{\theta}) = -\frac{\partial V}{\partial \theta_i} - \beta\dot{\theta}_i,$$

where β is the damping coefficient. Other forces acting on the manipulator, such as forces applied at the end-effector, can also be included by reflecting them to the joints (via the transpose of the appropriate Jacobian).

In order to put the equations of motion back into vector form, we define the matrix $C(\theta, \dot{\theta}) \in \mathbb{R}^{n \times n}$ as

$$C_{ij}(\theta, \dot{\theta}) = \sum_{k=1}^{n} \Gamma_{ijk}\dot{\theta}_k = \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial M_{ij}}{\partial \theta_k} + \frac{\partial M_{ik}}{\partial \theta_j} - \frac{\partial M_{kj}}{\partial \theta_i}\right)\dot{\theta}_k. \tag{4.23}$$

We call the matrix C the *Coriolis matrix* for the manipulator; the vector $C(\theta, \dot{\theta})\dot{\theta}$ gives the Coriolis and centrifugal force terms in the equations of motion. Note that there are other ways to define the matrix $C(\theta, \dot{\theta})$ such that $C_{ij}(\theta, \dot{\theta})\dot{\theta}_j = \Gamma_{ijk}\dot{\theta}_j\dot{\theta}_k$. However, this particular choice has important properties which we shall later exploit.

Equation (4.21) can now be rewritten as

$$\boxed{M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) = \tau} \tag{4.24}$$

where τ is the vector of actuator torques and $N(\theta, \dot{\theta})$ includes gravity terms and other forces which act at the joints. This is a second-order vector differential equation for the motion of the manipulator as a function of the applied joint torques. The matrices M and C, which summarize the inertial properties of the manipulator, have some important properties which we shall use in the sequel:

Lemma 4.2. Structural properties of the robot equations of motion

Equation (4.24) *satisfies the following properties:*

1. *$M(\theta)$ is symmetric and positive definite.*
2. *$\dot{M} - 2C \in \mathbb{R}^{n \times n}$ is a skew-symmetric matrix.*

Proof. Positive definiteness of the inertia matrix follows directly from its definition and the fact that the kinetic energy of the manipulator is zero only if the system is at rest. To show property 2, we calculate the components of the matrix $\dot{M} - 2C$:

$$\begin{aligned}
(\dot{M} - 2C)_{ij} &= \dot{M}_{ij}(\theta) - 2C_{ij}(\theta) \\
&= \sum_{k=1}^{n} \frac{\partial M_{ij}}{\partial \theta_k}\dot{\theta}_k - \frac{\partial M_{ij}}{\partial \theta_k}\dot{\theta}_k - \frac{\partial M_{ik}}{\partial \theta_j}\dot{\theta}_k + \frac{\partial M_{kj}}{\partial \theta_i}\dot{\theta}_k \\
&= \sum_{k=1}^{n} \frac{\partial M_{kj}}{\partial \theta_i}\dot{\theta}_k - \frac{\partial M_{ik}}{\partial \theta_j}\dot{\theta}_k.
\end{aligned}$$

Switching i and j shows $(\dot{M} - 2C)^T = -(\dot{M} - 2C)$. Note that the skew-symmetry property depends upon the particular definition of C given in equation (4.23). □

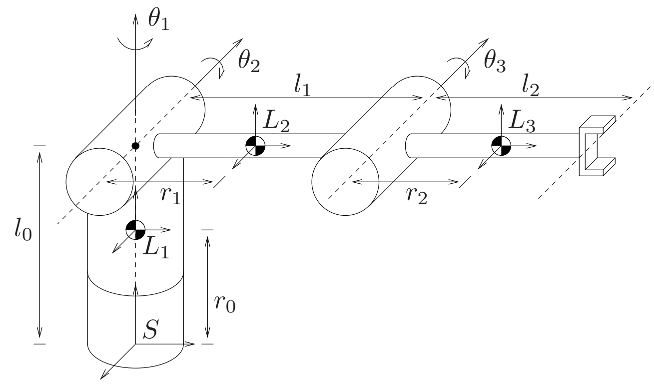


Figure 4.5: Three-link, open-chain manipulator.

Property 2 is often referred to as the *passivity* property since it implies, among other things, that in the absence of friction the net energy of the robot system is conserved (see Exercise 8). The passivity property is important in the proof of many control laws for robot manipulators.

Example 4.3. Dynamics of a three-link manipulator
To illustrate the formulation presented above, we calculate the dynamics of the three-link manipulator shown in Figure 4.5. The joint twists were computed in Chapter 3 (for the elbow manipulator) and are given by

$$\xi_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad \xi_2 = \begin{bmatrix} 0 \\ -l_0 \\ 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} \qquad \xi_3 = \begin{bmatrix} 0 \\ -l_0 \\ l_1 \\ -1 \\ 0 \\ 0 \end{bmatrix}.$$

To each link we attach a frame L_i at the center of mass and aligned with principle inertia axes of the link, as shown in the figure:

$$g_{sl_1(0)} = \begin{bmatrix} I & \begin{pmatrix} 0 \\ 0 \\ r_0 \end{pmatrix} \\ 0 & 1 \end{bmatrix} \quad g_{sl_2(0)} = \begin{bmatrix} I & \begin{pmatrix} 0 \\ r_1 \\ l_0 \end{pmatrix} \\ 0 & 1 \end{bmatrix} \quad g_{sl_3(0)} = \begin{bmatrix} I & \begin{pmatrix} 0 \\ l_1+r_2 \\ l_0 \end{pmatrix} \\ 0 & 1 \end{bmatrix}.$$

With this choice of link frames, the link inertia matrices have the general form

$$\mathcal{M}_\rangle = \left[\begin{array}{c|c} \begin{matrix} m_i & & 0 \\ & m_i & \\ 0 & & m_i \end{matrix} & 0 \\ \hline 0 & \begin{matrix} I_{xi} & & 0 \\ & I_{yi} & \\ 0 & & I_{zi} \end{matrix} \end{array}\right],$$

where m_i is the mass of the object and I_{xi}, I_{yi}, and I_{zi} are the moments of inertia about the x-, y-, and z-axes of the ith link frame.

To compute the manipulator inertia matrix, we first compute the body Jacobians corresponding to each link frame. A detailed, but straightforward, calculation yields

$$J_1 = J^b_{sl_1(0)} = \begin{bmatrix} 0&0&0\\ 0&0&0\\ 0&0&0\\ 0&0&0\\ 0&0&0\\ 1&0&0 \end{bmatrix} \qquad J_2 = J^b_{sl_2(0)} = \begin{bmatrix} -r_1c_2 & 0 & 0\\ 0 & 0 & 0\\ 0 & -r_1 & 0\\ 0 & -1 & 0\\ -s_2 & 0 & 0\\ c_2 & 0 & 0 \end{bmatrix}$$

$$J_3 = J^b_{sl_3(0)} = \begin{bmatrix} -l_2c_2 - r_2c_{23} & 0 & 0\\ 0 & l_1s_3 & 0\\ 0 & -r_2 - l_1c_3 & -r_2\\ 0 & -1 & -1\\ -s_{23} & 0 & 0\\ c_{23} & 0 & 0 \end{bmatrix}.$$

The inertia matrix for the system is given by

$$M(\theta) = \begin{bmatrix} M_{11} & M_{12} & M_{13}\\ M_{21} & M_{22} & M_{23}\\ M_{31} & M_{32} & M_{33} \end{bmatrix} = J_1^T \mathcal{M}_\infty \mathcal{J}_\infty + \mathcal{J}_\in^{\mathcal{T}} \mathcal{M}_\in \mathcal{J}_\in + \mathcal{J}_\ni^{\mathcal{T}} \mathcal{M}_\ni \mathcal{J}_\ni.$$

The components of M are given by

$$\begin{aligned} M_{11} &= I_{y2}s_2^2 + I_{y3}s_{23}^2 + I_{z1} + I_{z2}c_2^2 + I_{z3}c_{23}^2\\ &\qquad + m_2r_1^2c_2^2 + m_3(l_1c_2 + r_2c_{23})^2\\ M_{12} &= 0\\ M_{13} &= 0\\ \\ M_{21} &= 0\\ M_{22} &= I_{x2} + I_{x3} + m_3l_1^2 + m_2r_1^2 + m_3r_2^2 + 2m_3l_1r_2c_3\\ M_{23} &= I_{x3} + m_3r_2^2 + m_3l_1r_2c_3\\ \\ M_{31} &= 0\\ M_{32} &= I_{x3} + m_3r_2^2 + m_3l_1r_2c_3\\ M_{33} &= I_{x3} + m_3r_2^2. \end{aligned}$$

Note that several of the moments of inertia of the different links do not appear in this expression. This is because the limited degrees of freedom of the manipulator do not allow arbitrary rotations of each joint around each axis.

The Coriolis and centrifugal forces are computed directly from the inertia matrix via the formula

$$C_{ij}(\theta, \dot\theta) = \sum_{k=1}^{n} \Gamma_{ijk}\dot\theta_k = \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial M_{ij}}{\partial \theta_k} + \frac{\partial M_{ik}}{\partial \theta_j} - \frac{\partial M_{kj}}{\partial \theta_i}\right)\dot\theta_k.$$

A very messy calculation shows that the nonzero values of Γ_{ijk} are given by:

$$\begin{aligned}
\Gamma_{112} &= (I_{y2} - I_{z2} - m_2 r_1^2) c_2 s_2 + (I_{y3} - I_{z3}) c_{23} s_{23} \\
&\quad - m_3 (l_1 c_2 + r_2 c_{23})(l_1 s_2 + r_2 s_{23}) \\
\Gamma_{113} &= (I_{y3} - I_{z3}) c_{23} s_{23} - m_3 r_2 s_{23} (l_1 c_2 + r_2 c_{23}) \\
\Gamma_{121} &= (I_{y2} - I_{z2} - m_2 r_1^2) c_2 s_2 + (I_{y3} - I_{z3}) c_{23} s_{23} \\
&\quad - m_3 (l_1 c_2 + r_2 c_{23})(l_1 s_2 + r_2 s_{23}) \\
\\
\Gamma_{131} &= (I_{y3} - I_{z3}) c_{23} s_{23} - m_3 r_2 s_{23} (l_1 c_2 + r_2 c_{23}) \\
\\
\Gamma_{211} &= (I_{z2} - I_{y2} + m_2 r_1^2) c_2 s_2 + (I_{z3} - I_{y3}) c_{23} s_{23} \\
&\quad + m_3 (l_1 c_2 + r_2 c_{23})(l_1 s_2 + r_2 s_{23}) \\
\Gamma_{223} &= -l_1 m_3 r_2 s_3 \\
\Gamma_{232} &= -l_1 m_3 r_2 s_3 \\
\Gamma_{233} &= -l_1 m_3 r_2 s_3 \\
\\
\Gamma_{311} &= (I_{z3} - I_{y3}) c_{23} s_{23} + m_3 r_2 s_{23} (l_1 c_2 + r_2 c_{23}) \\
\\
\Gamma_{322} &= l_1 m_3 r_2 s_3
\end{aligned}$$

Finally, we compute the effect of gravitational forces on the manipulator. These forces are written as

$$N(\theta, \dot{\theta}) = \frac{\partial V}{\partial \theta},$$

where $V : \mathbb{R}^n \to \mathbb{R}$ is the potential energy of the manipulator. For the three-link manipulator under consideration here, the potential energy is given by

$$V(\theta) = m_1 g h_1(\theta) + m_2 g h_2(\theta) + m_3 g h_3(\theta),$$

where h_i is the the height of the center of mass for the ith link. These can be found using the forward kinematics map

$$g_{sl_i}(\theta) = e^{\widehat{\xi}_1 \theta_1} \dots e^{\widehat{\xi}_i \theta_i} g_{sl_i}(0),$$

which gives

$$\begin{aligned}
h_1(\theta) &= r_0 \\
h_2(\theta) &= l_0 - r_1 \sin\theta_2 \\
h_3(\theta) &= l_0 - l_1 \sin\theta_2 - r_2 \sin(\theta_2 + \theta_3).
\end{aligned}$$

Substituting these expressions into the potential energy and taking the derivative gives

$$N(\theta, \dot{\theta}) = \frac{\partial V}{\partial \theta} = \begin{bmatrix} 0 \\ -(m_2 g r_1 + m_3 g l_1)\cos\theta_2 - m_3 r_2 \cos(\theta_2 + \theta_3)) \\ -m_3 g r_2 \cos(\theta_2 + \theta_3)) \end{bmatrix}. \tag{4.25}$$

This completes the derivation of the dynamics.

3.3 Robot dynamics and the product of exponentials formula

The formulas and properties given in the last section hold for any mechanical system with Lagrangian $L = \frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta} - V(\theta)$. If the forward kinematics are specified using the product of exponentials formula, then it is possible to get more explicit formulas for the inertia and Coriolis matrices. In this section we derive these formulas, based on the treatments given by Brockett et al. [15] and Park et al. [87].

In addition to the tools introduced in Chapters 2 and 3, we will make use of one additional operation on twists. Recall, first, that in $so(3)$ the cross product between two vectors $\omega_1, \omega_2 \in \mathbb{R}^3$ yields a third vector, $\omega_1 \times \omega_2 \in \mathbb{R}^3$. It can be shown by direct calculation that the cross product satisfies

$$(\omega_1 \times \omega_2)^\wedge = \widehat{\omega}_1\widehat{\omega}_2 - \widehat{\omega}_2\widehat{\omega}_1.$$

By direct analogy, we define the *Lie bracket* on $se(3)$ as

$$[\widehat{\xi}_1, \widehat{\xi}_2] = \widehat{\xi}_1\widehat{\xi}_2 - \widehat{\xi}_2\widehat{\xi}_1.$$

A simple calculation verifies that the right-hand side of this equation has the form of a twist, and hence $[\widehat{\xi}_1, \widehat{\xi}_2] \in se(3)$.

If $\xi_1, \xi_2 \in \mathbb{R}^6$ represent the coordinates for two twists, we define the bracket operation $[\cdot,\cdot] : \mathbb{R}^6 \times \mathbb{R}^6 \to \mathbb{R}^6$ as

$$[\xi_1, \xi_2] = \left(\widehat{\xi}_1\widehat{\xi}_2 - \widehat{\xi}_2\widehat{\xi}_1\right)^\vee. \tag{4.26}$$

This is a generalization of the cross product on $\mathbb{R}^3$ to vectors in $\mathbb{R}^6$. The following properties of the Lie bracket are also generalizations of properties of the cross product:

$$= -[\xi_2, \xi_1]$$
$$[\xi_1, [\xi_2, \xi_3]] + [\xi_2, [\xi_3, \xi_1]] + [\xi_3, [\xi_1, \xi_2]] = 0.$$

A more detailed (and abstract) description of the Lie bracket operation on $se(3)$ is given in Appendix A. For this chapter we shall only need the formula given in equation (4.26)

We now define some additional notation which we use in the sequel. Let $A_{ij} \in \mathbb{R}^{6\times 6}$ represent the adjoint transformation given by

$$A_{ij} = \begin{cases} \mathrm{Ad}^{-1}_{\left(e^{\xi_{j+1}\theta_{j+1}} \cdots e^{\xi_i\theta_i}\right)} & i > j \\ I & i = j \\ 0 & i < j. \end{cases} \tag{4.27}$$

Using this notation, the jth column of the body Jacobian for the ith link is given by $\mathrm{Ad}_{g^{-1}_{sl_i}} A_{ij}\xi_j$:

$$J_i(\theta) = \mathrm{Ad}_{g^{-1}_{sl_i(0)}} \begin{bmatrix} A_{i1}\xi_1 & \cdots & A_{ii}\xi_i & 0 & \cdots & 0 \end{bmatrix}.$$

We combine $\mathrm{Ad}_{g^{-1}_{sl_i(0)}}$ with the link inertia matrix by defining the transformed inertia matrix for the ith link:

$$\mathcal{M}'_i = \mathrm{Ad}^T_{g^{-1}_{sl_i(0)}} \mathcal{M}_i \, \mathrm{Ad}_{g^{-1}_{sl_i(0)}}. \tag{4.28}$$

The matrix $\mathcal{M}'_i$ represents the inertia of the ith link reflected into the base frame of the manipulator.

Using these definitions, we can obtain formulas for the inertial quantities which appear in the equation of motion. We state the results as a proposition.

Proposition 4.3. Formulas for inertia and Coriolis matrices
Using the notation defined above, the inertia and Coriolis matrices for an open-chain manipulator are given by

$$\begin{aligned} M_{ij}(\theta) &= \sum_{l=\max(i,j)}^{n} \xi_i^T A_{li}^T \mathcal{M}'_l A_{lj}\xi_j \\ C_{ij}(\theta) &= \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial M_{ij}}{\partial\theta_k} + \frac{\partial M_{ik}}{\partial\theta_j} - \frac{\partial M_{kj}}{\partial\theta_i}\right)\dot\theta_k, \end{aligned} \tag{4.29}$$

where

$$\frac{\partial M_{ij}}{\partial\theta_k} = \sum_{l=\max(i,j)}^{n}\Big([A_{ki}\xi_i, \xi_k]^T A_{lk}^T \mathcal{M}'_l A_{lj}\xi_j + \xi_i^T A_{li}^T \mathcal{M}'_l A_{lk}[A_{kj}\xi_j, \xi_k]\Big). \tag{4.30}$$

This proposition shows that all of the dynamic attributes of the manipulator can be determined directly from the joint twists ξ_i, the link frames $g_{sl_i(0)}$, and the link inertia matrices $\mathcal{M}_i$. The matrices A_{ij} are the only expressions in equations (4.29) and (4.30) which depend on the current configuration of the manipulator.

Proof. The only term which needs to be calculated in order to prove the proposition is $\frac{\partial}{\partial\theta_k}(A_{lj}\xi_j)$. For $i \geq j$, let $g_{ij} \in SE(3)$ be the rigid transformation given by

$$g_{ij} = \begin{cases} e^{-\widehat{\xi}_i\theta_i} \cdots e^{-\widehat{\xi}_{j+1}\theta_{j+1}} & i > j \\ I & i = j, \end{cases}$$

so that $A_{ij} = \mathrm{Ad}_{g_{ij}}$. Using this notation, if k is an integer such that $i \geq k \geq j$, then $g_{ij} = g_{ik}g_{kj}$. We now proceed to calculate $\frac{\partial}{\partial\theta_k}(A_{lj}\xi_j)$ for $i \geq k \geq j$:

$$\begin{aligned}
\frac{\partial}{\partial\theta_k}(A_{lj}\xi_j) &= \left(\frac{\partial}{\partial\theta_k}\left(g_{lj}\,\widehat{\xi}_j\,g_{lj}^{-1}\right)\right)^{\vee} = \left(\frac{\partial g_{lj}}{\partial\theta_k}\,\widehat{\xi}_j\,g_{lj}^{-1} + g_{lj}\,\widehat{\xi}_j\,\frac{\partial g_{lj}^{-1}}{\partial\theta_k}\right)^{\vee} \\
&= \left(-g_{l,k}\,\widehat{\xi}_k\,g_{kj}\,\widehat{\xi}_j\,g_{lj}^{-1} + g_{lj}\,\widehat{\xi}_j\,g_{kj}^{-1}\,\widehat{\xi}_k\,g_{lk}^{-1}\right)^{\vee} \\
&= \mathrm{Ad}_{g_{lk}}\left(-\widehat{\xi}_k\,g_{kj}\,\widehat{\xi}_j\,g_{kj}^{-1} + g_{kj}\,\widehat{\xi}_j\,g_{kj}^{-1}\,\widehat{\xi}_k\right)^{\vee} \\
&= A_{lk}[A_{kj}\xi_j,\,\xi_k].
\end{aligned}$$

For all other values of k, $\frac{\partial}{\partial\theta_k}(A_{lj}\xi_j)$ is zero. The proposition now follows by direct calculation. □

Example 4.4. Dynamics of an idealized SCARA manipulator
Consider the SCARA manipulator shown in Figure 4.6. The joint twists are given by

$$\xi_1 = \begin{bmatrix}0\\0\\0\\0\\0\\1\end{bmatrix} \qquad \xi_2 = \begin{bmatrix}l_1\\0\\0\\0\\0\\1\end{bmatrix} \qquad \xi_3 = \begin{bmatrix}l_1+l_2\\0\\0\\0\\0\\1\end{bmatrix} \qquad \xi_4 = \begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix}.$$

Assuming that the link frames are initially aligned with the base frame and are located at the centers of mass of the links, the transformed link inertia matrices have the form

$$\mathcal{M}_\rangle' = \begin{bmatrix} I & 0 \\ -\widehat{p}_i & I \end{bmatrix}\begin{bmatrix} m_i I & 0 \\ 0 & \mathcal{I} \end{bmatrix}\begin{bmatrix} I & \widehat{p}_i \\ 0 & I \end{bmatrix} = \begin{bmatrix} m_i I & m_i\widehat{p}_i \\ -m_i\widehat{p}_i & \mathcal{I} \end{bmatrix},$$

where p_i is the location of the origin of the ith link frame relative to the base frame S.

Given the joint twists ξ_i and transformed link inertias $\mathcal{M}_\rangle'$, the dynamics of the manipulator can be computed using the formulas in Proposition 4.3. This task is considerably simplified using the software described in Appendix B, so we omit a detailed computation and present only the final result. The inertia matrix $M(\theta) \in \mathbb{R}^{4\times4}$ is given by

$$M(\theta) = \begin{bmatrix} \alpha+\beta+2\gamma\cos\theta_2 & \beta+\gamma\cos\theta_2 & \delta & 0 \\ \beta+\gamma\cos\theta_2 & \beta & \delta & 0 \\ \delta & \delta & \delta & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix},$$

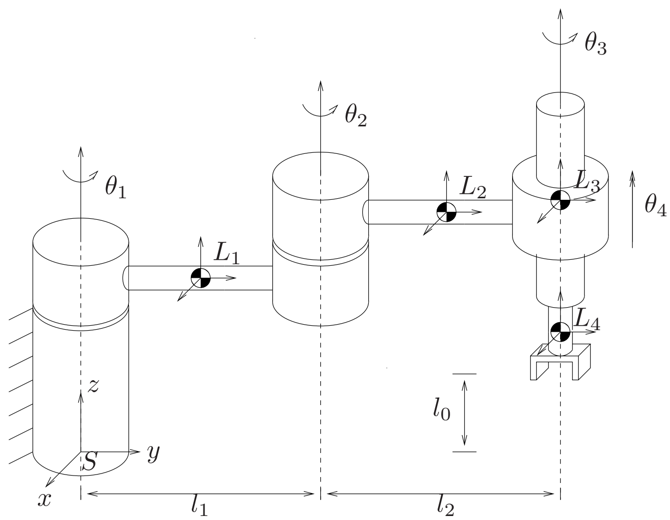


Figure 4.6: SCARA manipulator in its reference configuration.

where

$$\begin{aligned}
\alpha &= I_{z1} + r_1^2 m_1 + l_1^2 m_2 + l_1^2 m_3 + l_1^2 m_4 \\
\beta &= I_{z2} + I_{z3} + I_{z4} + l_2^2 m_3 + l_2^2 m_4 + m_2 r_2^2 \\
\gamma &= l_1 l_2 m_3 + l_1 l_2 m_4 + l_1 m_2 r_2 \\
\delta &= I_{z3} + I_{z4}.
\end{aligned}$$

The Coriolis matrix is given by

$$C(\theta, \dot{\theta}) = \begin{bmatrix} -\gamma \sin\theta_2\, \dot{\theta}_2 & -\gamma \sin\theta_2\, (\dot{\theta}_1 + \dot{\theta}_2) & 0 & 0 \\ \gamma \sin\theta_2\, \dot{\theta}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The only remaining term in the dynamics is the gravity term, which can be determined by inspection since only θ_4 affects the potential energy of the manipulator. Hence,

$$N(\theta, \dot{\theta}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ m_4 g \end{bmatrix}.$$

Friction and other nonconservative forces can also be included in N.

4 Lyapunov Stability Theory

In this section we review the tools of Lyapunov stability theory. These tools will be used in the next section to analyze the stability properties of a robot controller. We present a survey of the results that we shall need in the sequel, with no proofs. The interested reader should consult a standard text, such as Vidyasagar [118] or Khalil [49], for details.

4.1 Basic definitions

Consider a dynamical system which satisfies

$$\dot{x} = f(x,t) \qquad x(t_0) = x_0 \qquad x \in \mathbb{R}^n. \tag{4.31}$$

We will assume that $f(x,t)$ satisfies the standard conditions for the existence and uniqueness of solutions. Such conditions are, for instance, that $f(x,t)$ is Lipschitz continuous with respect to x, uniformly in t, and piecewise continuous in t. A point $x^* \in \mathbb{R}^n$ is an *equilibrium point* of (4.31) if $f(x^*,t) \equiv 0$. Intuitively and somewhat crudely speaking, we say an equilibrium point is *locally stable* if all solutions which start near x^* (meaning that the initial conditions are in a neighborhood of x^*) remain near x^* for all time. The equilibrium point x^* is said to be *locally asymptotically stable* if x^* is locally stable and, furthermore, all solutions starting near x^* tend towards x^* as $t \to \infty$. We say somewhat crude because the time-varying nature of equation (4.31) introduces all kinds of additional subtleties. Nonetheless, it is intuitive that a pendulum has a locally stable equilibrium point when the pendulum is hanging straight down and an unstable equilibrium point when it is pointing straight up. If the pendulum is damped, the stable equilibrium point is locally asymptotically stable.

By shifting the origin of the system, we may assume that the equilibrium point of interest occurs at $x^* = 0$. If multiple equilibrium points exist, we will need to study the stability of each by appropriately shifting the origin.

Definition 4.1. Stability in the sense of Lyapunov
The equilibrium point $x^* = 0$ of (4.31) is *stable (in the sense of Lyapunov)* at $t = t_0$ if for any $\epsilon > 0$ there exists a $\delta(t_0, \epsilon) > 0$ such that

$$\|x(t_0)\| < \delta \quad \Longrightarrow \quad \|x(t)\| < \epsilon, \quad \forall t \geq t_0. \tag{4.32}$$

Lyapunov stability is a very mild requirement on equilibrium points. In particular, it does not require that trajectories starting close to the origin tend to the origin asymptotically. Also, stability is defined at a time instant t_0. *Uniform stability* is a concept which guarantees that the equilibrium point is not losing stability. We insist that for a uniformly stable equilibrium point x^*, δ in the Definition 4.1 not be a function of t_0, so that equation (4.32) may hold for all t_0. Asymptotic stability is made precise in the following definition:

Definition 4.2. Asymptotic stability
An equilibrium point $x^* = 0$ of (4.31) is *asymptotically stable* at $t = t_0$ if

1. $x^* = 0$ is stable, and
2. $x^* = 0$ is locally attractive; i.e., there exists $\delta(t_0)$ such that

$$\|x(t_0)\| < \delta \quad \Longrightarrow \quad \lim_{t\to\infty} x(t) = 0. \tag{4.33}$$

As in the previous definition, asymptotic stability is defined at t_0. *Uniform asymptotic stability* requires:

1. $x^* = 0$ is uniformly stable, and
2. $x^* = 0$ is uniformly locally attractive; i.e., there exists δ independent of t_0 for which equation (4.33) holds. Further, it is required that the convergence in equation (4.33) is uniform.

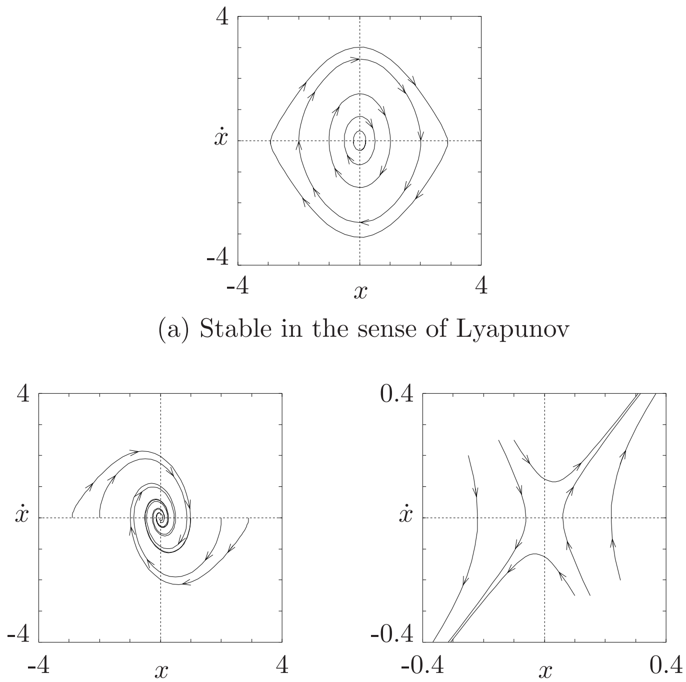
Finally, we say that an equilibrium point is *unstable* if it is not stable. This is less of a tautology than it sounds and the reader should be sure he or she can negate the definition of stability in the sense of Lyapunov to get a definition of instability. In robotics, we are almost always interested in uniformly asymptotically stable equilibria. If we wish to move the robot to a point, we would like to actually converge to that point, not merely remain nearby. Figure 4.7 illustrates the difference between stability in the sense of Lyapunov and asymptotic stability.

Definitions 4.1 and 4.2 are *local* definitions; they describe the behavior of a system near an equilibrium point. We say an equilibrium point x^* is *globally* stable if it is stable for all initial conditions $x_0 \in \mathbb{R}^n$. Global stability is very desirable, but in many applications it can be difficult to achieve. We will concentrate on local stability theorems and indicate where it is possible to extend the results to the global case. Notions of uniformity are only important for time-varying systems. Thus, for time-invariant systems, stability implies uniform stability and asymptotic stability implies uniform asymptotic stability.

It is important to note that the definitions of asymptotic stability do not quantify the rate of convergence. There is a strong form of stability which demands an exponential rate of convergence:

Definition 4.3. Exponential stability, rate of convergence
The equilibrium point $x^* = 0$ is an *exponentially stable* equilibrium point of (4.31) if there exist constants $m, \alpha > 0$ and $\epsilon > 0$ such that

$$\|x(t)\| \le m e^{-\alpha(t-t_0)} \|x(t_0)\| \tag{4.34}$$



(a) Stable in the sense of Lyapunov

(b) Asymptotically stable

(c) Unstable (saddle)

Figure 4.7: Phase portraits for stable and unstable equilibrium points.

for all $\|x(t_0)\| \leq \epsilon$ and $t \geq t_0$. The largest constant α which may be utilized in (4.34) is called the *rate of convergence*.

Exponential stability is a strong form of stability; in particular, it implies uniform, asymptotic stability. Exponential convergence is important in applications because it can be shown to be robust to perturbations and is essential for the consideration of more advanced control algorithms, such as adaptive ones. A system is *globally exponentially stable* if the bound in equation (4.34) holds for all $x_0 \in \mathbb{R}^n$. Whenever possible, we shall strive to prove global, exponential stability.

4.2 The direct method of Lyapunov

Lyapunov's direct method (also called the second method of Lyapunov) allows us to determine the stability of a system without explicitly integrating the differential equation (4.31). The method is a generalization of the idea that if there is some "measure of energy" in a system, then we can study the rate of change of the energy of the system to ascertain stability. To make this precise, we need to define exactly what one means by a "measure of energy." Let B_ϵ be a ball of size ϵ around the origin, $B_\epsilon = \{x \in \mathbb{R}^n : \|x\| < \epsilon\}$.

Definition 4.4. Locally positive definite functions (lpdf)
A continuous function $V : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ is a *locally positive definite function* if for some $\epsilon > 0$ and some continuous, strictly increasing function $\alpha : \mathbb{R}_+ \to \mathbb{R}$,

$$V(0,t) = 0 \quad \text{and} \quad V(x,t) \geq \alpha(\|x\|) \qquad \forall x \in B_\epsilon, \forall t \geq 0. \tag{4.35}$$

A locally positive definite function is locally like an energy function. Functions which are globally like energy functions are called positive definite functions:

Definition 4.5. Positive definite functions (pdf)
A continuous function $V : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ is a *positive definite function* if it satisfies the conditions of Definition 4.4 and, additionally, $\alpha(p) \to \infty$ as $p \to \infty$.

To bound the energy function from above, we define decrescence as follows:

Definition 4.6. Decrescent functions
A continuous function $V : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ is *decrescent* if for some $\epsilon > 0$ and some continuous, strictly increasing function $\beta : \mathbb{R}_+ \to \mathbb{R}$,

$$V(x,t) \leq \beta(\|x\|) \qquad \forall x \in B_\epsilon, \forall t \geq 0 \tag{4.36}$$

Using these definitions, the following theorem allows us to determine stability for a system by studying an appropriate energy function. Roughly, this theorem states that when $V(x,t)$ is a locally positive definite function and $\dot{V}(x,t) \leq 0$ then we can conclude stability of the equilibrium point. The time derivative of V is taken along the trajectories of the system:

$$\dot{V}\Big|_{\dot{x}=f(x,t)} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f.$$

In what follows, by $\dot{V}$ we will mean $\dot{V}|_{\dot{x}=f(x,t)}$.

Theorem 4.4. Basic theorem of Lyapunov
Let $V(x,t)$ be a non-negative function with derivative $\dot{V}$ along the trajectories of the system.

1. *If $V(x,t)$ is locally positive definite and $\dot{V}(x,t) \leq 0$ locally in x and for all t, then the origin of the system is locally stable (in the sense of Lyapunov).*

2. *If $V(x,t)$ is locally positive definite and decrescent, and $\dot{V}(x,t) \leq 0$ locally in x and for all t, then the origin of the system is uniformly locally stable (in the sense of Lyapunov).*

Table 4.1: Summary of the basic theorem of Lyapunov.

	Conditions on $V(x,t)$	Conditions on $-\dot{V}(x,t)$	Conclusions
1	lpdf	≥ 0 locally	Stable
2	lpdf, decrescent	≥ 0 locally	Uniformly stable
3	lpdf, decrescent	lpdf	Uniformly asymptotically stable
4	pdf, decrescent	pdf	Globally uniformly asymptotically stable

3. *If $V(x,t)$ is locally positive definite and decrescent, and $-\dot{V}(x,t)$ is locally positive definite, then the origin of the system is uniformly locally asymptotically stable.*

4. *If $V(x,t)$ is positive definite and decrescent, and $-\dot{V}(x,t)$ is positive definite, then the origin of the system is globally uniformly asymptotically stable.*

The conditions in the theorem are summarized in Table 4.1.

Theorem 4.4 gives sufficient conditions for the stability of the origin of a system. It does not, however, give a prescription for determining the Lyapunov function $V(x,t)$. Since the theorem only gives sufficient conditions, the search for a Lyapunov function establishing stability of an equilibrium point could be arduous. However, it is a remarkable fact that the converse of Theorem 4.4 also exists: if an equilibrium point is stable, then there exists a function $V(x,t)$ satisfying the conditions of the theorem. However, the utility of this and other converse theorems is limited by the lack of a computable technique for generating Lyapunov functions.

Theorem 4.4 also stops short of giving explicit rates of convergence of solutions to the equilibrium. It may be modified to do so in the case of exponentially stable equilibria.

Theorem 4.5. Exponential stability theorem
$x^* = 0$ is an exponentially stable equilibrium point of $\dot{x} = f(x,t)$ if and only if there exists an $\epsilon > 0$ and a function $V(x,t)$ which satisfies

$$\alpha_1 \|x\|^2 \leq V(x,t) \leq \alpha_2 \|x\|^2$$
$$\dot{V}|_{\dot{x}=f(x,t)} \leq -\alpha_3 \|x\|^2$$
$$\|\frac{\partial V}{\partial x}(x,t)\| \leq \alpha_4 \|x\|$$

for some positive constants $\alpha_1, \alpha_2, \alpha_3, \alpha_4$, and $\|x\| \leq \epsilon$.

The rate of convergence for a system satisfying the conditions of Theorem 4.5 can be determined from the proof of the theorem [102]. It can be shown that

$$m \leq \left(\frac{\alpha_2}{\alpha_1}\right)^{1/2} \qquad \alpha \geq \frac{\alpha_3}{2\alpha_2}$$

are bounds in equation (4.34). The equilibrium point $x^* = 0$ is globally exponentially stable if the bounds in Theorem 4.5 hold for all x.

4.3 The indirect method of Lyapunov

The indirect method of Lyapunov uses the linearization of a system to determine the local stability of the original system. Consider the system

$$\dot{x} = f(x,t) \tag{4.37}$$

with $f(0,t) = 0$ for all $t \geq 0$. Define

$$A(t) = \left.\frac{\partial f(x,t)}{\partial x}\right|_{x=0} \tag{4.38}$$

to be the Jacobian matrix of $f(x,t)$ with respect to x, evaluated at the origin. It follows that for each fixed t, the remainder

$$f_1(x,t) = f(x,t) - A(t)x$$

approaches zero as x approaches zero. However, the remainder may not approach zero *uniformly*. For this to be true, we require the stronger condition that

$$\lim_{\|x\| \to 0} \sup_{t \geq 0} \frac{\|f_1(x,t)\|}{\|x\|} = 0. \tag{4.39}$$

If equation (4.39) holds, then the system

$$\dot{z} = A(t)z \tag{4.40}$$

is referred to as the (uniform) *linearization* of equation (4.31) about the origin. When the linearization exists, its stability determines the local stability of the original nonlinear equation.

Theorem 4.6. Stability by linearization
Consider the system (4.37) *and assume*

$$\lim_{\|x\| \to 0} \sup_{t \geq 0} \frac{\|f_1(x,t)\|}{\|x\|} = 0.$$

Further, let $A(\cdot)$ defined in equation (4.38) *be bounded. If 0 is a uniformly asymptotically stable equilibrium point of* (4.40) *then it is a* locally *uniformly asymptotically stable equilibrium point of* (4.37).

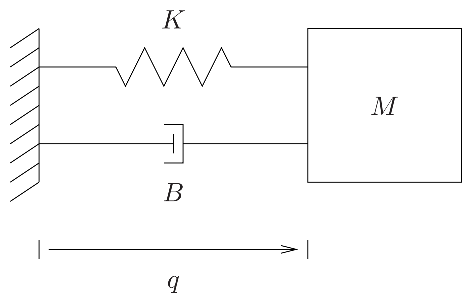


Figure 4.8: Damped harmonic oscillator.

The preceding theorem requires *uniform* asymptotic stability of the linearized system to prove uniform asymptotic stability of the nonlinear system. Counterexamples to the theorem exist if the linearized system is not uniformly asymptotically stable.

If the system (4.37) is time-invariant, then the indirect method says that if the eigenvalues of

$$A = \left. \frac{\partial f(x)}{\partial x} \right|_{x=0}$$

are in the open left half complex plane, then the origin is asymptotically stable.

This theorem proves that *global* uniform asymptotic stability of the linearization implies *local* uniform asymptotic stability of the original nonlinear system. The estimates provided by the proof of the theorem can be used to give a (conservative) bound on the domain of attraction of the origin. Systematic techniques for estimating the bounds on the regions of attraction of equilibrium points of nonlinear systems is an important area of research and involves searching for the "best" Lyapunov functions.

4.4 Examples

We now illustrate the use of the stability theorems given above on a few examples.

Example 4.5. Linear harmonic oscillator
Consider a damped harmonic oscillator, as shown in Figure 4.8. The dynamics of the system are given by the equation

$$M\ddot{q} + B\dot{q} + Kq = 0, \tag{4.41}$$

where M, B, and K are all positive quantities. As a state space equation we rewrite equation (4.41) as

$$\frac{d}{dt}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ -(K/M)q - (B/M)\dot{q} \end{bmatrix}. \tag{4.42}$$

Define $x = (q, \dot{q})$ as the state of the system.

Since this system is a linear system, we can determine stability by examining the poles of the system. The Jacobian matrix for the system is

$$A = \begin{bmatrix} 0 & 1 \\ -K/M & -B/M \end{bmatrix},$$

which has a characteristic equation

$$\lambda^2 + (B/M)\lambda + (K/M) = 0.$$

The solutions of the characteristic equation are

$$\lambda = \frac{-B \pm \sqrt{B^2 - 4KM}}{2M},$$

which always have negative real parts, and hence the system is (globally) exponentially stable.

We now try to apply Lyapunov's direct method to determine exponential stability. The "obvious" Lyapunov function to use in this context is the energy of the system,

$$V(x,t) = \frac{1}{2}M\dot{q}^2 + \frac{1}{2}Kq^2. \tag{4.43}$$

Taking the derivative of V along trajectories of the system (4.41) gives

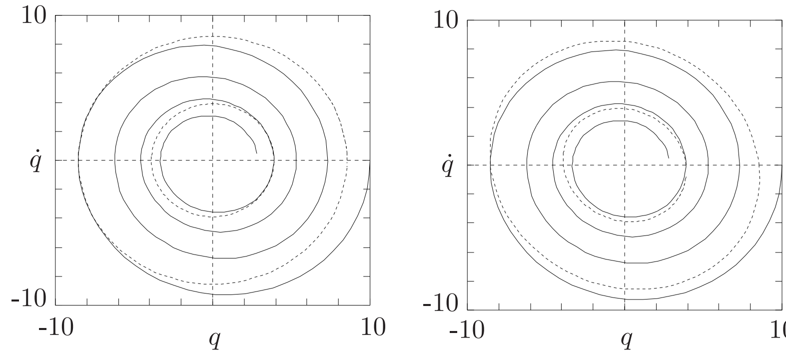
$$\dot{V} = M\dot{q}\ddot{q} + Kq\dot{q} = -B\dot{q}^2. \tag{4.44}$$

The function $-\dot{V}$ is quadratic but not locally positive definite, since it does not depend on q, and hence we cannot conclude exponential stability. It is still possible to conclude *asymptotic* stability using Lasalle's invariance principle (described in the next section), but this is obviously conservative since we already know that the system is exponentially stable.

The reason that Lyapunov's direct method fails is illustrated in Figure 4.9a, which shows the flow of the system superimposed with the level sets of the Lyapunov function. The level sets of the Lyapunov function become tangent to the flow when $\dot{q} = 0$, and hence it is not a valid Lyapunov function for determining exponential stability.

To fix this problem, we skew the level sets slightly, so that the flow of the system crosses the level surfaces transversely. Define

$$V(x,t) = \frac{1}{2}\begin{bmatrix} q \\ \dot{q} \end{bmatrix}^T \begin{bmatrix} K & \epsilon M \\ \epsilon M & M \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \frac{1}{2}\dot{q}M\dot{q} + \frac{1}{2}qKq + \epsilon\dot{q}Mq,$$



(a) (b)

Figure 4.9: Flow of damped harmonic oscillator. The dashed lines are the level sets of the Lyapunov function defined by (a) the total energy and (b) a skewed modification of the energy.

where ϵ is a small positive constant such that V is still positive definite. The derivative of the Lyapunov function becomes

$$\begin{aligned}\dot{V} &= \dot{q}M\ddot{q} + qK\dot{q} + \epsilon M\dot{q}^2 + \epsilon qM\ddot{q} \\ &= (-B + \epsilon M)\dot{q}^2 + \epsilon(-Kq^2 - Bq\dot{q}) = -\begin{bmatrix} q \\ \dot{q} \end{bmatrix}^T \begin{bmatrix} \epsilon K & \frac{1}{2}\epsilon B \\ \frac{1}{2}\epsilon B & B - \epsilon M \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix}.\end{aligned}$$

The function $\dot{V}$ can be made negative definite for ϵ chosen sufficiently small (see Exercise 11) and hence we can conclude *exponential* stability. The level sets of this Lyapunov function are shown in Figure 4.9b.

This same technique is used in the stability proofs for the robot control laws contained in the next section.

Example 4.6. Nonlinear spring mass system with damper
Consider a mechanical system consisting of a unit mass attached to a *nonlinear* spring with a velocity-dependent damper. If x_1 stands for the position of the mass and x_2 its velocity, then the equations describing the system are:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -f(x_2) - g(x_1).\end{aligned} \tag{4.45}$$

Here f and g are smooth functions modeling the friction in the damper and restoring force of the spring, respectively. We will assume that f, g are both passive; that is,

$$\begin{aligned}\sigma f(\sigma) &\geq 0 \quad \forall \sigma \in [-\sigma_0, \sigma_0] \\ \sigma g(\sigma) &\geq 0 \quad \forall \sigma \in [-\sigma_0, \sigma_0]\end{aligned}$$

and equality is only achieved when $\sigma = 0$. The candidate for the Lyapunov function is

$$V(x) = \frac{x_2^2}{2} + \int_0^{x_1} g(\sigma)\, d\sigma.$$

The passivity of g guarantees that $V(x)$ is a locally positive definite function. A short calculation verifies that

$$\dot{V}(x) = -x_2 f(x_2) \leq 0 \quad \text{when } |x_2| \leq \sigma_0.$$

This establishes the stability, but not the asymptotic stability of the origin. Actually, the origin is asymptotically stable, but this needs Lasalle's principle, which is discussed in the next section.

4.5 Lasalle's invariance principle

Lasalle's theorem enables one to conclude asymptotic stability of an equilibrium point even when $-\dot{V}(x,t)$ is not locally positive definite. However, it applies only to autonomous or periodic systems. We will deal with the autonomous case and begin by introducing a few more definitions. We denote the solution trajectories of the autonomous system

$$\dot{x} = f(x) \tag{4.46}$$

as $s(t, x_0, t_0)$, which is the solution of equation (4.46) at time t starting from x_0 at t_0.

Definition 4.7. ω limit set
The set $S \subset \mathbb{R}^n$ is the ω *limit set* of a trajectory $s(\cdot, x_0, t_0)$ if for every $y \in S$, there exists a strictly increasing sequence of times t_n such that

$$s(t_n, x_0, t_0) \to y$$

as $t_n \to \infty$.

Definition 4.8. Invariant set
The set $M \subset \mathbb{R}^n$ is said to be an (positively) *invariant set* if for all $y \in M$ and $t_0 \geq 0$, we have

$$s(t, y, t_0) \in M \quad \forall t \geq t_0.$$

It may be proved that the ω limit set of every trajectory is closed and invariant. We may now state Lasalle's principle.

Theorem 4.7. Lasalle's principle
Let $V : \mathbb{R}^n \to \mathbb{R}$ be a locally positive definite function such that on the compact set $\Omega_c = \{x \in \mathbb{R}^n : V(x) \leq c\}$ we have $\dot{V}(x) \leq 0$. Define

$$S = \{x \in \Omega_c : \dot{V}(x) = 0\}.$$

As $t \to \infty$, the trajectory tends to the largest invariant set inside S; i.e., its ω limit set is contained inside the largest invariant set in S. In particular, if S contains no invariant sets other than $x = 0$, then 0 is asymptotically stable.

A global version of the preceding theorem may also be stated. An application of Lasalle's principle is as follows:

Example 4.7. Nonlinear spring mass system with damper
Consider the same example as in equation (4.45), where we saw that with

$$V(x) = \frac{x_2^2}{2} + \int_0^{x_1} g(\sigma)\, d\sigma,$$

we obtained

$$\dot{V}(x) = -x_2 f(x_2).$$

Choosing $c = \min(V(-\sigma_0, 0), V(\sigma_0, 0))$ so as to apply Lasalle's principle, we see that

$$\dot{V}(x) \le 0 \quad \text{for } x \in \Omega_c := \{x : V(x) \le c\}.$$

As a consequence of Lasalle's principle, the trajectory enters the largest invariant set in $\Omega_c \cap \{x_1, x_2 : \dot{V} = 0\} = \Omega_c \cap \{x_1, 0\}$. To obtain the largest invariant set in this region, note that

$$x_2(t) \equiv 0 \quad \Longrightarrow \quad x_1(t) \equiv x_{10} \quad \Longrightarrow \quad \dot{x}_2(t) = 0 = -f(0) - g(x_{10}),$$

where x_{10} is some constant. Consequently, we have that

$$g(x_{10}) = 0 \quad \Longrightarrow \quad x_{10} = 0.$$

Thus, the largest invariant set inside $\Omega_c \cap \{x_1, x_2 : \dot{V} = 0\}$ is the origin and, by Lasalle's principle, the origin is locally asymptotically stable.

There is a version of Lasalle's theorem which holds for periodic systems as well. However, there are no significant generalizations for nonperiodic systems and this restricts the utility of Lasalle's principle in applications.

5 Position Control and Trajectory Tracking

In this section, we consider the position control problem for robot manipulators: given a desired trajectory, how should the joint torques be chosen so that the manipulator follows that trajectory. We would like to choose a control strategy which is robust with respect to initial condition errors, sensor noise, and modeling errors. We ignore the problems of actuator dynamics, and assume that we can command arbitrary torques which are exerted at the joints.

5.1 Problem description

We are given a description of the dynamics of a robot manipulator in the form of the equation

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + N(\theta,\dot{\theta}) = \tau, \tag{4.47}$$

where $\theta \in \mathbb{R}^n$ is the set of configuration variables for the robot and $\tau \in \mathbb{R}^n$ denotes the torques applied at the joints. We are also given a joint trajectory $\theta_d(\cdot)$ which we wish to track. For simplicity, we assume that θ_d is specified for all time and that it is at least twice differentiable.

If we have a perfect model of the robot and $\theta(0) = \theta_d(0), \dot{\theta}(0) = \dot{\theta}_d(0)$, then we may solve our problem by choosing

$$\tau = M(\theta_d)\ddot{\theta}_d + C(\theta_d,\dot{\theta}_d)\dot{\theta}_d + N(\theta_d,\dot{\theta}_d).$$

Since both θ and θ_d satisfy the same differential equation and have the same initial conditions, it follows from the uniqueness of the solutions of differential equations that $\theta(t) = \theta_d(t)$ for all $t \geq 0$. This an example of an *open-loop* control law: the current state of the robot is not used in choosing the control inputs.

Unfortunately, this strategy is not very robust. If $\theta(0) \neq \theta_d(0)$, then the open-loop control law will never correct for this error. This is clearly undesirable, since we almost never know the current position of a robot *exactly*. Furthermore, we have no guarantee that if our starting configuration is near the desired initial configuration that the trajectory of the robot will stay near the desired trajectory for all time. For this reason, we introduce feedback into our control law. This feedback must be chosen such that the actual robot trajectory converges to the desired trajectory. In particular, if our trajectory is a single setpoint, the closed-loop system should be asymptotically stable about the desired setpoint.

There are several approaches for designing stable robot control laws. Using the structural properties of robot dynamics, we will be able to prove stability of these control laws for *all* robots having those properties. Hence, we do not need to design control laws for a specific robot; as long as we show that stability of a particular control algorithm requires only those properties given in Lemma 4.2 on page 171, then our control law will work for general open-chain robot manipulators. Of course, the performance of a given control law depends heavily on the particular manipulator, and hence the control laws presented here should only be used as a starting point for synthesizing a feedback compensator.

5.2 Computed torque

Consider the following refinement of the open-loop control law presented above: given the current position and velocity of the manipulator, cancel all nonlinearities and apply exactly the torque needed to overcome the inertia of the actuator,

$$\tau = M(\theta)\ddot{\theta}_d + C(\theta,\dot{\theta})\dot{\theta} + N(\theta,\dot{\theta}).$$

Substituting this control law into the dynamic equations of the manipulator, we see that

$$M(\theta)\ddot{\theta} = M(\theta)\ddot{\theta}_d,$$

and since $M(\theta)$ is uniformly positive definite in θ, we have

$$\ddot{\theta} = \ddot{\theta}_d. \tag{4.48}$$

Hence, if the initial position and velocity of the manipulator matches the desired position and velocity, the manipulator will follow the desired trajectory. As before, this control law will not correct for any initial condition errors which are present.

The tracking properties of the control law can be improved by adding state feedback. The linearity of equation (4.48) suggests the following control law:

$$\boxed{\tau = M(\theta)\left(\ddot{\theta}_d - K_v\dot{e} - K_p e\right) + C(\theta,\dot{\theta})\dot{\theta} + N(\theta,\dot{\theta})} \tag{4.49}$$

where $e = \theta - \theta_d$, and K_v and K_p are constant gain matrices. When substituted into equation (4.47), the error dynamics can be written as:

$$M(\theta)\left(\ddot{e} + K_v\dot{e} + K_p e\right) = 0.$$

Since $M(\theta)$ is always positive definite, we have

$$\ddot{e} + K_v\dot{e} + K_p e = 0. \tag{4.50}$$

This is a linear differential equation which governs the error between the actual and desired trajectories. Equation (4.49) is called the *computed torque* control law.

The computed torque control law consists of two components. We can write equation (4.49) as

$$\tau = \underbrace{M(\theta)\ddot{\theta}_d + C\dot{\theta} + N}_{\tau_{\mathrm{ff}}} + \underbrace{M(\theta)\left(-K_v\dot{e} - K_p e\right)}_{\tau_{\mathrm{fb}}}.$$

The term τ_{ff} is the *feedforward* component. It provides the amount of torque necessary to drive the system along its nominal path. The term τ_{fb} is the *feedback* component. It provides correction torques to reduce any errors in the trajectory of the manipulator.

Since the error equation (4.50) is linear, it is easy to choose K_v and K_p so that the overall system is stable and $e \to 0$ exponentially as $t \to \infty$.

Moreover, we can choose K_v and K_p such that we get independent exponentially stable systems (by choosing K_p and K_v diagonal). The following proposition gives one set of conditions under which the computed torque control law (4.49) results in exponential tracking.

Proposition 4.8. Stability of the computed torque control law *If $K_p, K_v \in \mathbb{R}^{n\times n}$ are positive definite, symmetric matrices, then the control law* (4.49) *applied to the system* (4.47) *results in exponential trajectory tracking.*

Proof. The error dynamics can be written as a first-order linear system:

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}}_{A} \begin{bmatrix} e \\ \dot{e} \end{bmatrix}.$$

It suffices to show that each of the eigenvalues of A has negative real part. Let $\lambda \in \mathbb{C}$ be an eigenvalue of A with corresponding eigenvector $v = (v_1, v_2) \in \mathbb{C}^{2n}$, $v \neq 0$. Then,

$$\lambda \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} v_2 \\ -K_p v_1 - K_v v_2 \end{bmatrix}.$$

It follows that if $\lambda = 0$ then $v = 0$, and hence $\lambda = 0$ is not an eigenvalue of A. Further, if $\lambda \neq 0$, then $v_2 = 0$ implies that $v_1 = 0$. Thus, $v_1, v_2 \neq 0$ and we may assume without loss of generality that $\|v_1\| = 1$. Using this, we write

$$\begin{aligned} \lambda^2 &= v_1^* \lambda^2 v_1 = v_1^* \lambda v_2 \\ &= v_1^*(-K_p v_1 - K_v v_2) = -v_1^* K_p v_1 - \lambda v_1^* K_v v_1, \end{aligned}$$

where $*$ denotes complex conjugate transpose. Since $\alpha = v_1^* K_p v_1 > 0$ and $\beta = v_1^* K_v v_1 > 0$, we have

$$\lambda^2 + \alpha\lambda + \beta = 0 \qquad \alpha, \beta > 0$$

and hence the real part of λ is negative. □

The power of the computed torque control law is that it converts a nonlinear dynamical system into a linear one, allowing the use of any of a number of linear control synthesis tools. This is an example of a more general technique known as *feedback linearization*, where a nonlinear system is rendered linear via full-state nonlinear feedback. One disadvantage of using feedback linearization is that it can be demanding (in terms of computation time and input magnitudes) to use feedback to globally convert a nonlinear system into a single linear system. For robot manipulators, unboundedness of the inputs is rarely a problem since the inertia matrix of the system is bounded and hence the control torques which must be exerted always remain bounded. In addition, experimental results show that the computed torque controller has very good performance characteristics and it is becoming increasingly popular.

5.3 PD control

Another approach to controller synthesis for nonlinear systems is to design a linear controller based on the linearization of the system about an operating point. Since the linearization of a system locally determines the stability of the full system, this class of controllers is guaranteed to be locally stable. In many situations, it is possible to prove global stability for a linear controller by explicit construction of a Lyapunov function.

An example of this design methodology is a proportional plus derivative (PD) control law for a robot manipulator. In its simplest form, a PD control law has the form

$$\tau = -K_v \dot{e} - K_p e, \tag{4.51}$$

where K_v and K_p are positive definite matrices and $e = \theta - \theta_d$. Since this control law has no feedforward term, it can never achieve exact tracking for non-trivial trajectories. A common modification is to add an integral term to eliminate steady-state errors. This introduces additional complications since care must be taken to maintain stability and avoid integrator windup.

Before adding a feedforward term, we first show that the PD controller gives asymptotic setpoint stabilization.

Proposition 4.9. *If $\dot{\theta}_d \equiv 0$ and $K_v, K_p > 0$, the control law (4.51) applied to the system (4.47) renders the equilibrium point $\theta = \theta_d$ globally asymptotically stable.*

Proof. For $\theta_d \equiv 0$, the closed-loop system is

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + K_v \dot{\theta} + K_p(\theta - \theta_d) = 0. \tag{4.52}$$

Without loss of generality, we assume that $\theta_d = 0$ (if not, redefine $\theta' = \theta - \theta_d$). We choose the total energy of the system as our Lyapunov function,

$$V(\theta, \dot{\theta}) = \frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta} + \frac{1}{2}\theta^T K_p \theta.$$

The function V is (globally) positive definite and decresent. Evaluating $\dot{V}$ along trajectories of (4.52),

$$\begin{aligned} \dot{V}(\theta, \dot{\theta}) &= \dot{\theta}^T M \ddot{\theta} + \frac{1}{2}\dot{\theta}^T \dot{M}\dot{\theta} + \dot{\theta}^T K_p \theta \\ &= -\dot{\theta}^T K_v \dot{\theta} + \frac{1}{2}\dot{\theta}^T(\dot{M} - 2C)\dot{\theta}, \end{aligned}$$

and since $\dot{M} - 2C$ is skew-symmetric, we have

$$\dot{V} = -\dot{\theta}^T K_v \dot{\theta}.$$

Although K_v is positive definite, the function $\dot{V}$ is only negative *semi*-definite, since $\dot{V} = 0$ for $\dot{\theta} = 0$ and $\theta \neq 0$. Hence from Lyapunov's basic theorem, we can concluded only stability of the equilibrium point.

To check for asymptotic stability, we appeal to Lasalle's principle. The set S for which $\dot{V} \equiv 0$ is given by

$$S = \{(\theta, \dot{\theta}) : \dot{\theta} \equiv 0\}.$$

To find the largest invariant set contained in S, we substitute $\dot{\theta} \equiv 0$ into the closed loop equations 4.52. This gives

$$K_p \theta = 0$$

(recalling that $\theta_d = 0$) and since K_p is positive definite, it follows that the largest invariant set contained within S is the single point $\theta = 0$. Hence, the equilibrium point $\theta = 0$ is asymptotically stable. □

Since we are primarily interested in tracking, we consider a modified version of the PD control law:

$$\boxed{\tau = M(\theta)\ddot{\theta}_d + C(\theta, \dot{\theta})\dot{\theta}_d + N(\theta, \dot{\theta}) - K_v \dot{e} - K_p e} \tag{4.53}$$

We call this controller the *augmented PD control law.* Note that the second term in equation (4.53) is different from the Coriolis term $C(\theta, \dot{\theta})\dot{\theta}$. The reason for this difference is found in the proof of the following theorem.

Proposition 4.10. Stability of the PD control law
The control law (4.53) *applied to the system* (4.47) *results in exponential trajectory tracking if* $K_v, K_p > 0$.

Proof. The closed-loop system is

$$M(\theta)\ddot{e} + C(\theta, \dot{\theta})\dot{e} + K_v \dot{e} + K_p e = 0. \tag{4.54}$$

As in the proof of the previous proposition, using the energy of the system as a Lyapunov function does not allow us to conclude exponential stability because $\dot{V}$ is only negative semi-definite. Furthermore, since the system is time-varying (due to the $\theta_d(\cdot)$ terms), we cannot apply Lasalle's principle.

To show exponential stability, we adopt the same approach as with the spring mass system of the previous section. Namely, we skew the level sets of the energy function by choosing the Lyapunov function candidate

$$V(e, \dot{e}, t) = \frac{1}{2}\dot{e}^T M(\theta)\dot{e} + \frac{1}{2}e^T K_p e + \epsilon e^T M(\theta)\dot{e},$$

which is positive definite for ϵ sufficiently small since $M(\theta) > 0$ and $K_p > 0$. Evaluating $\dot{V}$ along trajectories of (4.54):

$$\begin{aligned}
\dot{V} &= \dot{e}^T M \ddot{e} + \frac{1}{2}\dot{e}^T \dot{M}\dot{e} + \dot{e}^T K_p e + \epsilon \dot{e}^T M \dot{e} + \epsilon e^T \big(M\ddot{e} + \dot{M}\dot{e}\big) \\
&= -\dot{e}^T (K_v - \epsilon M)\dot{e} + \frac{1}{2}\dot{e}^T (\dot{M} - 2C)\dot{e} + \epsilon e^T \big(-K_p e - K_v \dot{e} - C\dot{e} + \dot{M}\dot{e}\big) \\
&= -\dot{e}^T (K_v - \epsilon M)\dot{e} - \epsilon e^T K_p e + \epsilon e^T (-K_v + \frac{1}{2}\dot{M})\dot{e}
\end{aligned}$$

Choosing $\epsilon > 0$ sufficiently small insures that $\dot{V}$ is negative definite (see Exercise 11) and hence the system is exponentially stable using Theorem 4.5. □

If $\dot{\theta}_d \equiv 0$, i.e., we wish to stabilize a point, the control law (4.53) simplifies to the original PD control law (4.51). We also note that asymptotic tracking requires exact cancellation of friction and gravity forces and relies on accurate models of these quantities as well as the manipulator inertia matrix. In practice, errors in modeling will result in errors in tracking.

A further difficulty in using the PD control law is choosing the gains K_p and K_v. The linearization of the system about a given operating point θ_0 gives error dynamics of the form

$$M(\theta_0)\ddot{e} + K_v \dot{e} + K_p e = 0.$$

Since this is a linear system, it is possible to choose K_v and K_p to achieve a given performance specification using linear control theory. However, if we are tracking a trajectory, then there is no guarantee that we will remain near θ_0 and the chosen gains may not be appropriate. In practice, one can usually get reasonable results by choosing the gains based on the linearization about an equilibrium point in the middle of the robot's workspace.

5.4 Workspace control

Suppose we are given a path $g_d(t) \in SE(3)$ which represents the desired configuration of the end-effector as a function of time. One way to move the manipulator along this path is to solve the inverse kinematics problem at each instant in time and generate a desired joint angle trajectory $\theta_d(t) \in Q$ such that $g(\theta_d(t)) = g_d(t)$. The methods of the previous sections can then be used to generate a feedback controller which follows this path.

There are several disadvantages to solving the feedback control problem in this manner. Since solving the inverse kinematics problem is a time-consuming task, systems in which g_d is specified in real-time must use powerful computers to compute θ_d at a rate suitable for control. Furthermore, it may be difficult to choose the feedback gains in joint space in a meaningful way, since the original task was given in terms of the end-effector trajectory. For example, a joint-space, computed torque controller with diagonal gain matrices (K_p and K_v) will generate a decoupled response in joint space, resulting in straight line trajectories in θ if the setpoint of the manipulator is changed. However, due to the nonlinear nature of the kinematics, this will *not* generate a straight line trajectory in $SE(3)$. For many tasks, this type of behavior is undesirable.

To overcome these disadvantages, we consider formulating the problem directly in end-effector coordinates. In doing so, we will eliminate the need to solve the inverse kinematics and also generate controllers whose gains have a more direct connection with the task performance. However, in order to use the tools developed in Section 4, we must choose a set of local coordinates for $SE(3)$, such as parameterizing orientation via Euler angles. This limits the usefulness of the technique somewhat, although for many practical applications this limitation is of no consequence. This approach to writing controllers is referred to as *workspace control*, since x represents the configuration of the end-effector in the workspace of the manipulator.

Let $f : Q \to \mathbb{R}^p$ be a smooth and invertible mapping between the joint variables $\theta \in Q$ and the workspace variables $x \in \mathbb{R}^p$. In particular, this requires that $n = p$ so that the number of degrees of freedom of the robot equals the number of workspace variables x. We allow for the possibility that $p < 6$, in which case the workspace variables may only give a partial parameterization of $SE(3)$. An example of this situation is the SCARA robot, for which the position of the end-effector and its orientation with respect to the z-axis form a natural set of coordinates for specifying a task.

The dynamics of the manipulator in joint space has the form

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) = \tau,$$

where τ is the vector of joint torques and M, C, and N describe the dynamic parameters of the system, as before.

We can rewrite the dynamics in terms of $x \in \mathbb{R}^p$ by using the Jacobian of the mapping $f : \theta \mapsto x$,

$$\dot{x} = J(\theta)\dot{\theta} \qquad J(\theta) = \frac{\partial f}{\partial \theta}.$$

Note that J is the Jacobian of the *mapping* $f : Q \to \mathbb{R}^p$ and not the manipulator Jacobian. Under the assumption that f is smooth and invertible, we can write

$$\dot{\theta} = J^{-1}\dot{x} \qquad \text{and} \qquad \ddot{\theta} = J^{-1}\ddot{x} + \frac{d}{dt}(J^{-1})\dot{x}.$$

We can now substitute these expressions into the manipulator dynamics and pre-multiply by $J^{-T} := (J^{-1})^T$ to obtain

$$J^{-T}M(\theta)J^{-1}\ddot{x} + \left(J^{-T}C(\theta,\dot{\theta})J^{-1} + J^{-T}M(\theta)\frac{d}{dt}(J^{-1})\right)\dot{x} \\ + J^{-T}N(\theta,\dot{\theta}) = J^{-T}\tau.$$

We can write this in a more familiar form by defining

$$\begin{aligned}\tilde{M} &= J^{-T}MJ^{-1} \\ \tilde{C} &= J^{-T}\left(CJ^{-1} + M\frac{d}{dt}\left(J^{-1}\right)\right) \\ \tilde{N} &= J^{-T}N \\ F &= J^{-T}\tau,\end{aligned}$$

in which case the dynamics become

$$\tilde{M}(\theta)\ddot{x} + \tilde{C}(\theta,\dot{\theta})\dot{x} + \tilde{N}(\theta,\dot{\theta}) = F. \tag{4.55}$$

This equation represents the dynamics in terms of the workspace coordinates x and the robot configuration θ. We call $\tilde{M}$, $\tilde{C}$, and $\tilde{N}$ the *effective* parameters of the system. They represent the dynamics of the system as viewed from the workspace variables. Since f is locally invertible, we can in fact eliminate θ from these equations, and we see that equation (4.55) is nothing more than Lagrange's equations relative to the generalized coordinates x. However, since for most robots we measure θ directly and compute x via the forward kinematics, it is convenient to leave the θ dependence explicit.

Equation (4.55) has the same basic structure as the dynamics for an open-chain manipulator written in joint coordinates. In order to exploit this structure in our control laws, we must verify that some of the properties which we used in proving stability of controllers are also satisfied. The following lemma verifies that this is indeed the case.

Lemma 4.11. Structural properties of the workspace dynamics *Equation* (4.55) *satisfies the following properties:*

1. $\tilde{M}(\theta)$ *is symmetric and positive definite.*
2. $\dot{\tilde{M}} - 2\tilde{C} \in \mathbb{R}^{n\times n}$ *is a skew-symmetric matrix.*

Proof. Since J is an invertible matrix, property 1 follows from its definition. To show property 2, we calculate the $\dot{\tilde{M}} - 2\tilde{C}$:

$$\dot{\tilde{M}} - 2\tilde{C} = J^{-T}(\dot{\tilde{M}} - 2\tilde{C})J^{-1} + \frac{d}{dt}(J^{-T})\tilde{M}J^{-1} - J^{-T}\tilde{M}\frac{d}{dt}(J^{-1}).$$

A direct calculation shows that this matrix is indeed skew-symmetric. □

These two properties allow us to immediately extend the control laws in the previous section to workspace coordinates. For example, the computed torque control law becomes

$$F = \tilde{M}(\theta)\left(\ddot{x}_d - K_v\dot{e} - K_p e\right) + C(\theta, \dot{\theta})\dot{x} + N(\theta, \dot{\theta})$$
$$\tau = J^T F,$$

where x_d is the desired workspace trajectory and $e = x - x_d$ is the workspace error. The proof of stability for this control law is identical to that given previously. Namely, using the fact that $M(\theta)$ is positive definite, we can write the workspace error dynamics as

$$\ddot{e} + K_v\dot{e} + K_p e = 0$$

which is again a linear differential equation whose stability can be verified directly. The PD control law can be similarly extended to workspace coordinates.

The advantage of writing the control law in this fashion is that the matrices K_v and K_p now specify the gains directly in workspace coordinates. This simplifies the task of choosing the gains that are needed to accomplish a specific task. Furthermore, it eliminates the need to solve for the inverse mapping f^{-1} in order to control the robot. Instead, we only have to calculate the Jacobian matrix for f and its (matrix) inverse.

Notice that when the manipulator approaches a singular configuration relative to the coordinates x, the effective inertia $\tilde{M}$ gets very large. This is an indication that it is difficult to move in some directions and hence large forces produce very little motion. It is important to note that this singularity is strictly a function of our choice of parameterization. Such singularities never appear in the joint space of the robot.

Example 4.8. Comparison of joint space and workspace controllers

To illustrate some of the differences between implementing a controller in joint space versus workspace, we consider the control of a planar two degree of freedom robot. We take as our workspace variables the xy position of the end-effector.

Figure 4.10 shows the step response of a computed torque control law written in joint coordinates. Note that the trajectory of the end-effector, shown on the right, follows a curved path. The time response of the joint trajectories is a classical linear response for an underdamped mechanical system.

Figure 4.11 shows the step response of a computed torque control law written in workspace coordinates. Now the trajectory of the end-effector, including the overshoot, follows a straight line in the workspace and a curved line in the joint space.

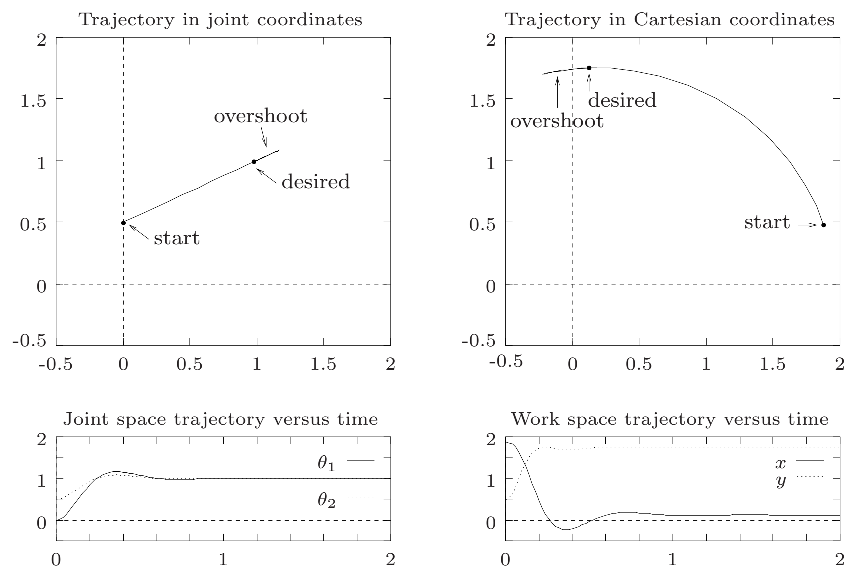


Figure 4.10: Step response of a joint space, computed torque controller.

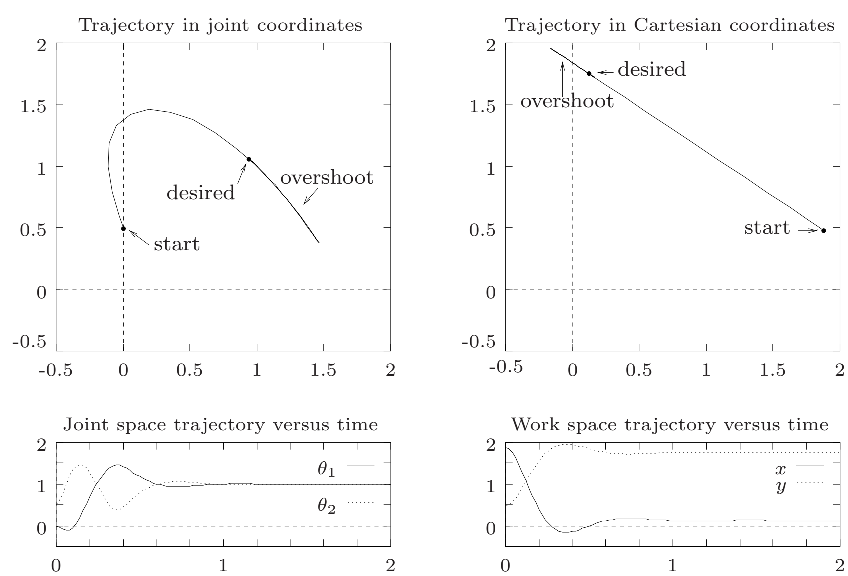


Figure 4.11: Step response of a workspace, computed torque controller.

6 Control of Constrained Manipulators

In this short section, we provide a brief treatment of the control of constrained manipulators. A more thorough development is given in Chapter 6.

6.1 Dynamics of constrained systems

Consider a problem in which we wish to move the tip of a robot along a surface and apply a force against that surface. For simplicity, we assume the surface is frictionless, although the analysis presented here can be readily extended to the more general case. We suppose that the surface we wish to move along can be described by a set of independent, smooth constraints

$$h_j(\theta_1, \ldots, \theta_n) = 0 \qquad j = 1, \ldots, k, \tag{4.56}$$

and that there exists a smooth, injective map $f : \mathbb{R}^{n-k} \to \mathbb{R}^n$ such that

$$h_j(f_1(\phi), \cdots, f_n(\phi)) = 0. \tag{4.57}$$

That is, $\phi \in \mathbb{R}^{n-k}$ parameterizes the allowable motion on the surface and $\theta = f(\phi)$ corresponds to a configuration in which the robot is in contact with the surface.

The control task is to follow a given trajectory $\phi_d(t)$ while applying a force against the surface. Since the surface is represented in joint space as the level set of the map $h(\theta) = 0$, the normal vectors to this surface are given by the span of the gradients of ∇h_i. (Since the surface is $n-k$ dimensional, the dimension of the space of normal vectors is k.) Any torques of the form

$$\tau_N = \sum \lambda_j \nabla h_j(\theta) \tag{4.58}$$

correspond to normal forces applied against the surface. In the absence of friction, the work done by these torques is given by

$$\begin{aligned} \tau_N \cdot \dot{\theta} &= \sum \lambda_i \nabla h_i \cdot \dot{\theta} = \sum \lambda_i \left(\frac{\partial h_i}{\partial \theta} \dot{\theta} \right) \\ &= \sum \lambda_i \frac{d}{dt} \left(h(\theta) \right) = 0. \end{aligned}$$

Hence the normal forces do no work on the system and therefore cause no motion in the system. We assume that a desired normal force, specified by $\lambda_1(t), \ldots, \lambda_k(t)$, is given as part of the task description.

If the robot remains in contact with the surface, as desired, then the dynamics of the manipulator can be written in terms of ϕ. Differentiating

$\theta = f(\phi)$, we have

$$\begin{aligned}\dot{\theta} &= \frac{\partial f}{\partial \phi}\dot{\phi} \\ \ddot{\theta} &= \frac{\partial f}{\partial \phi}\ddot{\phi} + \frac{d}{dt}\left(\frac{\partial f}{\partial \phi}\right)\dot{\phi}.\end{aligned} \tag{4.59}$$

These equations can be substituted into the robot equations of motion,

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) = \tau$$

to yield

$$M(\theta)\frac{\partial f}{\partial \phi}\ddot{\phi} + \left(C(\theta, \dot{\theta})\frac{\partial f}{\partial \phi} + M(\theta)\frac{d}{dt}\left(\frac{\partial f}{\partial \phi}\right)\right)\dot{\phi} + N(\theta, \dot{\theta}) = \tau, \tag{4.60}$$

where we have left M, C, and N in terms of θ to simplify notation.

Equation (4.60) can be made symmetric by multiplying both sides by $\frac{\partial f}{\partial \phi}^T$. Letting $J = \frac{\partial f}{\partial \phi}(\phi)$, we define

$$\begin{aligned}\tilde{M}(\phi) &= J^T M(f(\phi))J \\ \tilde{C}(\phi, \dot{\phi}) &= J^T \left(C(f(\phi), J\dot{\phi})J + M(f(\phi))\dot{J}\right) \\ \tilde{N}(\phi, \dot{\phi}) &= J^T N(f(\phi), J\dot{\phi}) \\ F &= J^T \tau.\end{aligned} \tag{4.61}$$

Using these definitions, the projected equations of motion can be written as

$$\tilde{M}(\phi)\ddot{\phi} + \tilde{C}(\phi, \dot{\phi})\dot{\phi} + \tilde{N}(\phi, \dot{\phi}) = F. \tag{4.62}$$

This equation has the same form as the equation for an unconstrained manipulator. We shall show in Chapter 6 that equation (4.62) also satisfies the properties in Lemma 4.2. This is not particularly surprising since the coordinates ϕ were chosen to be a set of generalized coordinates under the assumption that the robot maintains contact with the surface.

It is important to keep in mind that equation (4.62) represents the dynamics of the system only along the surface given by the level sets $h(\theta) = 0$. By pre-multiplying by J^T, we have eliminated the information about the forces of constraint. For many applications, we are interested in regulating the forces of constraint and hence we must use the full equations of motion given in equation (4.60).

6.2 Control of constrained manipulators

The control task for a constrained robot system is to simultaneously regulate the position of the system along the constraint surface and regulate the forces of the system applied against this surface. In terms of analyzing stability, it is enough to analyze only the motion along the surface, since no movement occurs perpendicular to the surface. Of course, implicit in this point of view is that we maintain contact with the surface. If the manipulator is not physically constrained, this may require that we regulate the forces so as to insure that we are always pushing against the surface and never pulling away from it.

In this section we show how to extend the computed torque formalism presented earlier to regulate the position and force of the manipulator. We give only a sketch of the approach, leaving a more detailed discussion until Chapter 6, where we shall see that hybrid position/force control is just one example of the more general problem of controlling single and multiple robots interacting with each other and their environment.

We take as given a path on the constraint surface, specified by $\phi_d(t)$, and a normal force to be applied against the surface, specified by the Lagrange multipliers $\lambda_1(t), \ldots, \lambda_k(t)$ as in equation (4.58). Since we are interested in regulating the force applied against the constraint, it is important to insure that the position portion of the controller does not push against the constraint. Define

$$\begin{aligned}\tau_\phi = M(\theta)\frac{\partial f}{\partial \phi}(\ddot{\phi}_d - K_v \dot{e}_\phi - K_p e_\phi) \\ + \left(C(\theta,\dot{\theta})\frac{\partial f}{\partial \phi} + M(\theta)\frac{d}{dt}\left(\frac{\partial f}{\partial \phi}\right)\right)\dot{\phi} + N(\theta,\dot{\theta}),\end{aligned}$$

where $e_\phi = \phi - \phi_d$. This is the torque required to move the manipulator along the surface while applying no force against the surface. In other words, if we apply $\tau = \tau_\phi$ and remove the constraint completely, the manipulator will follow the correct path, as if the constraint were present.

To apply the appropriate normal forces, we simply add τ_N as defined in equation (4.58) to τ_ϕ. Since τ_N is in the normal direction to the constraint, it does not affect the position portion of the controller. Of course, this requires that the constraint surface actually be present to resist the normal forces applied to it. The complete control law is given by

$$\tau = \tau_\phi + \sum_{i=1}^{k} \lambda_i(t)\nabla h_i \tag{4.63}$$

where τ_ϕ is given above. We defer the analysis and proof of convergence for this control law until Chapter 6.

As in the previous control laws, the force control law presented here relies on accurate models of the robot and the surface. In particular, we note that the applied normal force does not use feedback to correct for model error, sensor noise, or other non-ideal situations.

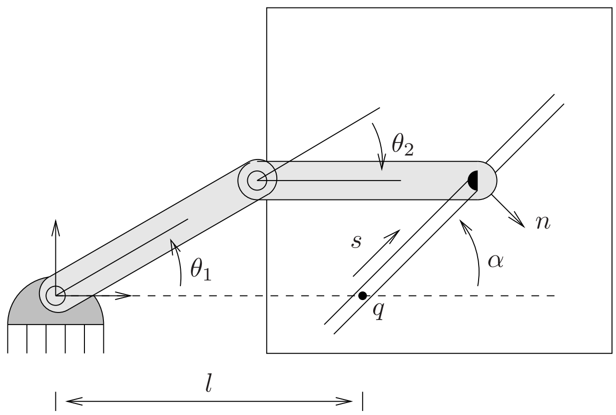


Figure 4.12: Planar manipulator moving in a slot.

6.3 Example: A planar manipulator moving in a slot

As a simple example of a constrained manipulator, consider the control of a two degree of freedom, planar manipulator whose end-effector is forced to lie in a slot, as shown in Figure 4.12. This system resembles a slider-crank mechanism, except that we are allowed to apply torques on both revolute joints, allowing us to control both the motion of the slider as well as the force applied against the slot. This example is easily adapted to a robot pushing against a wall, in which case the forces against the slot must always be pointed in a preferred direction.

We take the slot to be a straight line passing through the point $q = (l, 0)$ and making an angle α with respect to the x-axis of the base frame. The vector normal to the direction of the slot is given by

$$n = \begin{bmatrix} \sin\alpha \\ -\cos\alpha \end{bmatrix},$$

and the slot can be described as the set of all points $p \in \mathbb{R}^2$ such that $(p - q) \cdot n = 0$.

The constraint on the manipulator is obtained by requiring that the position of the end-effector remain in the slot. Letting $p(\theta) \in \mathbb{R}^2$ represent the position of the tool frame, this constraint becomes

$$h(\theta) = \left(p(\theta) - \begin{bmatrix} l \\ 0 \end{bmatrix} \right) \cdot \begin{bmatrix} \sin\alpha \\ -\cos\alpha \end{bmatrix} = 0.$$

Substituting the forward kinematics of the manipulator yields

$$\begin{aligned} h(\theta) &= (l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) - l) \sin\alpha \\ &\qquad - (l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)) \cos\alpha \\ &= -l_1 \sin(\theta_1 - \alpha) - l_2 \sin(\theta_1 + \theta_2 - \alpha) - l \sin\alpha. \end{aligned}$$

The gradient of the constraint, which gives the direction of the normal force, is given by

$$\nabla h(\theta) = \begin{bmatrix} -l_1 \cos(\theta_1 - \alpha) - l_2 \cos(\theta_1 + \theta_2 - \alpha) \\ -l_2 \cos(\theta_1 + \theta_2 - \alpha) \end{bmatrix}.$$

Note that this is the direction of the normal force in *joint coordinates*. That is, joint torques applied in this direction will cause no motion, only forces against the side of the slot.

To parameterize the allowable motion along the slot, we let $s \in \mathbb{R}$ represent the position along the slot, with $s = 0$ denoting the point $q = (l, 0)$. Finding a function $f(s)$ such that $h(f(s)) = 0$ involves solving the inverse kinematics of the manipulator: given the position along the slot, we must find joint angles which achieve that position.

If the end of the manipulator is at a position s along the slot, then the xy coordinates of the end-effector are

$$\begin{aligned} x(s) &= l + s \cos\alpha \\ y(s) &= s \sin\alpha. \end{aligned}$$

Solving the inverse kinematics (see Chapter 3, Section 3) and assuming the elbow down solution, we have

$$f(s) = \begin{bmatrix} \theta_1(s) \\ \theta_2(s) \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\frac{s \sin\alpha}{l + s\cos\alpha}\right) + \cos^{-1}\left(\frac{s^2 + 2ls\cos\alpha + l^2 + l_1^2 - l_2^2}{2l_1\sqrt{s^2 + 2ls\cos\alpha + l^2}}\right) \\ \pi + \cos^{-1}\left(\frac{l_1^2 + l_2^2 - s^2 - 2ls\cos\alpha - l^2}{2l_1 l_2}\right) \end{bmatrix}.$$

The Jacobian of the mapping is given by

$$J = \begin{bmatrix} \frac{-(s + l\cos\alpha)(s^2 + 2ls\cos\alpha + l^2 - l_1^2 + l_2^2)}{2l_1(s^2 + 2ls\cos\alpha + l^2)^{\frac{3}{2}}\sqrt{1 - \frac{(s^2 + 2ls\cos\alpha + l^2 + l_1^2 - l_2^2)}{4l_1^2(s^2 + 2ls\cos\alpha + l^2)}}} + \frac{l\sin\alpha}{s^2 + l^2 + 2ls\cos\alpha} \\ \frac{2(s + l\cos\alpha)}{\sqrt{4l_1^2 l_2^2 - (s^2 + 2ls\cos\alpha + l^2 - l_1^2 - l_2^2)^2}} \end{bmatrix}$$

(after some simplification).

This matrix can now be used to compute the equations of motion and derive an appropriate control law. In particular, the computed torque controller has the form

$$\tau = M(\theta)J\left(\ddot{s}_d - K_v\dot{e}_s - K_p e_s\right) + \left(C(\theta,\dot{\theta})J + M(\theta)\dot{J}\right)\dot{s} + \lambda n,$$

where $e_s = s - s_d$; λ is the desired force against the slot; $K_v, K_p \in \mathbb{R}$ are the gain and damping factors; and M and C are the generalized inertial and Coriolis matrices. The inertial parameters were calculated in Section 2.3 and are given by

$$M(\theta) = \begin{bmatrix} \alpha + \beta c_2 & \delta + \frac{1}{2}\beta c_2 \\ \delta + \frac{1}{2}\beta c_2 & \delta \end{bmatrix} \quad C(\theta,\dot{\theta}) = \begin{bmatrix} -\frac{1}{2}\beta s_2\dot{\theta}_2 & -\frac{1}{2}\beta s_2(\dot{\theta}_1 + \dot{\theta}_2) \\ \frac{1}{2}\beta s_2\dot{\theta}_1 & 0 \end{bmatrix},$$

where

$$
\begin{aligned}
\alpha &= \mathcal{I}_{z1} + \mathcal{I}_{z2} + m_1 r_1^2 + m_2(l_1^2 + r_2^2) \\
\beta &= m_2 l_1 l_2 \\
\delta &= \mathcal{I}_{z2} + m_2 r_2^2.
\end{aligned}
$$

It is perhaps surprising that such a simple problem can have such an unwieldy solution. The difficulty is that we have cast the entire problem into the joint space of the manipulator, where the constraint $\theta = f(s)$ is a very complex looking curve.

A better way of deriving the equations of motion for this system is to rewrite the dynamics of the system in terms of workspace variables which describe the position of the end-effector (see Exercise 12). Once written in this way, the constraint that the end of the manipulator remain in the slot is a very simple one. This is the basic approach used in Chapter 6, where we present a general framework which incorporates this example and many other constrained manipulation systems.

7 Summary

The following are the key concepts covered in this chapter:

1. The equations of motion for a mechanical system with Lagrangian $L = T(q, \dot{q}) - V(q)$ satisfies *Lagrange's equations*:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \Upsilon_i,$$

where $q \in \mathbb{R}^n$ is a set of generalized coordinates for the system and $\Upsilon \in \mathbb{R}^n$ represents the vector of generalized external forces.

2. The equations of motion for a rigid body with configuration $g(t) \in SE(3)$ are given by the *Newton-Euler equations*:

$$\begin{bmatrix} mI & 0 \\ 0 & \mathcal{I} \end{bmatrix} \begin{bmatrix} \dot{v}^b \\ \dot{\omega}^b \end{bmatrix} + \begin{bmatrix} \omega^b \times mv^b \\ \omega^b \times \mathcal{I}\omega^b \end{bmatrix} = F^b,$$

where m is the mass of the body, $\mathcal{I}$ is the inertia tensor, and $V^b = (v^b, \omega^b)$ and F^b represent the instantaneous body velocity and applied body wrench.

3. The equations of motion for an open-chain robot manipulator can be written as

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) = \tau$$

where $\theta \in \mathbb{R}^n$ is the set of joint variables for the robot and $\tau \in \mathbb{R}^n$ is the set of actuator forces applied at the joints. The dynamics of a robot manipulator satisfy the following properties:

   (a) $M(\theta)$ is symmetric and positive definite.

   (b) $\dot{M} - 2C \in \mathbb{R}^{n\times n}$ is a skew-symmetric matrix.

4. An equilibrium point x^* for the system $\dot{x} = f(x, t)$ is *locally asymptotically stable* if all solutions which start near x^* approach x^* as $t \to \infty$. Stability can be checked using the *direct method of Lyapunov*, by finding a locally positive definite function $V(x, t) \geq 0$ such that $-\dot{V}(x, t)$ is a locally positive definite function along trajectories of the system. In situations in which $-\dot{V}$ is only positive semi-definite, *Lasalle's invariance principle* can be used to check asymptotic stability. Alternatively, the *indirect method of Lyapunov* can be employed by examining the linearization of the system, if it exists. Global exponential stability of the linearization implies local exponential stability of the full nonlinear system.

5. Using the form and structure of the robot dynamics, several control laws can be shown to track arbitrary trajectories. Two of the most common are the *computed torque control law*,

$$\tau = M(\theta)(\ddot{\theta}_d + K_v \dot{e} + K_p e) + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}),$$

   and an *augmented PD control law*,

$$\tau = M(\theta)\ddot{\theta}_d + C(\theta, \dot{\theta})\dot{\theta}_d + N(\theta, \dot{\theta}) + K_v \dot{e} + K_p e.$$

   Both of these controllers result in exponential trajectory tracking of a given joint space trajectory. Workspace versions of these control laws can also be derived, allowing end-effector trajectories to be tracked without solving the inverse kinematics problem. Stability of these controllers can be verified using Lyapunov stability.

8 Bibliography

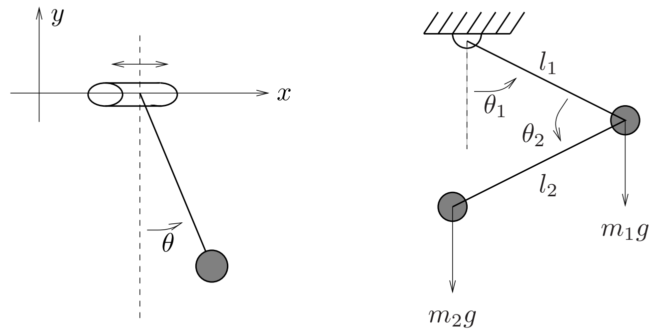
The Lagrangian formulation of dynamics is classical; a good treatment can be found in Rosenberg [99] or Pars [89]. Its application to the dynamics of a robot manipulator can be found in most standard textbooks on robotics, for example [2, 21, 35, 52, 110].

The geometric formulation of the equations of motion for kinematic chains presented in Section 3.3 is based on the recent work of Brockett, Stokes, and Park [15, 87]. This is closely related to the spatial operator algebra formulation of Rodriguez, Jain, and Kreutz-Delgado [45, 98], in which the tree-like nature of the system is more fully exploited in computing inertial properties of the system.

The literature on control of robot manipulators is vast. An excellent treatment, covering many of the different approaches to robot control, is given by Spong and Vidyasagar [110]. The collection [109] also provides a good survey of recent research in this area. The modified PD control law presented in Section 5 was originally formulated by Koditschek [51]. For a survey of manipulator control using exact linearization techniques, see Kreutz [53]. The use of skew terms in Lyapunov functions to prove exponential stability for PD controllers has been pointed out, for example, by Wen and Bayard [120].

9 Exercises

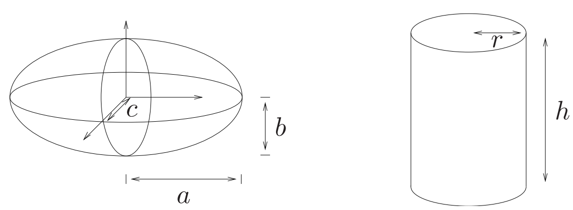
1. Derive the equations of motion for the systems shown below.



(a) (b)

   (a) Pendulum on a wire: an idealized planar pendulum whose pivot is free to slide along a horizontal wire. Assume that the top of the pendulum can move freely on the wire (no friction).
   (b) Double pendulum: two masses connected together by massless links and revolute joints.

2. Compute the inertia tensor for the objects shown below.



(a) Ellipsoid (b) Cylinder

3. *Transformation of the generalized inertia matrix*
   Show that under a change of body coordinate frame from B to C, the generalized inertia matrix for a rigid body is given by

$$\mathcal{M}_c = \mathrm{Ad}_{g_{bc}}^T \, \mathcal{M}_b \, \mathrm{Ad}_{g_{bc}} = \begin{bmatrix} mI & mR_{bc}^T \widehat{p}_{bc} R_{bc} \\ -mR_{bc}^T \widehat{p}_{bc} R_{bc} & R_{bc}^T(\mathcal{I} - m\widehat{p}_{bc}^2) R_{bc} \end{bmatrix},$$

   where g_{bc} denotes the rigid motion taking C to B, and $\mathcal{M}_b$ and $\mathcal{M}_c$ are the generalized inertia matrices expressed in frame B and frame C.

4. Show that Euler's equation written in spatial coordinates is given by

$$\mathcal{I}'\dot{\omega}^s + \omega^s \times \mathcal{I}'\omega^s = \tau,$$

where $\mathcal{I}' = \mathcal{R}\mathcal{I}\mathcal{R}^{\mathcal{T}}$ and τ is the torque applied to the center of mass of the rigid body, written in spatial coordinates.

5. Calculate the Newton-Euler equations in spatial coordinates.

6. Show that it is possible to choose M and C such that the Newton-Euler equations can be written as

$$M\dot{V}^b + C(g, \dot{g})V^b = F^b,$$

where $M > 0$ and $\dot{M} - 2C$ is a skew-symmetric matrix.

7. Verify that the equations of motion for a planar, two-link manipulator, as given in equation (4.11), satisfy the properties of Lemma 4.2.

8. *Passivity of robot dynamics*
Let $H = T + V$ be the total energy for a rigid robot. Show that if $\dot{M} - 2C$ is skew-symmetric, then energy is conserved, i.e., $\dot{H} = \dot{\theta} \cdot \tau$.

9. Show that the workspace version of the PD control law results in exponential trajectory tracking.

10. Show that the control law

$$\tau = M(\theta)(\ddot{\theta}_d + \lambda\dot{e}) + C(\theta, \dot{\theta})(\dot{\theta}_d + \lambda e) + N(\theta, \dot{\theta}) + K_v\dot{e} + K_p e$$

results in exponential trajectory tracking when $\lambda \in \mathbb{R}$ is positive and $K_v, K_p \in \mathbb{R}^{n\times n}$ are positive definite [107].

11. Show that the matrix

$$\begin{bmatrix} \epsilon A & \epsilon B \\ \epsilon B^T & C + \epsilon D \end{bmatrix}$$

is positive definite if A and C are symmetric, positive definite, and $\epsilon > 0$ is chosen sufficiently small.

12. *Hybrid control using workspace coordinates*
Consider the constrained manipulation problem described in Section 6.3. Let $p_{st}(\theta) \in \mathbb{R}^2$ be the coordinates of the end-effector and let $w = p(\theta)$ represent a set of workspace coordinates for the system.

    (a) Compute the matrix $J(\theta)$ which is used to convert the joint space dynamics into workspace dynamics (as in Section 5.4).

    (b) Compute the constraint function in terms of the workspace variables and find a parameterization $f : \mathbb{R} \to \mathbb{R}^2$ which maps the slot position to the workspace coordinates. Let $K(s)$ represent the Jacobian of the mapping $w = f(s)$.

(c) Write the dynamics of the constrained system in terms of ω and its derivatives, the dynamic parameters of the unconstrained system, and the matrices $J(\theta)$ and $K(s)$.

(d) Verify that the equations of motion derived in step (c) are the same as the equations of motion derived in Section 6.3. In particular, show that τ_N and the inertia matrix $\tilde{M}(s)$ are the same in both cases.