Control of Robot Manipulators

F. L. Lewis
University of Texas at Arlington

C. T. Abdallah
University of New Mexico

D. M. Dawson
Clemson University

1993

MACMILLAN PUBLISHING COMPANY
New York

MAXWELL MACMILLAN CANADA
Toronto

MAXWELL MACMILLAN INTERNATIONAL
New York   Oxford   Singapore   Sydney

2.4 State-Variable Representations and Feedback Linearization

The robot arm dynamical equation in Table 2.3-1 is

$$M(q)\ddot{q} + V(q,\dot{q}) + F_v\dot{q} + F_d(\dot{q}) + G(q) + \tau_d = \tau, \tag{2.4-1}$$

with $q(t) \varepsilon \mathbb{R}^n$ the joint variable vector and $\tau(t)$ the control input. $M(q)$ is the inertia matrix, $V(q,\dot{q})$ the Coriolis/centripetal vector, $F_v\dot{q}$ the viscous friction, $F_d(\dot{q})$ the dynamic friction, $G(q)$ the gravity, and τ_d a disturbance. These terms satisfy the properties shown in Table 2.3-1. We may also write the dynamics as

$$M(q)\ddot{q} + N(q,\dot{q}) + \tau_d = \tau, \tag{2.4-2}$$

with the nonlinear terms represented by

$$N(q,\dot{q}) \equiv V(q,\dot{q}) + F_v\dot{q} + F_d(\dot{q}) + G(q). \tag{2.4-3}$$

In this section we intend to show some equivalent formulations of the arm dynamical equation.

The nonlinear state-variable representation discussed in Chapter 1,

$$\dot{x} = f(x,u,t) \tag{2.4-4}$$

has many properties which are useful from a controls point of view. The function $u(t)$ is the control input and $x(t)$ is the state vector, which describes how the energy is stored in a system. We show here how to place (2.4-1) into such a form. In Chapter 3 we show how to use computers to *simulate* the behavior of a robot arm using this nonlinear state-variable form. Throughout the book we shall use the state-space formulation repeatedly for controls design, either in the nonlinear form or in the linear form

$$\dot{x} = Ax + Bu. \tag{2.4-5}$$

In this section we also present a general approach to *feedback linearization* for the nonlinear robot equation, which involves redefining variables in a methodical way to yield a linear state equation in terms of a dynamical variable we are interested in. This variable could be, for instance, the joint variable $q(t)$, a Cartesian position, or the position in a camera frame of reference.

Hamiltonian Formulation

The arm equation was derived using Lagrangian mechanics. Here, let us use Hamiltonian mechanics [Marion 1965] to derive a state-variable formulation of the manipulator dynamics [Arimoto and Miyazaki 1984, Gu and Loh 1985]. Let us neglect the friction terms $F(\dot{q}) = F_v\dot{q} + F_d(\dot{q})$ and the disturbance τd for simplicity; they may easily be added at the end of our development.

In Section 2.2 we expressed the arm Lagrangian as

$$L = K - P = \tfrac{1}{2}\dot{q}^T M(q)\dot{q} - P(q) \tag{2.4-6}$$

with $q(t)\,\varepsilon R^n$ the joint variable, K the kinetic energy, P the potential energy, and $M(q)$ the arm inertia matrix. Define the *generalized momentum* by

$$p \equiv \frac{\partial L}{\partial \dot{q}} = M(q)\dot{q}. \tag{2.4-7}$$

Then we have

$$\dot{q} = M^{-1}(q)p \tag{2.4-8}$$

and the kinetic energy in terms of $p(t)$ is

$$K = \tfrac{1}{2}p^T M^{-1}(q)p. \tag{2.4-9}$$

It is worth noting that

$$K = \tfrac{1}{2}p^T\dot{q}. \tag{2.4-10}$$

Defining the *manipulator Hamiltonian* by

$$H = p^T\dot{q} - L, \tag{2.4-11}$$

Hamilton's equations of motion are

$$\dot{q} = \frac{\partial H}{\partial p} \tag{2.4-12}$$

$$-\dot{p} = \frac{\partial H}{\partial q} - \tau. \tag{2.4-13}$$

Note that

$$H = \tfrac{1}{2}p^T M^{-1}(q)p + P(q) = K + P. \tag{2.4-14}$$

Evaluating (2.4-13) yields

$$\dot{p} = \frac{1}{2}\frac{\partial}{\partial q}(p^T M^{-1}(q)p) - \frac{\partial P}{\partial q} + \tau,$$

which may be expressed (see the Problems) as

$$\dot{p} = -\frac{1}{2}(I_n \otimes p^T)\frac{\partial M^{-1}(q)}{\partial q}p - G(q) + \tau, \tag{2.4-15}$$

where $G(q)$ is the gravity vector and $\otimes$ is the Kronecker product (see Section 2.3).

Defining the state vector $x \in R^{2n}$ as

$$x = [q^T \quad p^T]^T, \tag{2.4-16}$$

we see that the arm dynamics may be expressed as

$$\frac{d}{dt}\begin{bmatrix} q \\ p \end{bmatrix} = \begin{bmatrix} M^{-1}(q)p \\ -\frac{1}{2}\,(I_n \otimes p^T)\,\dfrac{\partial M^{-1}(q)}{\partial q}\,p \end{bmatrix} + \begin{bmatrix} 0 \\ I_n \end{bmatrix} u, \tag{2.4-17}$$

with the control input defined by

$$u(t) = \tau - G(q). \tag{2.4-18}$$

This is a nonlinear state equation of the form (2.4-4). It is important to note that this dynamical equation is *linear* in the control input u, which excites *each component* of the generalized momentum $p(t)$.

This *Hamiltonian state-space formulation* was used to derive a PID control law using the Lyapunov approach in [Arimoto and Miyazaki 1984] and to derive a trajectory-following control in [Gu and Loh 1985].

Position/Velocity Formulations

Alternative state-space formulations of the arm dynamics may be obtained by defining the position/velocity state $x \in R^{2n}$ as

$$x = [q^T \quad \dot{q}^T]^T. \tag{2.4-19}$$

For simplicity, neglect the disturbance τ_d and friction $F_v\dot{q} + F_d(\dot{q})$ and note that according to (2.4-2), we may write

$$\frac{d}{dt}\,\dot{q} = -M^{-1}(q)N(q,\dot{q}) + M^{-1}(q)\tau. \tag{2.4-20}$$

Now, we may directly write the position/velocity state-space representation

$$\dot{x} = \begin{bmatrix} \dot{q} \\ -M^{-1}(q)N(q,\dot{q}) \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}(q) \end{bmatrix} \tau, \tag{2.4-21}$$

which is in the form of (2.4.4) with $u(t) = \tau(t)$.

An alternative *linear* state equation of the form (2.4-5) may be written as

$$\dot{x} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ I \end{bmatrix} u, \tag{2.4-22}$$

with control input defined by

$$u(t) = -M^{-1}(q)N(q,\dot{q}) + M^{-1}(q)\tau. \tag{2.4-23}$$

Both of these position/velocity state-space formulations will prove useful in later chapters.

Feedback Linearization

Let us now develop a general approach to the determination of *linear* state-space representations of the arm dynamics (2.4-1)–(2.4-2). The technique involves a linearization transformation that removes the manipulator nonlinearities. It is a simplified version of the *feedback linearization* technique in [Hunt et al. 1983, Gilbert and Ha 1984]. See also [Kreutz 1989].

The robot dynamics are given by (2.4-2) with $q \in \mathbb{R}^n$. Let us define a general sort of output by

$$y = h(q) + s(t), \tag{2.4-24}$$

with $h(q)$ a general predetermined function of the joint variable $q \in \mathbb{R}^n$ and $s(t)$ a general predetermined time function. The control problem, then, will be to select the joint torque and force inputs $\tau(t)$ in order to make the output $y(t)$ go to zero.

The selection of $h(q)$ and $s(t)$ is based on the control objectives we have in mind. For instance, if $h(q) = -q$ and $s(t) = q_d(t)$, the desired joint space trajectory we would like the arm to follow, then $y(t) = q_d(t) - q(t) \equiv e(t)$ the *joint space tracking error.* Forcing $y(t)$ to zero in this case would cause the joint variables $q(t)$ to track their desired values $q_d(t)$, resulting in arm trajectory following.

As another example, $y(t) = [e_p^T \ \ e_0^T]^T$ could represent the *Cartesian space* tracking error, with $e_p \in \mathbb{R}^3$ the position error and $e_0 \in \mathbb{R}^3$ the orientation error. Controlling $y(t)$ to zero would then result in trajectory following directly in *Cartesian space,* which is, after all, where the desired motion is usually specified.

Finally, $-h(q)$ could represent the nonlinear transformation to a *camera frame of reference* and $s(t)$ the desired trajectory in that frame. Then $y(t)$ is the camera frame tracking error. Forcing $y(t)$ to zero would then result in tracking motion in *camera space.*

Feedback Linearizing Transformation. To determine a linear state-variable model for robot controller design, let us simply differentiate the output $y(t)$ twice to obtain

$$\dot{y} = \frac{\partial h}{\partial q}\dot{q} + \dot{s} \equiv J\dot{q} + \dot{s} \tag{2.4-25}$$

$$\ddot{y} = \dot{J}\dot{q} + J\ddot{q} + \ddot{s}, \tag{2.4-26}$$

where we have defined the Jacobian

$$J(q) \equiv \frac{\partial h(q)}{\partial q}. \tag{2.4-27}$$

If $y \varepsilon \mathbb{R}^p$, the Jacobian is a $p \times n$ matrix of the form

$$J(q) = \frac{\partial h(q)}{\partial q} = \left[\frac{\partial h}{\partial q_1}\frac{\partial h}{\partial q_2}\cdots\frac{\partial h}{\partial q_n}\right]. \tag{2.4-28}$$

Given the function $h(q)$, it is straightforward to compute the Jacobian $J(q)$ associated with $h(q)$. In the special case where $\dot{y}$ represents the Cartesian velocity, $J(q)$ is the arm Jacobian discussed in Appendix A. Then, if all joints are revolute, the units of J are those of length.

According to (2.4-2),

$$\ddot{q} = M^{-1}(-N - \tau_d + \tau), \tag{2.4-29}$$

so that (2.4-26) yields

$$\ddot{y} = \ddot{s} + \dot{J}\dot{q} + JM^{-1}(-N - \tau_d + \tau). \tag{2.4-30}$$

Define the *control input* function

$$u(t) = \ddot{s} + \dot{J}\dot{q} + JM^{-1}(-N + \tau) \tag{2.4-31}$$

and the *disturbance* function

$$v(t) = -JM^{-1}\tau_d. \tag{2.4-32}$$

Now we may define a state $x(t) \varepsilon \mathbb{R}^{2p}$ by

$$x = [y^T \quad \dot{y}^T]^T \tag{2.4-33}$$

and write the robot dynamics as

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & I_p \\ 0 & 0 \end{bmatrix}\begin{bmatrix} y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ I_p \end{bmatrix} u + \begin{bmatrix} 0 \\ I_p \end{bmatrix} v. \tag{2.4-34}$$

This is a linear state-space system of the form

$$\dot{x} = Ax + Bu + Dv, \tag{2.4-35}$$

driven both by the control input $u(t)$ and the disturbance $v(t)$. Due to the special form of A and B, this system is said to be in *Brunovsky canonical form* (Chapter 1). The reader should determine the controllability matrix to verify that it is always controllable from $u(t)$.

Equation (2.4-31) is said to be a *linearizing transformation* for the robot dynamical equation. We may invert this transformation to obtain

$$\tau = MJ^+[u - \ddot{s} - \dot{J}\dot{q}] + N, \tag{2.4-36}$$

where J^+ is the *Moore–Penrose inverse* [Rao and Mitra 1971] of the Jacobian $J(q)$. If $J(q)$ is square (i.e., $p = n$) and nonsingular, then $J^+(q) = J^{-1}(q)$ and we may write

$$\tau = MJ^{-1}[u - \ddot{s} - \dot{J}\dot{q}] + N. \tag{2.4-37}$$

As we shall see in Chapter 3, feedback linearization provides a powerful controls design technique. In fact, if we select $u(t)$ so that (2.4-34) is stable (e.g., a possibility is the PD feedback $u = -K_v\dot{y} - K_p y$), then the control input torque $\tau(t)$ defined by (2.4-36) makes the robot arm move in such a way that $y(t)$ goes to zero.

In the special case $y(t) = q(t)$, then $J = I$ and (2.4-34) reduces to the linear position/velocity form (2.4-22).

2.5 Cartesian and Other Dynamics

In Section 2.2 we derived the robot dynamics in terms of the time behavior of $q(t)$. According to Table 2.3-1,

$$M(q)\ddot{q} + V(q,\dot{q}) + F_v\dot{q} + F_d(\dot{q}) + G(q) + \tau_d = \tau \tag{2.5-1}$$

or

$$M(q)\ddot{q} + N(q,\dot{q}) + \tau_d = \tau, \tag{2.5-2}$$

where the nonlinear terms are

$$N(q,\dot{q}) \equiv V(q,\dot{q}) + F_v\dot{q} + F_d(\dot{q}) + G(q). \tag{2.5-3}$$

We call this the dynamics of the arm formulated in joint space, or simply the *joint-space dynamics.*

Cartesian Arm Dynamics

It is often useful to have a description of the dynamical development of variables other than the joint variable $q(t)$. Consequently, define

$$y = h(q) \tag{2.5-4}$$

with $h(q)$ a generally nonlinear transformation. Although $y(t)$ could be any variable of interest, let us think of it here as the Cartesian or task space position of the end effector (i.e., position and orientation of the end effector in base coordinates).

The derivation of the Cartesian dynamics from the joint-space dynamics is akin to the feedback linearization in Section 2.4. Differentiating (2.5-4) twice yields

$$\dot{y} = J\dot{q} \tag{2.5-5}$$

$$\ddot{y} = J\ddot{q} + \dot{J}\dot{q}, \tag{2.5-6}$$