Katie Byl

Robot Dynamics and Control

October 2, 2012

For Pieter. Never stop smiling.

Preface

These are the course notes for UCSB class ECE/ME179D, which addresses problems of modeling and controlling the dynamic response of a robot. Material is aimed at advanced undergraduate or early graduate students studying either electrical or mechanical engineering.

This is an early draft and is therefore almost guaranteed to contain many errors and typos. I encourage you to contact me with corrections or other suggestions at my email address below.

UC Santa Barbara, CA

Katie Byl
katiebyl@gmail.com

Contents

Chapter 1
Introduction

Abstract This chapter introduces concepts and terminology in robotics and defines the scope of the text.

1.1 History of Robotics

The word "robot" was first used in the 1920 play "Rossum's Universal Robots" (English translation title) by Czech author Karel Čapek (1890-1938), who gratiously credits the invention of the word to his close brother, Josef. In Czech, the word "robota" refers to unfree labor. Its origins are thought to derive from the Proto-Slavic language root "orbota", meaning work or slavery, from which are derived both the German noun Arbeit, meaning work, and the Russian word работа [rabota], which is an often perjoratively-used term for work or labor. Thus,"Roboti" (robots) originally evoked some notion of unfree labor at a time (note that serfdom was abolished for the Czechs in Bohemia in 1848 and abolished in Russian by Alexander II in 1861, while slavery was abolished by the 13^{th} Amendent in the US in 1865) when these practices were still within living memory.

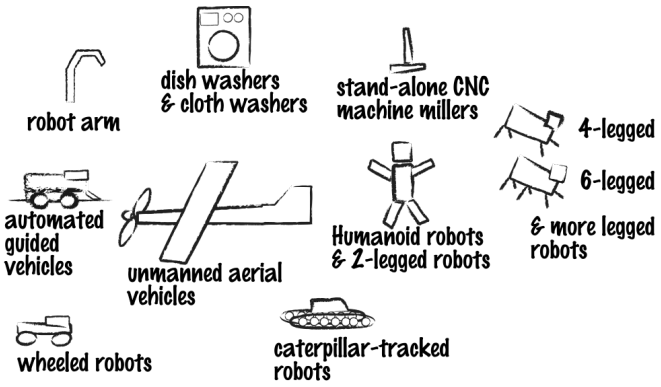

Fig. 1.1 Robot Examples. A robot is a machine that can perform physical tasks in an automated way. By this definition, robots can take many forms, as shown above.

By today's definition, robots are machines that perform various types of **work**, usually with some degree of **autonomy**. From an engineering perspective, "work" involves forces and displacements: things are moved about in the physical world. Robots manipulate other objects and/or manipulate themselves (i.e., locomote).

Not surprisingly, the first practical "robots" to be developed were arms, rather than full human-like forms. A multi-link arm can be modeled as a dynamic system that is rigidly mounted to a base (e.g., the body of a human, or the floor of a factory), such that each degree of freedom (DOF) if fully actuated (as opposed to being underactuated). This simply means that the actuators are capable of creating an arbitrary combination of accelerations on each of the degrees of freedom of the robot – at least within the limits of power of the system. Full actuation simplifies the problem of manipulation to a great extent.

The terms "underactuated" and "under-powered" can bring on tremendous confusion, so a few words are required to distinguish the two. Under-powered systems simply cannot provide a large enough force (or torque) to complete a desired task. Underactuation, by contrast, means that one is trying to control N degrees of freedom (say, joint angles) of a robot by using, effectively, fewer than N independent actuators.

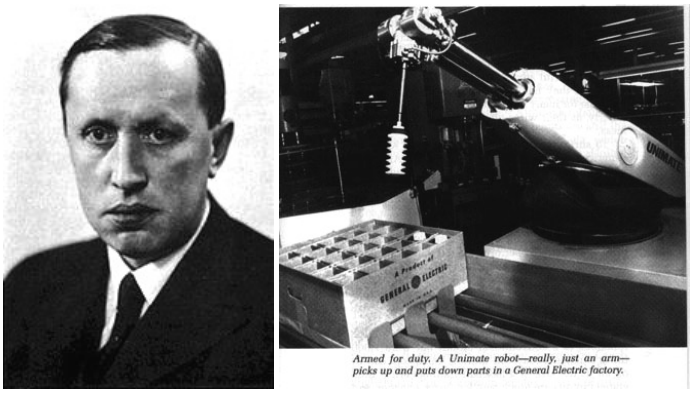

Fig. 1.2 Early robot history. Karel Čapek (at left), who conceptualized the "robot" as a humanoid worker, and the first industrial robot, the 1961 Unimate, which took the practical approach of operating as a robot arm, rather than copying a full and complex anthropomorphic form.

1.2 Scope of the Text

This text focuses on Dynamics and Control, presenting material for a 10-week course aimed at advanced undergraduate and early graduate students. By **Dynamics**, we mean the equations of motion describing how forces result in motion, and by **Control**, we mean a set of man-made rules by which a system's natural dynamics are replaced by the dynamics we wish to have. Thus, this course (unfortunately) cannot practically cover a large volume of the material needed for expertise in robotics. Some topics, in particular **Kinematics**, must be addressed at least briefly, but the coverage is minimal, toward developing concepts in dynamics and control, only.

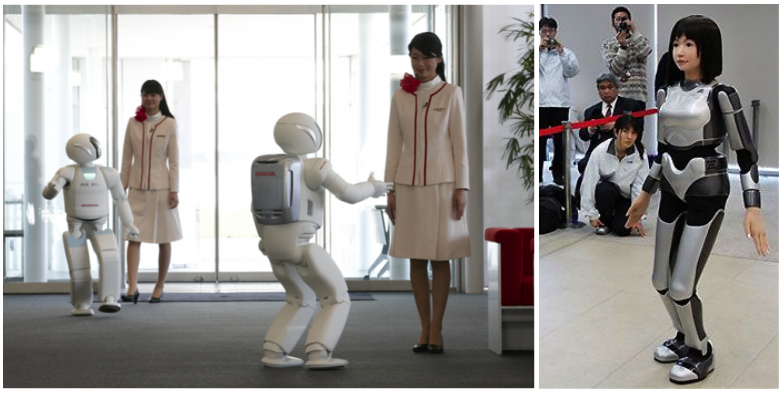

Fig. 1.3 Humanoid robots. ASIMO robots (left) and an HRP-4C ("c" for cybernetic) robot (right).

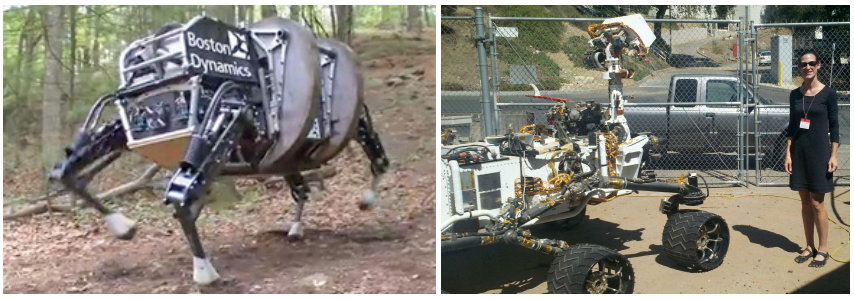

Fig. 1.4 Rough terrain mobile robots. An LS3 (Legged Squad Support System) quadruped robot (left) and an on-earth twin of the Curiosity Mars rover (right), with six articulated wheels. (The author is at right, for scale.)

Fig. 1.5 Autonomous vehicle robots. A Google self-driving car (left) and a Predator drone in Iraq (right). As such robots become more wide-spread, a large range of ethical questions arise in determining both their appropriate use and who is responsible when technical (and contextual) mistakes inevitably occur.

1.3 Terminology: Joint space, task space and end effector

With regard to dynamics and control, the first distinction to make is whether the coordinates used to describe motions relate directly to the **actuators** or to the **end effectors**. An actuator could include, for example, an electric motor that moves a particular joint of the robot. And end effector refers to a part of the robot that is intended to interact with the rest of the world in some way. Since sensors (e.g., encoders on a motor) often measure actuator motion directly, it is convenient *for*

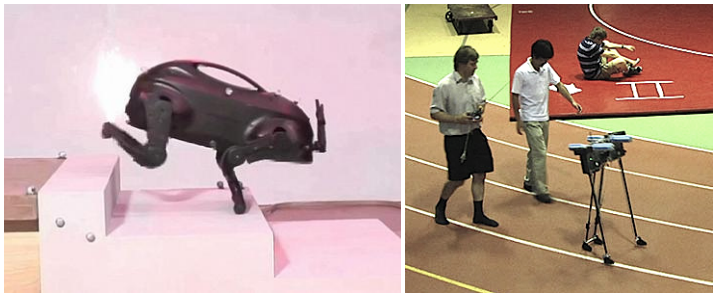

Fig. 1.6 Robots in academia. Littledog (at left), demonstrating underactuated locomotion by performing a dynamic lunge at MIT, and Cornell's Ranger (at right), which currently holds the world distance record for a legged robot that must carry its own power, without recharging/refueling: walking 40.5 miles via remote control guidance.

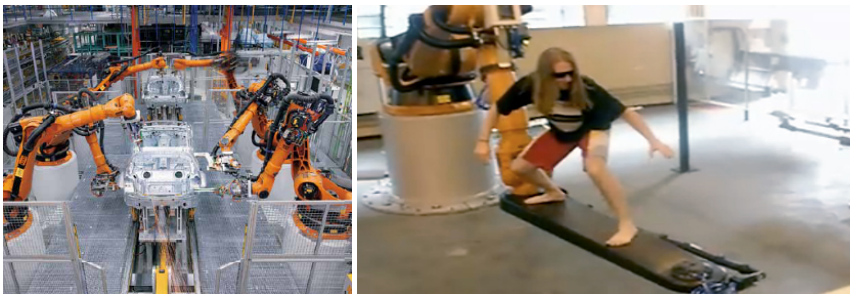

Fig. 1.7 Modern industrial robot arms. The multi-link robot arm is still an important participant in modern manufacturing. Such arms are typically designed for high positional accuracy, which results in "stiff" dynamics. In other words, these robots are quite strong (and a bit scary to be near). Above, a team of Kuka robot arms are at work in an automotive plant (at left), and an engineer demonstrates the strength and scale of a Kuka arm by "surfing" on a board held by one.

an engineer to control theses coordinates directly. But since the goal of a robot is, essentially, manipulation (moving) of objects, and since the end effector is the part interacting with the world, it is typically more desirable *for the user of the robot* to have dynamics that are controlled with respect to the positions and orientation (and the velocities of these coordinates, too), of the end effector.

When we discuss displacements and velocities of actuated joints, this is referred to as the "joint space" of the robot. When we discuss displacements and velocities of the end effector, which is controlled to complete some specific task, this is referred to as the "task space".

Robotics involves a large repertoire of terms that are specific to the field. Many of these are defined – concisely – in the remainder of this Section.

Definitions:

Manipulator : a multi-link arm.
Configuration : a specification of all points, geometrically, on a robot.
Configuraton Space : the set of all possible configurations.
Kinematics : Geometric relationship of motion, e.g., from joint angles to end effector position and orientation.
Degrees of Freedom (DOFs) : Independent rotational or translational directions of motion that are presently possible.

Revolute Joint : A rotational DOF.
Prismatic Joint : A translational DOF.

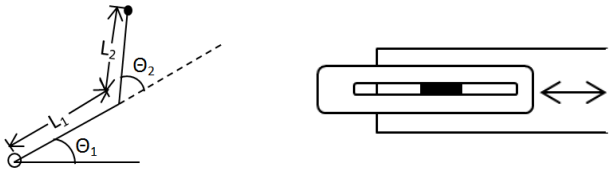

Fig. 1.8 Two-link planar arm with two revolute joints (left) and a prismatic joint (right).

Pitch, theta, θ : Rotation about y axis.
Roll, phi, ϕ : Rotation about x axis.
Yaw, psi, ψ : Rotation about z axis.

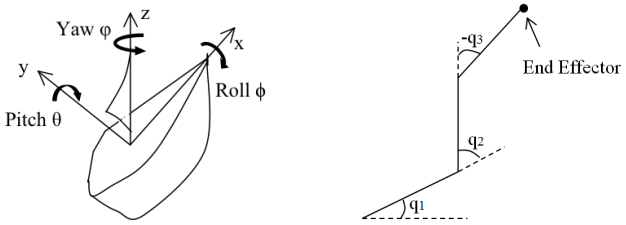

Fig. 1.9 At left: Euler angle representation for the orientation (i.e., 3D rotations) of an object: roll, pitch, and yaw. At right: A simple, 3-link planar arm, showing an "end effector". The joint DOFs are q_1, q_2, and q_3, each measured as positive in the counter-clockwise direction.

Joint Variables, q : DOFs for joints, which may depend on the robot configuration.
Joint Space : The set of all possible joint variable combinations.
State Space : The set of all combinations of geometry AND dynamics, i.e., positions and velocities, that are possible.
Dynamics : Relationships between forces and motions, e.g., Newtons Laws.
End Effector : The point(s) of the robot (and most typically, of a robot manipulator) designed to interact with the environment, e.g., a tool tip. (See Fig. 1.9.)
Workspace : The set of all points reachable by the end effector.
Reachable Workspace : See above definition for Workspace.
Dextrous Workspace : The set of all points that the end effector can reach at any, arbitrary orientation (e.g., roll, pitch, yaw in 3D space; or just angle for a 2D robot).
Kinematic Redundancy : When multiple joint solutions exist to put an end effector either at a particular position or a particular position and orientation (depending on whether dexterity is required for a given task).
Forward Kinematics : Mapping from joint geometry to end effector geometry. One unique forward kinematic mapping exists, always.
Inverse Kinematics : Mapping from end effector to joint geometry. There may be multiple solutions (i.e., kinematically redundant). There may be no solution (i.e., kinematically infeasible).

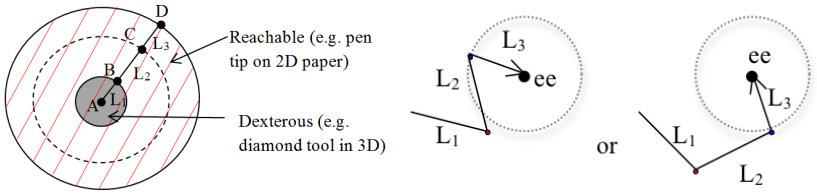

Fig. 1.10 Left: Reachable versus Dexterous Workspace. Middle and Right: For a test point to be within the Dexterous Workspace, the first two links, L_1 and L_2, must be capable of reaching every point a distance L_3 from the desired end effector location.

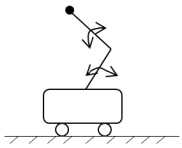

Fig. 1.11 Kinematic Redundancy Example: Mobile Cart with Arm. For many desired end effector locations, there may be an infinite number of combinations of cart position and arm orientation to reach this (x,y) coordinate with the end effector. To visualize this, picture the cart moving left to right, while the end effector points at the same, fixed point in space.

Singularity : Loss of an independent DOF (i.e., direction of instantaneous motion) of the end effector. [To be discussed at length in Section 1.4.]

The Jacobian : A matrix representation of the velocity relationships between joint DOFs and end effector DOFs, for a given geometry, which usually depends on the configuration of a robot. [See Section 1.4.]

Backlash : The small motion that occurs when the gears in a gearbox reverse direction, before the gears re-engage with one another fully. Backlash is an undesirable "wiggle" in many types of gearboxes that can cause some uncertainty between measured encoder positions and actual joint and end effector locations.

Accuracy : How close to a desired target one can get (i.e., mean error with respect to a goal). The difference between the average trial value and the desired target is called a **bias**.

Repeatability : How closely repeated trials are to one another (i.e., variance in error). Also called **precision**.

1.3.1 Robot Control Overview

Much of the development in robot control was driven by the increase in practical use of industrial robots throughout the 1980's for factory automation. These robots are multi-link arms, programmed for highly repetitive tasks. They operate in a well-known environment, usually without people nearby who can be harmed by their motions, and they can be quite powerful and strong. The high-gain feedback control increases accuracy of planned trajectories, but it also makes such robots potentially dangerous in environments that are not perfectly known, or where people are interacting with them. Often, more recent robotsfor non-factory use are designed to be

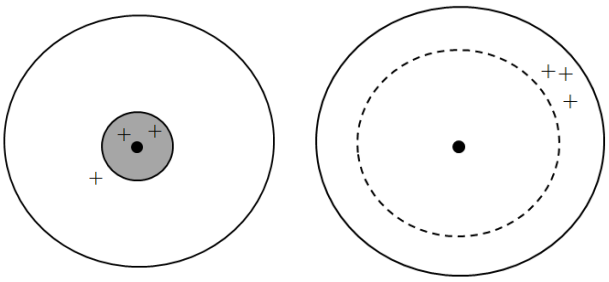

Fig. 1.12 Accuracy vs Repeatability. Accuracy (left) refers to repeated trials being close on average to a desired goal, without much bias. Repeatability (right) refers to repeated trials always giving close to the same result, regardless of whether it is accurately aimed at the target or not. Backlash of gears, the weight of gravity, and improperly modeled kinematics can all cause a bias, resulting in low accuracy of a robot. Noisy measurements or variability in the loads being manipulated can cause variability in output, resulting in low repeatability.

much more gentle. Setting the trade-offs between accuracy and gentleness are a significant challenge in the field of Robot Dynamics and Control. The first commercial, industrial robot arm was the Unimate robot arm, shown in Figure 1.2; examples of modern industrial robot arms are shown in Figure 1.7.

It is important to note that the dynamics of such robot manipulators depend on the **configuration** of the joints and on the **load** (e.g., mass and inertia of objects) being manipulated. As a result, these robot arms are:

1. Not linear (i.e., nonlinear).
2. Varying of time (time variable)

For example, sin and cos functions and velocity-squared terms typically appear in the equations of motion, making them nonlinear, and the loads being manipulated change over time, making the dynamics time-varying.

By contrast, introductory controls classes start by assuming systems are:

1. Linear.
2. Time invariant.

Both classic and modern control theory begins by assuming the dynamic system to be controlled (which is referred to as the **plant**) is "linear, time-invariant" (LTI). The dynamics of such systems can be described using constant-coefficient differential equations. The most typical goal in controls is to **modify the natural dynamics** of a plant such that the controlled (aka, **closed-loop**) plant behaves either a first-order or second-order system.

As an aside, note that real-world systems can have a very large number of poles. The poles describe the natural response the system will have if you start it with some initial condition. When we observe such outputs, it turns out the ones that dominate the response are those fursthest to the right on the s-plane. If the poles are all to the left of the imaginary axis, their real parts all correspond to stable, decaying exponential envelopes in the system response. The slowest of these decay envelope(s) is the one closest to the imaginary axis, and it "dominates" because the response"lingers" the longest over time, due to the slow decay rate.

By the way, note that if any pole(s) are to the right of the imaginary axis, the system is of course not stable, and the pole(s) furthest to the right still dominate.

However, positive real parts for poles correspond to growing exponential envelopes. As the real part of an unstable pole (or pole pair) is moved to the right, its contribution to the system response grows unstable more and more rapidly!

So, by saying we wish a closed-loop system to behave like either a first- or a second-order system, what we really mean is that it should have either a single, real-valued **dominant pole** or a complex dominant pole pair. For the pole(s) to be dominant, there must be a significant separation (say, a factor of 10 or more) between the slowest pole (or pole pair) and all other closed-loop poles.

Recall that a first-order system has a transfer function of the following form:

$$\frac{Y(s)}{X(s)} = \frac{K}{\tau s + 1} \tag{1.1}$$

where, τ is the time constant of the system, and K is the final value of the output (as time goes to infinity) given a unit step input. Similarly, a second-order system transfer function can be written as:

$$\frac{Y(s)}{X(s)} = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1.2}$$

where ω_n is the (undamped) natural frequency, ζ is the damping ratio, and the final value of the system after a unit step input is K.

Here, the expression $Y(s)/X(s)$ is a **transfer function** describing the dynamic relationship between input $x(t)$ and output $y(t)$. In these notes, we assume you are familiar with the Laplace transform and with Laplace notation, in general. As a reminder, recall that the Laplace operator s can be thought, loosely, to represent the derivative operator d/dt in the time domain response. So, Equations 1.1 and 1.2 can be written in the time domain, respectively, as shown below:

$$\tau dy/dt + y = Kx \tag{1.3}$$

$$d^2y/dt^2 + 2\zeta\omega_n dy/dt + \omega_n^2 y = K\omega_n^2 x \tag{1.4}$$

From these expressions, one can imagine that if the system is stable (i.e., poles in lefthalf plane, only), then after the transient response dies away, all derivatives in y will be insignificant. It is left for the reader to verify what the final value of y must then be if the value of x is held constant at 1 (e.g., for a unit step in x).

1.3.2 Kinematics: Overview

From our earlier definition, kinematics describe robot geometry. Given some set of joint space coordinates (anglular rotations and/or prismatic joint translations), there will be one, unique solution for the **forward kinematics** giving the resulting task space coordinates, ξ. The task space coordinates for a 6-DOF end effector in 3D space would be:

$$\xi = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix} \tag{1.5}$$

with joint coordinates, q, being:

$$q = \begin{bmatrix} \theta_1 \\ \theta_2 \\ . \\ . \\ . \\ \theta_n \end{bmatrix} \tag{1.6}$$

where one would need $n \geq 6$ (i.e., at least 6 DOF in joint space) to allow the end effector to have 6 DOF (in task space). For a planar robot, where the end effector is constrained to a 2D plane, there would be only 3 DOFs:

$$\xi = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{1.7}$$

A two-link planar arm (see Fig. 1.8), with only θ_1 and θ_2,

$$q = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \tag{1.8}$$

cannot control all 3DOF of *xi* is Eq. 1.7. At least three links (3 DOF) are needed for a planar arm to allow for 3 independent DOFs in the task space of the end effector. Another planar robot configuration that could allow for 3 DOF at the end effector is the cart-arm system shown in Figure 1.11. Here, the cart acts as a prismatic joint, and there are two rotational joints, creating 3 independent DOFs, in all.

If a we are given some desired coordinates for *xi* of the end effector, solving for a set of joint coordinate to achieve this geometry is called the **inverse kinematics** problem. There may be no possible solution (infeasible), one solution (feasible and unique) or more than one possible solution (feasible and redundant). Forward kinematics require careful, consistent geometric calculations, but are relatively straightforward to calculate. Inverse kinematics can be much more challenging to solve.

1.4 Singularities and Jacobians

1.5 Work, Force, and Displacement: A Review

1.6 Mechanical Impedance

Problems

Chapter 2
SISO Control: Single-Joint Elements

Abstract This chapter reviews so-called "classical" methods of control, most particularly proportional-derivative (PD) and proportional-integral-derivative (PID) control. Here, the goal is to control a single-input single-output (SISO) system such that its response can be characterized by (i.e., is linear and has the poles of) either a first- or second-order dynamic system.

2.1 First- and Second-Order Dynamics: A Review

2.2 Reference Tracking

2.2.1 PD and lead control

2.2.2 PID and lag control

2.3 Feedforward Control

2.4 Effects of Nonlinearities

2.4.1 Actuator saturation

2.4.2 Higher-order dynamic modes

Problems

Chapter 3
The Jacobian

Abstract This chapter introduces concepts and terminology in robotics and defines the scope of the text.

3.1 Matrix Velocity Kinematics

Use the template *chapter.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style the various elements of your chapter content in the Springer layout.

3.2 Static Force-vs-Torque Relationships

3.3 Inverse Jacobian

3.4 Singularities

3.5 Manipulability

Problems

Chapter 4
Wheeled Vehicle Dynamics

Abstract The most common form of mobility for a robot is a wheeled vehicle base. This chapter focuses on basic concepts for planning and controlling motion of a wheeled robot, with an emphasis on the connection between Jacobian for a manipulator, presented in Chapter!3, and the Jacobian for a wheeled vehicle. A three-wheel omnibot is used as a case example.

4.1 Terminology and Introduction

4.2 Kinematic Constraints

4.2.1 Holonomic Constraints

4.2.2 Non-holonomic Constraints

4.3 Jacobian Revisited: Relating wheel velocities to chassis velocity

4.4 Omnibot Example

Problems

Chapter 5
The Lagrangian

Abstract The Lagriangian method for deriving equations of motion for a dynamics system is described.

5.1 Generalized coordinates and forces

5.2 Holonomic constraints revisited

5.3 D'Alembert's Principle

5.4 Deriving equations of motion: kinetic and potential energy

5.5 Inertia tensor: Inertia of a 3D object

5.6 Relative vs absolute coordinates: Representing non-conservative forces

5.7 Examples: common robot linkage configurations

Problems

Chapter 6
State Space Methods

Abstract This chapter introduces so-called "modern" control methods, as opposed to the classical control presented in Chapter 2. The state space formulation is presented and full state feedback is introduced. State space techniques are particularly useful in controlling multiple-input multiple-output (MIMO) system, as described in more detail in Chapter!7.

6.1 Matrix equations of motion

6.2 Controllability and observability

6.3 Controller and observer design

6.4 LQR design

6.5 Lyapunov stability

Problems

Chapter 7
MIMO Control: multi-link systems

Abstract The state space formulation is presented and full state feedback is introduced.

7.1 PD revisited

7.2 Matrix inertia and stiffness

7.3 Inverse dynamics: computed torque

7.4 Inner/outer loop strategy

Problems

Chapter 8
Force and Impedance Control

Abstract This chapter focuses on methods for modifying the dynamics to match a desired force output or mechanical impedance relationship.

8.1 Artificial Constraints

8.2 Contact Dynamics

8.3 Thevinin and Norton Equivalents

8.4 Force Feedback

8.5 Reference Tracking with Impedance Control

Problems

Chapter 9
Uncertainty: Sensors, models, and disturbances

Abstract This chapter focuses on methods for modifying the dynamics to match a desired force output or mechanical impedance relationship.

9.1 Robust Control

9.2 Adaptive Control

9.3 Kalman Filtering

Problems

Chapter 10
Underactuated systems

Abstract This chapter focuses on methods for modifying the dynamics to match a desired force output or mechanical impedance relationship.

10.1 Definition of underactuation

10.2 Robot locomotion

10.3 Partial feedback linearization

Problems

Appendix A
MATLAB: Brief Tutorial

Abstract This chapter provides a brief tutorial on some commands and tools useful for modeling and simulation of dynamic systems, for design and analysis of controllers, for plotting data and for drawing animations.

A.1 Getting started

A.2 m-files: scripts and functions

A.2.1 Script example

A.2.2 Function example

A.2.3 MATLAB help command

A.3 Figures: plotting and drawing

A.3.1 Plotting example

A.3.2 Drawing example

A.4 Glossary of commands and operators

Problems

A.1. A given problem or Excercise is described here. The problem is described here. The problem is described here.

A.2. Problem Heading
(a) The first part of the problem is described here.
(b) The second part of the problem is described here.

Appendix B
Kinematics: A Brief Tutorial

Appendix C
Modeling

Abstract Basic elements of actuation and transmission of force and of sensing of state are covered in this chapter.

C.1 Linear vs nonlinear dynamics

C.2 Actuator dynamics: standard DC motor model

C.2.1 Motor electromechanical transformation constant

C.2.2 Electrical time constant

C.3 Transmission dynamics: gear boxes and linkages

C.3.1 Geometry: kinematics and inverse kinematics

C.3.2 Reflected inertia

C.3.3 Shaft flexibility

C.3.4 Nonlinear losses: efficiency and backlash

C.4 Sensor dynamics: State estimation

C.4.1 Quantization

C.4.2 Time Delay

C.4.3 Sensor dynamics

Problems

Appendix D
Linear Algebra: A Review

Problems

D.1. A given problem or Excercise is described here. The problem is described here. The problem is described here.

D.2. Problem Heading
(a) The first part of the problem is described here.
(b) The second part of the problem is described here.