Personal Lecture Notes
MITx6.832x: Underactuated Robotics
Algorithms for Walking, Running, Swimming, Flying, and Manipulation

Julian Eßer

February 12, 2020

Contents

1. Introduction

The purpose of this document is to provide an brief overview of the essential knowledge of the underactuated robotics class [1] from MIT. Special emphasis is on the following topics:

- Understanding Control as Optimization
- Dynamics of Biped Locomotion
- Optimization of Biped Locomotion

2. Lecture 1: Why Study Robot Dynamics?

2.1 Background / Motivation

The **motivation** for this course is to

- Build great robots that can do amazing things
- Exploit natural dynamics of robots, not just doing dump control
- Achieve extraordinary performance in terms of speed, efficiency, or robustness (Honda's ASIMO vs. passive dynamic walkers)
- Controlling nonlinear systems without complete control authority
- View computation of challenging tasks in robotics (manipulation, autonomous driving) through the lense of dynamics.

This course is all about nonlinear dynamics and control of underactuated mechanical systems, with an emphasis on computational methods. Especially it covers the **topics**

- Nonlinear dynamics
- Applied optimal and robust control
- Motion planning
- Examples from biology and applications to legged locomotion, compliant manipulation, underwater robots, and flying machines

2.2 Definitions

Nonlinear differential equations typically take the form

$$\dot{x} = f(x, u)$$

where f is a vector valued function, x is the state vector and u is the vector of control input and $\dot{x} = \frac{dx}{dt}$ is the time derivative. Mechanical Systems are described by second order differential equations. When the state vector is defined as

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix},$$

where the system dynamics can be described as

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

Since mechanical systems are *control affine*, this specializes to

$$\ddot{q} = f_1(q, \dot{q}) + f_2(q, \dot{q})u.$$

A system of this form is called *underactuated* if $rank[f_2] <= n$. Other causes of underactuated include

- Input saturation (e.g. torque limits)
- State constraints (e.g. joint limits)
- Model uncertainty / state estimation

2.3 Manipulator Equations

The equations of motion for simple systems, e.g. a double pendulum, are quite simple to derive. Results, e.g. obtained from an Lagrangian calculation approach, can be expressed in the form of the standard "manipulator equations":

$$M(q)\ddot{q} + C(q, \dot{q})\ddot{q} = \tau_g(q) + Bu$$

where M is the Inertia matrix, C is the matrix of Coriolis terms τ_g covers gravitational torques, B maps inputs to generalized force and u is the control input (either force or torque).
The acceleration then is expressed as

$$\ddot{q} = M^{-1}(q)[\tau_g(q) + Bu - C(q, \dot{q})\dot{q}]. \tag{2.1}$$

With equation 2.1, the dynamics of the systems and accordingly the functions f_1 and f_2 are fully defined.
For simulating the dynamics of a robot, it is sufficient to provide the kinematics in form of a *URDF file*, pass it to an forward Dynamics solver and you get the resulting acceleration and its integrations.

2.4 Plan for the Course

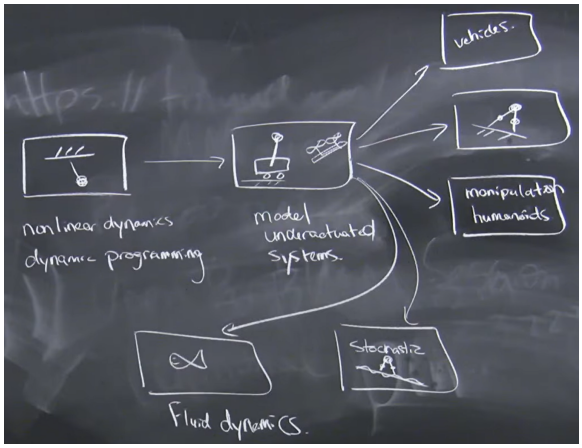

Figure 2.1: Course overview: Basics first, then simple systems and then various advanced systems.

Bibliography

[1] Russ Tedrake. Mitx6.832x: Official lecture website. http://underactuated.csail.mit.edu/Spring2019/index.htmltextbook/ass
2020.