Underactuated Robotics:
Learning, Planning, and Control for Efficient and Agile Machines

Course Notes for MIT 6.832

Russ Tedrake
Massachusetts Institute of Technology

Contents

Preface

This book is about building robots that move with speed, efficiency, and grace. The author believes that this can only be achieve through a tight coupling between mechanical design, passive dynamics, and nonlinear control synthesis. Therefore, these notes contain selected material from dynamical systems theory, as well as linear and nonlinear control.

These notes also reflect a deep belief in computational algorithms playing an essential role in finding and optimizing solutions to complex dynamics and control problems. Algorithms play an increasingly central role in modern control theory; nowadays even rigorous mathematicians use algorithms to develop mathematical proofs. Therefore, the notes also cover selected material from optimization theory, motion planning, and machine learning.

Although the material in the book comes from many sources, the presentation is targeted very specifically at a handful of robotics problems. Concepts are introduced only when and if they can help progress our capabilities in robotics. I hope that the result is a broad but reasonably self-contained and readable manuscript that will be of use to any robotics practicioner.

Organization

The material in these notes is organized into two main parts: "nonlinear dynamics and control", which introduces a series of increasingly complex dynamical systems and the associated control ideas, and "optimal control and motion planning", which introduces a series of general derivations and algorithms that can be applied to many, if not all of the problems introduced in the first part of the book. This second part of the book is organized by techniques; perhaps the most logical order when using the book as a reference. In teaching the course, however, I take a spiral trajectory through the material, introducing robot dynamics and control problems one at a time, and introducing only the techniques that are required to solve that particular problem. Finally, a third part of the book puts it all together through a few more complicated case studies and examples.

Exercises

The exercises in these notes come in a few varieties. The standard exercises are intended to be straight-forward extensions of the materials presented in the chapter. Some exercises are labeled as MATLAB exercises - these are computational investigations, which sometimes involve existing code that will help get you started. Finally, some exercises are labeled as CHALLENGE problems. These are problems that I have not yet seen or found the answers too, yet, but which I would very much like to solve. I cannot guarantee that they are unsolved in the literature, but the intention is to identify some problems which would advance the state-of-the-art.

Russ Tedrake, 2009

Notation

Dynamics and System Identification:

$\mathbf{q}$	Generalized coordinates (e.g., joint space)
$\mathbf{x}$	State space ($\mathbf{x} = [\mathbf{q}^T, \dot{\mathbf{q}}^T]^T$)
$\mathbf{u}$	Controllable inputs
$\bar{\alpha}$	Time trajectory of α
$s_i \in S$	s_i is a particular state in the set of all states S (for a discrete-state system)
$a_i \in A$	a_i is a particular action from the set of all actions A (for a discrete-action system)
$\mathbf{w}$	Uncontrollable inputs (disturbances)
$\mathbf{y}$	Outputs
$\mathbf{v}$	Measurement errors
$\mathbf{z}$	Observations
$\mathbf{H}$	Mass/Inertial Matrix
$\mathbf{C}$	Coriolis Matrix
$\mathbf{G}$	Gravity and potential terms
$\mathbf{f}$	First-order plant dynamics
T	Kinetic Energy
U	Potential Energy
L	Lagrangian ($L = T - U$)

Learning and Optimal Control:

π	Control policy
π^*	*Optimal* control policy
$\alpha, \beta, \gamma, ...$	Parameters
g	Instantaneous cost
h	Terminal cost
J	Long-term cost / cost-to-go function (value function)
J^*	*Optimal* cost-to-go function
$\mathbf{e}$	eligibility vector
η	learning rate

Basic Math:

$E[\mathbf{z}], \mu_z$	Expected value of $\mathbf{z}$
σ_z^2	Variance of z
$\sigma_{xy}, \mathbf{C}_{xy}$	Scalar covariance (and covariance matrix) between x and y
$\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$	Vector gradient ($= \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$)
$\delta(z)$	Continuous delta-function, defined via $\int_{-\infty}^{x} \delta(z) dz = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$
$\delta[z]$	Discrete delta-function, equals 1 when $z = 0$, zero otherwise

δ_{ij}	Shorthand for $\delta[i-j]$