

---

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

Course Notes for MIT 6.832

---

**Russ Tedrake**

*Massachusetts Institute of Technology*



# Contents

<b>Preface</b>	<b>vii</b>
<b>1 Fully Actuated vs. Underactuated Systems</b>	<b>3</b>
1.1 Motivation . . . . .	3
1.1.1 Honda’s ASIMO vs. Passive Dynamic Walkers . . . . .	3
1.1.2 Birds vs. modern aircraft . . . . .	4
1.1.3 The common theme . . . . .	5
1.2 Definitions . . . . .	5
1.3 Feedback Linearization . . . . .	7
1.4 Input and State Constraints . . . . .	8
1.5 Underactuated robotics . . . . .	8
1.6 Goals for the course . . . . .	9
<b>I Nonlinear Dynamics and Control</b>	<b>11</b>
<b>2 The Simple Pendulum</b>	<b>12</b>
2.1 Introduction . . . . .	12
2.2 Nonlinear Dynamics w/ a Constant Torque . . . . .	12
2.2.1 The Overdamped Pendulum . . . . .	13
2.2.2 The Undamped Pendulum w/ Zero Torque . . . . .	16
2.2.3 The Undamped Pendulum w/ a Constant Torque . . . . .	19
2.2.4 The Damped Pendulum . . . . .	19
2.3 The Torque-limited Simple Pendulum . . . . .	20
<b>3 The Acrobot and Cart-Pole</b>	<b>22</b>
3.1 Introduction . . . . .	22
3.2 The Acrobot . . . . .	22
3.2.1 Equations of Motion . . . . .	23
3.3 Cart-Pole . . . . .	23
3.3.1 Equations of Motion . . . . .	24
3.4 Balancing . . . . .	25
3.4.1 Linearizing the Manipulator Equations . . . . .	25
3.4.2 Controllability of Linear Systems . . . . .	26
3.4.3 LQR Feedback . . . . .	29
3.5 Partial Feedback Linearization . . . . .	29
3.5.1 PFL for the Cart-Pole System . . . . .	30
3.5.2 General Form . . . . .	31
3.6 Swing-Up Control . . . . .	33
3.6.1 Energy Shaping . . . . .	33
3.6.2 Simple Pendulum . . . . .	34
3.6.3 Cart-Pole . . . . .	35
3.6.4 Acrobot . . . . .	36

3.6.5	Discussion . . . . .	36
3.7	Other Model Systems . . . . .	37
<b>4</b>	<b>Manipulation</b>	<b>38</b>
4.1	Introduction . . . . .	38
4.2	Dynamics of Manipulation . . . . .	38
4.2.1	Form Closure . . . . .	38
4.2.2	Force Closure . . . . .	39
4.2.3	Active research topics . . . . .	40
4.3	Ground reaction forces in Walking . . . . .	40
4.3.1	ZMP . . . . .	41
4.3.2	Underactuation in Walking . . . . .	41
<b>5</b>	<b>Walking</b>	<b>42</b>
5.1	Limit Cycles . . . . .	42
5.2	Poincaré Maps . . . . .	43
5.3	The Ballistic Walker . . . . .	44
5.4	The Rimless Wheel . . . . .	44
5.4.1	Stance Dynamics . . . . .	45
5.4.2	Foot Collision . . . . .	45
5.4.3	Return Map . . . . .	46
5.4.4	Fixed Points and Stability . . . . .	47
5.5	The Compass Gait . . . . .	48
5.6	The Kneed Walker . . . . .	49
5.7	Numerical Analysis . . . . .	52
5.7.1	Finding Limit Cycles . . . . .	52
5.7.2	Local Stability of Limit Cycle . . . . .	53
<b>6</b>	<b>Running</b>	<b>55</b>
6.1	Introduction . . . . .	55
6.2	Comparative Biomechanics . . . . .	55
6.3	Raibert hoppers . . . . .	56
6.4	Spring-loaded inverted pendulum (SLIP) . . . . .	56
6.4.1	Flight phase . . . . .	56
6.4.2	Stance phase . . . . .	56
6.4.3	Transitions . . . . .	56
6.4.4	Approximate solution . . . . .	57
6.5	Koditschek's Simplified Hopper . . . . .	57
6.6	Lateral Leg Spring (LLS) . . . . .	57
<b>7</b>	<b>Flight</b>	<b>58</b>
7.1	Flate Plate Theory . . . . .	58
7.2	Simplest Glider Model . . . . .	58
7.3	Perching . . . . .	59
7.4	Swimming and Flapping Flight . . . . .	59
7.4.1	Swimming . . . . .	60
7.4.2	The Aerodynamics of Flapping Flight . . . . .	60

<b>8</b>	<b>Model Systems with Stochasticity</b>	<b>62</b>
8.1	Stochastic Dynamics . . . . .	62
8.1.1	The Master Equation . . . . .	62
8.1.2	Continuous Time, Continuous Space . . . . .	63
8.1.3	Discrete Time, Discrete Space . . . . .	63
8.1.4	Stochastic Stability . . . . .	63
8.1.5	Walking on Rough Terrain . . . . .	63
8.2	State Estimation . . . . .	63
8.3	System Identification . . . . .	63
<b>II</b>	<b>Optimal Control and Motion Planning</b>	<b>65</b>
<b>9</b>	<b>Dynamic Programming</b>	<b>66</b>
9.1	Introduction to Optimal Control . . . . .	66
9.2	Finite Horizon Problems . . . . .	67
9.2.1	Additive Cost . . . . .	67
9.3	Dynamic Programming in Discrete Time . . . . .	67
9.3.1	Discrete-State, Discrete-Action . . . . .	68
9.3.2	Continuous-State, Discrete-Action . . . . .	69
9.3.3	Continuous-State, Continuous-Actions . . . . .	69
9.4	Infinite Horizon Problems . . . . .	69
9.5	Value Iteration . . . . .	69
9.6	Value Iteration w/ Function Approximation . . . . .	69
9.6.1	Special case: Barycentric interpolation . . . . .	70
9.7	Detailed Example: the double integrator . . . . .	70
9.7.1	Pole placement . . . . .	70
9.7.2	The optimal control approach . . . . .	71
9.7.3	The minimum-time problem . . . . .	71
9.8	The quadratic regulator . . . . .	73
9.9	Detailed Example: The Simple Pendulum . . . . .	73
<b>10</b>	<b>Analytical Optimal Control with the Hamilton-Jacobi-Bellman Sufficiency Theorem</b>	<b>74</b>
10.1	Introduction . . . . .	74
10.1.1	Dynamic Programming in Continuous Time . . . . .	74
10.2	Infinite-Horizon Problems . . . . .	78
10.2.1	The Hamilton-Jacobi-Bellman . . . . .	79
10.2.2	Examples . . . . .	79
<b>11</b>	<b>Analytical Optimal Control with Pontryagin's Minimum Principle</b>	<b>81</b>
11.1	Introduction . . . . .	81
11.1.1	Necessary conditions for optimality . . . . .	81
11.2	Pontryagin's minimum principle . . . . .	82
11.2.1	Derivation sketch using calculus of variations . . . . .	82
11.3	Examples . . . . .	83

<b>12 Trajectory Optimization</b>	<b>85</b>
12.1 The Policy Space	85
12.2 Nonlinear optimization	85
12.2.1 Gradient Descent	86
12.2.2 Sequential Quadratic Programming	86
12.3 Shooting Methods	86
12.3.1 Computing the gradient with Backpropagation through time (BPTT)	86
12.3.2 Computing the gradient w/ Real-Time Recurrent Learning (RTRL)	88
12.3.3 BPTT vs. RTRL	89
12.4 Direct Collocation	89
12.5 LQR trajectory stabilization	90
12.5.1 Linearizing along trajectories	90
12.5.2 Linear Time-Varying (LTV) LQR	91
12.6 Iterative LQR	91
12.7 Real-time planning (aka receding horizon control)	92
<b>13 Feasible Motion Planning</b>	<b>93</b>
13.1 Artificial Intelligence via Search	93
13.1.1 Motion Planning as Search	93
13.1.2 Configuration Space	94
13.1.3 Sampling-based Planners	94
13.2 Rapidly-Exploring Randomized Trees (RRTs)	94
13.2.1 Proximity Metrics	94
13.2.2 Reachability-Guided RRTs	94
13.2.3 Performance	94
13.3 Probabilistic Roadmaps	94
13.3.1 Discrete Search Algorithms	94
<b>14 Global policies from local policies</b>	<b>95</b>
14.1 Real-time Planning	95
14.2 Multi-query Planning	95
14.2.1 Probabilistic Roadmaps	95
14.3 Feedback Motion Planning	95
<b>15 Stochastic Optimal Control</b>	<b>96</b>
15.1 Essentials	96
15.2 Implications of Stochasticity	96
15.3 Markov Decision Processes	96
15.4 Dynamic Programming Methods	96
15.5 Policy Gradient Methods	96
<b>16 Model-free Value Methods</b>	<b>97</b>
16.1 Introduction	97
16.2 Policy Evaluation	97
16.2.1 for known Markov Chains	97
16.2.2 Monte Carlo Evaluation	98
16.2.3 Bootstrapping	98

16.2.4	A Continuum of Updates . . . . .	99
16.2.5	The TD( $\lambda$ ) Algorithm . . . . .	99
16.2.6	TD( $\lambda$ ) with function approximators . . . . .	99
16.2.7	LSTD . . . . .	101
16.3	Off-policy evaluation . . . . .	101
16.3.1	Q functions . . . . .	101
16.3.2	TD for Q with function approximation . . . . .	102
16.3.3	Importance Sampling . . . . .	102
16.3.4	LSTDQ . . . . .	102
16.4	Policy Improvement . . . . .	102
16.4.1	Sarsa( $\lambda$ ) . . . . .	102
16.4.2	Q( $\lambda$ ) . . . . .	102
16.4.3	LSPI . . . . .	102
16.5	Case Studies: Checkers and Backgammon . . . . .	102
<b>17</b>	<b>Model-free Policy Search</b>	<b>103</b>
17.1	Introduction . . . . .	103
17.2	Stochastic Gradient Descent . . . . .	103
17.3	The Weight Perturbation Algorithm . . . . .	103
17.3.1	Performance of Weight Perturbation . . . . .	105
17.3.2	Weight Perturbation with an Estimated Baseline . . . . .	107
17.4	The REINFORCE Algorithm . . . . .	108
17.4.1	Optimizing a stochastic function . . . . .	109
17.4.2	Adding noise to the outputs . . . . .	109
17.5	Episodic REINFORCE . . . . .	109
17.6	Infinite-horizon REINFORCE . . . . .	110
17.7	LTI REINFORCE . . . . .	110
17.8	Better baselines with Importance sampling . . . . .	110
<b>18</b>	<b>Actor-Critic Methods</b>	<b>111</b>
18.1	Introduction . . . . .	111
18.2	Pitfalls of RL . . . . .	111
18.2.1	Value methods . . . . .	111
18.2.2	Policy Gradient methods . . . . .	111
18.3	Actor-Critic Methods . . . . .	111
18.4	Case Study: Toddler . . . . .	112
<b>III</b>	<b>Applications and Extensions</b>	<b>113</b>
<b>19</b>	<b>Learning Case Studies and Course Wrap-up</b>	<b>114</b>
19.1	Learning Robots . . . . .	114
19.1.1	Ng's Helicopters . . . . .	114
19.1.2	Schaal and Atkeson . . . . .	114
19.1.3	AIBO . . . . .	114
19.1.4	UNH Biped . . . . .	114
19.1.5	Morimoto . . . . .	114

19.1.6	Heaving Foil . . . . .	114
19.2	Optima for Animals . . . . .	114
19.2.1	Bone geometry . . . . .	115
19.2.2	Bounding flight . . . . .	115
19.2.3	Preferred walking and running speeds/transitions . . . . .	115
19.3	RL in the Brain . . . . .	115
19.3.1	Bird-song . . . . .	115
19.3.2	Dopamine TD . . . . .	115
19.4	Course Wrap-up . . . . .	115
<b>IV</b>	<b>Appendix</b>	<b>117</b>
<b>A</b>	<b>Robotics Preliminaries</b>	<b>118</b>
A.1	Deriving the equations of motion (an example) . . . . .	118
A.2	The Manipulator Equations . . . . .	119
<b>B</b>	<b>Machine Learning Preliminaries</b>	<b>121</b>
B.1	Function Approximation . . . . .	121



# Preface

This book is about building robots that move with speed, efficiency, and grace. The author believes that this can only be achieved 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 practitioner.

## 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 to, 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 $\alpha$
$s_i \in S$	$s_i$ 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:

---

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

## Basic Math:

---

$E[\mathbf{z}], \mu_z$	Expected value of $\mathbf{z}$
$\sigma_z^2$	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 ( = $\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$ )
$\delta(z)$	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

2

$\delta_{ij}$       Shorthand for  $\delta[i - j]$

MIT OpenCourseWare  
<http://ocw.mit.edu>

6.832 Underactuated Robotics  
Spring 2009

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.