# Controlling Self-Landing Rockets Using CVXPY

Philipp Schiele    Steven Diamond

Eric Luxenberg    Stephen Boyd

Austin, TX
July 10, 2023

# Outline

# Landing a rocket is hard

# Landing a rocket is hard

# Landing a rocket is hard

# Landing a rocket is hard

# Background on this tutorial

- SpaceX has used CVXGEN, a code generator for convex problems, as part of their control system for landing Falcon 9 rockets [Boy21]
- this tutorial is based on work by Thomas Lipp, Lars Blackmore, and Yoshi Kuwata (see, *e.g.*, [LB16, ACB13, OPKB15])
- a simplified version of the problem was added as an exercise to Convex Optimization [BV04]
- we are not involved in landing actual rockets
- we are not affiliated with SpaceX

## Landing a rocket

- we are looking actions $x$ that will land our rocket, *e.g.*,
  - thrust of the rocket engines at each time step
  - changing pitch and heading of the rocket via fins, thrusters, or engine gimballing
- these actions should be good, or even optimal, with respect to some criterion
- constraints limit our actions or impose conditions on the outcome, such as,
  - physics constraints
  - maximum thrust level
  - limited fuel
  - some engines can only be ignited once

## Optimization problem

$$
\begin{array}{ll}
\text{minimize} & f_0(x) \\
\text{subject to} & f_i(x) \leq 0, \quad i = 1, \dots, m \\
& g_i(x) = 0, \quad i = 1, \dots, p
\end{array}
$$

- $x \in \mathbf{R}^n$ is (vector) variable to be chosen
- $f_0$ is the *objective function*, to be minimized
- $f_1, \dots, f_m$ are the *inequality constraint functions*
- $g_1, \dots, g_p$ are the *equality constraint functions*

- variations: maximize objective, multiple objectives, ...

## Application areas

Optimization problems arise in many other areas (more than previously thought), including

- **control**
- machine learning, statistics
- finance
- supply chain, revenue management, advertising
- signal and image processing, vision
- networking
- circuit design
- combinatorial optimization
- quantum mechanics
- flux-based analysis

## Summary

- **summary**: optimization arises *everywhere*
- **the bad news**: most optimization problems are *intractable*
  *i.e.*, *we cannot solve them*
- **an exception**: *convex optimization problems are tractable*
  *i.e.*, we (generally) *can* solve them

## Convex optimization problem

convex optimization problem:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax = b \end{array}$$

- variable $x \in \mathbf{R}^n$
- equality constraints are linear
- $f_0, \ldots, f_m$ are **convex**: for $\theta \in [0, 1]$,

$$f_i(\theta x + (1 - \theta)y) \leq \theta f_i(x) + (1 - \theta) f_i(y)$$

*i.e.*, $f_i$ have nonnegative (upward) curvature

## Why

- beautiful, nearly complete theory
  - duality, optimality conditions, ...
- effective algorithms, methods (in theory and practice)
  - get **global solution** (and optimality certificate)
  - polynomial complexity
  - extremely robust (no need to tune parameters)
- conceptual unification of many methods

## The approach

- try to formulate your optimization problem as convex
- if you succeed, you can (usually) solve it (numerically)
  - using generic software if your problem is not really big
  - by developing your own software otherwise

- some tricks:
  - change of variables
  - approximation of true objective, constraints
  - *relaxation*: ignore terms or constraints you can't handle

## Medium-scale solvers

- 1k – 100k variables, constraints
- reliably solved by interior-point methods on single machine (especially for problems in standard cone form)
- exploit problem sparsity
- very solid technology
- used in control, finance, engineering design, ...

## Large-scale solvers

- 1M – 1B variables, constraints
- solved using custom (often problem specific) methods
  - limited memory BFGS
  - stochastic subgradient
  - block coordinate descent
  - operator splitting methods
- require custom implementation, tuning for each problem
- used in machine learning, image processing, ...

## Modeling languages

- high level language support for convex optimization
  - describe problem in high level language
  - description automatically transformed to a standard form
  - solved by standard solver, transformed back to original form
- implementations:
  - CVXPY (Python)
  - YALMIP, CVX (Matlab)
  - Convex.jl (Julia)
  - CVXR (R)

# CVXPY

a modeling language in Python for convex optimization

- developed since 2014
- verifies convexity via signed *disciplined* convex programming
- open source all the way to the solvers
- supports parameters
- mixes easily with general Python code, other libraries
- used in many research projects, classes, companies
- many extensions available
- tens of thousands of users

## CVXPY

- example: the regularized least squares problem

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 + \gamma\|x\|_1 \\ \text{subject to} & \|x\|_\infty \leq 1 \end{array}$$

translates to the following CVXPY problem, where `A`, `b`, `gamma` are constants (`gamma` nonnegative)

```python
import cvxpy as cp
x = cp.Variable(n)
cost = cp.sum_squares(A@x-b) + gamma*cp.norm(x,1)
prob = cp.Problem(cp.Minimize(cost),
                  [cp.norm(x,"inf") <= 1])
opt_val = prob.solve()
solution = x.value
```

- `solve` method converts problem to standard form, solves, assigns `value` attributes

# Modeling languages

- make convex optimization accessible to non-experts
- easy to experiment with different formulations
- enable more complex models

- slower than custom methods, but often not much
- ongoing work to extend to very large problems

## Summary

- convex optimization problems **arise in many applications**
- convex optimization problems **can be solved effectively**
  - using generic methods for not huge problems
  - by developing custom methods for huge problems
- high level language support (CVXPY/CVX/Convex.jl/CVXR) makes prototyping easy

## Resources

*many* researchers have worked on the topics covered

- *Convex Optimization* (book)
- *EE364a* (course slides, videos, code, homework, . . . )
- software CVXPY, CVX, Convex.jl, CVXR
- *convex optimization short course*

all available online

## References I

B. Açıkmeşe, J. Carson, and L. Blackmore.
Lossless Convexification of Nonconvex Control Bound and Pointing Constraints in the Soft Landing Optimal Control Problem.
*Control Systems Technology, IEEE Transactions on*, 2013.

S. Boyd.
Stephen Boyd's CVXGEN software helps guide SpaceX Falcon.
*Stanford SystemX Alliance*, January 5 2021.

S. Boyd and L. Vandenberghe.
*Convex Optimization*.
Cambridge University Press, 2004.

# References II


T. Lipp and S. Boyd.
Variations and extension of the convex–concave procedure.
*Optimization and Engineering*, 2016.

M. Ono, M. Pavone, Y. Kuwata, and J. Balaram.
Chance-constrained dynamic programming with application to risk-aware robotic space exploration.
*Autonomous Robots*, 39:555–571, 2015.