Lecture 4
ROB-GY 7863 / CSCI-GA 3033 7863: Planning, Learning, and Control for Space Robotics

Benjamin Riviere

October 5, 2025

Logistics

- Proposals are due tomorrow (postponed in case you want to do something rockets related)
- Project 1 Report/Presentation due in two weeks

Recap Last Week

- Contraction Theory:

$$\frac{\partial f}{\partial x}^T M + M\frac{\partial f}{\partial x} + \dot{M} + 2\alpha M \preceq 0 \implies \delta x(t) \leq e^{-\alpha t}\delta x(0)$$
$$\implies \|x_2(t) - x_1(t)\| \leq e^{-\alpha t}\|x_2(0) - x_1(0)\|$$

- Sliding Mode Control:

$$S = \{(q, \dot{q}) \mid \dot{q} = \dot{q}_r\}$$
$$q \to S \text{ at rate } K$$
$$e \to 0 \text{ at rate } \Lambda$$

- Consensus and Formation Flying:

$$\dot{x} = -Lx$$

Space Culture I

- falcon 9
- starship

Space Culture II

Autonomous Precision Landing of Space Rockets

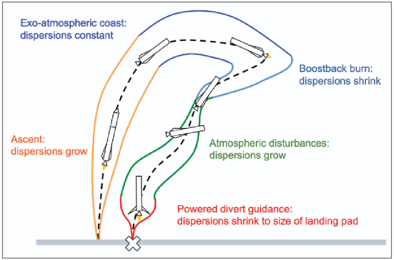
Lars Blackmore

Lars Blackmore is principal rocket landing engineer at SpaceX.

Landing an autonomous spacecraft or rocket is very challenging, and landing one with precision close to a prescribed target even more so. Precision landing has the potential to improve exploration of the solar system and to enable rockets that can be refueled and reused like an airplane.

This paper reviews the challenges of precision landing, recent advances that have enabled precision landing on Earth for commercial reusable rockets, and what is required to extend this to landing on planets such as Mars.

Brief History of Autonomous Space Landings



Space Culture III

ing without exceeding the capabilities of the hardware.

The computation must be done autonomously, in a fraction of a second. Failure to find a feasible solution in time will crash the spacecraft into the ground. Failure to find the optimal solution may use up the available propellant, with the same result. Finally, a hardware failure may require replanning the trajectory multiple times.

A general solution to such problems has existed in one dimension since the 1960s (Meditch 1964), but not in three dimensions. Over the past decade, research has shown how to use modern mathematical optimization techniques to solve this problem for a Mars landing, with guarantees that the best solution can be found in time

too can be solved. SpaceX uses CVXGEN (Mattingley and Boyd 2012) to generate customized flight code, which enables very high-speed onboard convex optimization.

Space Culture IV

Journal of Guidance, Control, and Dynamics
Vol. 33, No. 4, July–August 2010

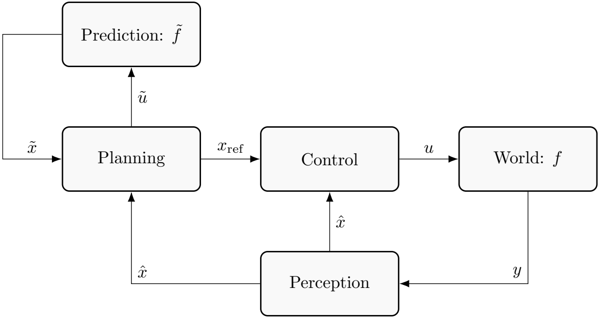
Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization

Lars Blackmore,* Behçet Açikmeşe,† and Daniel P. Scharf‡
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California 91109

DOI: 10.2514/1.47202

To increase the science return of future missions to Mars and to enable sample return missions, the accuracy with which a lander can be delivered to the Martian surface must be improved by orders of magnitude. The prior work developed a convex-optimization-based minimum-fuel powered-descent guidance algorithm. In this paper, this convex-optimization-based approach is extended to handle the case when no feasible trajectory to the target exists. In this case, the objective is to generate the minimum-landing-error trajectory, which is the trajectory that minimizes the distance to the prescribed target while using the available fuel optimally. This problem is inherently a nonconvex optimal control problem due to a nonzero lower bound on the magnitude of the feasible thrust vector. It is first proven that an optimal solution of a convex relaxation of the problem is also optimal for the original nonconvex problem, which is referred to as a lossless convexification of the original nonconvex problem. Then it is shown that the minimum-landing-error trajectory generation problem can be posed as a convex optimization problem and solved to global optimality with known bounds on convergence time. This makes the approach amenable to onboard implementation for real-time applications.

Agenda for Next Few Lectures I



Agenda for Next Few Lectures II

- Convex Optimization:
  - Introduction: Convex Functions and Sets
  - Algorithms and Convergence:
    - Unconstrained Optimization
    - Equality Constraints
    - Inequality Constraints
  - Examples
  - Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization Blackmore et al., 2010.
- Sampling-Based Planning
- Search-Based Planning

High Level Intuition Going into This Chapter

- Convex optimization problems are those that are easily solved with descent methods. In this lecture, we show why that is the case.
- Often, non-convex optimization problems can be relaxed or reformulated as convex optimization problems. In next lecture, we show examples of convex programming, common patterns of reformulation,
- Both lectures build towards understanding the minimum landing error powered descent problem.

Mathematical Optimization

- Mathematical Optimization Problem:

$$\begin{aligned} \min\ & f(x) \\ \text{s.t.}\ & g_i(x) \leq 0,\ i \in \{1, ..., m\} \\ & h_i(x) = 0,\ i \in \{1, ..., p\} \end{aligned}$$

  where $x \in \mathbb{R}^n$ is the optimization variable, $f : \mathbb{R}^n \to \mathbb{R}$ is the objective and $g_i : \mathbb{R}^n \to \mathbb{R}$ for $i \in \{1, ..., m\}$ are the constraint functions. A vector x^* is called optimal if it solves this optimization problem.
- A convex optimization problem is a mathematical optimization problem with the following requirements:
  - f must be convex
  - g_i must be convex
  - h_i must be affine

Optimal Control Problem

- Optimal Control Problem = Trajectory Optimization $\subseteq$ Planning

$$
\begin{aligned}
\min_{x_{1:K}, u_{1:K}} \quad & \sum_{k=1}^{K} c(x_k, u_k) + d(x_K) \\
\text{s.t.} \quad & x_{k+1} = f(x_k, u_{k+1}) \\
& x_k \in X_{\text{safe}} \\
& u_k \in U, \ \forall k \in \{1, \ldots, K\}
\end{aligned}
$$

Convex Functions I

- A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex iff $\forall x, y \in \mathrm{dom}(f)$ and $\theta \in [0, 1]$:

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

- First order condition: If f is differentiable, f is convex iff $\forall x, y \in \mathrm{dom}(f)$:

$$f(y) \geq f(x) + \nabla f(x)^T(y - x)$$

- Second order condition: If f is twice differentiable, f is convex iff $\forall x \in \mathrm{dom}(f)$:

$$\nabla^2 f(x) \succeq 0$$

- A function is **strongly convex** if $\exists m, M > 0$ such that: $mI \preceq \nabla^2 f \preceq MI$.

Convex Functions II

- A function is **concave** if these inequalities are flipped. If f is convex, $-f$ is concave.
- A function f is **affine** if it can be represented as $f(x) = Ax + b$.
- Examples of Convex functions: exponential e^{ax}, powers: x^a for $a \in (-\infty, 0) \cup (1, \infty)$, negative entropy $x\log(x)$, norms: $\|x\|$, max function: $\max\{x_i\}$, log-sum-exp: $f(x) = \log(\sum_i e^{x_i})$)
- Examples of concave functions: logarithm, geometric mean, log-determinant
- Operations that preserve convexity: nonnegative weighted sums, composition with an affine mapping, pointwise maximum and supremum,

Convex Functions III

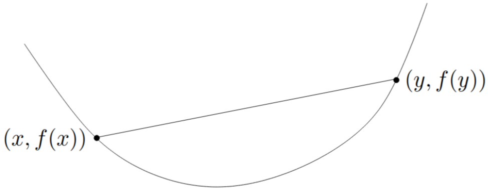


Figure 3.1 Graph of a convex function. The chord (*i.e.*, line segment) between any two points on the graph lies above the graph.

Convex Functions IV

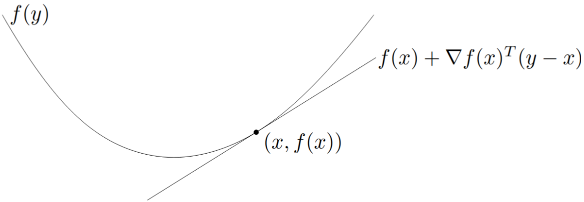


Figure 3.2 If f is convex and differentiable, then $f(x)+\nabla f(x)^T(y-x) \leq f(y)$ for all $x,\ y \in \mathbf{dom}\, f$.

Convex Sets I

- A set is convex if the line segment between any two points in C lies in C: if $\forall x_1, x_2 \in C$, $\theta \in [0,1]$: $\theta x_1 + (1-\theta)x_2 \in C$.
- Norm Balls: $C = \{x | \|x\|_2 \leq 1\}$, $\|x\|_1 \leq 1$, $\|x\|_M \leq 1$
- Polyhedra: $C = \{a_i^T x \leq b_i\}$
- Cones: $\forall x_1, x_2 \in C$, $\forall \theta \in [0,1]$, $\theta_1 x_1 + \theta_2 x_2 \in C$. Important example, cone of positive semi-definite matrices.
- Second order cones: $C = \{(x,t) \in \mathbb{R}^{n+1} \mid \|x_2\| \leq t\}$
- Relation between convex function and convex set:

$$\text{epigraph}(f) = \{(x,t) \in \mathbb{R}^{n+1} \| \ f(x) \leq t\}$$

  A function f is convex iff its epigraph is a convex set.
- **TODO:** pictures

Unconstrained Optimization

- Let $f : X \to \mathbb{R}$ be strongly convex and smooth. We seek to solve:

$$x^* = \min_x f(x)$$

- Basic algorithm is gradient descent (GD):

$$x_{k+1} = x_k - \alpha \nabla f(x)$$

- Basic analysis assumes m-strongly convex and M-smooth:

$$mI \preceq \nabla^2 f(x) \preceq MI$$

Continuous-time GD

- Continuous-time dynamics are gradient flow:

$$\dot{x} = F(x) = -\nabla f(x)$$

- Strong convexity and smoothness assumptions on f means the dynamics are contracting in identity metric with rate $2m$:

$$\frac{\partial F}{\partial x}^T + \frac{\partial F}{\partial x} = -2\nabla^2 f(x) \preceq 2mI$$

- Note that x^* is a solution of the flow, contraction implies all solutions globally contract to it:

$$\|x(t) - x^*\|_2 \leq e^{-mt}\|x(0) - x^*\|$$

- For convex functions, continuous time GD is a contracting flow with a unique minimum

Discrete-time Contraction I

- Before we analyze gradient descent algorithm, we need discrete time contraction results.
- Define familiar objects:

$$
\begin{aligned}
x_{k+1} &= F(x_k) && \text{dynamics map} \\
x_k(x_0) &= \underbrace{F(\dots F}_{k\text{times}}(x_0)) && \text{trajectory map} \\
x_0^\epsilon &= x_0 + \epsilon h && \text{finite perturbation} \\
\delta x_k &= \frac{\partial}{\partial \epsilon}\Big|_{\epsilon=0} \; x_k(x_0^\epsilon) && \text{variation}
\end{aligned}
$$

Discrete-time Contraction II

- Derive variation dynamics:

$$x_{k+1}(x_0^\epsilon) = F(x_k(x_0^\epsilon))$$
$$\frac{\partial}{\partial \epsilon}|_{\epsilon=0} x_{k+1}(x_0^\epsilon) = \frac{\partial}{\partial \epsilon}|_{\epsilon=0} F(x_k(x_0^\epsilon))$$
$$\delta x_{k+1} = \frac{\partial F}{\partial x} \delta x_k$$

- Contraction condition. Let $V_k = \delta x_k^T M_k \delta x_k$ and $V_{k+1} = \delta x_{k+1}^T M_{k+1} \delta x_{k+1}$, and compare V_{k+1} and V_k:

$$V_{k+1} \leq \rho^2 V_k \iff \frac{\partial F}{\partial x}^T M_{k+1} \frac{\partial F}{\partial x} - \rho^2 M_k \preceq 0$$

where $\rho \in [0, 1]$. Then, for any two solutions of the flow $x^{(1)}$, $x^{(2)}$:

$$\|x_k^{(2)} - x_k^{(1)}\|_M \leq \rho^k \|x_0^{(2)} - x_0^{(1)}\|_M$$

Contraction Analysis of Gradient Descent I

- Start from discrete time dynamics and variation dynamics:

$$x_{k+1} = x_k - \alpha \nabla f(x)$$
$$\delta x_{k+1} = (I - \alpha \nabla^2 f(x))\delta_k = \frac{\partial F}{\partial x}\delta x_k$$

- Let λ_i, v_i be eigenvalues/vectors of $\nabla^2 f$. The eigenvalues of $\frac{\partial F}{\partial x}$ are $1 - \alpha\lambda_i$:

$$\frac{\partial F}{\partial x} v_i = (I - \alpha \nabla^2 f(x)) v_i = (1 - \alpha\lambda_i) v_i$$

- Pick identity metric $M_k = I$, and compute LHS of contraction condition:

$$\frac{\partial F}{\partial x}^T \frac{\partial F}{\partial x} \preceq \max_{i,j} (1 - \alpha\lambda_i)(1 - \alpha\lambda_j) I = (\max_i |1 - \alpha\lambda_i|)^2 I = \rho^2 I$$

Contraction Analysis of Gradient Descent II

- We can pick α that minimizes ρ to give us the fastest contraction possible, i.e.:

$$\alpha^* = \min \rho = \min\{|1 - \alpha m|, |1 - \alpha M|\} = \frac{2}{m + M}$$

Then, we can compute the contraction rate:

$$\rho^* = 1 - \frac{2m}{m + M} = \frac{M - m}{M + m} = \frac{\kappa - 1}{\kappa + 1}$$

where $\kappa = M/m$ is the condition number. Then we can apply the standard contraction result:

$$\|x_k - x^*\|_2 \leq \rho^k \|x_0 - x^*\|_2$$

Contraction Analysis of Gradient Descent III

- Now we can compute **sample complexity** for ϵ-error:

$$k \geq \frac{\log(\epsilon/\epsilon_0)}{\log(\rho)} = \frac{\log(\epsilon_0/\epsilon)}{\log(1/\rho)}$$

  where $\epsilon_0 = \|x_0 - x^*\|_2$.
- Note: the convergence rate of the algorithm is dependent on the inverse of the condition number κ: if $\kappa = 1$, the solution is reached with one step, but usually $\kappa \gg 1$, in which case convergence slows down. This motivates the need for Newton Method.
- Note: In algorithms, we usually do not assume knowledge of m and M, so this variant of selecting the best timestep is not always available. This analysis gives an idea for the limits in ideal performance of gradient descent. For real algorithms, see Boyd Boyd et al., 2004.

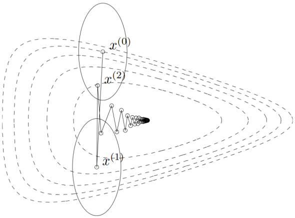
Examples

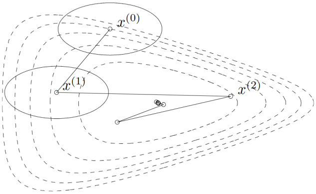
- Objective function:

$$f(x,y) = e^{x+3y-0.1} + e^{x-3y-0.1} + e^{-x-0.1}$$

- Steepest Descent:

$$x_{k+1} = x_k - \alpha P^{-1}\nabla f(x), \quad P_1 = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix} \quad P_2 = \begin{bmatrix} 8 & 0 \\ 0 & 2 \end{bmatrix}$$





Newton Method

- Let $f : X \to \mathbb{R}$ be strongly convex and smooth. We seek to solve:

$$x^* = \min_x f(x)$$

- Newton method:

$$x_{k+1} = x_k - \alpha(\nabla^2 f)^{-1} \nabla f$$

- Analysis: e.g. see Desoer et al., 1972
  - Quadratic convergence: for some region R, $x_k \in R$ implies:

$$\|x_{k+1} - x^*\| \leq C \|x_k - x^*\|^2$$

Equality Constrained Optimization

- Let $f : X \to \mathbb{R}$ be strongly convex and smooth. We seek to solve:

$$\begin{aligned} x^* = &\min_x f(x) \\ \text{s.t. } &Ax = b \end{aligned}$$

- The Lagrangian:

$$L(x, \lambda) = f(x) + \lambda^T (Ax - b)$$

- Primal-dual step:

$$\begin{aligned} x_{k+1} &= x_k - \alpha \nabla_x L = x_k - \alpha(\nabla f + A^T \lambda_k) \\ \lambda_{k+1} &= \lambda_k + \beta \nabla_\lambda L = \lambda_k + \beta(Ax_{k+1} - b) \end{aligned}$$

Analysis of Primal Dual I

- Augmented state $z_k = [x_k, \lambda_k]$:

$$z_{k+1} = F(z_k) = z_k - \begin{bmatrix} \alpha \nabla_x L \\ -\beta \nabla_\lambda L \end{bmatrix} \tag{1}$$

- For a fixed timestep $\alpha = \beta$, you can use same analysis as unconstrained gradient descent in higher dimensions. However, you need to prove that L (not f) is strongly convex and smooth, which typically requires dual regularization or augmentation.
- For mixed timesteps $\alpha \neq \beta$, you need a change of variables, something like $\lambda' = \frac{\alpha}{\beta}\lambda$

Analysis of Primal Dual II

- Conclusion:

$$\alpha = \frac{2}{M_\gamma + m_\gamma}, \quad \rho_x = \frac{\kappa_x - 1}{\kappa_x + 1} \tag{2}$$

$$\beta = \frac{2}{\sigma_{\max}^2(A) + \sigma_{\min}^2(A)}, \quad \rho_y = \frac{\kappa_y - 1}{\kappa_y + 1} \tag{3}$$

Inequality Constraints

- Constrained problem with inequality constraints:

$$\min_x f(x) \quad \text{s.t. } g_i(x) \leq 0, \; i = 1, \dots, m$$

- Introduce barrier function with parameter $\mu > 0$:

$$\phi_\mu(x) = -\mu \sum_{i=1}^{m} \log(-g_i(x))$$

- **TODO:** picture: log(x) vs $-g_i(x)$ with $i \in \{1\}$
- Inner loop: optimize $f_\mu = f + \phi_\mu$
- Outer loop: bring μ to zero.

Inner Loop Dynamics

- Discrete-time dynamics for fixed μ:

$$x_{k+1} = x_k - \alpha \nabla f_\mu(x_k)$$

- Variation dynamics:

$$\delta x_{k+1} = \left(I - \alpha \nabla^2 \phi_\mu(x_k)\right) \delta x_k$$

- Derivatives of ϕ_μ:

$$\nabla \phi_\mu = \sum_i \frac{1}{-g_i(x)} \nabla g_i(x)$$

$$\nabla^2 \phi_\mu = \sum_i \frac{1}{-g_i(x)} \nabla^2 g_i(x) + \frac{1}{g_i(x)^2} \nabla g_i(x) \nabla g_i(x)^T$$

- Inside feasible region $g_i(x) < 0$, so $\nabla^2 \phi_\mu(x) \succ 0$ bc g_i are convex and outer products are always positive semi-definite.

Discrete Contraction for IPM

- In identity metric:

$$\|\delta x_{k+1}\| \leq \rho_\mu \|\delta x_k\|$$

- Contraction factor:

$$\rho_\mu = \max_{\lambda \in [\mu_\phi, L_\phi]} |1 - \alpha\lambda|$$

where $[\mu_\phi, L_\phi]$ are eigenvalue bounds of $\nabla^2 f_\mu(x)$.

- Optimal constant step size:

$$\alpha^* = \frac{2}{L_\phi + \mu_\phi}, \qquad \rho_\mu^* = \frac{L_\phi - \mu_\phi}{L_\phi + \mu_\phi}$$

- The same form as previous results, but on the barrier function.

Example of Contraction for Time-Varying Optimization

Parameter-varying case: Consider the *parameter-dependent equality-constrained minimization problem*:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f_\theta(x), \\ \text{s.t.} \quad & Ax = b_\theta, \end{aligned} \tag{19}$$

where for each $\theta \in \Theta$, $b_\theta \in \mathbb{R}^m$ and the map f_θ is continuously differentiable, strongly convex, and strongly smooth with parameters ρ and ℓ, respectively. We also assume that A is full row rank.

The *parameter-varying primal-dual dynamics* are

$$\begin{aligned} \dot{x} &= -\nabla f_\theta(x) - A^\top \lambda, \\ \dot{\lambda} &= Ax - b_\theta, \end{aligned} \tag{20}$$

- "Time-Varying Convex Optimization: A Contraction and Equilibrium Tracking Approach" Davydov et al., 2025

Standard Forms of Optimization Problems I

- Linear Program:

$$\begin{aligned} \min_x \ & c^T x \\ \text{s.t.} \ & Ax = b \\ & x_i \geq 0, \ \forall i \in \{1, ..., n\} \end{aligned}$$

  Examples: Diet problem, Chebyshev center of polyhedron, Dynamic activity planning, Chevyshev inequalities, Piecewise-linear minimization, linear assignment problem.

- Quadratic Program:

$$\begin{aligned} \min_x \ & x^T P x + q^T x + r \\ \text{s.t.} \ & (Ax)_i \leq b_i \end{aligned}$$

  Examples:

Standard Forms of Optimization Problems II

- Second-Order Cone Program:

$$
\begin{aligned}
\min_{x} \ & h^T x \\
\text{s.t.} \ & \|A_i x + b_i\|_2 \leq c_i^T x + d_i \\
& Fx = g
\end{aligned}
$$

Examples:

- Semidefinite Program:

$$
\begin{aligned}
\min_{X} \ & \operatorname{trace}(C^T X) \\
\text{s.t.} \ & \operatorname{trace}(A_i^T X) \leq b_i \\
& X = X^T \succeq 0
\end{aligned}
$$

Examples:

Complexity Across Problem Classes I

- We presented sample-complexity results for first-order methods (i.e. not Newton's method, although you can imagine how to do that now).
- All share the same iteration bound form:

$$k = O(\log(1/\epsilon)/\log(1/\rho))$$

- What differs is the cost of each step:
- Practical implication:
  - LP solvers scale to millions of variables.
  - SDP solvers handle only a few thousand variables.

References I

- Convex Optimization Boyd et al., 2004
- Lectures on Convex Optimization Nesterov et al., 2018
- Spacecraft Trajectory Optimization *Spacecraft Trajectory Optimization* 2010

Blackmore, Lars, Behçet Açikmeşe, and Daniel P Scharf (2010). "Minimum-landing-error powered-descent guidance for Mars landing using convex optimization". In: *Journal of guidance, control, and dynamics* 33.4, pp. 1161–1171 (cit. on p. 9).

Boyd, Stephen P and Lieven Vandenberghe (2004). *Convex optimization*. Cambridge university press (cit. on pp. 24, 37).

References II

- Davydov, Alexander, Veronica Centorrino, Anand Gokhale, Giovanni Russo, and Francesco Bullo (2025). "Time-varying convex optimization: A contraction and equilibrium tracking approach". In: *IEEE Transactions on Automatic Control* (cit. on p. 33).
- Desoer, C. and H. Haneda (1972). "The measure of a matrix as a tool to analyze computer algorithms for circuit analysis". In: *IEEE Transactions on Circuit Theory* 19.5, pp. 480–486. DOI: 10.1109/TCT.1972.1083507 (cit. on p. 26).
- Nesterov, Yurii et al. (2018). *Lectures on convex optimization*. Vol. 137. Springer (cit. on p. 37).
- *Spacecraft Trajectory Optimization* (2010). Cambridge Aerospace Series. Cambridge University Press (cit. on p. 37).