Lecture # 8

Goals
1) Controls Project
2) Motivating Optimization for Control
3) Convexity
4) Convex Optimization
5) QP Based MPC

I Controls Project

1. nonlinear dynamics
2. complex track
3. two tasks
   a. get around track
   b. get around track w/ obstacles
4. how do we test
   a. integrate forward and verify using ode45.
   b. several test days
5. hints
   a. feedback
   b. use linearizations cleverly

II. Motivation for Optimization Based Control

Need algorithmic intelligence to generate controllers in unforeseen environments, obstacles, and input constraints and to deal w/ modeling and state estimation errors.

Note to Ram: describe how to use LQR for nonlinear dynamics.

Methods	Dynamics	Input/State constraints	Real-Time	Feedback?
Ackerman's Equation	LTI	no	yes	yes
LQR	LTV	no*	yes	yes
LTV Model Predictive Control	LTV	yes*	yes	yes*
Nonlinear Trajectory Design	Nonlinear	yes	no	no

We formulate the control design problem as an optimization problem:

Def: An optimization problem is:

$$\min_{z \in Z} J(z)$$

subject to $g_i(z) \leq 0 \quad \forall i \in \{1, \dots, p_i\}$
$h_i(z) = 0 \quad \forall i \in \{1, \dots, p_e\}$

where Z is the decision set, $J : Z \to \mathbb{R}$ is the objective function or cost is minimized over the decision variable z that satisfy the inequality constraints $g_i : Z \to \mathbb{R}$ and equality constraints $h_i : Z \to \mathbb{R}$.
The set of points that satisfy the constraints ($\{z \mid g_i(z) \leq 0 \ \forall i,\ h_i(z) = 0 \ \forall i\}$) is called the feasible set.

Note: We focus on Z that is $\mathbb{R}^p$ where $p \in \mathbb{N}$.

Ex: Consider the kinematic steering model

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix} = f(x, u)$$

x: p_x, p_y, θ $\qquad$ u: v, ω

Suppose we start at $(p_x, p_y, \theta) = (0, 0, 0)$ at $t = 0$ and want to find a control that drives the system to $p_x = 1$, $p_y = 1$ at $t = 1$.
In addition, suppose we want to avoid an obstacle that is a circle w/ radius 0.1 at $(0.5, 0.5)$ under input constraints $v \in [-1, 1]$, $\omega \in [-1, 1]$.

Step 1: Discretize dynamics
Use Eulers Integration to represent solution:
Select step size $h < 1$ such that $\frac{1}{h} \in \mathbb{N}$, then

$$x(h(k+1)) = x(hk) + h f(x(kh), u(kh))$$

for each $k \in \{0, \dots, \frac{1}{h} - 1\}$

Step 2: Decision variables become $\{x(kh)\}_{k=0}^{\frac{1}{h}}$ and $\{u(kh)\}_{k=0}^{\frac{1}{h}-1}$

Question: What is Z?

Step 3: Check constraints at the discretization points

a) Equality constraints represent the dynamics of the system, and initial condition

b) Inequality constraints

i.) ensure input constraints are satisfied:

$$-1 \leq u(kh) \leq 1 \qquad \forall k \in \{0, \dots, \tfrac{1}{h}-1\}$$

ii) check obstacle constraint

$$(p_x(kh) - 0.5)^2 + (p_y(kh) - 0.5)^2 - (0.1)^2 \geq 0$$

for each $k \in \{1, \dots, \frac{1}{h}\}$.

Step 4: Cost function:

$$J\left(\{x(kh)\}_{k=0}^{\frac{1}{h}}, \{u(kh)\}_{k=0}^{\frac{1}{h}-1}\right)$$

$$= (p_x(1) - 1)^2 + (p_y(1) - 1)^2$$

$$\min_{\{x(kh)\}_{k=0}^{\frac{1}{h}}, \{u(kh)\}_{k=0}^{\frac{1}{h}-1}} J\left(\{x(kh)\}_{k=0}^{\frac{1}{h}}, \{u(kh)\}_{k=0}^{\frac{1}{h}-1}\right)$$

subject to
$$x(k(h+1)) = x(kh) + hf(x(kh), u(kh))$$
$$\forall k \in \{0, \dots, \tfrac{1}{h}-1\}$$

$$x(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$-(p_x(kh) - 0.5)^2 - (p_y(kh) - 0.5)^2 + (0.1)^2 \leq 0$$
$$\forall k \in \{0, \dots, \tfrac{1}{h}\}$$

$$\left.\begin{aligned} v(kh) - 1 &\leq 0 \\ -v(kh) - 1 &\leq 0 \\ w(kh) - 1 &\leq 0 \\ -w(kh) - 1 &\leq 0 \end{aligned}\right\} \forall k \in \{0, \dots, \tfrac{1}{h}-1\}$$

Ram Note: 1) fill in table.
2) describe how solvers work.

- When is this problem easy to solve?
  ↳ solution (if slow to generate) cannot be generated rapidly enough to use as feedback

Our approach will rely on using derivatives to solve the optimization problem.

Def: 1) Let J^* be the optimal value of the optimization problem. A global optimizer is a feasible z^* such that $J(z^*) = J^*$

2) A feasible $\bar{z}$ is a local optimizer if there exists an $R>0$ such that

$$J(\bar{z}) = \min_{z \in Z} J(z)$$

subject to
$$g_i(z) \leq 0 \quad \forall i \in \{1, \dots, n_{ie}\}$$
$$h_i(z) = 0 \quad \forall i \in \{1, \dots, n_e\}$$
$$\|z - \bar{z}\| \leq R$$

Since we rely on derivative information local minimizers will be easy to compute, but global minima will be hard to compute. However sometimes all local minima are global minima:

this lecture: LTV Model Predictive Control
(for this problem local = global)

next lecture: Nonlinear Trajectory Design
(for this problem local $\neq$ global)

objectives for today

1. why is LTV-MPC easy to solve globally, but nonlinear optimization is not?
2. how to formulate easy optimization problems?

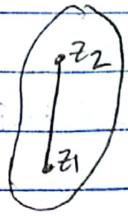
   Note: we will not speak about how to solve them; however, the state of algorithms is so robust all major solvers have the same interface (i.e. easy problems are described the same way)
3. how to formulate obstacle avoidance problems as an easy problem?

III Convexity

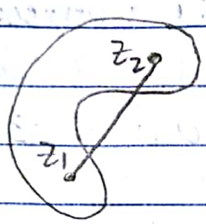
Def: a) A set $Z \subset \mathbb{R}^s$ is convex if
$$\lambda z_1 + (1-\lambda) z_2 \in Z$$
for all $z_1, z_2 \in Z$, $\lambda \in [0,1]$.

b) A function $f: Z \to \mathbb{R}$ is convex if Z is convex and
$$f(\lambda z_1 + (1-\lambda) z_2) \leq \lambda f(z_1) + (1-\lambda) f(z_2)$$
for all $z_1, z_2 \in Z$, $\lambda \in [0,1]$.

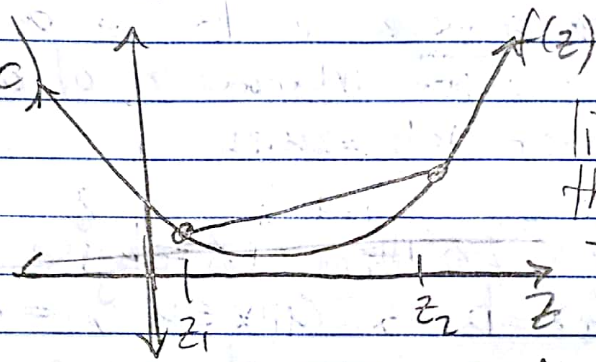
c) A function $f: Z \to \mathbb{R}$ is strictly convex if Z is convex and
$$f(\lambda z_1 + (1-\lambda) z_2) < \lambda f(z_1) + (1-\lambda) f(z_2)$$
for all $z_1, z_2 \in Z$, $\lambda \in [0,1]$.

Ex: a) 

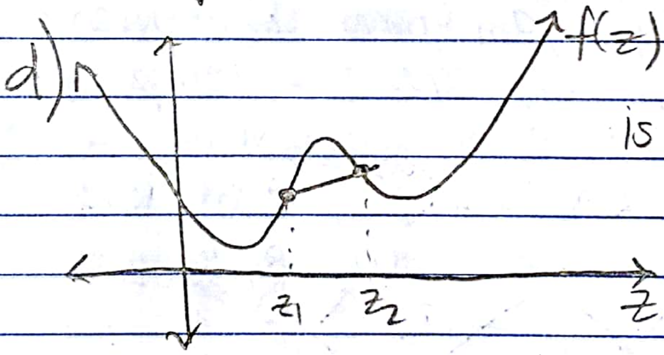
is a convex set

b) 

is not a convex set

c) 

line is above the function, so this function is convex.

d) 

is not convex

e) An affine function $f(z)=c'z+d$ is convex

f) A quadratic function $f(z)=z'Qz+2sz+r$ is convex iff Q is positive semidefinite.

g) A quadratic function $f(z)=z'Qz+2sz+r$ is strictly convex iff Q is positive definite

Theorem: 1) The intersection of an arbitrary number of convex sets is a convex set.

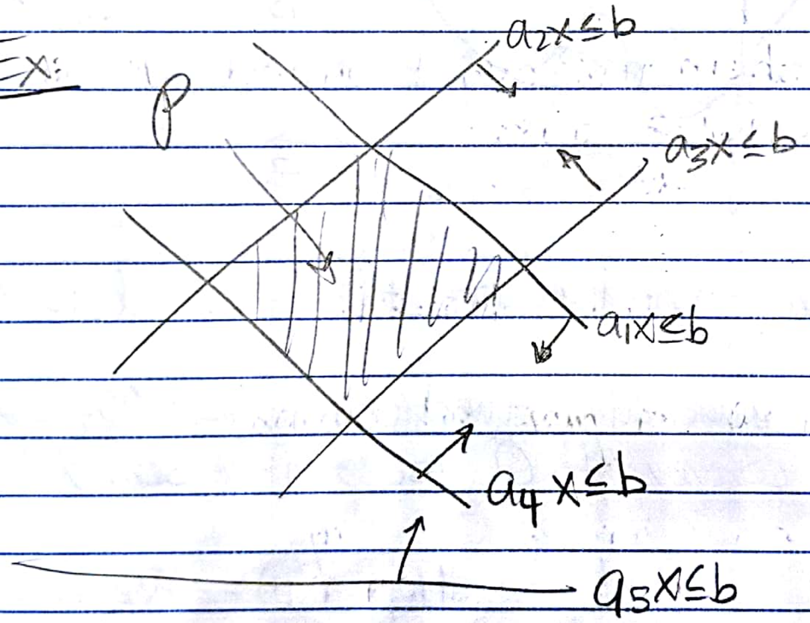
2) If $f_1, \ldots, f_N$ are convex functions, then $\sum \alpha_i f_i$ is a convex function for all $\alpha_i \geq 0$. $\forall i \in \{1, \ldots, N\}$.

Def: A convex polytope $P \subset \mathbb{R}^s$ is a set defined as the intersection of a finite number of half spaces:

$$P = \{x \in \mathbb{R}^s \mid Ax \leq b\}$$

where $Ax \leq b$ is $a_i x \leq b_i$, $i = 1, \ldots, m$ where $a_1, \ldots, a_m$ are the rows of A and $b_1, \ldots, b_m$ are the components of b.

Ex:



Note: A convex polytope is a convex set.

IV Convex Optimization

Def: An optimization problem is convex if the objective function is convex and the feasible set is convex.

Theorem: In convex optimization problems local minimizers are global optimizers.

Note: As a result, we can just use derivative information to find local minima $\Rightarrow$ global minima!

A. Linear Programming

Def: A linear program is an optimization problem that can be written in the following form:

$$\min_{z \in \mathbb{R}^s} \; c'z$$

$$\text{s.t. } Az \leq b$$

Theorem: Linear programs are convex.

Ex: A farmer has a piece of land L km^2 to be planted w/ wheat or barley or some combination of the two. The farmer has a limited amount of fertilizer, F kilograms, and pesticide, P kilograms. Every square kilometer of wheat requires f_1 kg of fertilizer and P_1 kg of pesticide, while every square kilometer of barley requires F_2 kg of fertilizer and P_2 kg of pesticide. Let S_1 be the selling price of wheat/km^2 and S_2 be the selling price of barley/km^2. Determine the optimal amount of wheat x_1 and barley x_2 to plant:

$$\min_{x_1, x_2 \in \mathbb{R}} \; -S_1 x_1 - S_2 x_2 \qquad \text{(max revenue)}$$

$$x_1 + x_2 \leq L \qquad \text{(limit on total area)}$$

$$F_1 x_1 + F_2 x_2 \leq F \qquad \text{(limit on fertilizer)}$$

$$P_1 x_1 + P_2 x_2 \leq P \qquad \text{(limit on pesticide)}$$

$$x_1, x_2 \geq 0 \qquad \text{(cannot plant a negative area)}$$

<u>Note:</u> We can use linprog in MATLAB to solve LP but many commercial solvers exist (CPLEX, GUROBI, MOSEK, etc.)

B. Quadratic Programming

<u>Def:</u> A <u>quadratic program</u> is an optimization problem ~~that~~ can be written as:

$$\min_{z \in \mathbb{R}^s} \; \frac{1}{2} z' Q z + c' z$$

$$\text{subject to } \; Az \leq b$$

where Q is symmetric.

Theorem: If $Q \succeq 0$, then the quadratic program is convex.

Note: We can use quadprog to solve in MATLAB.

V. QP Based Model Predictive Control

Basic Idea:

1. Rely on fast speed of QP solvers
2. At each sampling time k, starting at current state, compute an optimal controller at a sequence of time steps over a finite horizon $[k, k+K]$
3. Use the sample for the first computed sampling instance $[k, k+1]$
4. Repeat step 2 at new location $[k+1, K+1]$

Note: 1) This approach is used everywhere!
2) Allows us to do feedback!
3) If we have an LTV continuous time system, we can apply Euler integration to create a discrete-time system:

$$x(k+1) = A(k)x(k) + B(k)u(k)$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$, $A(k) \in \mathbb{R}^{n \times n}$, and $B(k) \in \mathbb{R}^{n \times m}$ for each $k \in \mathbb{N}$.

Notation: 1) Let $u_{k+k'|k}$ be the input at time step $k+k'$ computed at time step k.

2) Given $\{u_{k+k'|k}\}_{k'=0}^{K-1}$ let $x_{k+k'|k}$ be the state at time step $k+k'$.

Consider the problem of ensuring that the origin of a discrete-time LTV system

$$x(k+1) = A(k)x(k) + B(k)u(k)$$

subject to constraints that $x(k) \in \mathcal{X}$ and $u(k) \in \mathcal{U}$ $\forall k$ where $\mathcal{X}$ and $\mathcal{U}$ are convex polytopes. We solve the problem iteratively as:

$$\min_{\{u_{k+k'|k}\}_{k'=0}^{K-1},\ \{x_{k+k'|k}\}_{k'=0}^{K}} \sum_{k'=0}^{K} x_{k+k'|k}^T Q(k') x_{k+k'|k} + \sum_{k'=0}^{K-1} u_{k+k'|k}^T R(k') u_{k+k'|k}$$

$$\left.\begin{aligned} x_{k+k'+1|k} &= A(k+k')x_{k+k'|k} + B(k+k')u_{k+k'|k} \\ u_{k+k'|k} &\in \mathcal{U} \end{aligned}\right\} \forall k' \in \{0, \ldots, K-1\}$$

$$x_{k+k'|k} \in \mathcal{X} \quad \forall k' \in \{0, \ldots, K\}.$$

Algorithm for Model Predictive Control

1) Measure the state $x_{k|k}$ at time k
2) Solve QP
3) If no solution exists, then stop and quit.
4) Apply $u_{k|k}$ at time step k
5) Wait for new sampling time $k+1$, go to step 1

Ex: Consider the system

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

Try to regulate system to origin while ensuring that

$$-10 \leq u(k) \leq 10$$

$$\begin{bmatrix} -2 \\ -2 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$