SECOND NATURE

Goals
1. Motivation for Optimization Based Control
2. Convexity
3. Convex Optimization
4. QP Based MPC
5. Spatial Discretization for Obstacle Avoidance.

I Motivation for Optimization Based Control

Need algorithmic intelligence to generate controllers in unforeseen environments, obstacles and input constraints are important to satisfy.

Methods	Dynamics	Input/State Constraints	Time Discretized Constraint Check	Real-Time?
Ackermann's Equation	LTI	no	NA	yes
LQR	LTV	no*	NA	yes*
LTV-MPC	LTV	yes	yes	yes*
Nonlinear Trajectory Design	Nonlinear	yes	yes	no

show this afterwards jump to page 2 description first

Scanned by CamScanner

put this after table w/ LTI and LQR

We will formulate our control design problem as an optimization problem:

Def: An optimization problem is:

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

decision variables

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

where Z is the decision set, $J: Z \to \mathbb{R}$ is the objective function which is minimized over 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 in Z that satisfy the constraints (i.e. $g_i(z) \leq 0 \ \forall i \in \{1, \ldots, p_e\}$ and $h_i(z) \leq 0 \ \forall i \in \{1, \ldots, p_e\}$) is called the feasible set.

or cost

Note: we focus on Z that looks like $\mathbb{R}^m$

Ex: Lets suppose we have our kinematic steering model:

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ w \end{bmatrix} = f(\underbrace{x}_{p_x, p_y, \theta}, \underbrace{u}_{v, w})$$

Scanned by CamScanner

Lets suppose we want to drive this model to $(p_x, p_y) = (1, 1)$ by $t=1$ starting from $(p_x, p_y, \theta) = (0, 0, 0)$ at $t=0$.

In addition suppose we want to avoid an obstacle that is a circle with radius 0.1 centered at $(0.5, 0.5)$.

Under input constraints on $v \in [-1, 1]$, $w \in [-1, 1]$

1) Discretize dynamics

Use Euler integration to represent dynamics

Select step size h, then

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

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

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

3) Check constraints at the discretization points. Let $g_k(\{x(kh)\}_{k=0}^{1/h}, \{u(kh)\}_{k=0}^{1/h-1})$

$$= -(p_x(kh) - 0.5)^2 - (p_y(kh) - 0.5)^2 + (0.1)^2$$

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

4) Cost function: $J(\{x(kh)\}_{k=0}^{1/h}, \{u(kh)\}_{k=0}^{1/h-1})$

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

$\downarrow$

Scanned by CamScanner

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

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

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

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

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

When is this problem easy to solve?

→ Fill in the remainder of the table

→ We use derivatives to find solutions to opt. problems.

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 $R > 0$ s.t.

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

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

$h_i(z) = 0 \quad \forall i \in \{1, \dots, p_e\}$

$\|z - \bar{z}\| \leq R$

Scanned by CamScanner

Local optima is easy to compute
Global optima is not...

this lecture → LTV-MPC = linear time varying model predictive control

Nonlinear Trajectory Design = nonlinear optimization based

next lecture

objective for today:

1). why is LTV-MPC easy to solve but nonlinear optimization isn't (i.e why are certain optimization problems easy and others are not?) ?

2) understanding how to formulate such easy optimization problems. We will not speak about how they are solved; however the state of algorithms is so robust that all major solvers have the same interface (i.e. easy problems are described in the same way)

3). Understand how to obstacle avoidance using an easy problem.

I Convexity

Def: a) A set $Z \in \mathbb{R}^s$ 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 $f: Z \rightarrow \mathbb{R}$ 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 $f: Z \rightarrow \mathbb{R}$ is strictly convex if Z is convex and

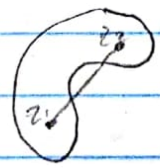
$$f(\lambda z_1 + (1-\lambda) z_2) < \lambda f(z_1) + (1-\lambda) f(z_2)$$

for all $z_1, z_2 \in Z$, $\lambda \in (0,1)$

Ex: a)

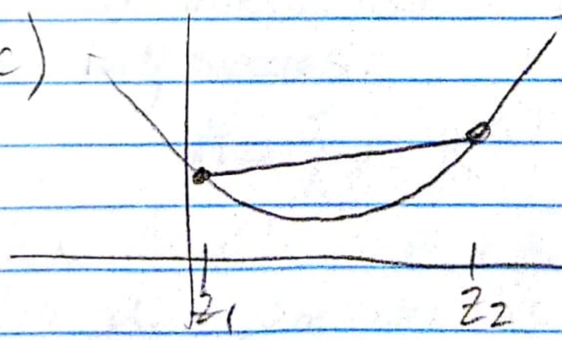
is a convex set

b)



isn't a convex set

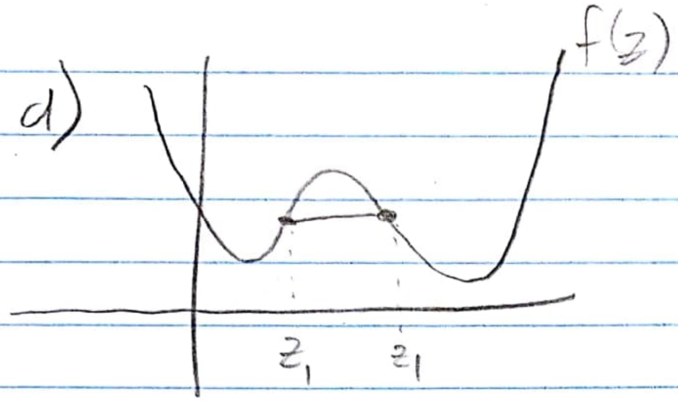
c)



$f(z)$

line is above the function this function is convex

z

d) 

is not convex

e) A linear function $f(z) = c'z + d$ is convex

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

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

<u>Theorem:</u> 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 $\sum_{i=1}^{N} \alpha_i f_i$ is a convex function for all $\alpha_i \geq 0$, $i = 1, \ldots, N$.

<u>Def:</u> A <u>convex polytope</u>, $\mathcal{P} \subseteq \mathbb{R}^s$, is a set defined as the intersection of a finite number of half spaces:

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

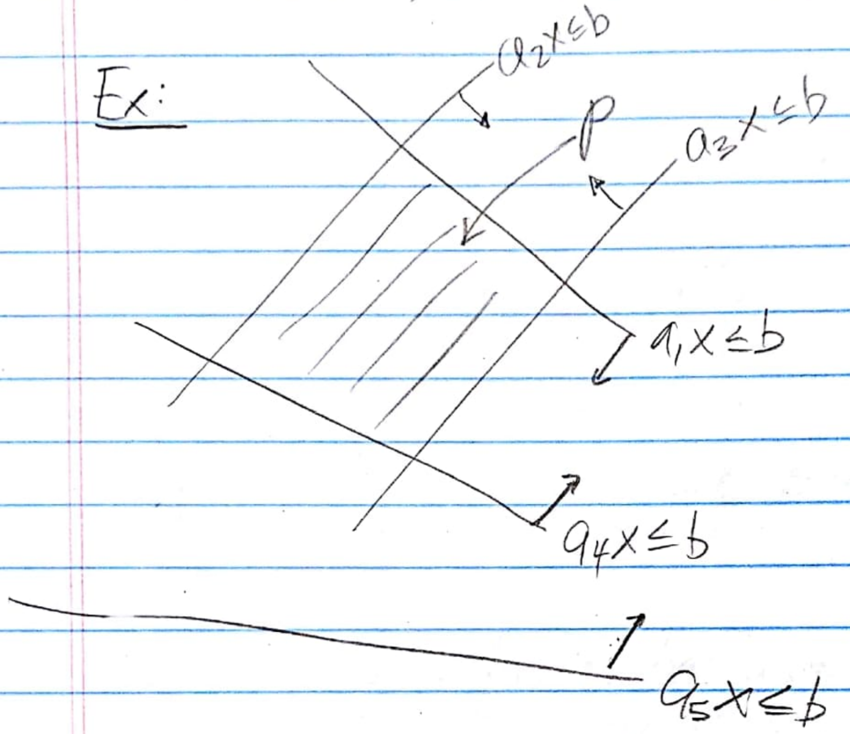
where $Ax \leq b$ is $a_i x \leq b_i$, $i = 1, \ldots, m$

where $a_1, \ldots, a_m$ are the rows of A

Scanned by CamScanner

and $b_1, \ldots, b_m$ are the components of b.

Ex:



Note: A convex polytope is a convex set.

III Convex Optimization

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

Theorem: In convex optimization problems local optimizers 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:

$$\min_{z \in \mathbb{R}^s} c'z$$
$$\text{subject to } Ax \leq b$$

Theorem: Linear programs are convex optimization problems.

Ex: Suppose a farmer has a piece of land say L km², to be planted w/ wheat or barley or some combination of the two. 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/km² and S_2 be the selling price of barley/km². Determine optimal amount of wheat x_1 and barley x_2 to plot:

Scanned by CamScanner

$$\min_{x_1, x_2 \in \mathbb{R}^2} \; -S_1 x_1 - S_2 x_2 \qquad \text{(max revenue)}$$

$$x_1 + x_2 \leq L \qquad \text{(limit on total area)}$$
$$F_1 x_1 + F_2 x_2 \leq F \qquad \text{(limit on fertilizer)}$$
$$P_1 x_1 + P_2 x_2 \leq P \qquad \text{(limit on pesticide)}$$
$$x_1 \geq 0, \; x_2 \geq 0 \qquad \text{(cannot plant a negative area)}$$

Note: We can use linprog to solve in MATLAB, but many solvers exist commercially (CPLEX, GUROBI, MOSEK, etc.).

B. Quadratic Programming

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

$$\min_{z \in \mathbb{R}^5} \; \frac{1}{2} z' Q z + c' z$$

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

where Q is a symmetric matrix

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

Note: We can use quadprog to solve in MATLAB

Scanned by CamScanner

IV. QP Based Model Predictive Control

Basic Idea of Model Predictive Control

1. Rely on fast speed of QP solvers.
2. At each sampling time, starting at current state, compute an optimal controlles at a sequence of time steps over 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 technique is used extensively in real world control.

2) Suppose we have an LTV continuous time system; we can apply Euler integration to create a discrete time system:

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

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

$$(\Delta t A(k) + I)x(k) + \Delta t B(k)u(k)$$
$$\Delta t(A(k)x(k) + B(k)u(k))$$

$Ax \leq b$

$-Ax \leq -b$

Scanned by CamScanner

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 $k+k'$ under $\{u_{k+k'|k}\}_{k'=0}^{K-1}$.

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

$$x(k+1) = A(k)\,x(k) + B(k)u(k)$$
$$y(k) = C(k)x(k)$$

subject to constraints that $x(k) \in \mathcal{X}$ and $u(k) \in \mathcal{U}$ $\forall k$ where $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$ are convex polytopes. Fix a horizon $K > 0$. We solve this problem by solving the following problem iteratively

$$\min_{\{u_{k+k'|k}\}_{k'=0}^{K-1},\ \{x_{k+k'|k}\}_{k'=0}^{K}} \sum_{k'=0}^{K} x_{k+k'|k}' Q(k') x_{k+k'|k} + u_{k+k'|k}' R(k') u_{k+k'|k}$$

subject to

$$x_{k+k'+1|k} = A(k+k')x_{k+k'|k} + B(k+k')u_{k+k'|k}$$

$$x_{k+k'|k} \in \mathcal{X}, \quad u_{k+k'|k} \in \mathcal{U}$$

$$\forall k' = 0, \dots, K-1 \qquad \forall k' = 0, \dots, K$$

Scanned by CamScanner

Algorithm for Model Predictive Control

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

Ex: Consider the system:

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

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

Try to regulate system to origin while ensuring that

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

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

$x \leq 5$

$-x \leq 5$

$x \geq -5$

Scanned by CamScanner