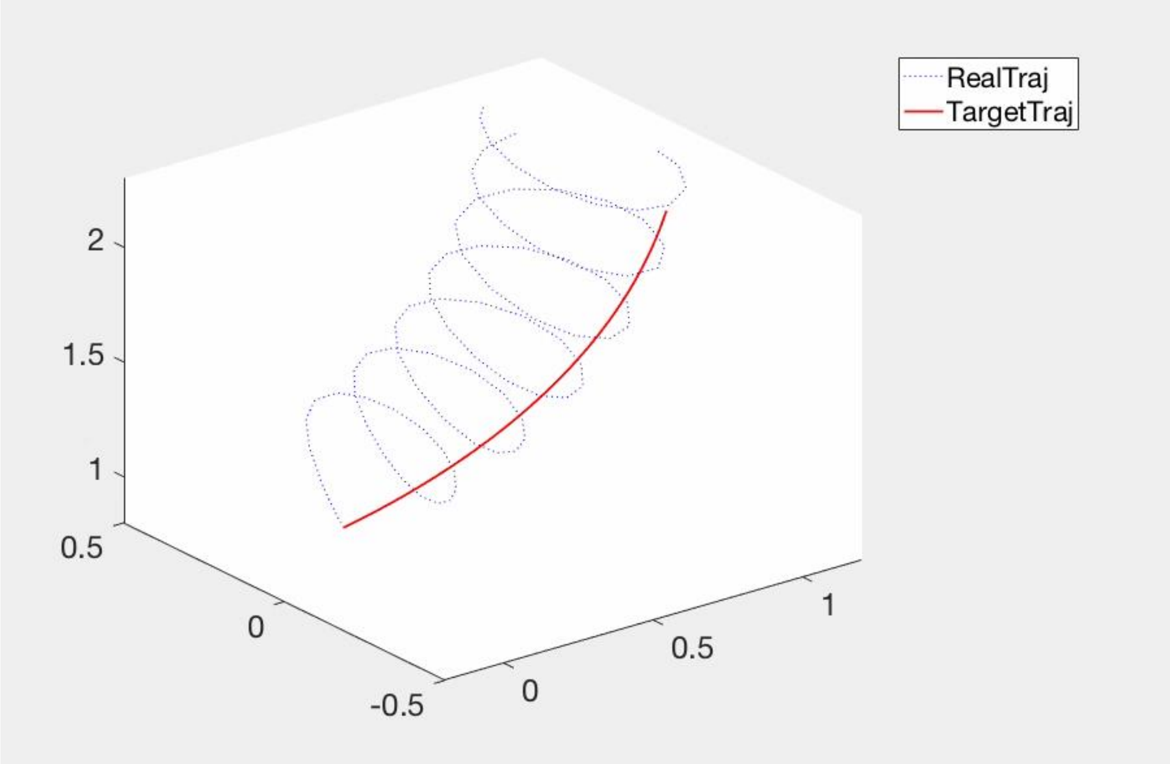
A Homotopy method for motion planning

M.-A. Belabbas
Electrical and Computer Engineering Department
Coordinated Science Lab
University of Illinois, Urbana-Champaign

Motion planning can mean a lot of things to a lot of people

- Changing, unknown environment

- **Goal**: find controls to reach final destination
- High-level control: send ‘move to (x,y)’ commands, algorithmic solutions



- Addresses the *difficult* problem of planning with *non-holonomic* dynamics

$$\dot{x} = \sum_i u_i f_i(x)$$

$$\dot{x} = \sqrt{\frac{1}{\varepsilon}} \cos\left(\frac{t}{\varepsilon}\right) f_1(x) + \sqrt{\frac{1}{\varepsilon}} \sin\left(\frac{t}{\varepsilon}\right) f_2(x) \Leftrightarrow \dot{x} = [f_1, f_2](x)$$

- **Goal**: find *controls* to follow target trajectory.
- Provides: Analytic solutions

Motion planning = find controls to drive you to a desired destination

New method for motion planning for nonlinear systems

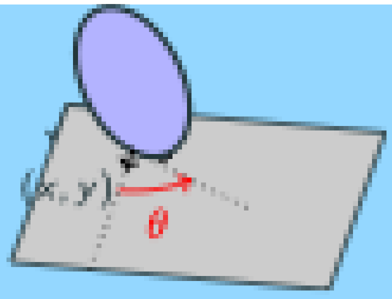
in:

1. Description of environment
2. Description of dynamics
3. Desired final location/state

out:

Controls that send you to destination

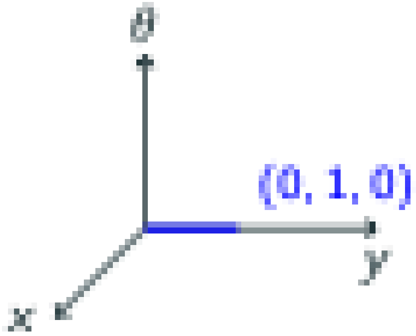
- Handles holonomic, nonholonomic, obstacle constraints
- Provides an algorithm, and strong guarantees of convergence
- Yields *natural* motions

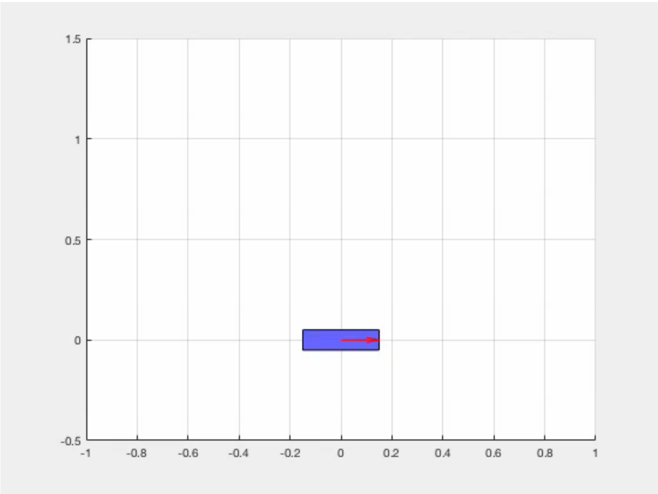


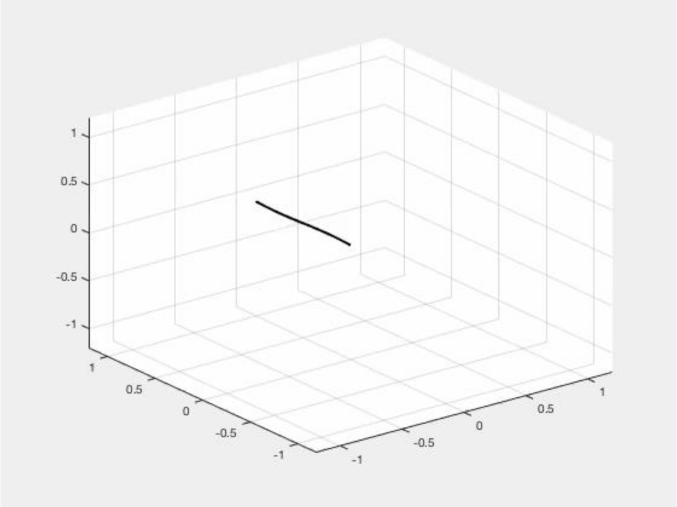
$$\dot{x} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_2$$

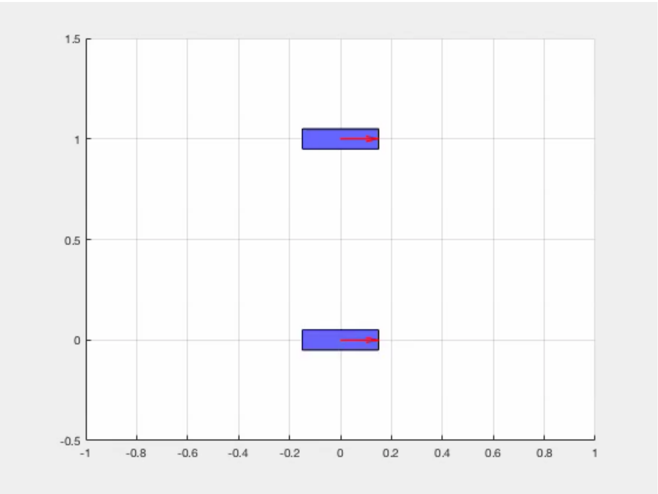
Goal: move unicycle from (0,0,0) to (0,1,0)

Initial trajectory



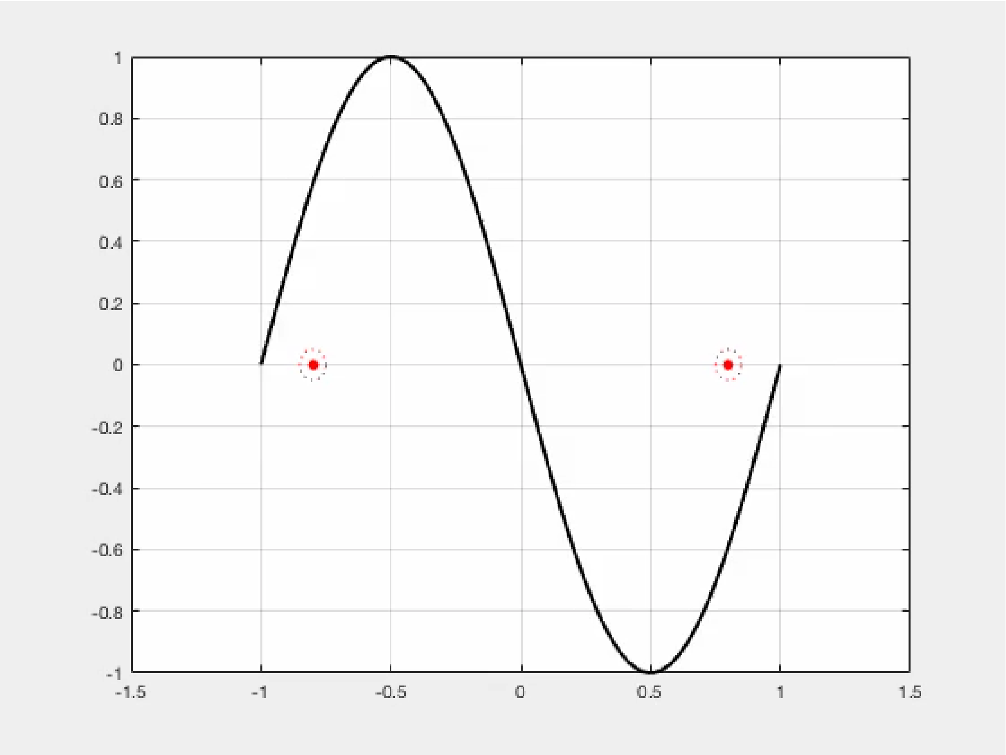


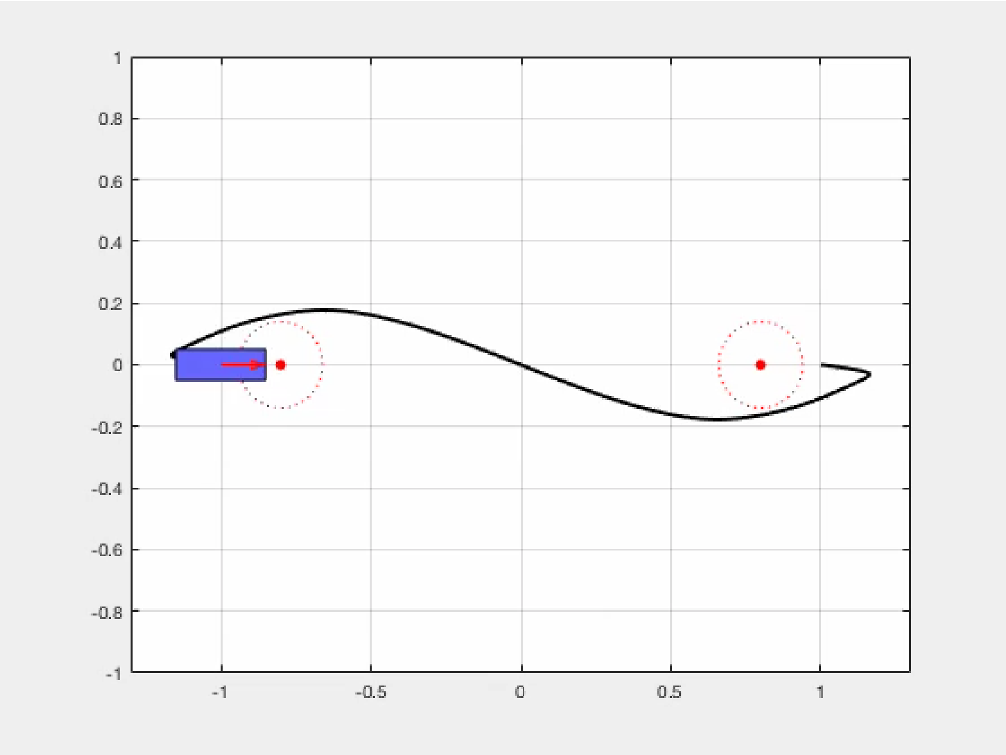




With obstacles

- Warning: homotopies always take place in configuration space and are differentiable
- We show a projection onto 2D workspace





Outline of the presentation

1. Introduction and motivation
2. Presentation of the method
3. Complexity, drift and bounded controls
4. Case study: wheeled vehicles
5. Sketch of proof of convergence

Three steps of the method

1. The constraints: think local
   When at a point in configuration space, what are good directions to follow?

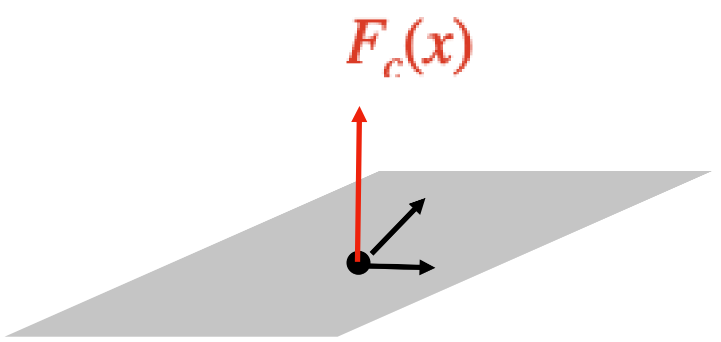
2. Homotopy
   Transform an initial path to one that meets constraints

3. Extract controls

Think local!

- From a local perspective: 2 types of constraints

Hard constraints



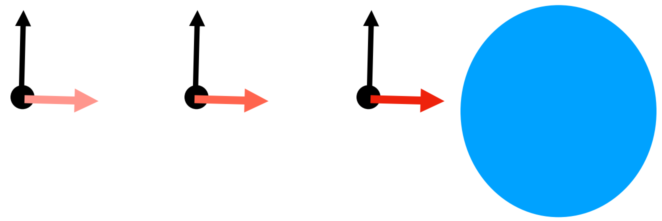
- Holonomic/non-holonomic:
  motion in some direction not allowed

Mathematically, create a matrix

$$F_c(x) = \begin{pmatrix} | & | & \cdots & | \\ f_1 & f_2 & \cdots & f_p \\ | & | & \cdots & | \end{pmatrix}$$

where the columns span the forbidden motion

Soft constraints



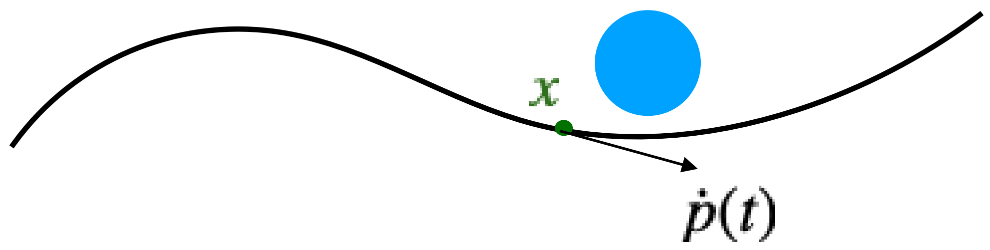
- Obstacle:
  unless exactly at boundary, no restriction on motion

Mathematically, described by "barrier" function *r(x)*

$$\mathcal{O} = \{x \in Q \mid r(x) \geq 0\}$$

Naive approach: minimize integral cost of forbidden directions

- Let *p(t)* be a trajectory in configuration space joining initial to desired final states
- How much is the trajectory using constrained motions at *x?*



- Motion at *x=p(t)* is given by $\dot{p}(t)$
- Constrained motions are $F_c(x)$
- Proximity to obstacle is given by $\left|\frac{1}{\varepsilon + r(x)}\right|$, where ε is a tolerance parameter

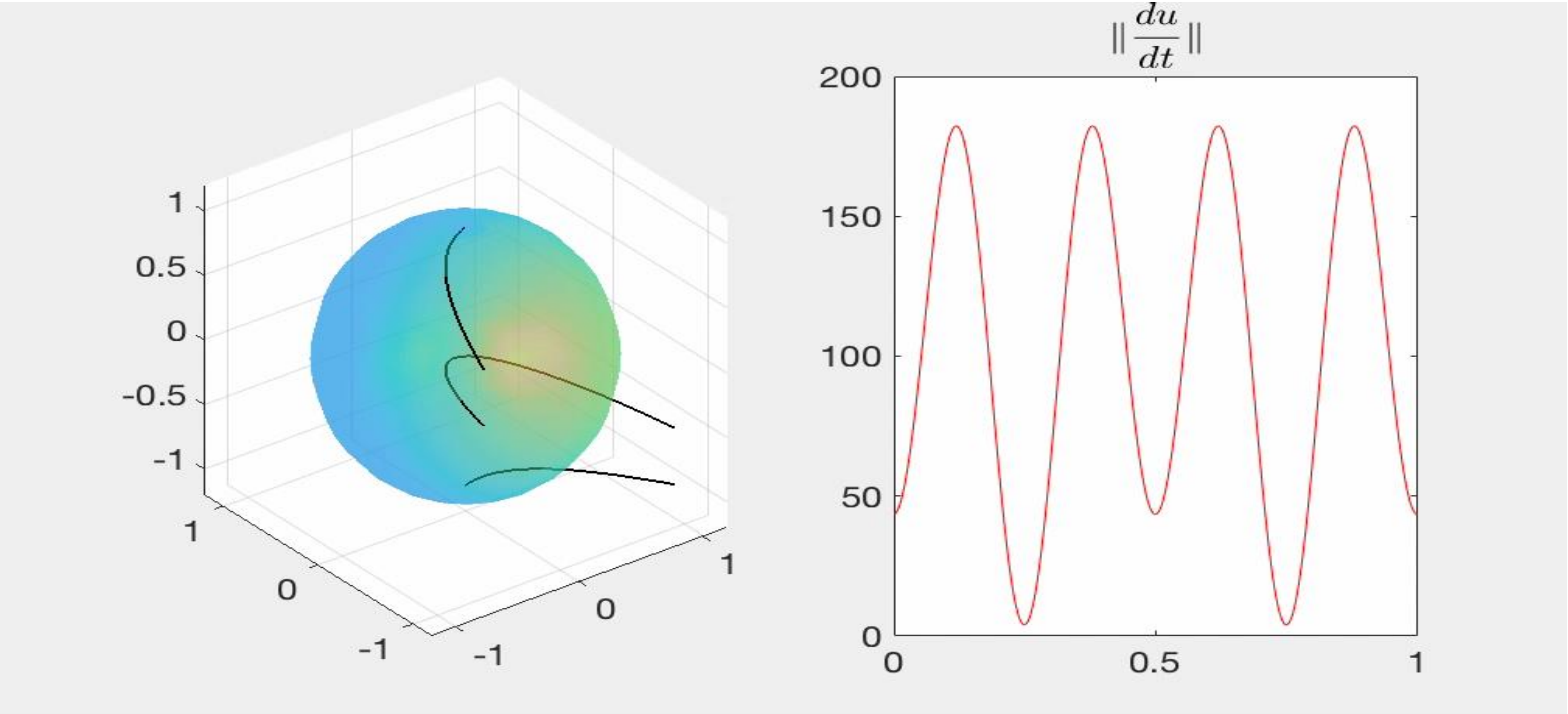
The quantity $\|F_c^\top(x)\dot{p}(t)\|^2 \left|\frac{1}{\varepsilon + r(x)}\right|^2$ measures how much "bad directions" are used at *p(t)*

Naive approach: minimize cost of forbidden directions

- We assign the cost J to a path $p(t)$:

$$J(p) := \int_0^{t_f} \|F_c^\top(p(t))\dot{p}(t)\|^2 \left|\frac{1}{\varepsilon + r(p(t))}\right|^2 dt$$

- Derive the *gradient flow* for this functional, in path space



- Converges to a *discontinuous* path in general

The fix: ``regularization''

- Discontinuous curve $\Leftrightarrow$ large derivative
- $\rightarrow$ Penalize the norm of the derivative

$$\text{minimize } E_k(p) := J(p) + \frac{1}{k}\int_0^{t_f} ||\dot{p}||^2 dt$$

- The good: minima are continuous
- The bad: when k<∞, we are *not* solving our original problem
- The ugly: we can try to solve for *k* large and hope to have *p* close to minima of *J*, but we know that for k=∞, we converge to a discontinuous *p.*

Theorem: Let $p_k(t)$ be a minimizer of E_k. Then $\lim_{k\to\infty} p_k(t)$ is continuous.

Riemannian geometry naturally appears when dealing with localized constraints

Observations: 1. we can always find a G so that the functional E can be expressed as

$$E(p) := J(p) + \frac{1}{k}\int_0^{t_f} \|\dot{p}\|^2 dt \quad \longrightarrow \quad E(p) = \int_0^{t_f} \dot{p}^\top G(p(t))\dot{p}\,dt$$

2. This is the energy functional, whose minimizers are geodesics for the metric associated with G!

Admissible paths ← geodesics for "good metric"

The method in three steps redux

➞

1. Localize the constraints
   ➞Encode in a positive definite matrix

2. Homotopy
   Transform an initial path which does *not* meet the constraints to one that meets constraints
   Becomes a Geometric Heat Flow (GHF)

3. Extract controls

Encoding the constraints in an inner product

Set $F_f(x) \longleftarrow F_c^{\perp}(x)$ and $F(x) = \begin{pmatrix} | & | \\ F_c & F_f \\ | & | \end{pmatrix}$

F is square and invertible.

- Denote by *p* the number of columns of $F_c(x)$
- Set $b(x) := \left| \dfrac{1}{\varepsilon + r(x)} \right|$
- Define the family of inner products (metrics)

$$G_k(x) := F(x) \begin{pmatrix} k & 0 & \cdots & & & 0 \\ 0 & k & 0 & \cdots & & 0 \\ \vdots & & \ddots & & & \vdots \\ 0 & \cdots & & b(x) & & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & \cdots & & & & b(x) \end{pmatrix} F^{\top}(x)$$

The method in three steps redux

1. Localize the constraints

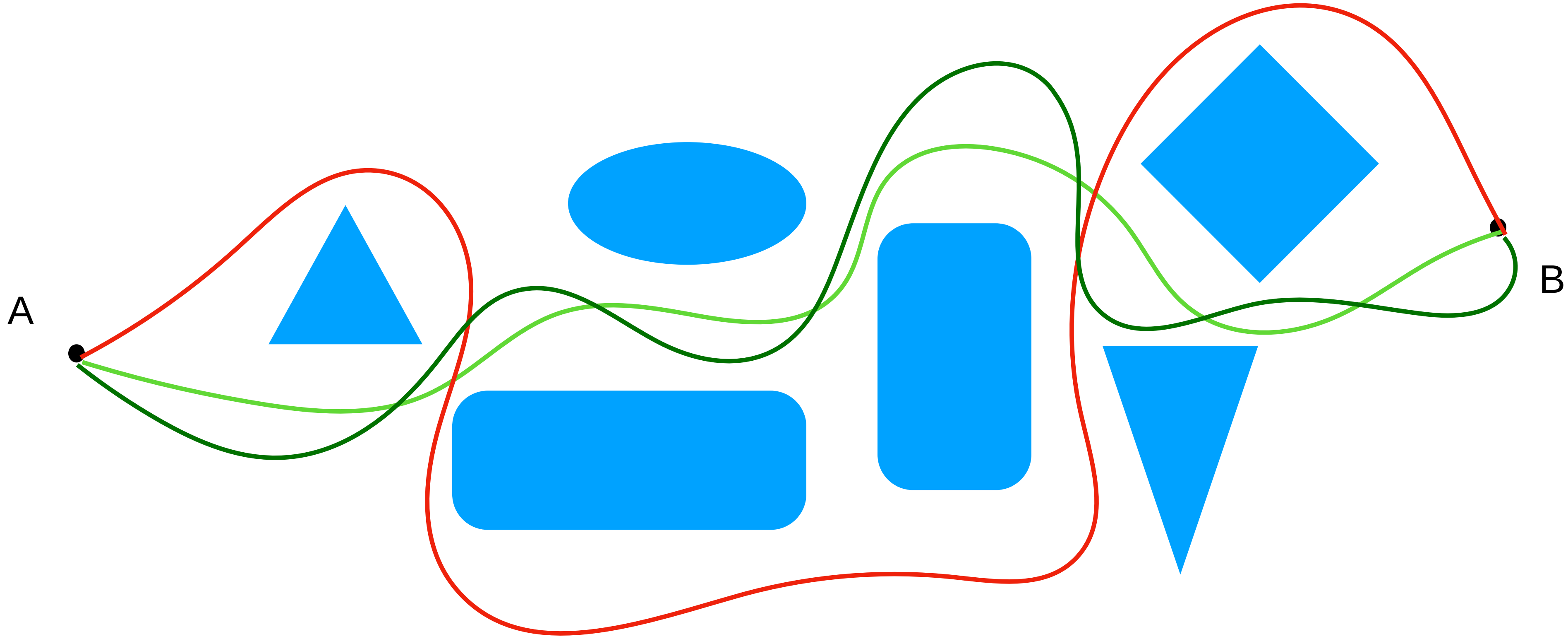
   When at a point in configuration space, what are good directions to follow?
   Add regularization for complementary directions
   ➞Riemannian inner product

2. Homotopy

   Transform an initial path which does *not* meet the constraints to one that meets constraints
   Becomes a Geometric Heat Flow (GHF)

3. Extract controls

Why use homotopy of paths?



Homotopic paths have similar “macroscopic” characteristics

1. + : if you need to insure properties of paths
   Integrate prior knowledge into initial trajectory

2. - : if you want to optimize global properties.
   Need preprocessing to find optimal homotopy class

Homotopy through geometric heat flow

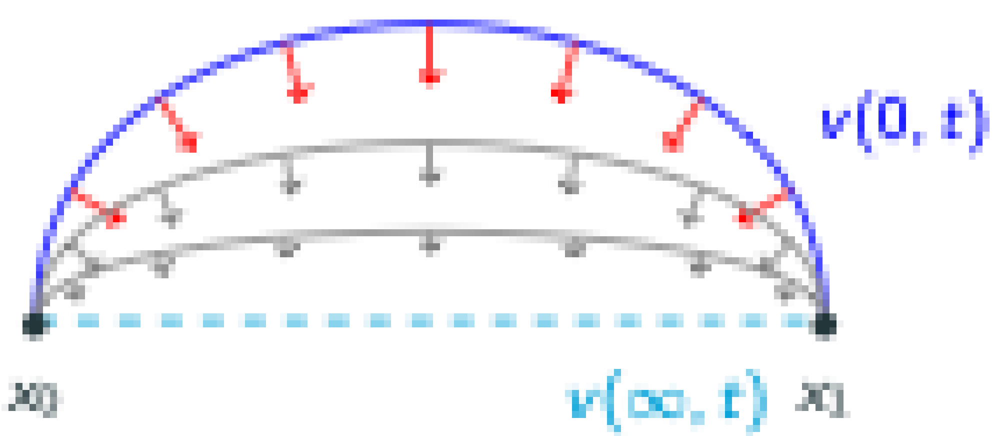
- In our problem, we are given 1. Fixed initial and final states

  2. An initial path between these states that avoid obstacles, but does not meet constraints necessarily.

- **Goal**: minimize length for *G(x)* amongst differentiable paths joining these states

Large body of research in the area, under the name of *curve shortening flows* or *Geometric heat flows*

[Angenent, Huisken, Grayson, Altschuler, Abresch, Langer (between 1986-1991)]



Homotopy

$$v(s,t) : [0,\infty] \times [0,t_f] \to Q \text{ with } v(s,0) = x_0, v(s,t_f) = x_1$$

Curve shortening: move in direction of inner normal

$$\frac{\partial v(s,t)}{\partial s} = \text{"}\dot{v}\text{"} = \kappa N$$

In general case

$$\boxed{\frac{\partial}{\partial s} v_i(s,t) = \frac{\partial^2}{\partial t^2} v_i(s,t) + \sum_{j,k} \Gamma^i_{jk} \frac{\partial v_j}{\partial t} \frac{\partial v_k}{\partial t}}$$

This is the geometric heat flow we solve

Where, we recall the *Christoffels symbols*

$$\Gamma^i_{jk}(x) := \frac{1}{2} \sum_{l=1}^{n} G^{-1}_{il} \left(\frac{\partial G_{lj}}{\partial x_k} + \frac{\partial G_{lk}}{\partial x_j} - \frac{\partial G_{jk}}{\partial x_l} \right)$$

The method in three steps redux

1. Localize the constraints

   When at a point in configuration space, what are good directions to follow?
   Add regularization for complementary directions
   ➞Riemannian inner product

2. Homotopy

   Transform an initial path which does *not* meet the constraints to one that meets constraints
   Becomes a Geometric Heat Flow (GHF)

3. Extract controls

Extract controls and convergence guarantee

- Assume system is given by $\dot{x} = \sum_{i=1}^{l} u_i b_i(x)$, and the desired final state is $\bar{x}_{t_f}$
- Let $B(x)$ be the matrix with columns $b_i(x)$

Let $p_{k,s}(t)$ be the solution using our method with integration time s and metric $G_k(x)$

Note that $\lim_{s\to\infty} p_{k,s}(t) = \arg\min \int_0^{t_f} \dot{p}^\top G_k \dot{p}\, dt$

Set $\bar{u}_{k,s}(t) = B^\dagger F_f F_f^\dagger \dot{p}_{k,s}(t)$ $\longrightarrow$ $\dot{x}^*_{s,k}(t) = \sum_{i=1}^{p} \bar{u}_{i,s,k} b_i(x^*)$

Under some assumptions (controllability), the following hold

Theorem : We have $\lim_{s,k\to\infty} p_{s,k}(t)$ is continuous

Theorem [Consistency]: We have $\lim_{s,k\to\infty} x^*_{s,k}(t) = \lim_{s,k\to\infty} p_{s,k}(t)$

Theorem: For all $\varepsilon > 0$, there exists s_m, k_m so that for all $s \geq s_m, k \geq k_m, \|x^*_{s,k}(t_f) - \bar{x}(t_f)\| < \varepsilon$

Outline of the presentation

1. Introduction and motivation
2. Presentation of the method
3. Complexity, drift and bounded controls
4. Case study: wheeled vehicles
5. Sketch of proof of convergence

On complexity

- Computationally intensive part: numerical solution of the GHF
- Good news: this is a *parabolic PDE.* Fast, parallel algorithm exist.
- It is of *much* lower complexity than solving the Hamilton-Jacobi-Bellman (HJB) equation

Hamilton-Jacobi-Bellman

$$-\frac{\partial V(t,x)}{\partial t} = -\min_{u}\left[\frac{\partial V}{\partial x} f(x,u) + c(x,u)\right]$$

- Complexity: dimension of *domain* of *V* increases linearly in dimension of Q
  - 1 equation with domain scaling with number of dimensions of Q
  - complexity is *exponential* in dimension of domain
- + : Provides *feedback* control
- - : Feedback control may not exist, no clean stopping rule

Geometric Heat Flow

$$\frac{\partial}{\partial s} v_i(s,t) = \frac{\partial^2}{\partial t^2} v_i(s,t) + \sum_{j,k} \Gamma^i_{jk} \frac{\partial v_j}{\partial t} \frac{\partial v_k}{\partial t}$$

- Complexity: domain is *(s,t)*: always two-dimensional.
  - Number of PDEs increases with dimension of Q
  - complexity increases *polynomially* in dimension Q
- Provides open loop control

Systems with drift and constrained controls

$$\dot{x} = f_d(x) + \sum_i u_i b_i(x)$$

$u \in \mathcal{U}$ a set of admissible controls

- Deciding controllability for systems with drift is *far* more complex

- From local point of view: set of allowable motions is not a *vector* space, but an *affine* space

- We use instead the functional:

$$\tilde{E}(p) := \int_0^{t_f} (\dot{p}^\top - f_{drift}) G(p(t)) (\dot{p} - f_{drift}) dt$$

- Constraints on the controls are handled via obstacles and added state variables

$$\begin{array}{l} \dot{x} = u_1 f_1 + u_2 f_2 \\ u_1 \geq 0 \end{array} \longrightarrow \begin{array}{l} \dot{x} = z_1 f_1 + u_2 f_2 \\ \dot{z}_1 = u_1 \\ \text{and obstacle } z_1 \geq 0 \end{array}$$

Systems with drift and constrained controls

Legged robot 2D planning // parallel legs

Configuration space: position and orientation (x,y,θ)

Inputs: ground reaction force ux uy

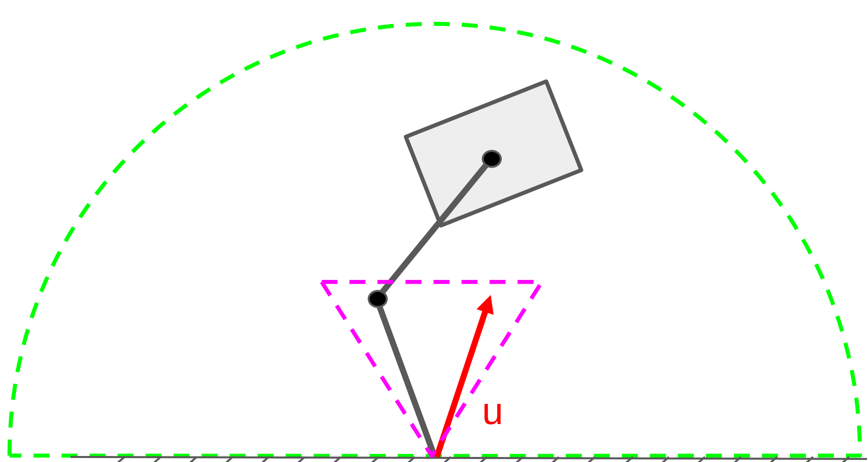
Massless: inputs to legs joint torque: $\tau = J^T \begin{bmatrix} u_x \\ u_y \end{bmatrix}$

Goal:

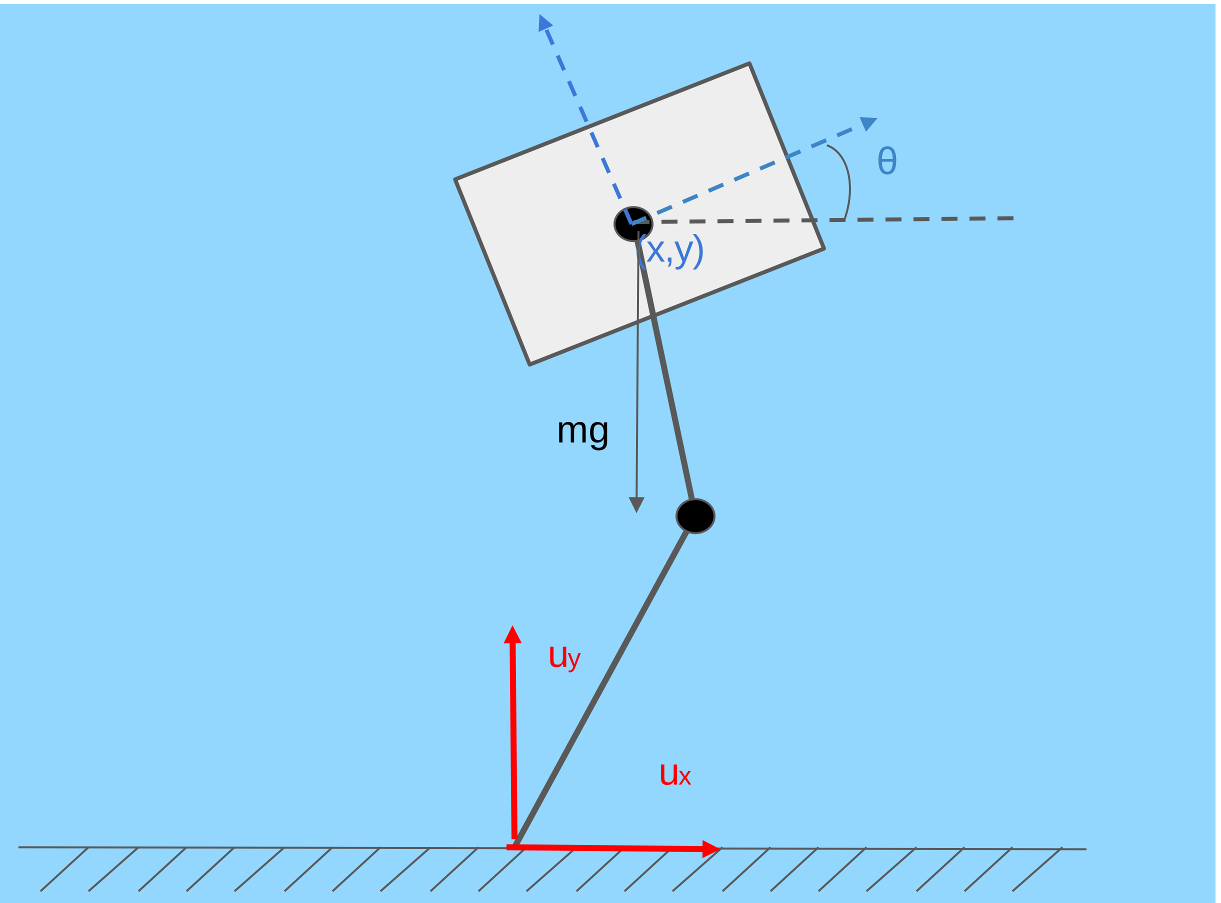
find a trajectory for x, y, θ, ux, uy with given initial and final condition.

Constraints:

- Kinematics constraints at joints
- Input constraints: friction cone



$u(t) \in$ cone



EoM: (assuming m=1)

$$\ddot{x} = u_x$$
$$\ddot{y} = u_y - g$$
$$\ddot{\theta} = u_x y - u_y x$$

$$X = \begin{bmatrix} x & y & \theta & \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T$$

$$\dot{X} = \underbrace{\begin{bmatrix} X(4) \\ X(5) \\ X(6) \\ 0 \\ -g \\ 0 \end{bmatrix}}_{\substack{f(X) \\ \text{drift term}}} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ X(2) & -X(1) \end{bmatrix}}_{\substack{F_f(X) \\ \text{admissible direction}}} u$$

Systems with drift and positivity constraints

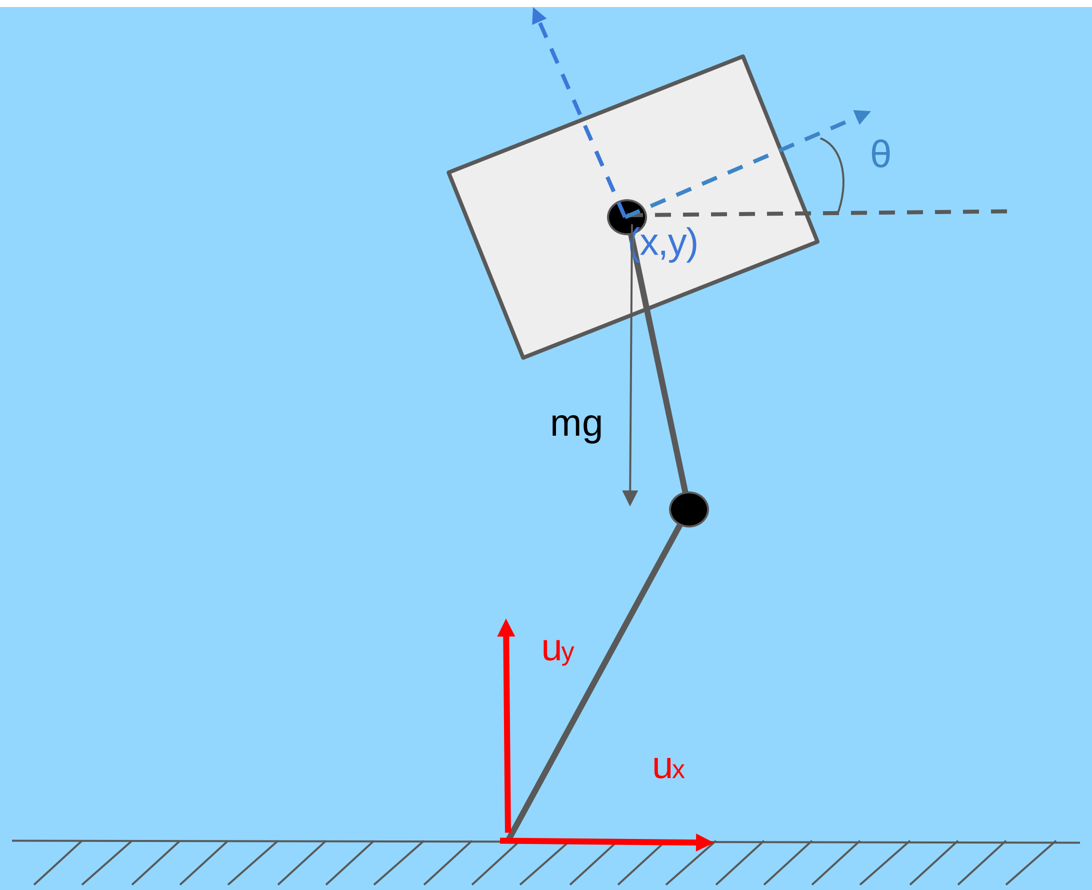
To implement the constraints, modify the system:
add the control as states

$$X = \begin{bmatrix} x & y & \theta & \dot{x} & \dot{y} & \dot{\theta} & \boxed{u_x \quad u_y} \end{bmatrix}^T$$

Update the dynamics

$$\dot{X} = \underbrace{\begin{bmatrix} X(4) \\ X(5) \\ X(6) \\ X(7) \\ X(8) - g \\ -X(1)X(8) + X(2)X(7) \\ 0 \\ 0 \end{bmatrix}}_{f(X)} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{F_f} u$$

Set obstacles on the control states to implement the friction cone



EoM: (assuming m=1)

$$\ddot{x} = u_x$$
$$\ddot{y} = u_y - g$$
$$\ddot{\theta} = u_x y - u_y x$$

$$X = \begin{bmatrix} x & y & \theta & \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T$$

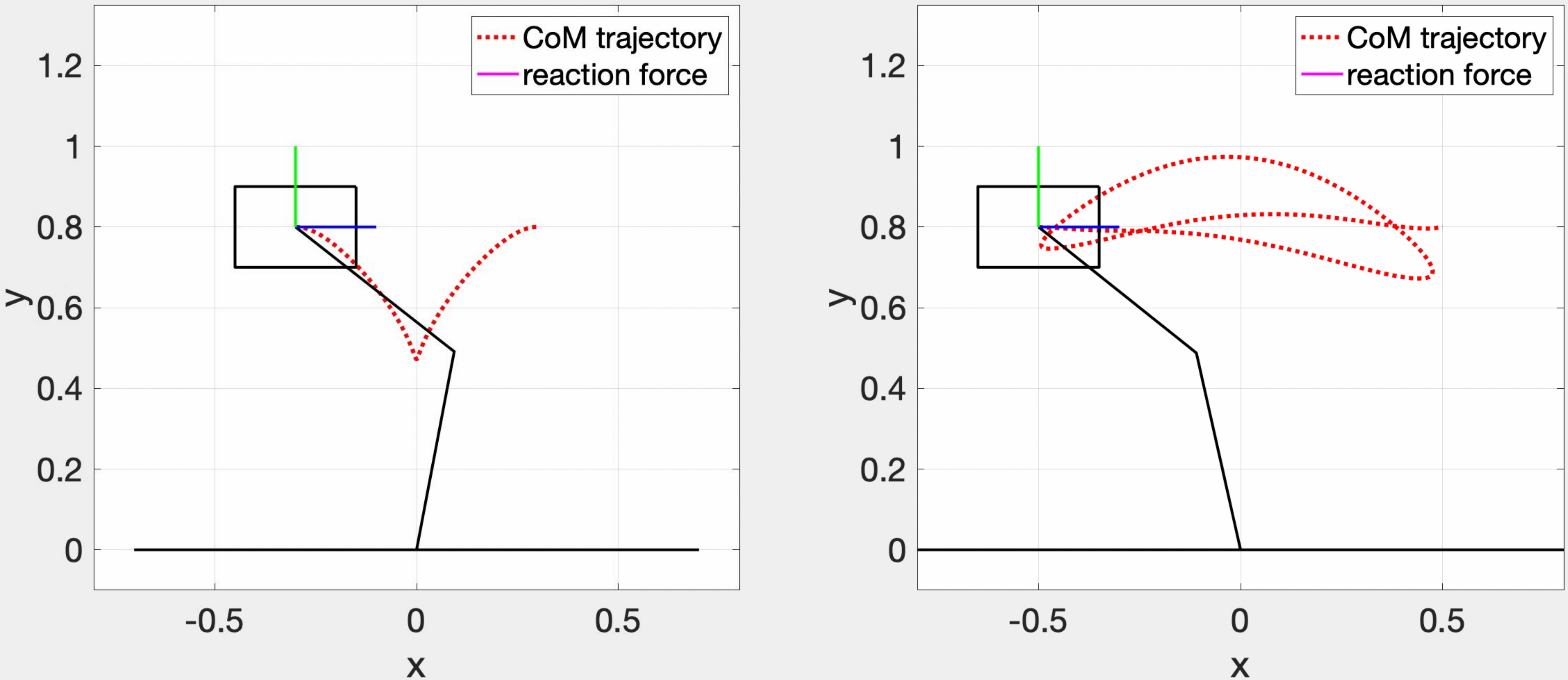
$$\dot{X} = \underbrace{\begin{bmatrix} X(4) \\ X(5) \\ X(6) \\ 0 \\ -g \\ 0 \end{bmatrix}}_{\substack{f(X) \\ \text{drift term}}} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ X(2) & -X(1) \end{bmatrix}}_{\substack{F_f(X) \\ \text{admissible direction}}} u$$

Systems with drift and positive constraints

Transfer body from (-.25,.8,0) to (.25,.8,0)

Transfer body from (-.5,.8,0) to (.5,.8,0)

Allowed time: 1.5 sec

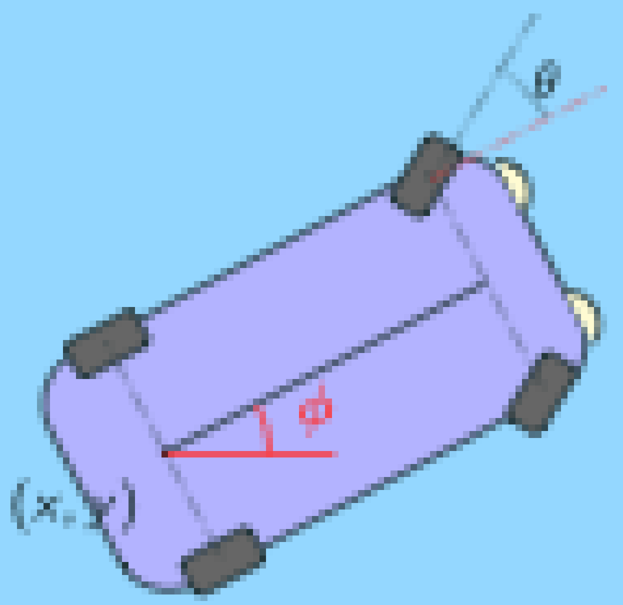


Outline of the presentation

1. Introduction and motivation
2. Presentation of the method
3. Complexity, drift and bounded controls
4. Case study: wheeled vehicles
5. Sketch of proof of convergence

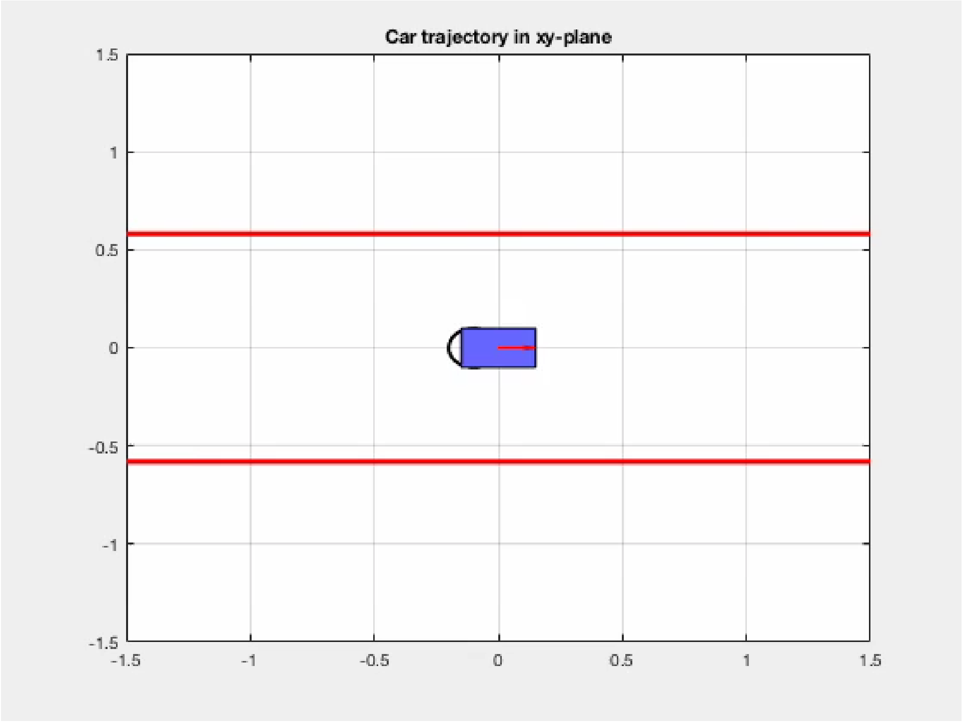
Case study: car maneuvers

Model

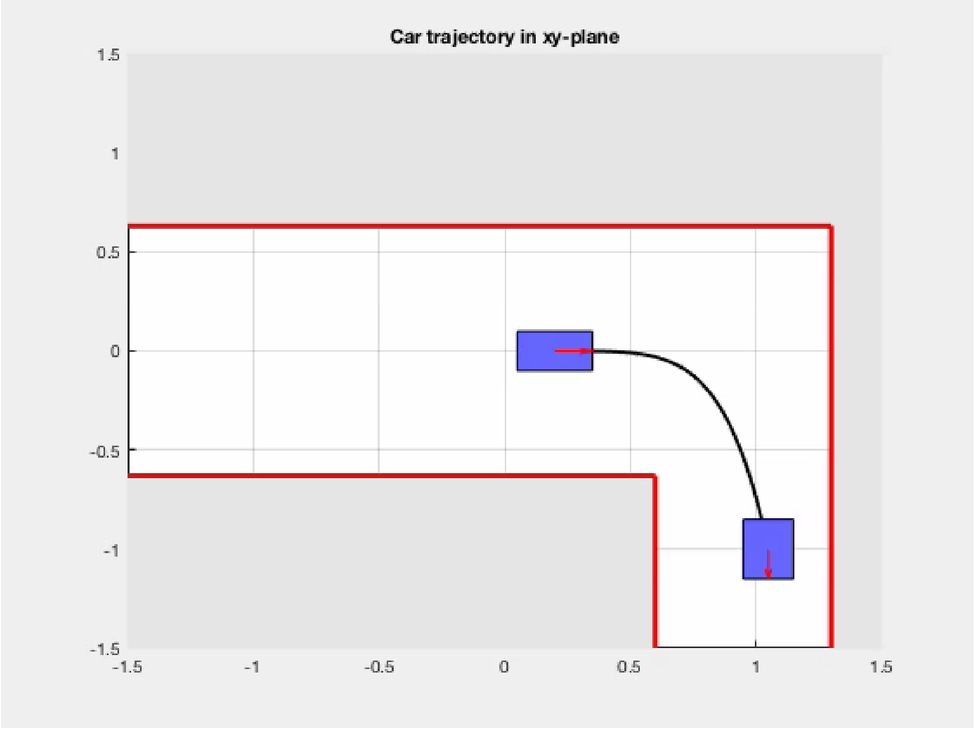


$$\frac{d}{dt}\begin{pmatrix} x \\ y \\ \theta \\ \phi \end{pmatrix} = u_1 \underbrace{\begin{pmatrix} \cos\phi \\ \sin\phi \\ 0 \\ \frac{1}{d}\sin\theta \end{pmatrix}}_{f_1} + u_2 \underbrace{\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}}_{f_2}.$$

Initial trajectories: 180 turn

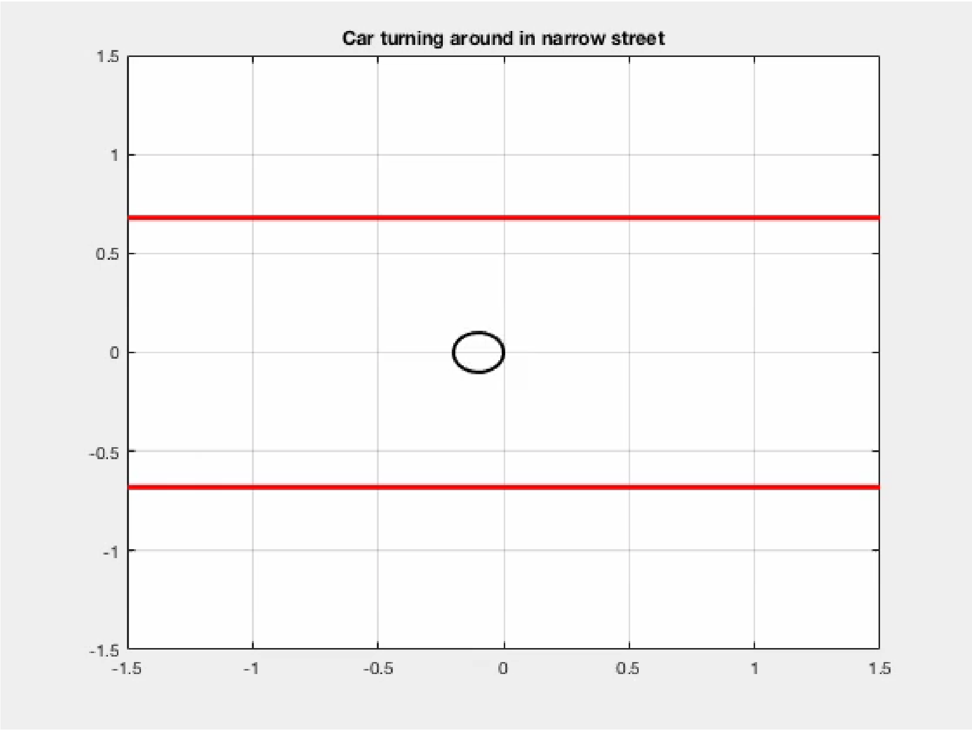


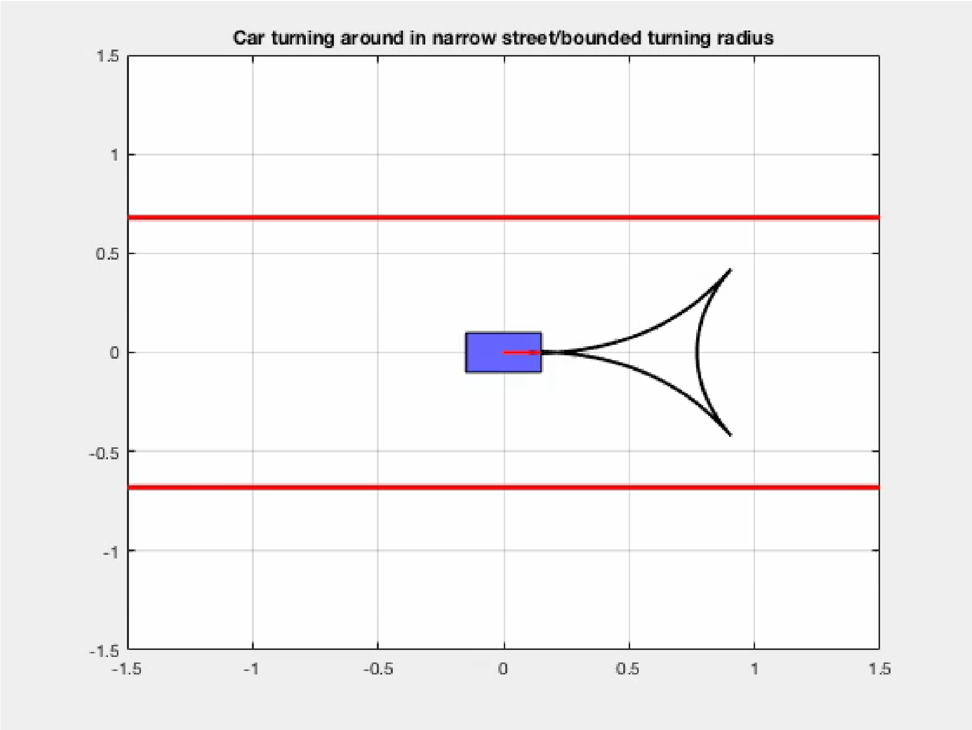
Initial trajectories: street turn



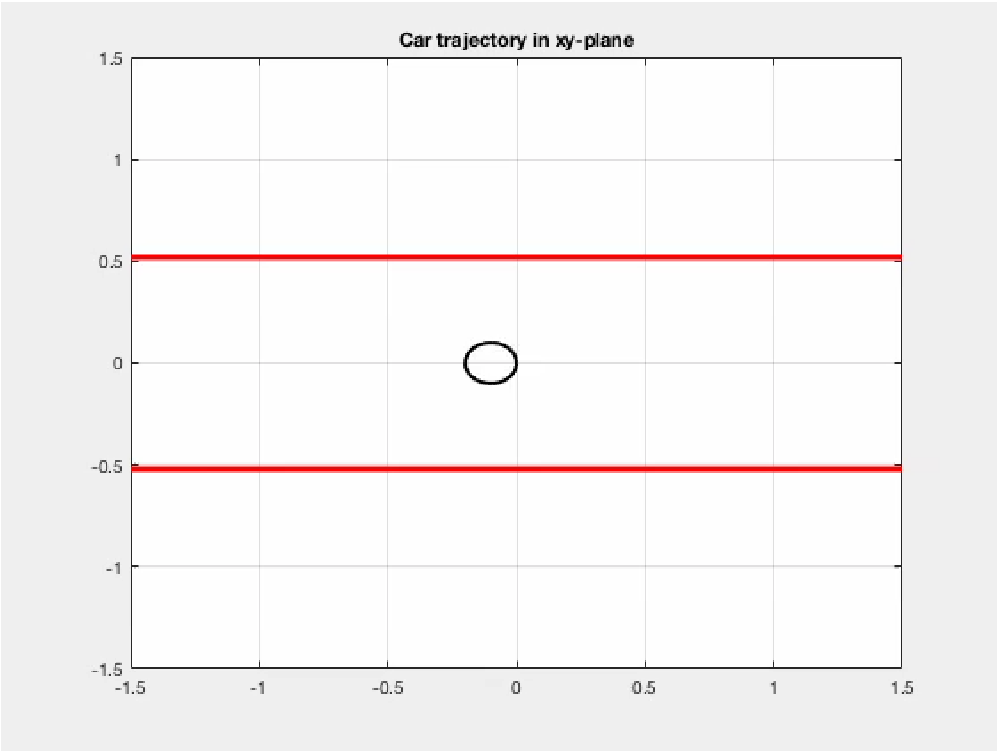
Car maneuvers: 180 turn

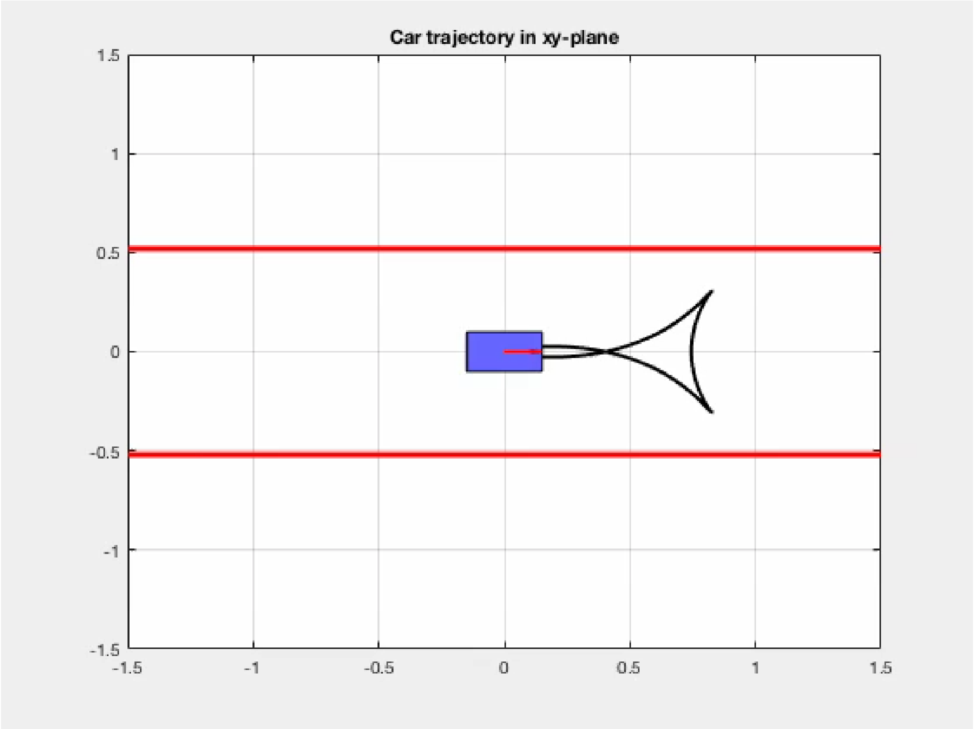
Wide street





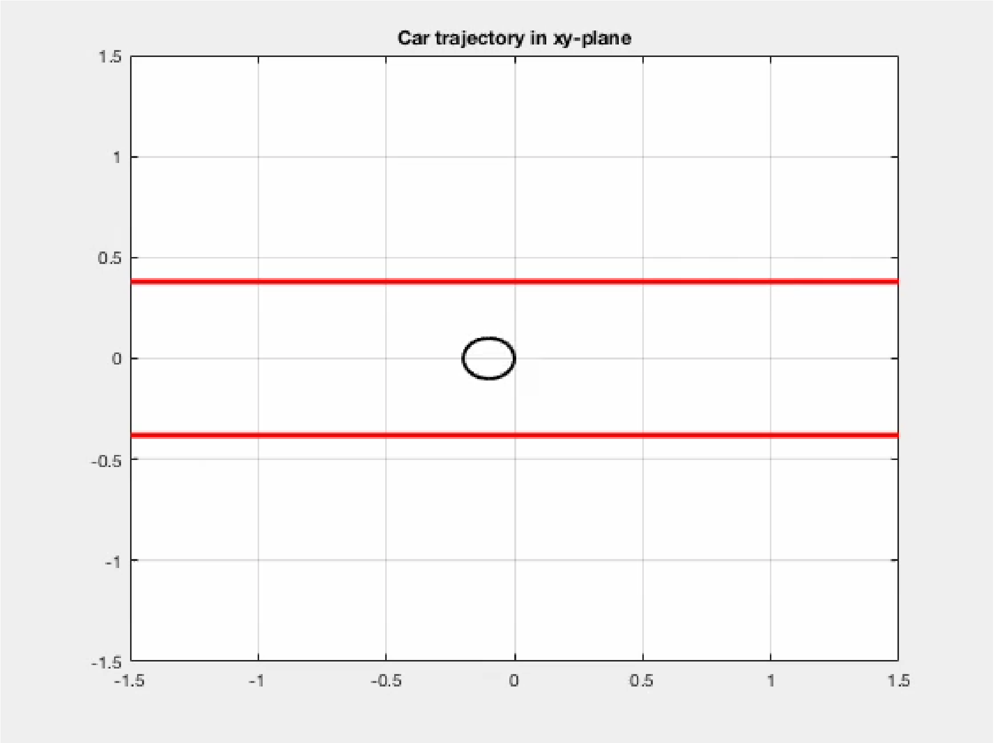
Narrow street

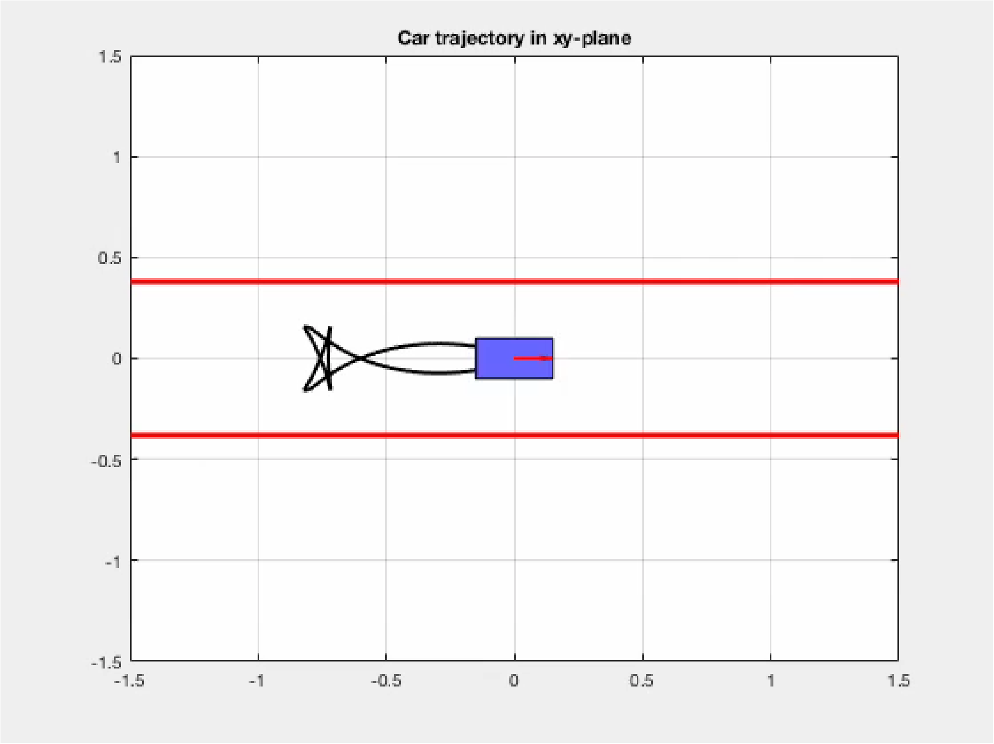




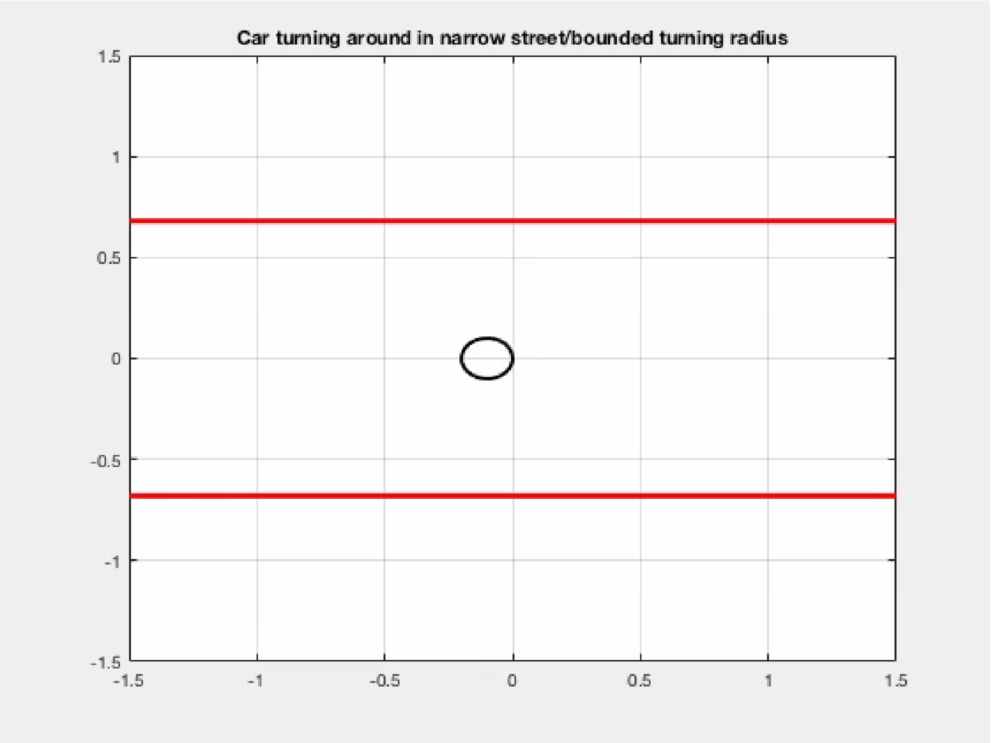
Car maneuvers: 180 turn

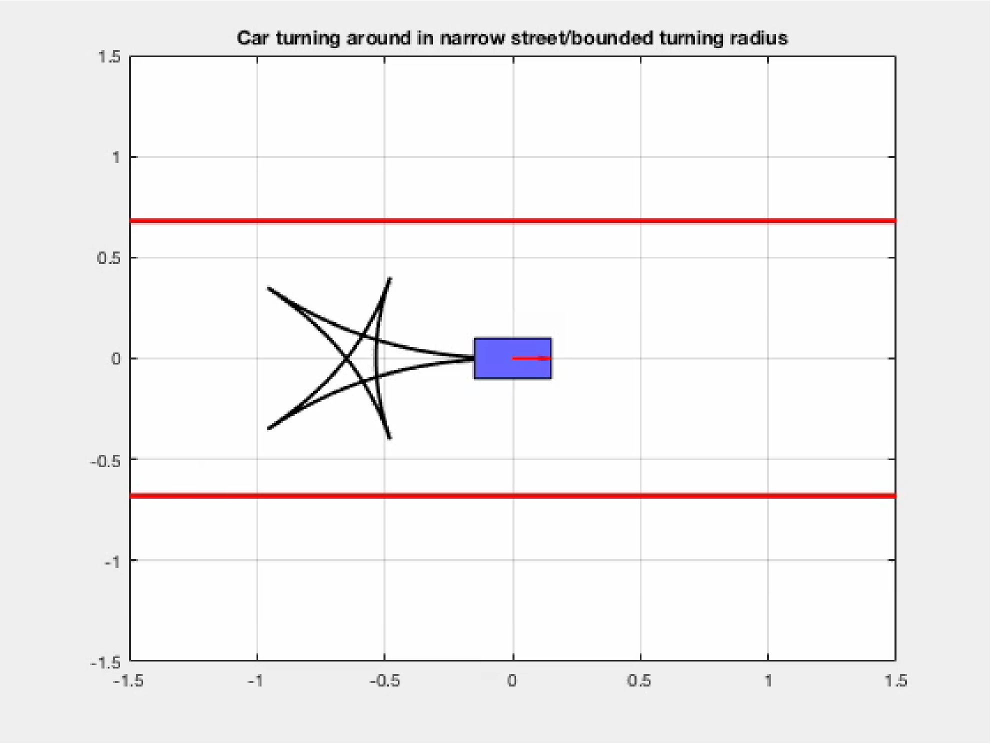
1. Narrower street





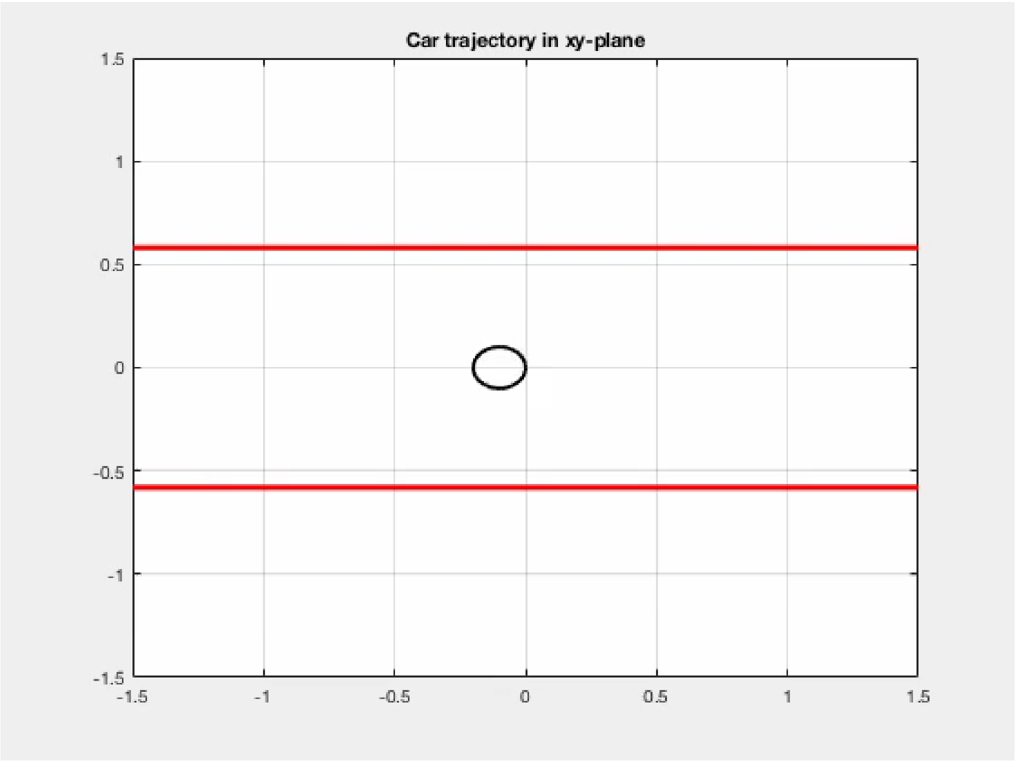
1. Bounding turning angle

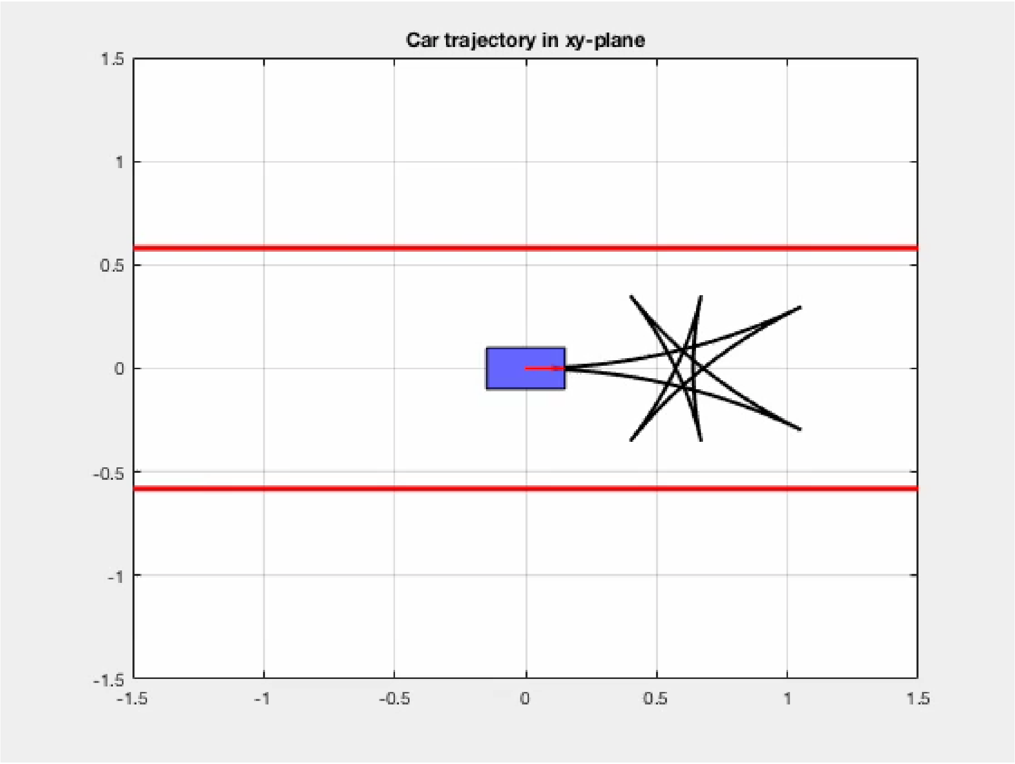




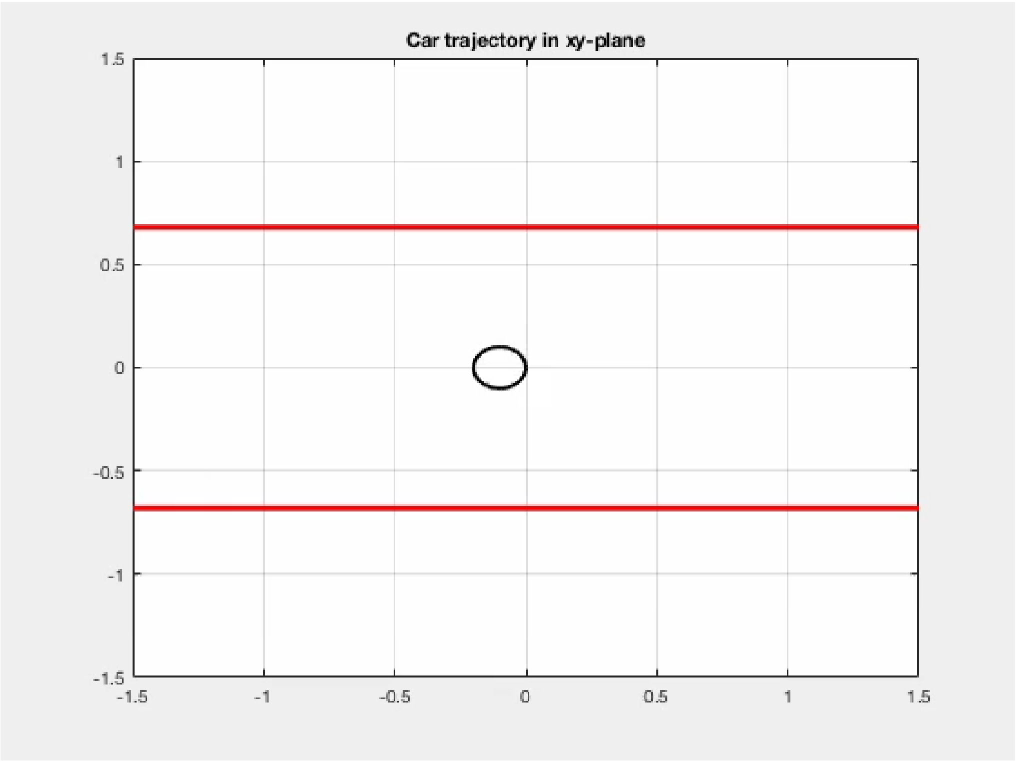
Car maneuvers: 180 turn

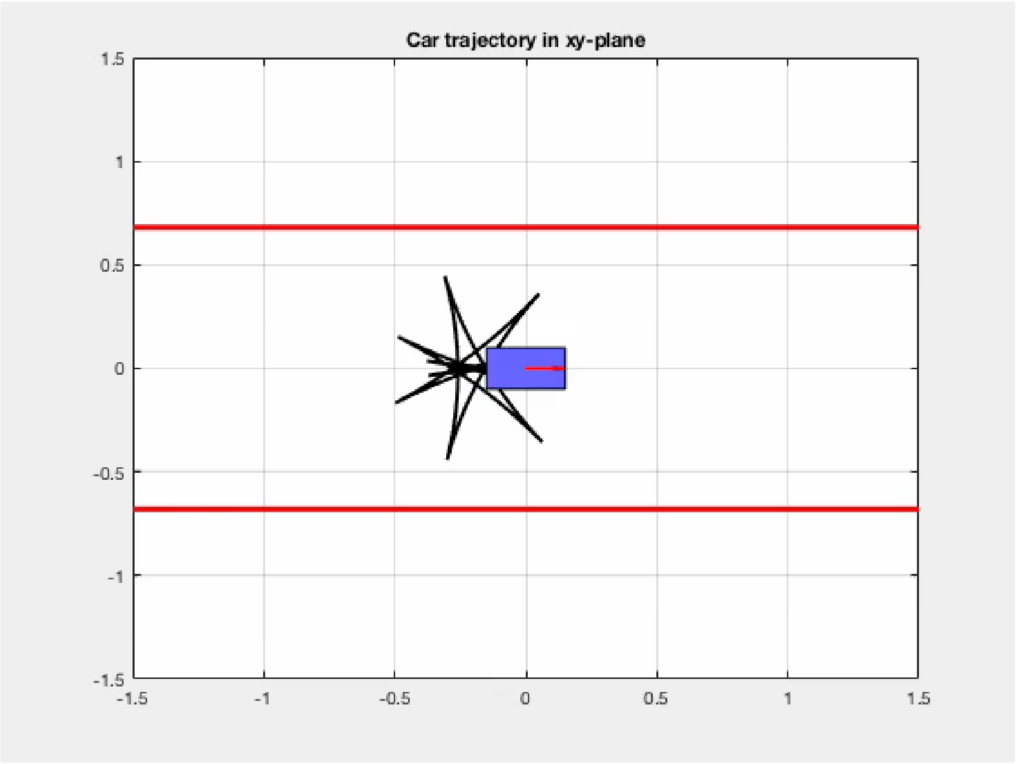
1. Smaller turning angle





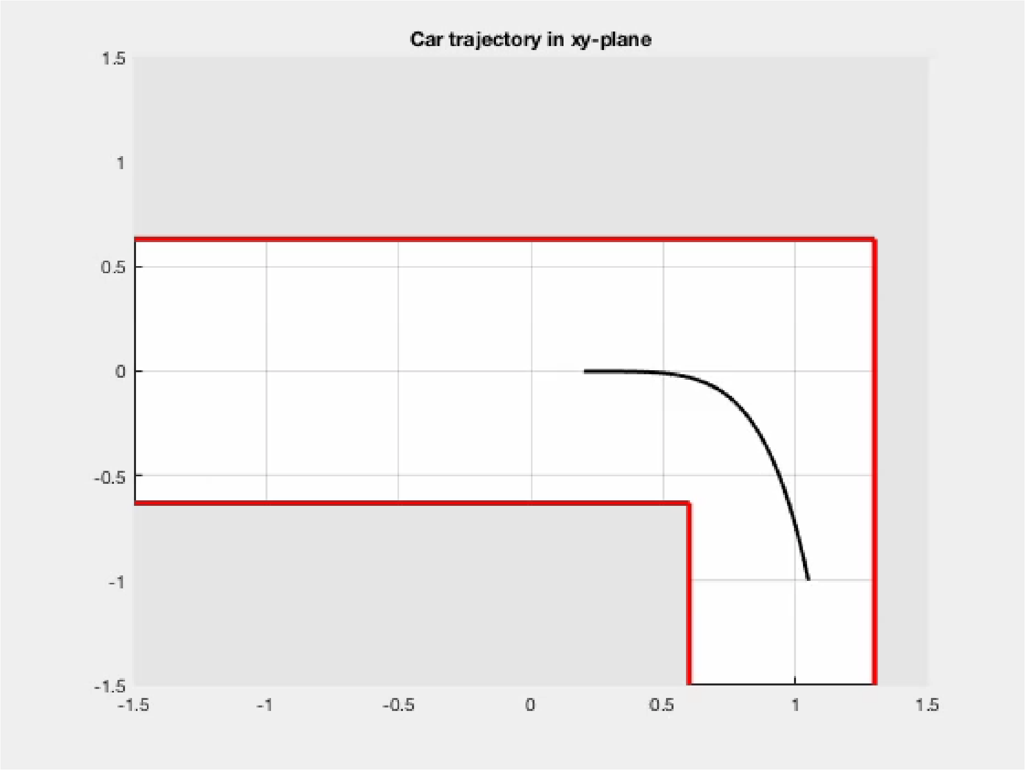
1. Even smaller turning angle

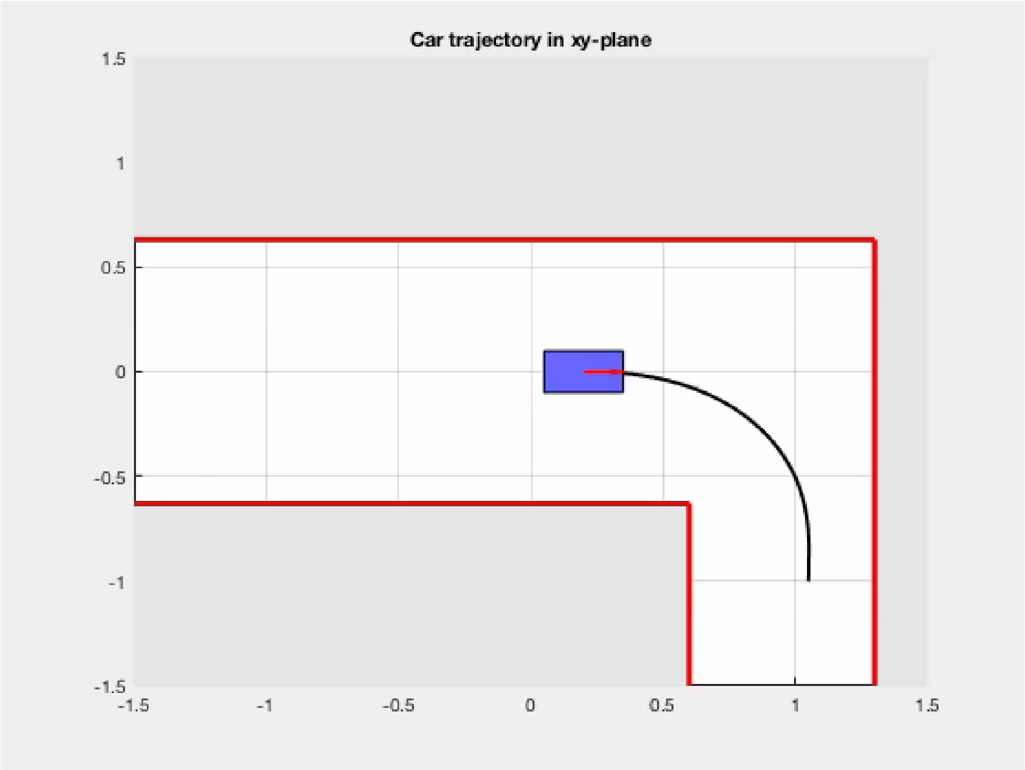




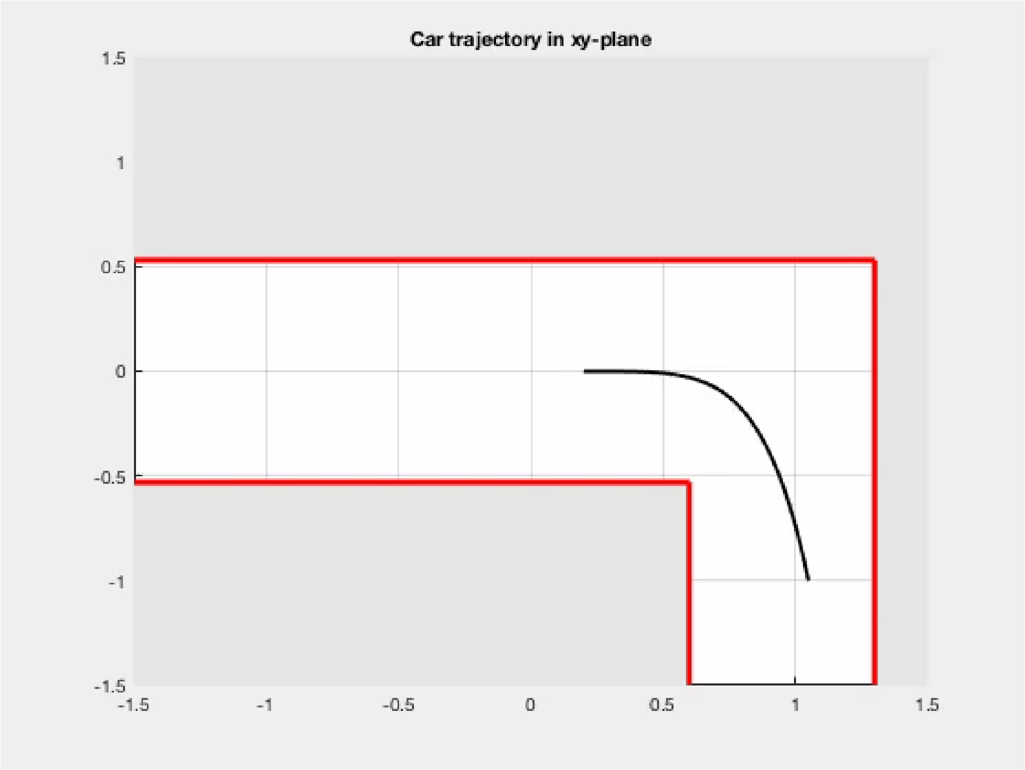
Car maneuvers: 180 turn

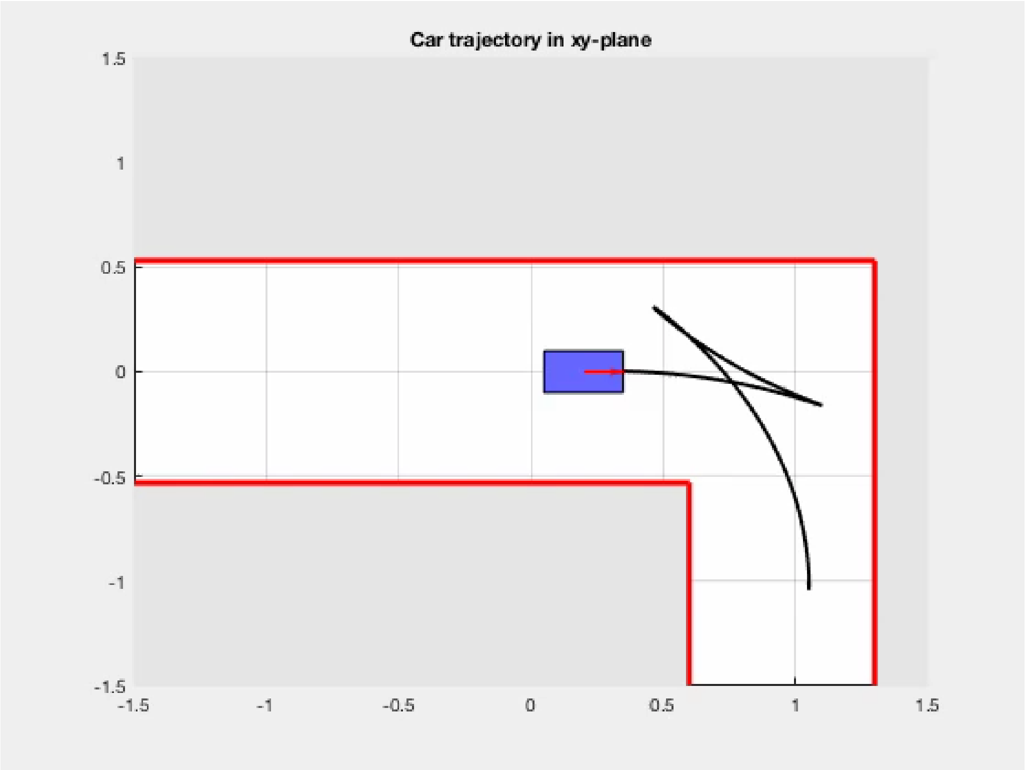
1. 90 degree turn





1. bounded turning angle





Car maneuvers: parking lot

This is obtained using fixed waypoints

Outline of the presentation

1. Introduction and motivation
2. Presentation of the method
3. Complexity, drift and bounded controls
4. Case study: wheeled vehicles
5. Sketch of proof of convergence

Theoretical guarantee

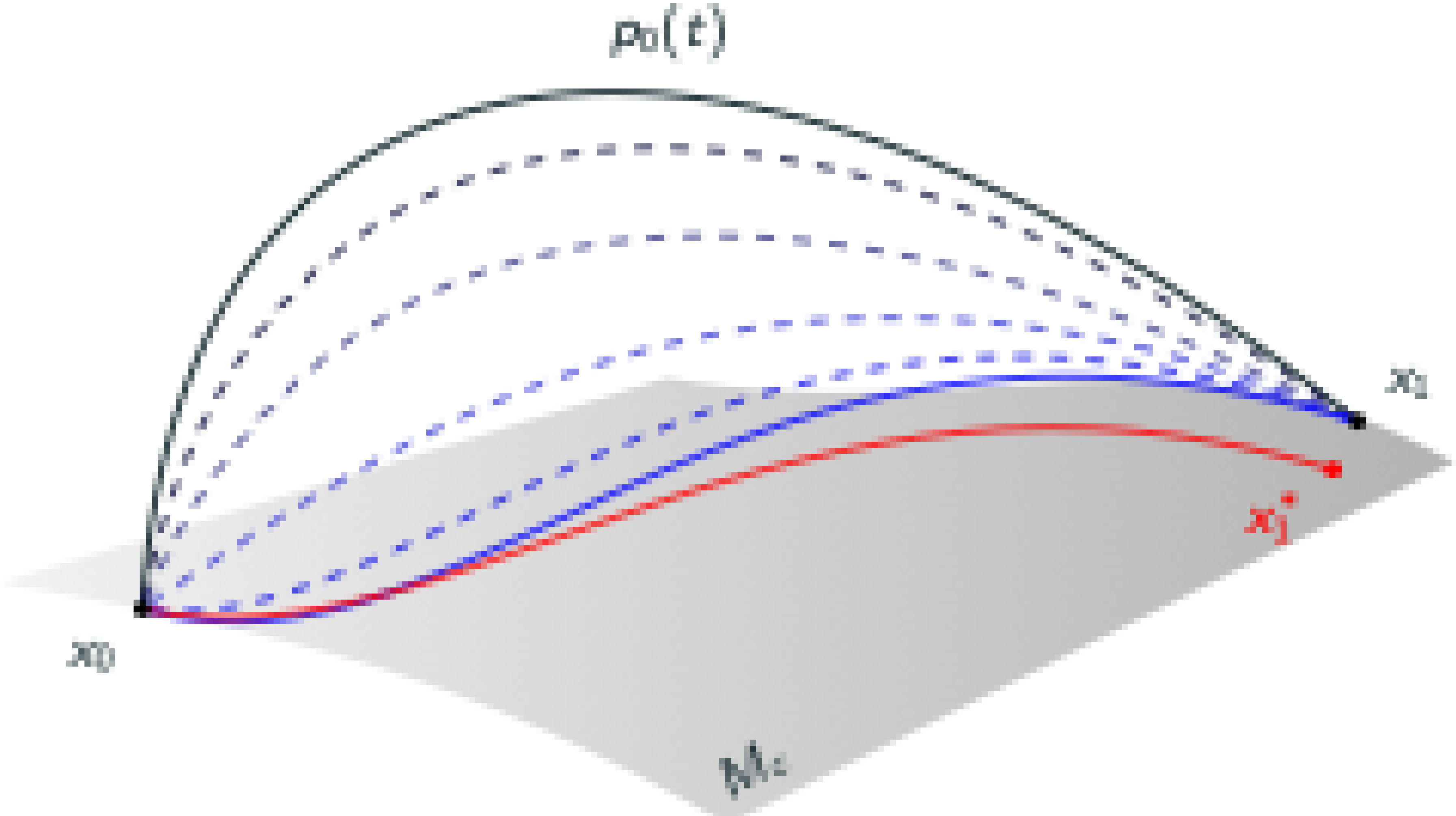
Set $\Delta = \text{span}\{f_{holon}\} \cap \text{span}\{f_{nonholon}\}$.
Definition: Constraints are satisfiable if the distribution Δ satisfies the LARC.

Theorem

Assume that the constraints are satisfiable and all functions/vector fields are smooth. Then our method provides controls $\bar{u}(t)$ so that the solution $x^*(t)$ of $\dot{x} = \sum_i \bar{u}_i f_i$ has the properties:

1. It satisfies the holonomic and non-holonomic constraints
2. For $\varepsilon > 0$, $\exists k > 0$ such that $||x^*(1) - x_1|| < \varepsilon$ and all obstacles are avoided.

Theoretical guarantee



$$p_0(t) \xrightarrow{\text{Eq. (3)}} v(\infty, t) \xrightarrow{\text{Extract controls}} x^*(t)$$

- ○ $p_0(t)$:init. curve
- ● $v(\infty, t)$:sol. of PDE
- ● $x^*(t)$:system traj.
- □ M_c:admissible traj.

The trajectory x^* is so that $\|x^*(1) - x_1\| \searrow 0$ as $k \nearrow \infty$.

Sketch of the proof

- Set F_f so that $\mathrm{span}\{F_f\} = \Delta$ and F_c to be the orthogonal complement of F_f.
- Set $p(t) := \lim_{s\to\infty} v(s,t)$: solution to the PDE (3).
- Define $z(t) := \mathrm{Proj}_\Delta\, \dot{p}(t)$.
- We can set $\bar{u}$ s.t. $z(t) = F_f(p)\bar{u}$.
- Because $\Delta \subseteq \mathrm{span}\{f_i\}$, there exists $\bar{u}(t)$ so that

$$z(t) = \sum_i \bar{u}_i f_i(p).$$

  $\rightarrow$ By construction, $x^*(t)$ meets both holonomic and non-holonomic constraints.
- We can express

$$\dot{p} = F_f(p)\bar{u} + F_c(p)u_c$$

Sketch of the proof

- Consider the energy of p:

$$
\begin{aligned}
E(p) &= \int_0^1 \dot{p}(t)^\top G(p(t))\dot{p}(t)dt \\
&= \int_0^1 |F_f(p)^\top \bar{u}(t)|^2 + k|F_c(p)^\top u_c(t)|^2 dt
\end{aligned}
$$

- By Chow-Rashevski, an admissible trajectory *ending* at x_1 exists. Denote it by $\bar{p}$. Notice that $E(\bar{p})$ is independent of k because $F_c^\top \dot{\bar{p}} = 0$
$\rightarrow E(\bar{p})$ is a finite constant.
- The solution of the PDE minimizes $E(\cdot)$, hence

$$
\int_0^1 |F_f(p)^\top \bar{u}(t)|^2 + k|F_c(p)^\top u_c(t)|^2 dt = E(p) \leq E(\bar{p})
$$

Sketch of the proof

- Without loss of generality with F_f, F_c are normalized, we have

$$\int_0^1 |\bar{u}(t)|^2 + k|u_c(t)|^2 dt \leq E(\bar{p}),$$

which leads to the two inequalities:

$$\int_0^1 |\bar{u}(t)|^2 dt \leq E(\bar{p}),$$

$$\int_0^1 |u_c(t)|^2 dt \leq \frac{E(\bar{p})}{k}$$

- Define the error $e(t) := p(t) - x^*(t)$. It satisfies

$$e(0) = 0, \dot{e}(t) = (F_f(p) - F_f(x^*))\bar{u}(t) + F_c(p)u_c(t)$$

Sketch of the proof

- We can show that F_f is Lipschitz with constant L in the domain of interest. Applying Cauchy-Schwartz inequality and Grönwall's lemma,

$$
\begin{aligned}
|e(t)|^2 &= \left| \int_0^t (F_f(p) - F_c(x^*))\bar{u}(\tau) + F_c(p)u_c(\tau)d\tau \right|^2 \\
&\leq 2t \int_0^t L^2|\bar{u}(\tau)|^2|e(\tau)|^2 d\tau + 2t \int_0^t |u_c(\tau)|^2 d\tau
\end{aligned}
$$

$$
\Rightarrow |e(t)|^2 \leq 2t \int_0^1 |u_c(\tau)|^2 d\tau \exp\left(2tL^2 \int_0^1 |\bar{u}(\tau)|^2 d\tau\right)
$$

- Plugin the two inequalities from previous slide,

$$
|e(t)| \leq \sqrt{\frac{2tE(\bar{p})}{k}} e^{tL^2E(\bar{p})}
$$

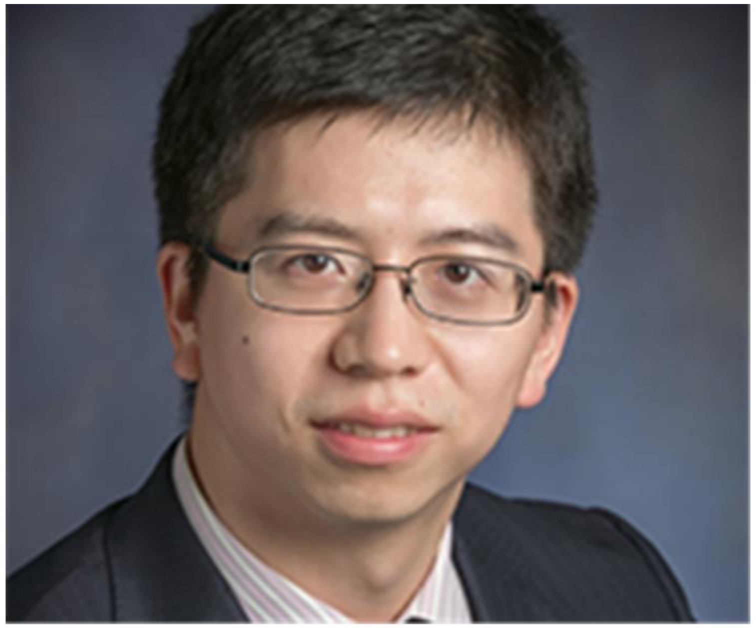
- In particular, $|x^*(1) - x_1| = |e(1)| \leq \sqrt{\frac{2E(\bar{p})}{k}} e^{L^2E(\bar{p})} \to 0$ as $k \to \infty$, meaning we get arbitrarily close to x_1 by taking k sufficiently large.

Summary and outlook

1. New method for motion planning for nonlinear systems
2. Handles holonomic/non-holonomic and obstacle constraints
3. Provides natural motions
4. Can plan motion for multi-vehicles systems
5. Theoretical guarantees of convergence are provided

1. Streamline encoding of constraints
2. Parallelize GHF solver
3. Investigate regularity properties of solutions
4. Ensemble control and harmonic surfaces

Yinai Fan



Shenyu Liu