OPTIMAL PLANNING STRATEGIES FOR MULTIPLE UAV MISSIONS

Venanzio Cichella

University of Iowa – Department of Mechanical Engineering

OUTLINE

- Introduction and general framework
- Optimal motion planning
- Coordinated tracking control
- Conclusions

OUTLINE

- Introduction and general framework
- Optimal motion planning
- Coordinated tracking control
- Conclusions

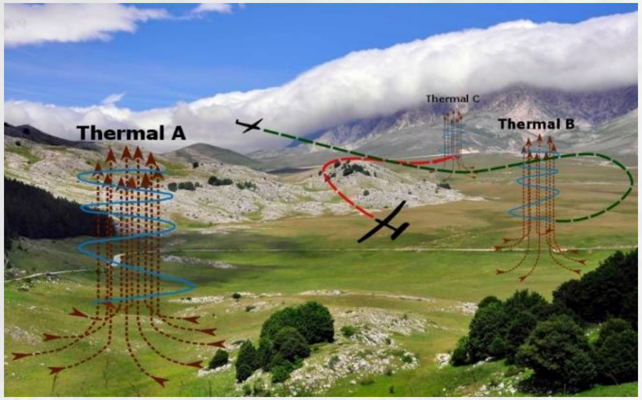
MOTIVATION

"In the long history of humankind (and animal kind, too) those who learned to collaborate and improvise most effectively have prevailed." - Charles Darwin (1809 – 1882)

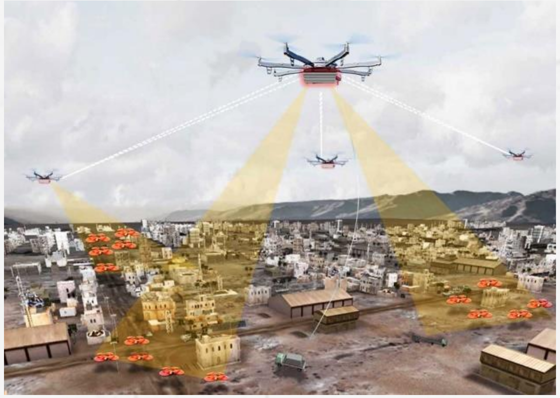
- ❑ Formation control
  - e.g. Murray et al. (2006), Egersted et al. (2001)
- ❑ Collective behavior/flocking
  - e.g. Jadbabaie et al. (2003), Shamma et al. (2007)
- ❑ Multi-agent differential games
  - e.g. Stipanovic et al. (2009), Astolfi et al. (2014)
- ❑ Multi-agent adaptive dynamic programming
  - e.g. Lewis et al. (2012)
- ❑ Coordination
  - e.g. Arcak et al. (2007)
- ❑ Optimal Control-Based Methods
  - e.g. How et al. (2011), D'Andrea et al. (2010), Beard et al. (2005)

A BROADER CLASS OF MISSIONS

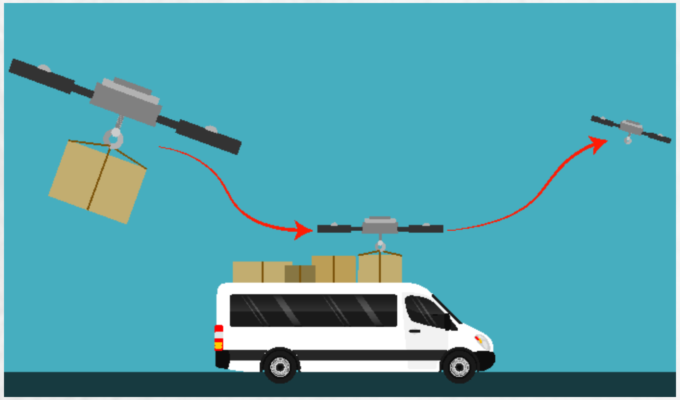
Steer a group of Unmanned Vehicle Systems (UxSs) along desired trajectories while meeting mission-specific requirements



Air sampling missions



Search and rescue missions



Autonomous delivery

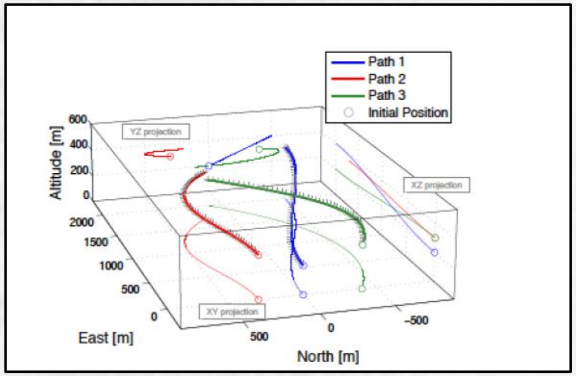
Entertainment

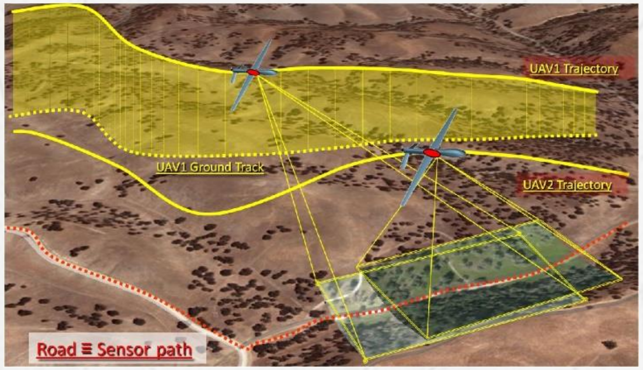
A REPRESENTATIVE EXAMPLE

Execute collision-free maneuvers and arrive at final destinations at the same time (or separated by pre-defined time intervals)

❑ **Time-critical** applications for multiple vehicles:

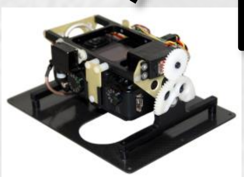
- Reaching formation
- Sequential auto-landing
- Coordinated road search

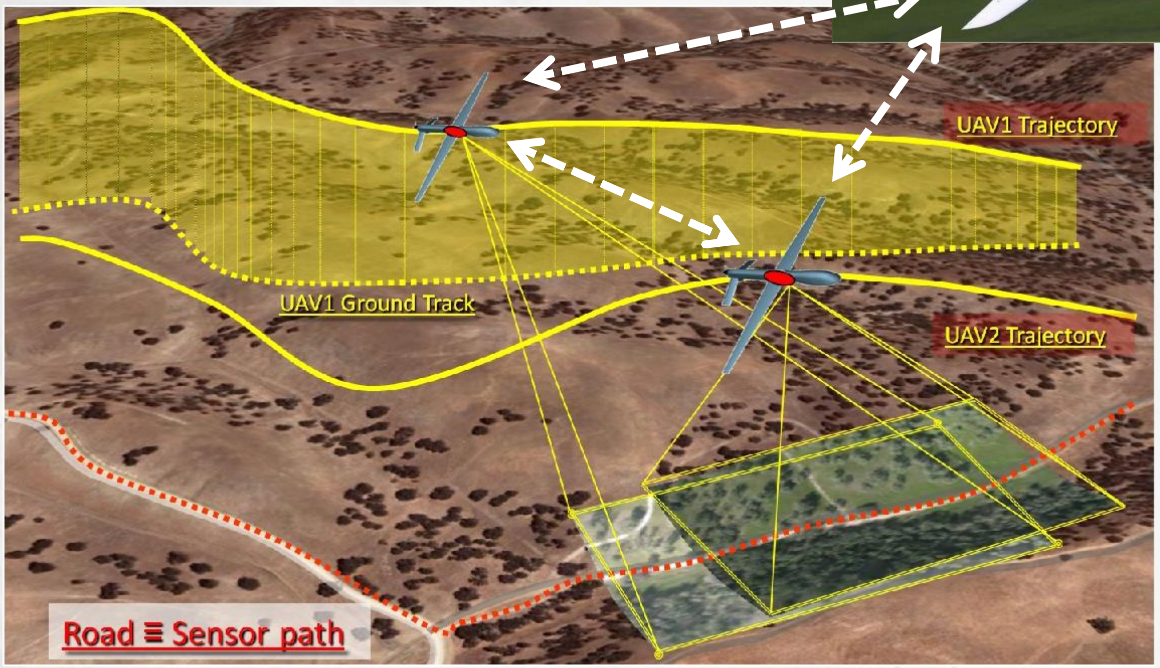


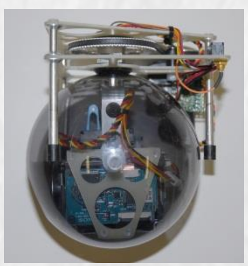


EXAMPLE: COOPERATIVE ROAD SEARCH

Single DOF gimbal with
high resolution camera
(satellite quality imagery)







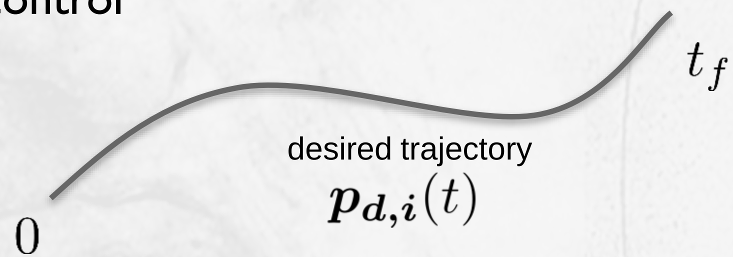
2 DOF pan/tilt gimbal with
video camera
(enabling vision-based guidance)

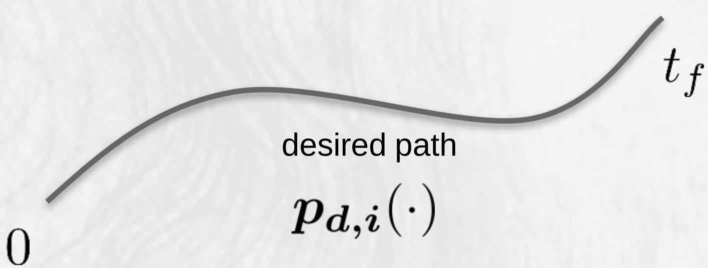
*Thermal seeking **soaring gliders***
are used as flying antennas to
extend communication range

DECOUPLING SPACE AND TIME

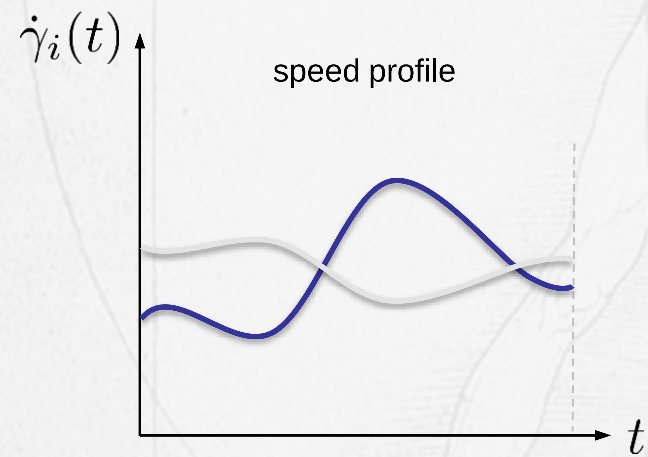
- Optimal Motion planning

- Coordinated tracking control
  - Virtual target tracking
  - Coordination control





decoupling

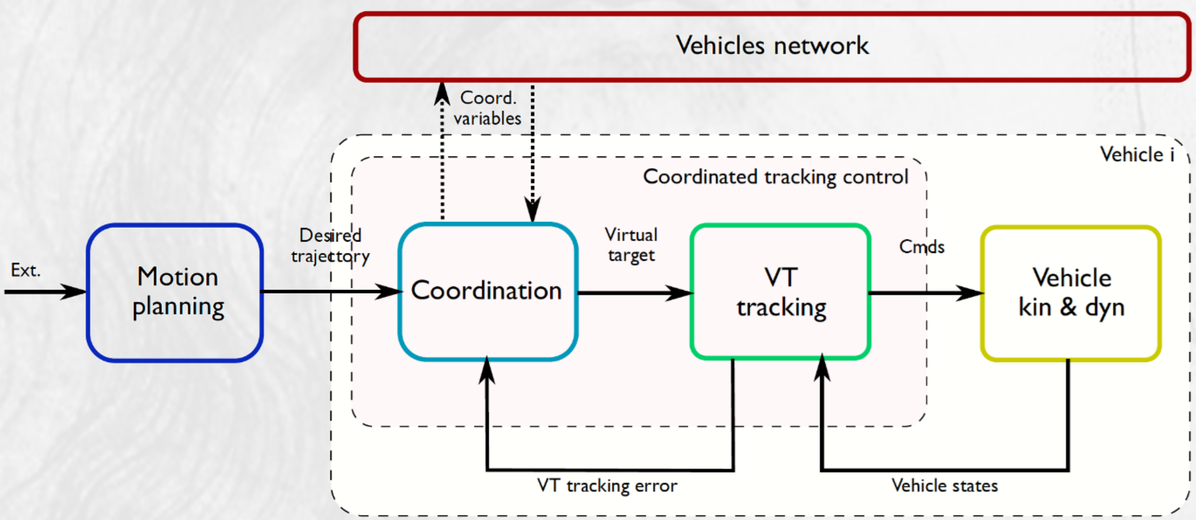


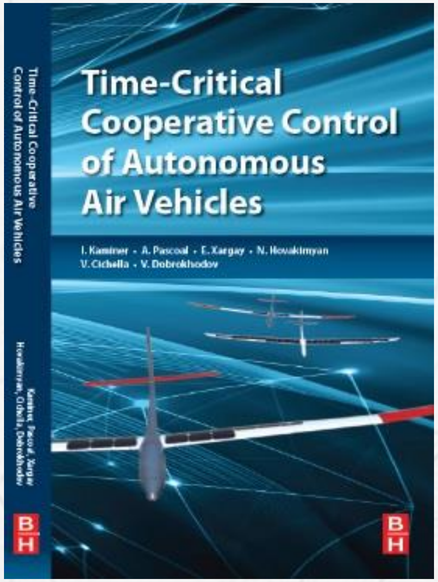
virtual target to be tracked: $\boldsymbol{p_{d,i}}(\gamma_i(t))$

MULTI-LOOP ARCHITECTURE

- Optimal Motion planning
  - Efficient and safe (guaranteed satisfaction of constraints)

- Coordinated tracking control
  - Virtual target tracking
    - Vehicle’s performance limitation
  - Coordination control
    - Communication network (drop-outs, switching topologies, …)





OUTLINE

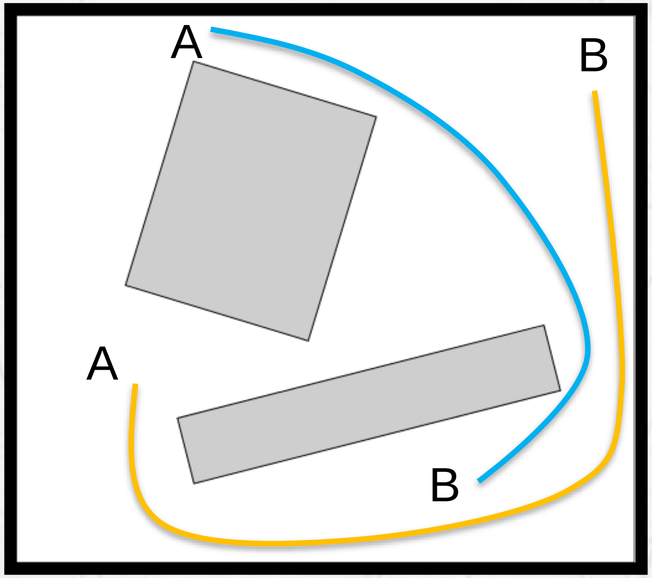
- Introduction and general framework
- Optimal motion planning
- Coordinated tracking control
- Conclusions

OPTIMAL MOTION PLANNING

OCP: determine $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$

that minimize $$E(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) + \int_0^{t_f} F(\boldsymbol{x}(t), \boldsymbol{u}(t))dt$$

subject to
$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \forall t \in [0, t_f]$$
$$\boldsymbol{e}(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) = \boldsymbol{0}$$
$$\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \leq \boldsymbol{0}, \quad \forall t \in [0, t_f]$$



OPTIMAL MOTION PLANNING

OCP: determine $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$

that minimize $$E(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) + \int_0^{t_f} F(\boldsymbol{x}(t), \boldsymbol{u}(t))dt$$

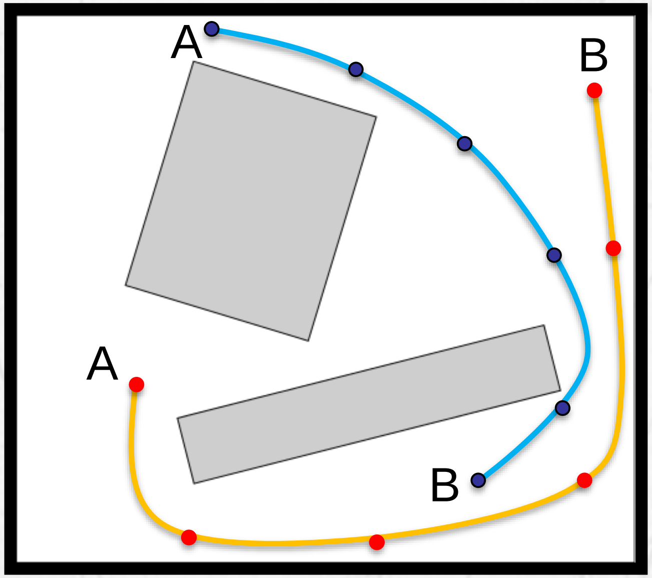
subject to
$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))\,, \quad \forall t \in [0, t_f]$$
$$\boldsymbol{e}(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) = \boldsymbol{0}$$
$$\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \leq \boldsymbol{0}\,, \quad \forall t \in [0, t_f]$$

$$\bar{\boldsymbol{t}} = [t_0, \ldots, t_N] \qquad \bar{\boldsymbol{x}} = [\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N] \qquad \bar{\boldsymbol{u}} = [\boldsymbol{u}_0, \ldots, \boldsymbol{u}_N]$$

NLP: determine $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{u}}$

that minimize $$E(\boldsymbol{x}_0, \boldsymbol{x}_N) + \sum_{j=0}^{N} w_j F(\boldsymbol{x}_j, \boldsymbol{u}_j)$$

subject to
$$\left\| \sum_{j=0}^{N} D_{ij}\boldsymbol{x}_j - \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{u}_i) \right\| \leq N^{-\delta}$$
$$\|\boldsymbol{e}(\boldsymbol{x}_0, \boldsymbol{x}_N)\| \leq N^{-\delta}$$
$$\boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{u}_i) \leq \boldsymbol{1}N^{-\delta}$$



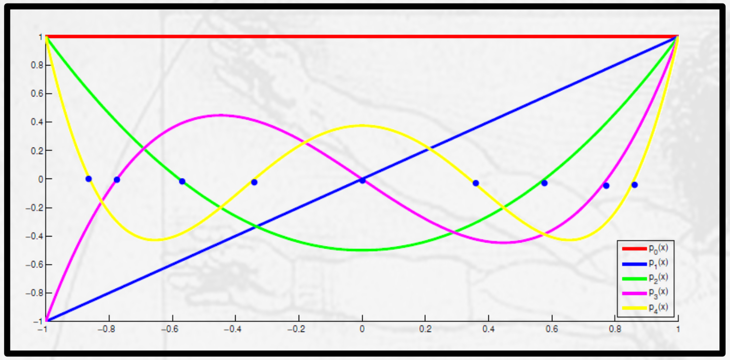
Approximate - Solve - Interpolate

LGL PSEUDOSPECTRAL

- Legendre-Gauss-Lobatto (LGL) nodes:

$$t_0 = -1\,,\ t_N = 1\,,\ \text{and } t_i \text{ are roots of } q(t) = \dot{L}_N$$

$$L_N(x) = \frac{1}{2^N N!}\frac{d^N}{dx^N}[(x^2-1)^N]$$



- Lagrange interpolation:

$$\boldsymbol{x}(t) \approx \boldsymbol{x}_N(t) = \sum_{k=0}^{N} \boldsymbol{x}(t_k)\ell_k(t) \qquad \ell_i = \prod_{k=0,k\neq i}^{N} \frac{t-t_k}{t_i-t_k}$$

- Differentiation:

$$\dot{\boldsymbol{x}}(t_k) \approx \dot{\boldsymbol{x}}_N(t_k) = \sum_{i=0}^{N} \boldsymbol{x}(t_i)\boldsymbol{D}_{ki} \qquad \boldsymbol{D} = \begin{bmatrix} \dot{\ell}_0(t_0) & \dot{\ell}_1(t_0) & \cdots & \dot{\ell}_N(t_0) \\ \dot{\ell}_0(t_1) & \dot{\ell}_1(t_1) & \cdots & \dot{\ell}_N(t_1) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{\ell}_0(t_N) & \dot{\ell}_1(t_N) & \cdots & \dot{\ell}_N(t_N) \end{bmatrix}$$

- Gaussian quadrature:

$$\int_{-1}^{1} \boldsymbol{x}(t)dt \approx \sum_{i=0}^{N} w_i \boldsymbol{x}(t_i) \qquad w_0 = w_N = \frac{2}{N(N+1)}\,, \quad w_i = \frac{2}{N(N+1)[L_N(t_i)]^2}$$

I. M. Ross and M. Karpenko, *A review of pseudospectral optimal control: from theory to flight*, Annual Reviews in Control, 36 (2012), pp. 182–197. 5, 6

LGL PSEUDOSPECTRAL

❑ Advantages

- Lagrange interpolation at Legendre nodes is robust – *for sufficiently smooth solutions*
- Consistency analysis [Polak, 1997]
  - NLP is feasible
  - Solutions to NLP converge to solutions to OCP
  - The proof heavily relies on orthogonal collocation property of Lagrange interpolants

$$\boldsymbol{x}(t) \approx \boldsymbol{x}_N(t) = \sum_{k=0}^{N} \boldsymbol{x}(t_k)\ell_k(t) \qquad \ell_i(t_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \implies \boldsymbol{x}_N(t_i) = \boldsymbol{x}(t_i)$$

- High rate of convergence

$$\boldsymbol{x} \in W^{m,\infty} \implies ||\boldsymbol{x}_N(t) - \boldsymbol{x}(t)||_{L^2} \leq \frac{C}{N^m}$$

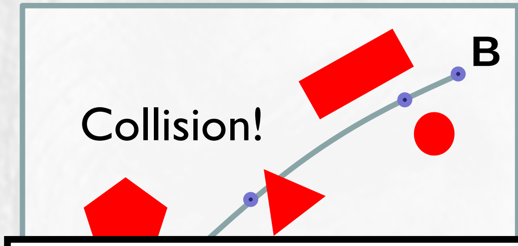
❑ Main disadvantage

- Constraints can be imposed only at the nodes
  - Efficient VS Safe

E. Polak, “Optimization: Algorithms and consistent approximations,” 1997, Springer Verlage Publications.

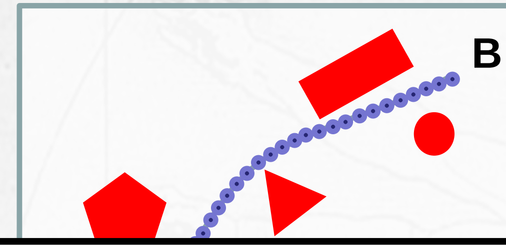
EFFICIENCY vs CONSTRAINTS SATISFACTION

Efficient - unsafe



Example

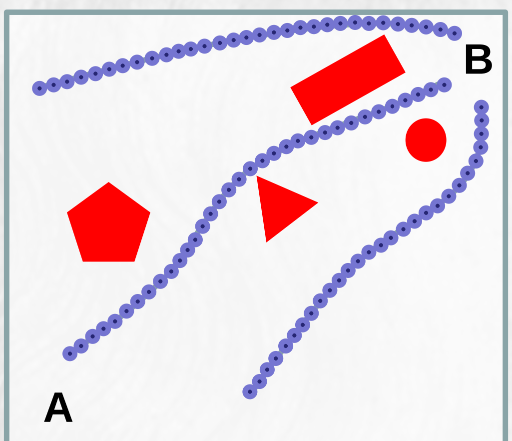
Inefficient - safe



We seek a class of polynomials with geometric properties that can be exploited in satisfying the set of imposed constraints:

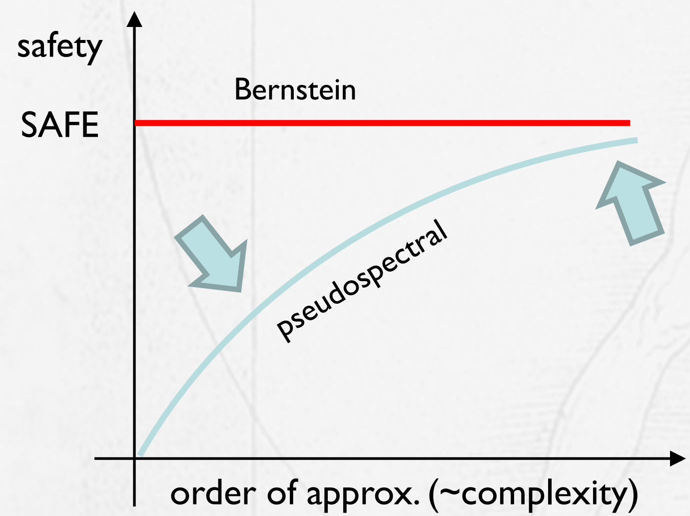
Bernstein polynomials

Unfeasible



$N = 50$

$\#\text{Constraints} = 8100$

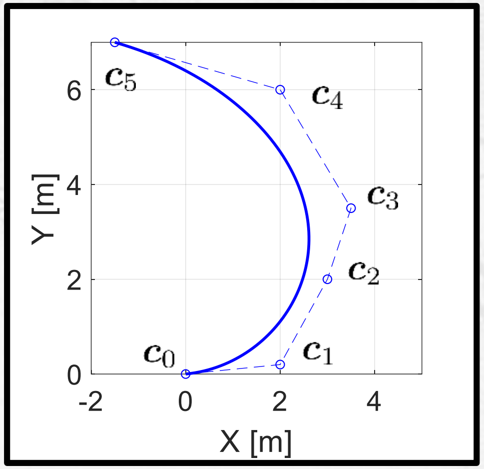


n^2*order of approx. (~complexity)

BERNSTEIN POLYNOMIALS

A degree *n* Bernstein polynomial is given by

$$\boldsymbol{x}_N(t) = \sum_{k=0}^{N} \boldsymbol{c}_k \, b_{k,N}(t)$$



where

- $b_{k,N}(t)$ are the Bernstein polynomial basis

$$b_{k,N} = \binom{N}{k} t^N (t_f - t)^{N-k}, \quad t \in [0, t_f]$$

- $\mathbf{c}_k \in \mathbb{R}^3$ are the *Bernstein coefficients*


Sergei Bernstein (1880-1968)

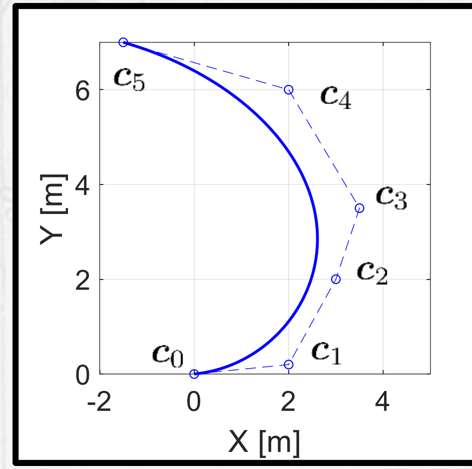

Paul de Casteljau (1930)


Pierre Bézier (1910-1999)

BERNSTEIN POLYNOMIAL APPROXIMATION

A degree *N* Bernstein polynomial is given by

$$\boldsymbol{x}_N(t) = \sum_{k=0}^{N} \boldsymbol{c}_k \, b_{k,N}(t)$$



- Bernstein approximation

$$t_j = 0, \quad j = 0, \ldots, N, \qquad \boldsymbol{c}_j = \boldsymbol{x}(t_j)$$

$$\boldsymbol{x}(t) \approx \boldsymbol{x}_N(t) = \sum_{j=0}^{N} \boldsymbol{c}_j b_{j,N}(t)$$

- Differentiation

$$\dot{\boldsymbol{x}}(t) \approx \dot{\boldsymbol{x}}_N(t) = \sum_{j=0}^{N-1} \left(\sum_{i=0}^{N} \boldsymbol{c}_i \boldsymbol{D}_{ij} \right) b_{j,N}(t)$$

$$\boldsymbol{D} = \begin{bmatrix} -\frac{N}{t_f} & 0 & \cdots & 0 \\ \frac{N}{t_f} & \ddots & \cdots & \vdots \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -\frac{N}{t_f} \\ 0 & \cdots & \cdots & \frac{N}{t_f} \end{bmatrix}$$

- Quadrature

$$\int_0^{t_f} \boldsymbol{x}(t) dt \approx \sum_{i=0}^{N} w_i \boldsymbol{x}(t_i) \qquad w_i = \frac{t_f}{N+1}$$

OPTIMAL MOTION PLANNING

OCP: determine $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$

that minimize
$$E(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) + \int_0^{t_f} F(\boldsymbol{x}(t), \boldsymbol{u}(t))dt$$

subject to
$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \forall t \in [0, t_f]$$
$$\boldsymbol{e}(\boldsymbol{x}(0), \boldsymbol{x}(t_f)) = \boldsymbol{0}$$
$$\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \leq \boldsymbol{0}, \quad \forall t \in [0, t_f]$$

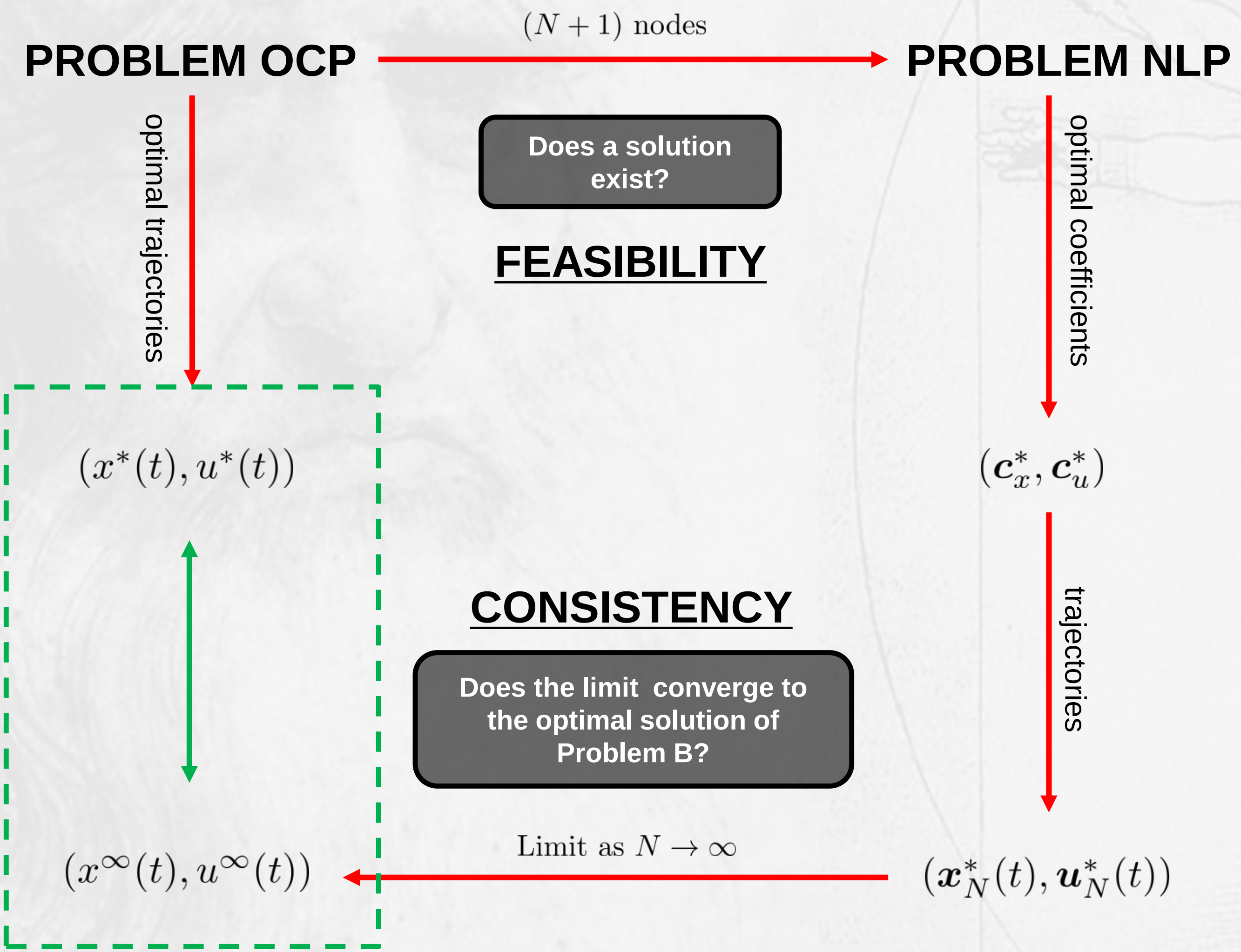
$$\boldsymbol{x}_N(t) = \sum_{k=0}^{N} \boldsymbol{c}_{x,k}\, b_{k,N}(t) \qquad \boldsymbol{u}_N(t) = \sum_{k=0}^{N} \boldsymbol{c}_{u,k}\, b_{k,N}(t)$$

NLP: Let $0 < \delta_P < 1$. Determine $\boldsymbol{c}_{x,k}$ and $\boldsymbol{c}_{u,k}$, $k = 0, \ldots, N$

that minimize
$$E(\boldsymbol{x}_N(0), \boldsymbol{x}_N(t_N)) + w \sum_{j=0}^{N} F(\boldsymbol{x}_N(t_j), \boldsymbol{u}_N(t_j))$$

subject to
$$\|\dot{\boldsymbol{x}}_N(t_j) - \boldsymbol{f}(\boldsymbol{x}_N(t_j), \boldsymbol{u}_N(t_j))\| \leq N^{-\delta_P} \quad \forall j = 0, \ldots, N$$
$$\boldsymbol{e}(\boldsymbol{x}_N(0), \boldsymbol{x}_N(t_N)) = \boldsymbol{0}$$
$$\boldsymbol{h}(\boldsymbol{x}_N(t_j), \boldsymbol{u}_N(t_j)) \leq N^{-\delta_P}\boldsymbol{1}, \quad \forall j = 0, \ldots, N$$

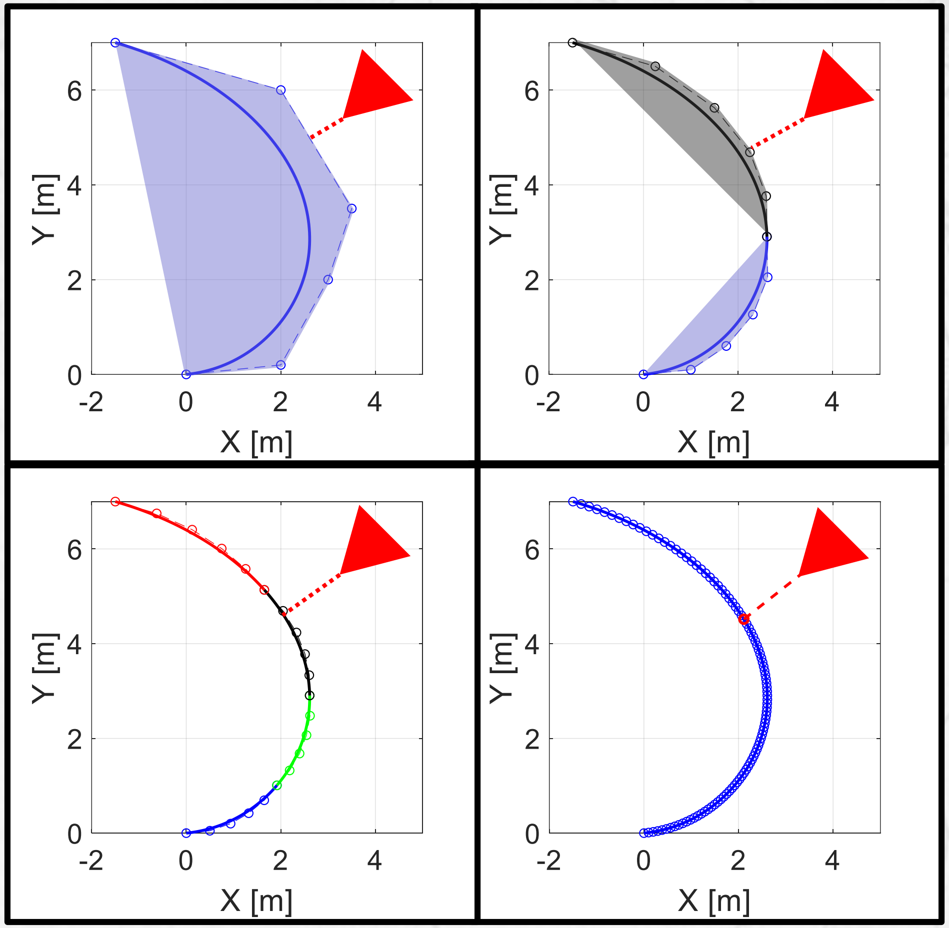
MAIN RESULT



E. Polak, "Optimization: Algorithms and consistent approximations," 1997, Springer Verlage Publications.

MINIMUM DISTANCE COMPUTATION

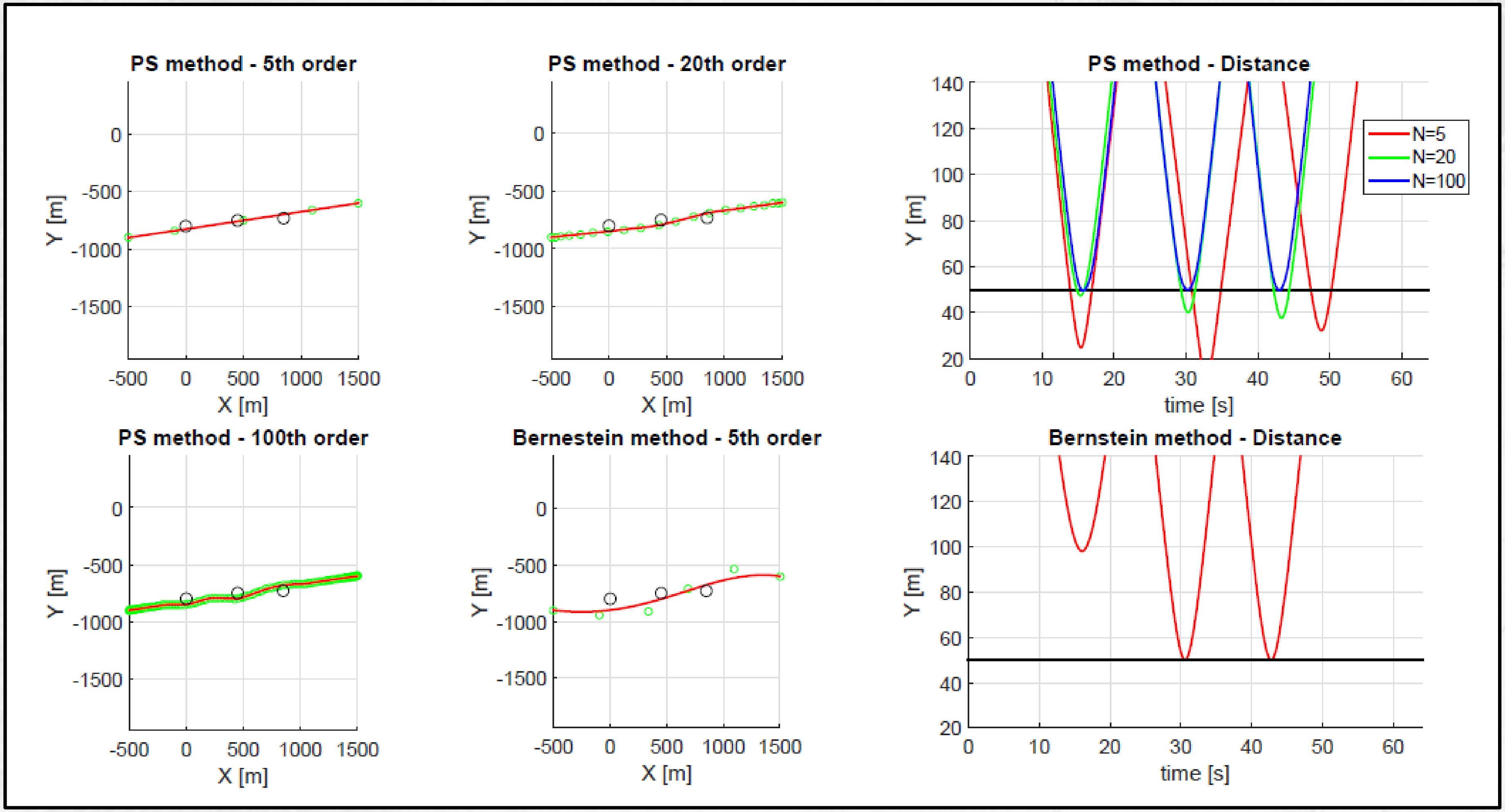
- Convex hull
  - A Bernstein polynomial is contained within the convex hull defined by its Bernstein coefficients
  - GJK algorithm computes distance between convex hulls (curve and obstacle)



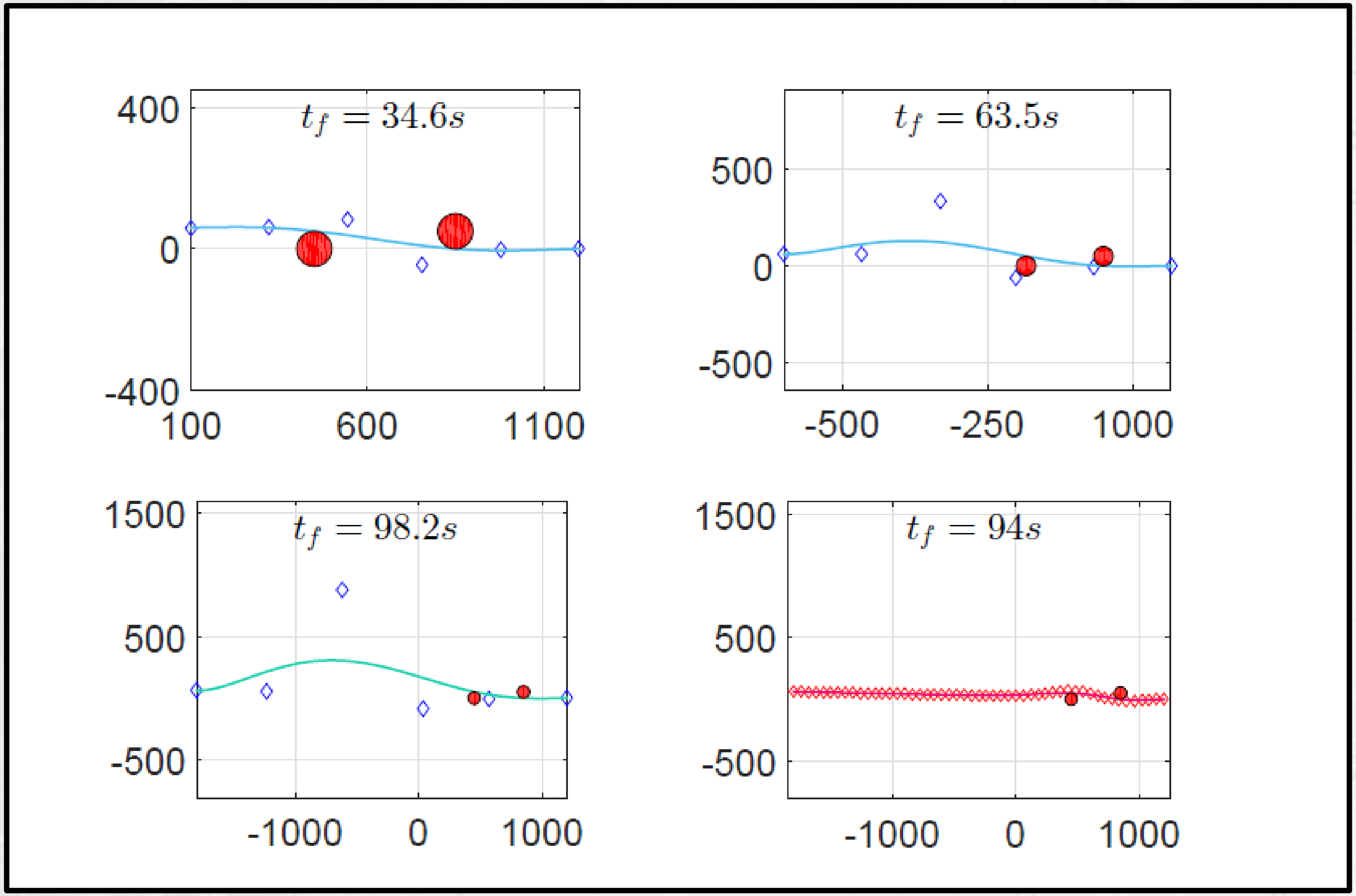
- de Casteljau algorithm
  - Subdivides Bernstein polynomials in multiple Bernstein polynomials

- Distance between 2 curves, min/max velocity, acceleration, etc.

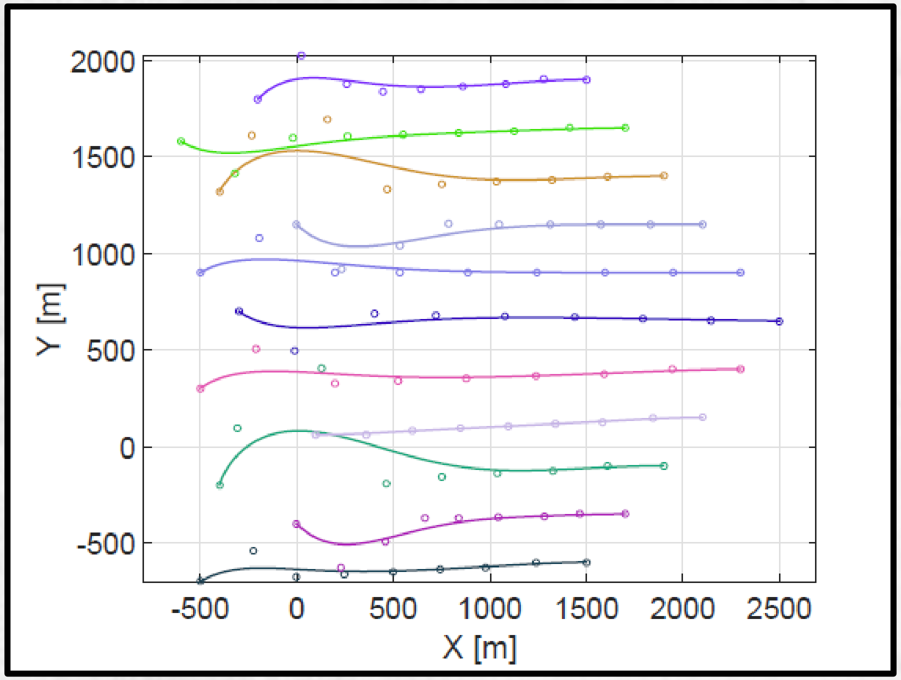
RESULTS: PS vs BERNSTEIN

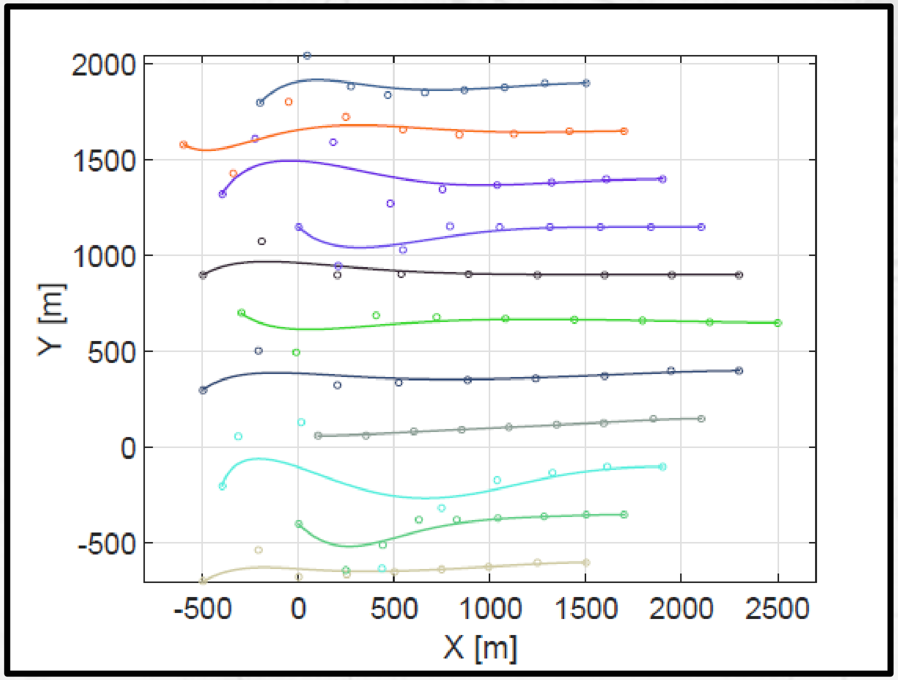


RESULTS: SCALABILITY



RESULTS :: MULTI-VEHICLE MISSIONS





Temporal separation

- Bernstein: <u>55 constraints</u>
- Pseudospectral: 550 constraints (<u>55*N</u>)

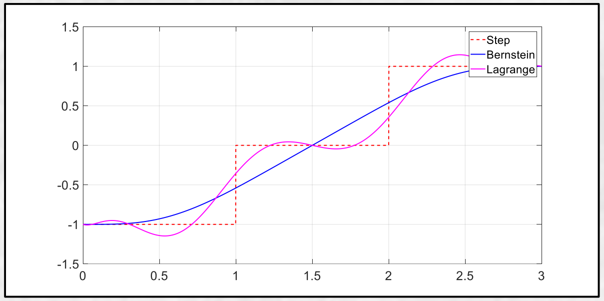
Spatial separation

- Bernstein: <u>55 constraints</u>
- Pseudospectral: 5500 constraints (<u>55*N^2</u>)

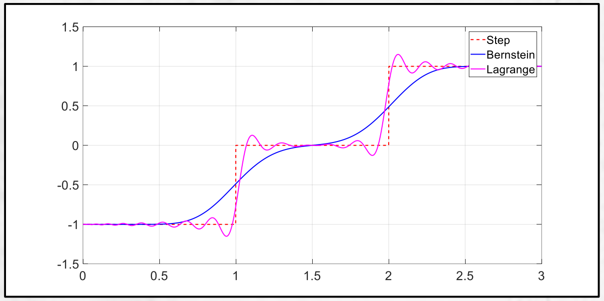
APPROXIMATING NONSMOOTH FUNCTIONS

- Bernstein approximations can be used to approximate piecewise continuous functions

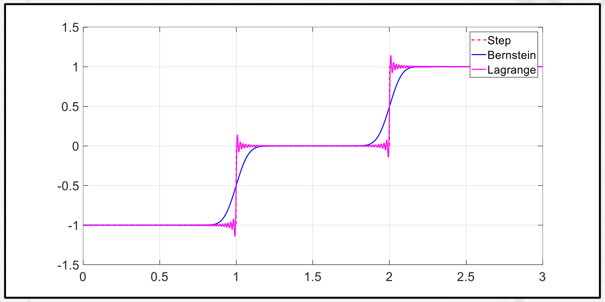
GIBBS PHENOMENON



N = 10



N = 50



N = 500

$$\lim_{n\uparrow\infty}\left(\max_{|x_0-x|\le\delta} f_n(x) - \min_{|x_0-x|\le\delta} f_n(x)\right) = C|f(x_0+0) - f(x_0-0)|.$$

$$C = \frac{2}{\pi}\int_0^{\pi}\left(\frac{\sin x}{x}\right)dx \approx 1.18.$$

Bernstein Approximation

$$\lim_{\delta\to 0}\lim_{n\to\infty}\left(\max_{|x_0-x|\le\delta} B_n f(x) - \min_{|x_0-x|\le\delta} B_n f(x)\right) = |f(x_0+0) - f(x_0-0)|.$$

Gzyl, Henryk, and José Luis Palacios. "On the approximation properties of Bernstein polynomials via probabilistic tools." *Boletın de la Asociación Matemática Venezolana* 10.1 (2003): 5-13.

APPROXIMATING NONSMOOTH FUNCTIONS

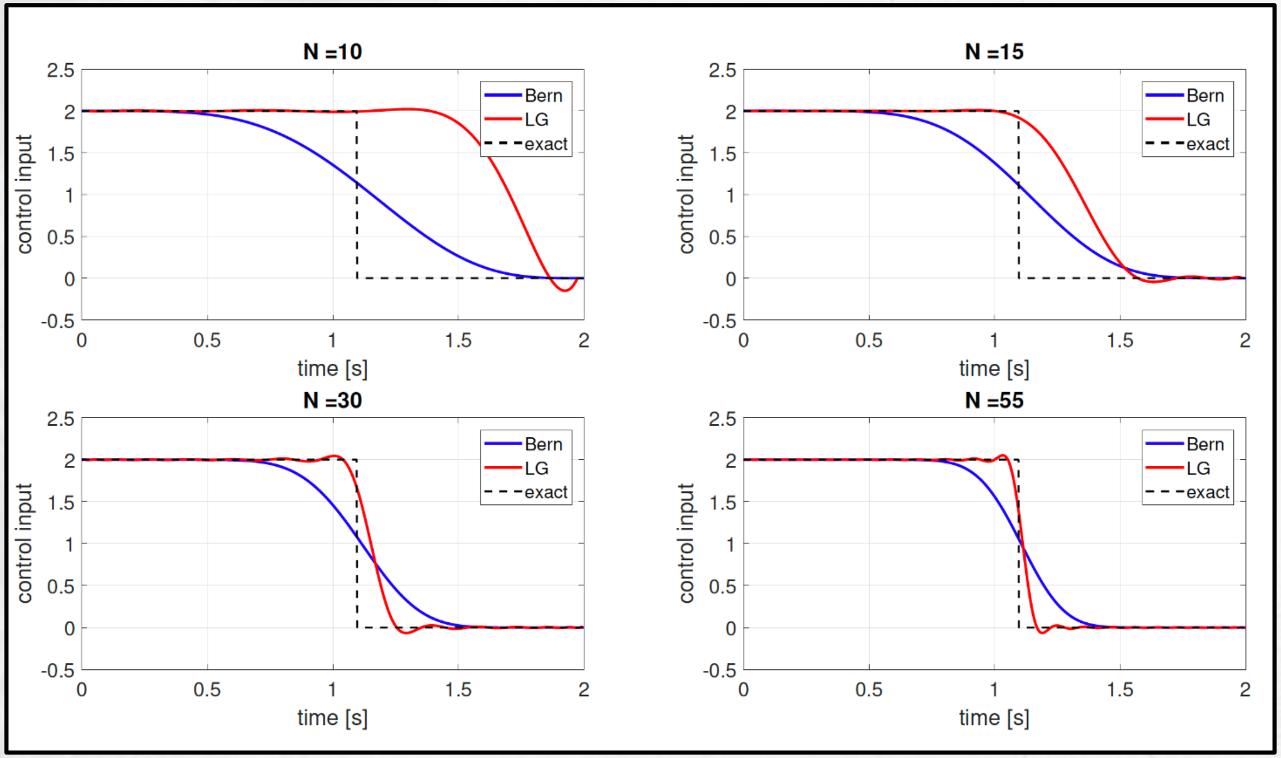
Minimize

$$I(y(t), u(t)) = \int_0^2 (3u(t) - 2y(t))dt\,,$$

subject to

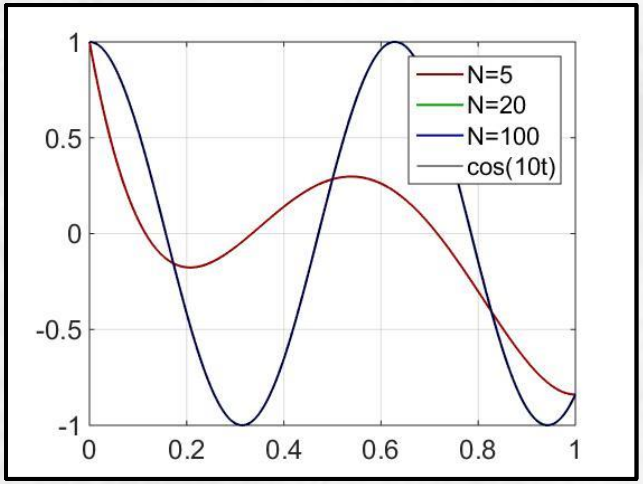
$$\dot{y}(t) = y(t) + u(t)\,, \quad \forall t \in [0,2],$$
$$y(0) = 4\,,$$
$$y(2) = 39.392\,,$$
$$0 \leq u(t) \leq 2 \quad \forall t \in [0,2]\,.$$

Optimal controller $u^*(t) = \begin{cases} 2 & 0 \leq t \leq 1.096 \\ 0 & 1.096 \leq t \leq 2. \end{cases}$

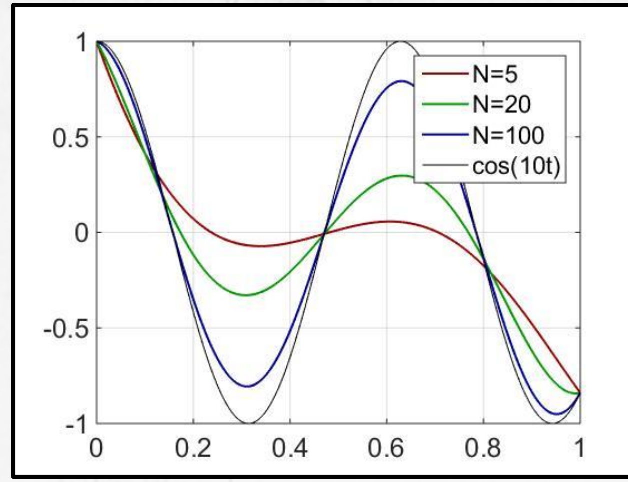


BERNSTEIN POLYNOMIAL APPROXIMATION

Lagrange interpolation (Legendre nodes)



Bernstein Approximation (equidistant nodes)



$$\boldsymbol{x} \in W^{m,\infty} \implies ||\boldsymbol{x}_N(t) - \boldsymbol{x}(t)||_{L^2} \leq \frac{C}{N^m}$$

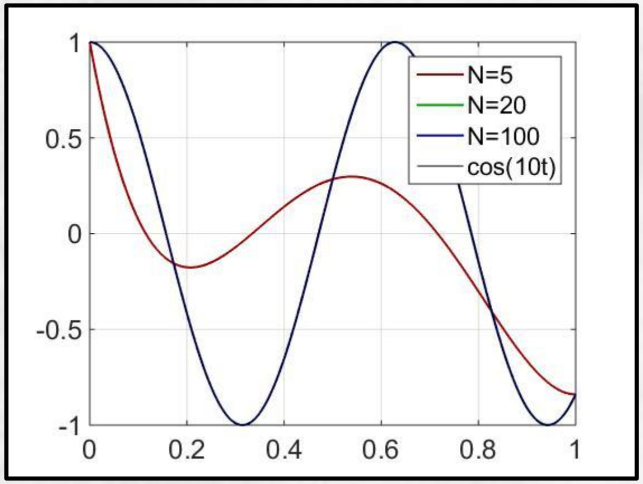
$$\boldsymbol{x} \in \mathcal{C}^2 \implies ||\boldsymbol{x}_N(t) - \boldsymbol{x}(t)|| \leq \frac{C}{N}$$

"The fact seems to have precluded any numerical application of Bernstein polynomials from having been made. Perhaps they will find application when the properties of the approximant in the large are of more importance than the closeness of the approximation."

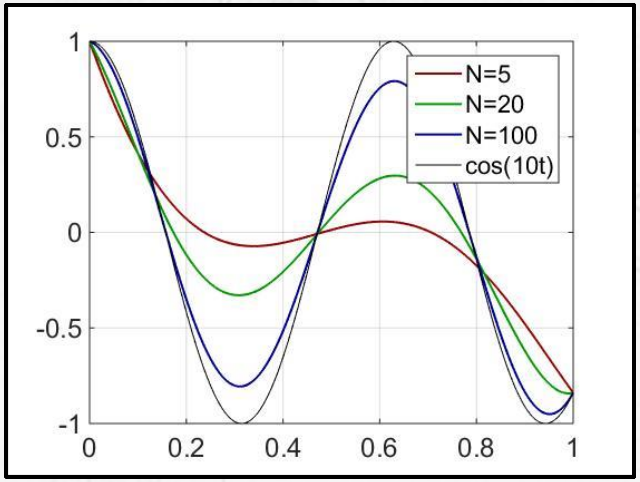
Davis, Philip J. *Interpolation and approximation*. Courier Corporation, 1975.

BERNSTEIN POLYNOMIAL APPROXIMATION

Lagrange interpolation (Legendre nodes)

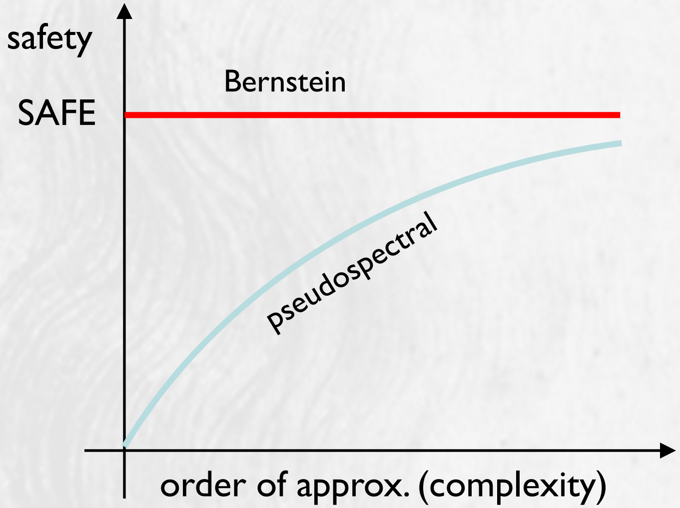


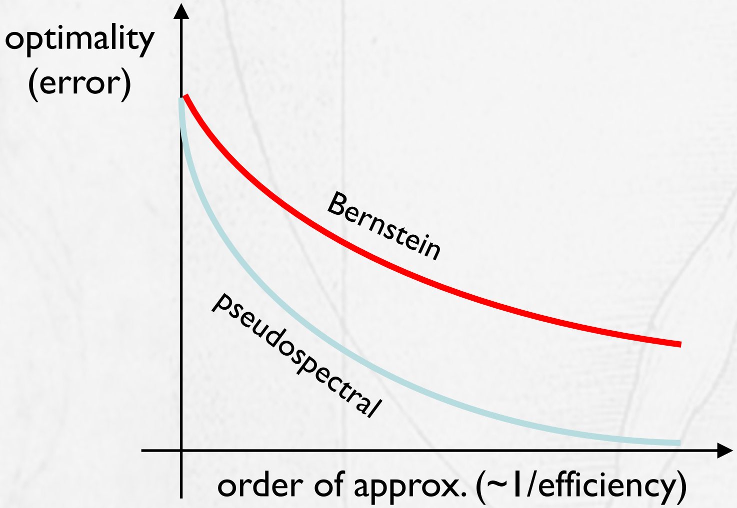
Bernstein Approximation (equidistant nodes)



$$\boldsymbol{x} \in W^{m,\infty} \implies ||\boldsymbol{x}_N(t) - \boldsymbol{x}(t)||_{L^2} \leq \frac{C}{N^m}$$

$$\boldsymbol{x} \in \mathcal{C}^2 \implies ||\boldsymbol{x}_N(t) - \boldsymbol{x}(t)|| \leq \frac{C}{N}$$



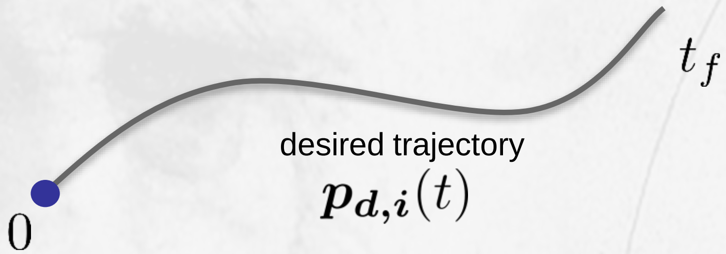


OUTLINE

- ❖ Introduction and general framework
- ❖ Optimal motion planning
- ❖ Coordinated tracking control
  - ❑ Virtual Target (VT) Tracking
  - ❑ Coordination control
- ❖ Conclusions

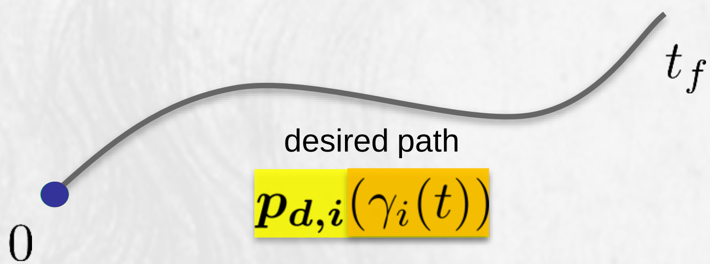
VT TRACKING vs TRAJECTORY TRACKING

A **trajectory** is a curve in space as a function of time: desired location of the vehicle at *any* point of time

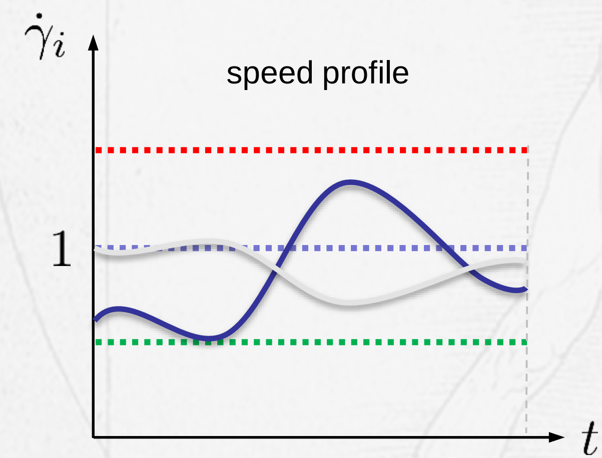


Trajectory tracking: enable vehicle *i* to track $p_{d,i}(t)$

A **path** is a curve in space, parameterized by an independent variable **(virtual time)** variable

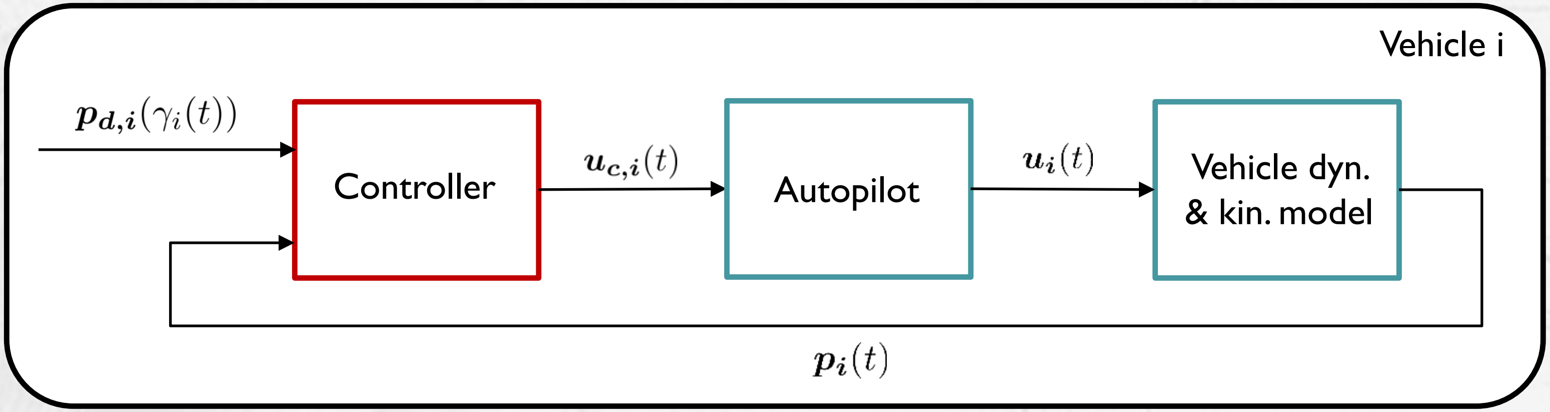


decoupling



VT tracking: enable vehicle *i* to track the **virtual target** $p_{d,i}(\gamma_i(t))$ independently on the speed profile

ASM: VT TRACKING



Assumption: $\|p_i(t) - p_{d,i}(\gamma_i(t))\| \leq B_{\mathrm{pf}}$




VT tracking algorithms are derived depending on the vehicle under consideration

VT TRACKING – FLIGHT TESTS

Cichella et al. 2012

$\dot{z}$	Vertical velocity
θ	Pitch
ϕ	Roll
$\dot{\psi}$	Yaw rate

Cichella et al. 2011

p	Roll rate
q	Pitch rate
v	Speed

Cichella et al. 2012

p	Roll rate
q	Pitch rate
r	Yaw rate
T	Total thrust

VT TRACKING vs AUTOPILOT

Ideal case

- Assume $\boldsymbol{p_{d,i}}(\gamma_i(t))$ is feasible
- Assume **ideal** performance of the A.P.

$$\|\boldsymbol{u}(t) - \boldsymbol{u}_c(t)\| \equiv 0$$

- Then, the path following error is locally **exponentially stable**

$$\|\boldsymbol{p_i}(t) - \boldsymbol{p_{d,i}}(\gamma_i(t))\| \leq k\|\boldsymbol{p_i}(0) - \boldsymbol{p_{d,i}}(\gamma_i(0))\|e^{-\lambda t}$$

$$k, \lambda > 0$$

Non-ideal case

- Assume $\boldsymbol{p_{d,i}}(\gamma_i(t))$ is feasible
- Assume **non-ideal** performance of the A.P.

$$\|\boldsymbol{u}(t) - \boldsymbol{u}_c(t)\| \leq B_{\mathrm{u}}$$

- Then, the path following error is locally **uniformly bounded**

$$\|\boldsymbol{p_i}(t) - \boldsymbol{p_{d,i}}(\gamma_i(t))\| \leq B_{\mathrm{pf}}(B_{\mathrm{u}})$$

COORDINATION

What about coordinating multiple vehicles?

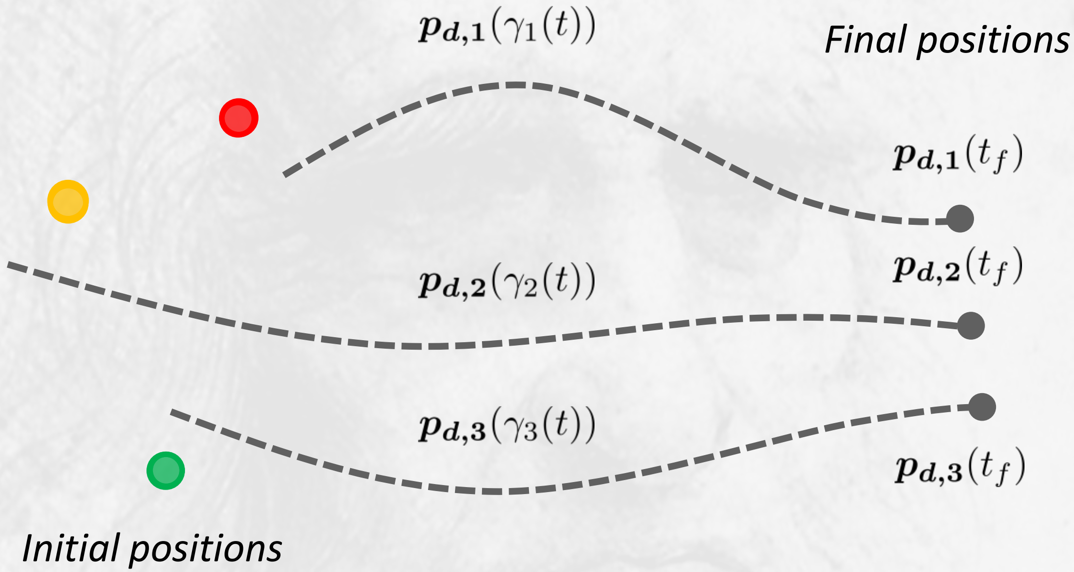
Adjust the progression of the **virtual time** $\gamma_i(t)$ in order to

- ❑ **achieve coordination between the vehicles**

  (???)

- ❑ while taking into account the feasibility constraints on $\boldsymbol{p_{d,i}}(\gamma_i(t))$

- ❑ and the path following error $\|\boldsymbol{p_i}(t) - \boldsymbol{p_{d,i}}(\gamma_i(t))\| \leq B_{\mathrm{pf}}(B_{\mathrm{u}})$



Simultaneous arrival

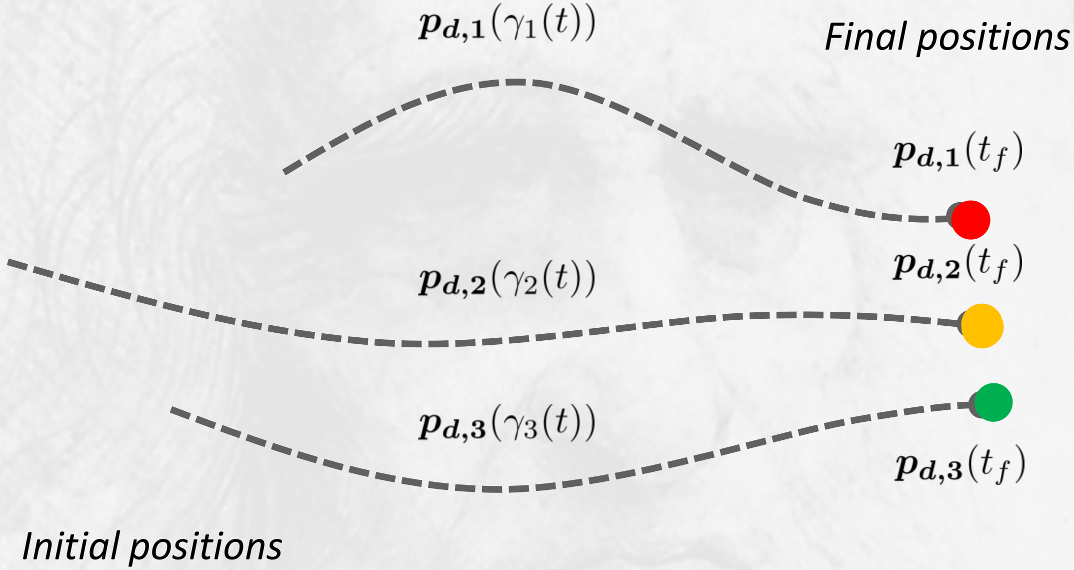
but…

Absolute time is <u>not</u> a priority

Consensus problem: reach an *agreement* on some distributed variables of interest *(coordination states)*

$$\gamma_i(t) - \gamma_j(t) \xrightarrow{t\to\infty} 0\,, \quad \forall\ i,j = 1,\ldots,n$$

COORDINATION OBJECTIVE



Simultaneous arrival

but…

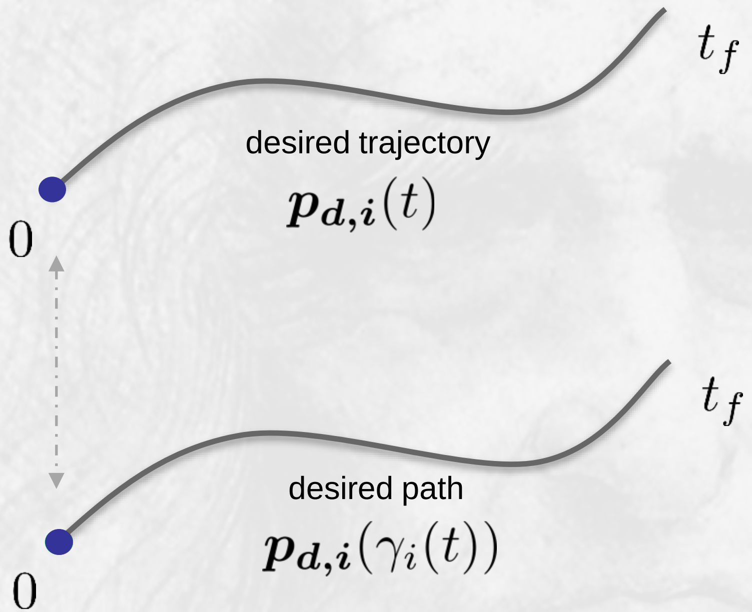
Absolute time is <u>not</u> a priority

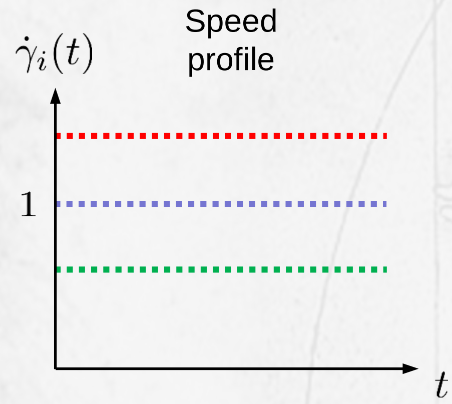
Consensus problem: reach an *agreement* on some distributed variables of interest *(coordination states)*

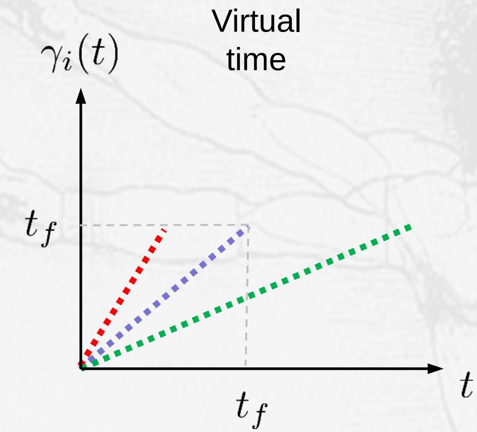
$$\gamma_i(t) - \gamma_j(t) \xrightarrow{t \to \infty} 0\,, \quad \forall\, i, j = 1, \ldots, n$$
$$\dot{\gamma}_i(t) \xrightarrow{t \to \infty} 1\,, \quad \forall\, i = 1, \ldots, n$$

Synchronize in both 'position' and 'speed'

COORDINATION OBJECTIVE







Consensus problem: reach an *agreement* on some distributed variables of interest *(coordination states)*

$$
\begin{aligned}
\gamma_i(t) - \gamma_j(t) &\xrightarrow{t\to\infty} 0\,, \quad \forall\ i,j = 1,\dots,n \\
\dot{\gamma}_i(t) &\xrightarrow{t\to\infty} 1\,, \quad \forall\ i = 1,\dots,n
\end{aligned}
$$

Synchronize in both 'position' and 'speed'

COORDINATION: PROBLEM FORMULATION

Coordinating multiple vehicles

Adjust $\ddot{\gamma}_i(t) = u_i(t)$ in order to

- **achieve coordination between the vehicles**

$$\begin{aligned} \gamma_i(t) - \gamma_j(t) &\overset{t\to\infty}{\longrightarrow} 0\,, \quad \forall\, i,j = 1,\ldots,n \\ \dot{\gamma}_i(t) &\overset{t\to\infty}{\longrightarrow} 1\,, \quad \forall\, i = 1,\ldots,n \end{aligned}$$

- while taking into account the feasibility constraints on $\boldsymbol{p_{d,i}}(\gamma_i(t))$

- and the path following error $\|\boldsymbol{p_i}(t) - \boldsymbol{p_{d,i}}(\gamma_i(t))\| \leq B_{\mathrm{pf}}(B_{\mathrm{u}})$

COORDINATION CONTROL LAW

- Distributed control law for group coordination:

$$\ddot{\gamma}_i(t) = -b(\dot{\gamma}_i(t)-1) - a\sum_{j\in\mathcal{N}_i}(\gamma_i(t)-\gamma_j(t)) - \alpha_i(\boldsymbol{p_{d,i}}(\gamma_i(t)) - \boldsymbol{p_i}(t)),$$

$$\alpha_i(\boldsymbol{p_{d,i}}(\gamma_i(t)) - \boldsymbol{p_i}(t)) = \frac{\dot{\boldsymbol{p}}_{\boldsymbol{d,i}}(\gamma_i(t))^\top(\boldsymbol{p_{d,i}}(\gamma_i(t)) - \boldsymbol{p_i}(t))}{||\dot{\boldsymbol{p}}_{\boldsymbol{d,i}}(\gamma_i(t))|| + \delta}$$

- Each vehicle exchanges only its **coordination state** with its neighbors
- Control law accounts for path following error

$$\begin{aligned}\gamma_i(t) - \gamma_j(t) &\overset{t\to\infty}{\longrightarrow} 0, \quad \forall\, i,j = 1,\ldots,n\\ \dot{\gamma}_i(t) &\overset{t\to\infty}{\longrightarrow} 1, \quad \forall\, i = 1,\ldots,n\end{aligned}$$

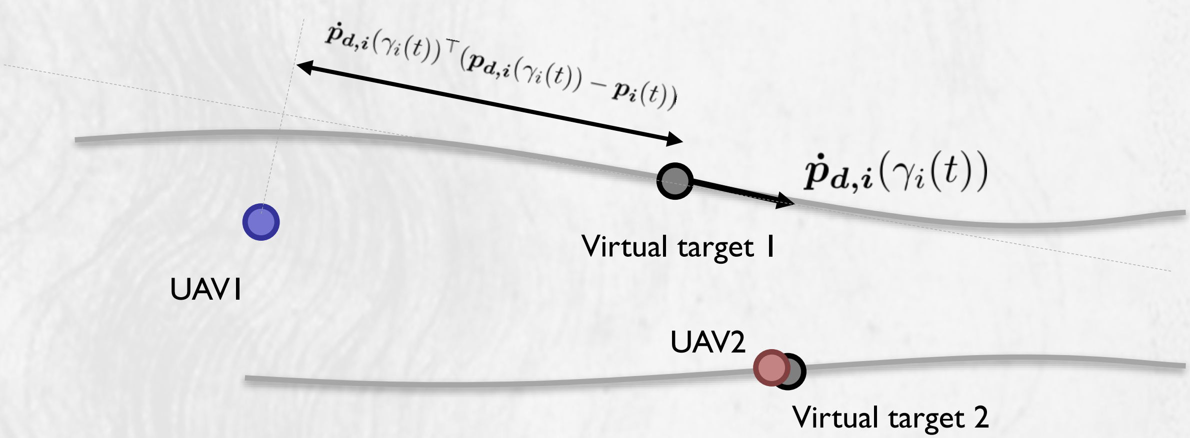
COORDINATION CONTROL LAW

- Distributed control law for group coordination:

$$\ddot{\gamma}_i(t) = -b(\dot{\gamma}_i(t) - 1) - a \sum_{j \in \mathcal{N}_i} (\gamma_i(t) - \gamma_j(t)) - \alpha_i(\boldsymbol{p_{d,i}}(\gamma_i(t)) - \boldsymbol{p_i}(t)) \,,$$

$$\alpha_i(\boldsymbol{p_{d,i}}(\gamma_i(t)) - \boldsymbol{p_i}(t)) = \frac{\dot{\boldsymbol{p}}_{\boldsymbol{d,i}}(\gamma_i(t))^\top (\boldsymbol{p_{d,i}}(\gamma_i(t)) - \boldsymbol{p_i}(t))}{||\dot{\boldsymbol{p}}_{\boldsymbol{d,i}}(\gamma_i(t))|| + \delta}$$

- Each vehicle exchanges only its **coordination state** with its neighbors
- Control law accounts for path following error



Virtual target 1 waits for UAV1

By virtue of coordination, also UAV2 waits for UAV1

COORDINATION CONTROL LAW

- Distributed control law for group coordination:

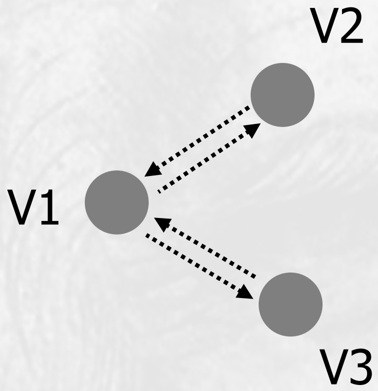
$$\ddot{\gamma}_i(t) = -b(\dot{\gamma}_i(t) - 1) - a \sum_{j \in \mathcal{N}_i} (\gamma_i(t) - \gamma_j(t)) - \alpha_i(\boldsymbol{p_{d,i}}(\gamma_i(t)) - \boldsymbol{p_i}(t)) \,,$$

$$\alpha_i(\boldsymbol{p_{d,i}}(\gamma_i(t)) - \boldsymbol{p_i}(t)) = \frac{\dot{\boldsymbol{p}}_{\boldsymbol{d,i}}(\gamma_i(t))^\top (\boldsymbol{p_{d,i}}(\gamma_i(t)) - \boldsymbol{p_i}(t))}{||\dot{\boldsymbol{p}}_{\boldsymbol{d,i}}(\gamma_i(t))|| + \delta}$$

- Each vehicle exchanges only its **coordination state** with its neighbors
- Control law accounts for path following error

Under which assumptions on the communication network this control law guarantees that the coordination objective is attained?

COMMUNICATION NETWORK



Laplacian Matrix $\boldsymbol{L} = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$

The graph is connected if

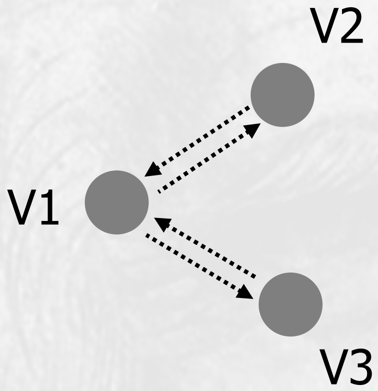
$$\text{rank}\boldsymbol{L} = n - 1 = 2$$

V1 receives info from *neighbours* V2 and V3

V2 receives info from *neighbour* V1

V3 receives info from *neighbour* V1

COMMUNICATION NETWORK

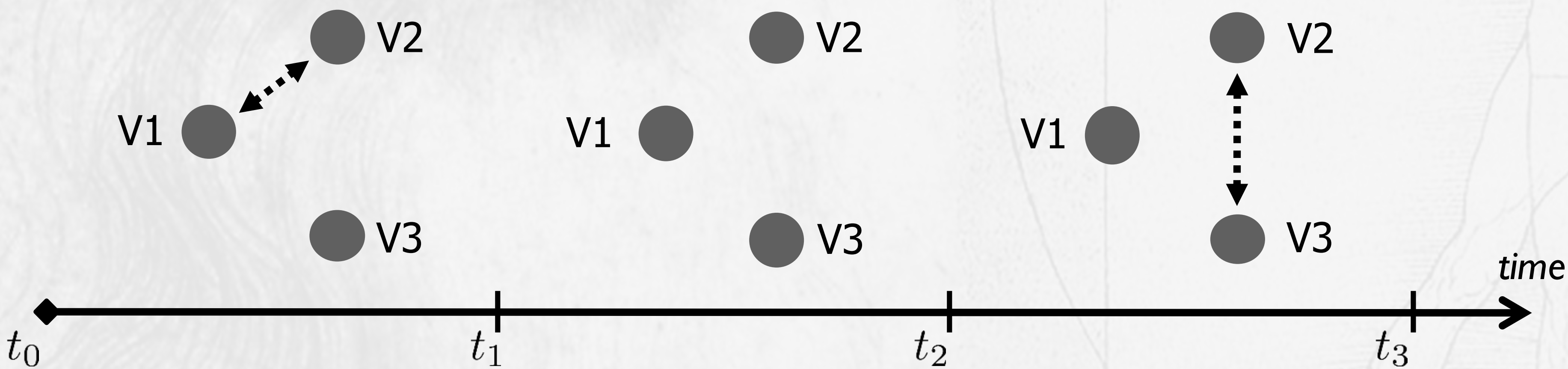


Laplacian Matrix $\boldsymbol{L} = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$

The graph is connected if

$$\mathrm{rank}\boldsymbol{L} = n - 1 = 2$$

Graph connected in the mean



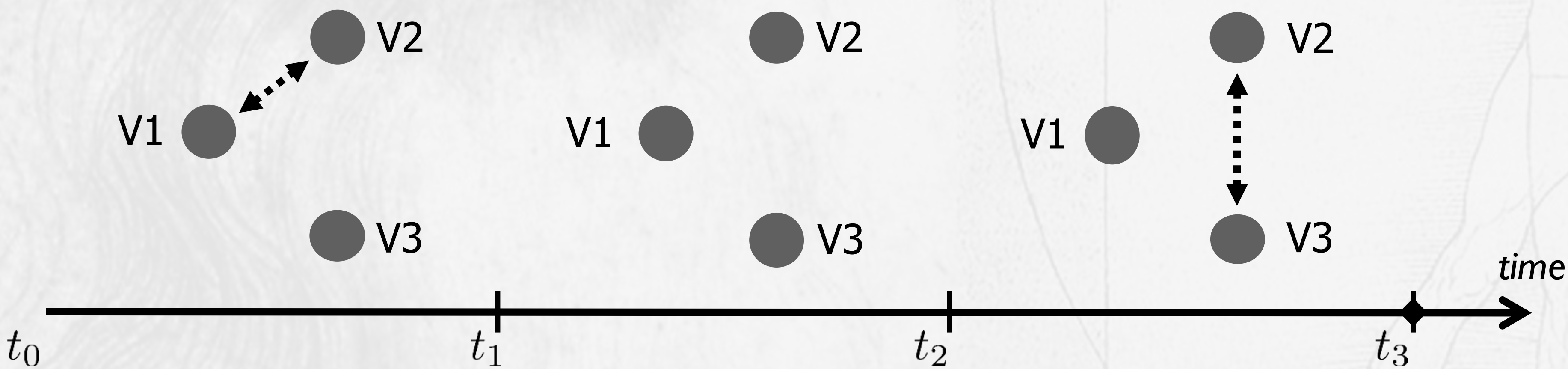
$$\boldsymbol{L}_T = \boldsymbol{L}_{[t_0,t_1]} + \boldsymbol{L}_{[t_1,t_2]} + \boldsymbol{L}_{[t_2,t_3]}\,, \qquad \mathrm{rank}\boldsymbol{L}_T = n - 1 = 2$$

COMMUNICATION NETWORK

$$\frac{1}{n}\frac{1}{T}\int_{t}^{t+T} \boldsymbol{Q}\,\boldsymbol{L}(\tau)\,\boldsymbol{Q}^{\top} d\tau \geq \mu\,\mathbb{I}_{\boldsymbol{n-1}}\,, \qquad \forall\, t \geq 0, \qquad \boldsymbol{Q}\boldsymbol{1}_{\boldsymbol{n}} = 0$$

Network connected in an integral sense, not pointwise in time (Arcak 2007)

Parameters μ and T characterize the **QoS of the network**



$$\boldsymbol{L}_T = \boldsymbol{L}_{[t_0,t_1]} + \boldsymbol{L}_{[t_1,t_2]} + \boldsymbol{L}_{[t_2,t_3]}\,, \qquad \mathrm{rank}\boldsymbol{L}_T = n-1 = 2$$

COORDINATION: MAIN RESULT

- Assume **network connectivity** satisfies

$$\frac{1}{n}\frac{1}{T}\int_{t}^{t+T} \boldsymbol{Q}\,\boldsymbol{L}(\tau)\,\boldsymbol{Q}^\top d\tau \geq \mu\,\mathbb{I}_{\boldsymbol{n-1}}\,, \qquad \forall\, t \geq 0$$

- The coordination states $\boldsymbol{x_{cd,i}}(t) = \left[\sum_{j=1}^{n}(\gamma_i(t) - \gamma_j(t))\,, \quad \dot{\gamma}_i(t) - 1\right]$ *satisfy*

$$\|\boldsymbol{x_{cd,i}}(t)\| \leq \kappa_1\|\boldsymbol{x_{cd,i}}(0)\|e^{-\lambda_{cd}t} + \kappa_2\text{sup}_{t\geq 0}\|\boldsymbol{p_{d,i}}(\gamma_i(t)) - \boldsymbol{p_i}(t)\|$$

AUTOPILOT PERFORMANCE

- For ideal performance of the autopilot *the coordination states* ***converge to zero exponentially*** *with rate of convergence*

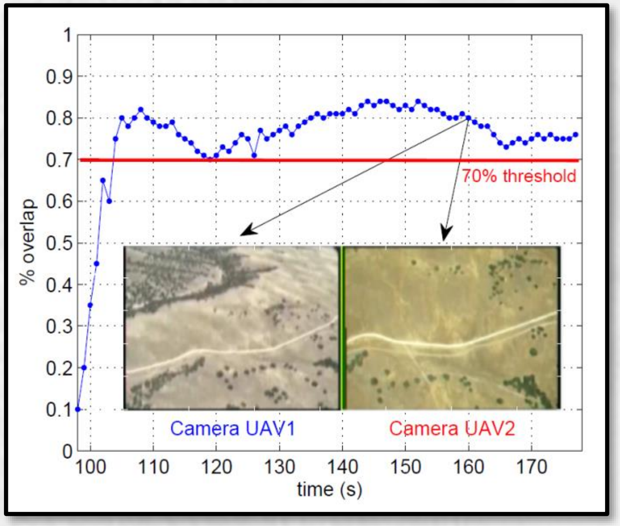
$$\lambda_{cd} \geq \bar{\lambda}_{cd} \triangleq \frac{a}{b}\frac{n\mu}{T\left(1 + \frac{a}{b}nT\right)^2}$$

QoS of the communication network

- Moreover, $\boldsymbol{p_{d,i}}(\gamma_i(t))$ is feasible.

V. Cichella, I. Kaminer, V. Dobrokhodov, E. Xargay, R. Choe, N. Hovakimyan, A. P. Aguiar, and A. M. Pascoal. "Cooperative path following of multiple multirotors over time-varying networks." *IEEE Transactions on Automation Science and Engineering* 12, no. 3 (2015): 945-957.

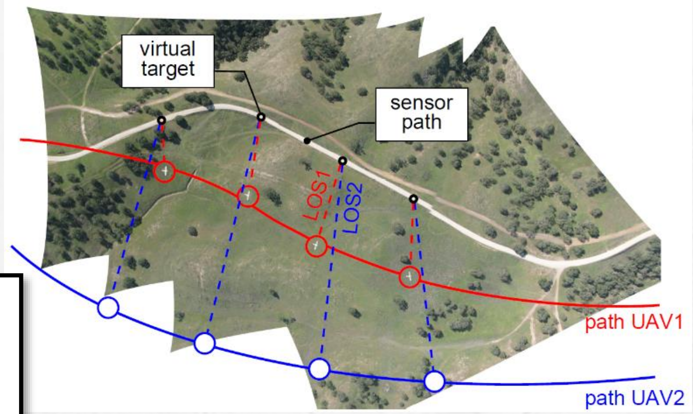
COOPERATIVE ROAD SEARCH: FLIGHT TESTS



Cooperation** ensures satisfactory overlap of the field-of-view footprints of the sensors, increasing the probability of **target detection

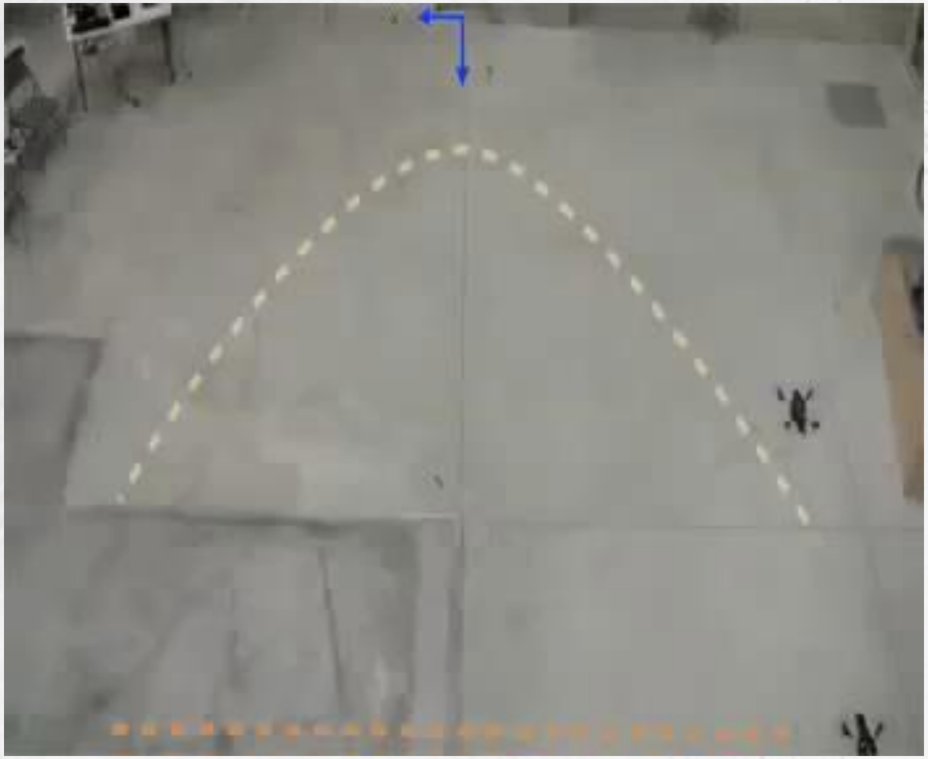
UAV 1

UAV 2



Mosaic of 4 consecutive high-resolution images

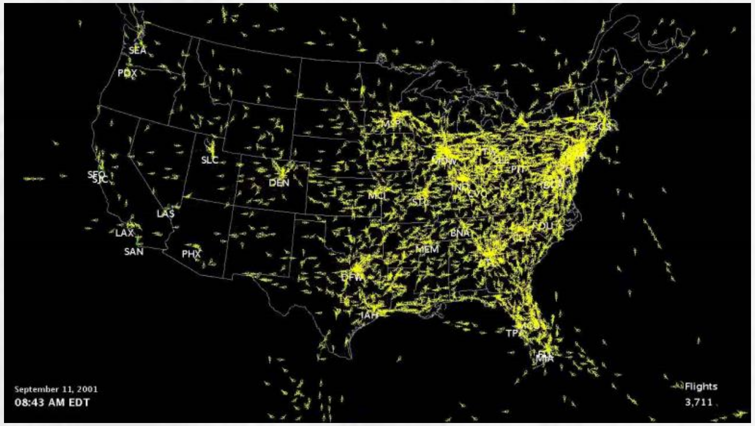
AR.DRONE: FLIGHT TESTS

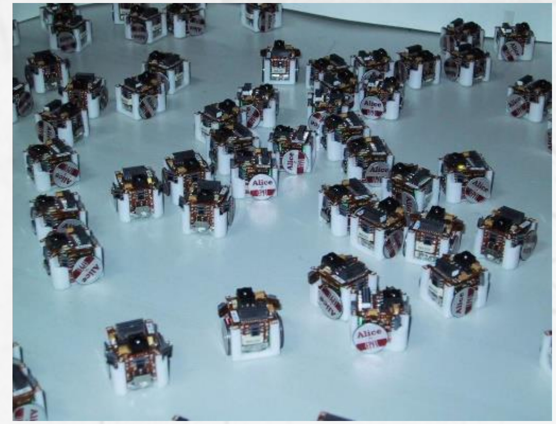


OUTLINE

- ❖ Introduction and general framework
- ❖ Optimal motion planning
- ❖ Coordinated tracking control
- ❖ Conclusions

OPTIMAL MOTION PLANNING





- **Implementation**
  - Develop a toolbox for trajectory generation
    - Python
    - Machine learning + Optimization
- **Uncertainties:**
  - Address generalized stochastic optimal control problems

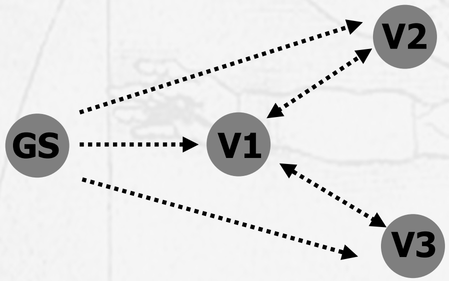
COORDINATED TRACKING CONTROL

- Previous work

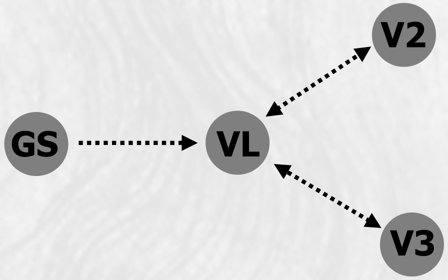
$$\gamma_i(t) - \gamma_j(t) \overset{t\to\infty}{\longrightarrow} 0\,, \qquad \forall\, i,j = 1,\dots,n$$

$$\dot{\gamma}_i(t) \overset{t\to\infty}{\longrightarrow} r(t)\,, \qquad \forall\, i = 1,\dots,n$$

$$\frac{1}{n}\frac{1}{T}\int_t^{t+T} \boldsymbol{Q}\,\boldsymbol{L}(\tau)\,\boldsymbol{Q}^\top d\tau \geq \mu\,\mathbb{I}_{\boldsymbol{n-1}}\,, \qquad \forall\, t \geq 0$$



- Future work



$$\ddot{\gamma}_1(t) = -k_D(\dot{\gamma}_1(t) - r(t)) - k_P \sum_{j\in\mathcal{N}_1} (\gamma_1(t) - \gamma_j(t))$$

$$\ddot{\gamma}_i(t) = -k_D(\dot{\gamma}_i(t) - \chi(t)) - k_P \sum_{j\in\mathcal{N}_i} (\gamma_i(t) - \gamma_j(t))\,, \qquad i = 2,3$$

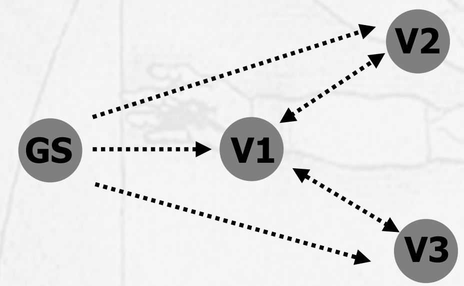
$$\dot{\chi}(t) = -k_I \sum_{j\in\mathcal{N}_i(t)} (\gamma_i(t) - \gamma_j(t))\,, \qquad i = 2,3$$

COORDINATED TRACKING CONTROL

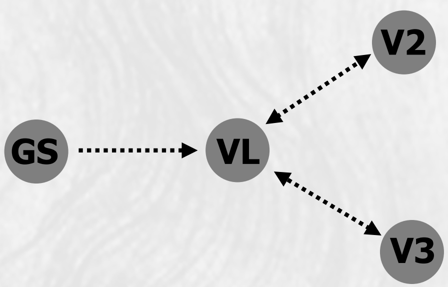
- Previous work

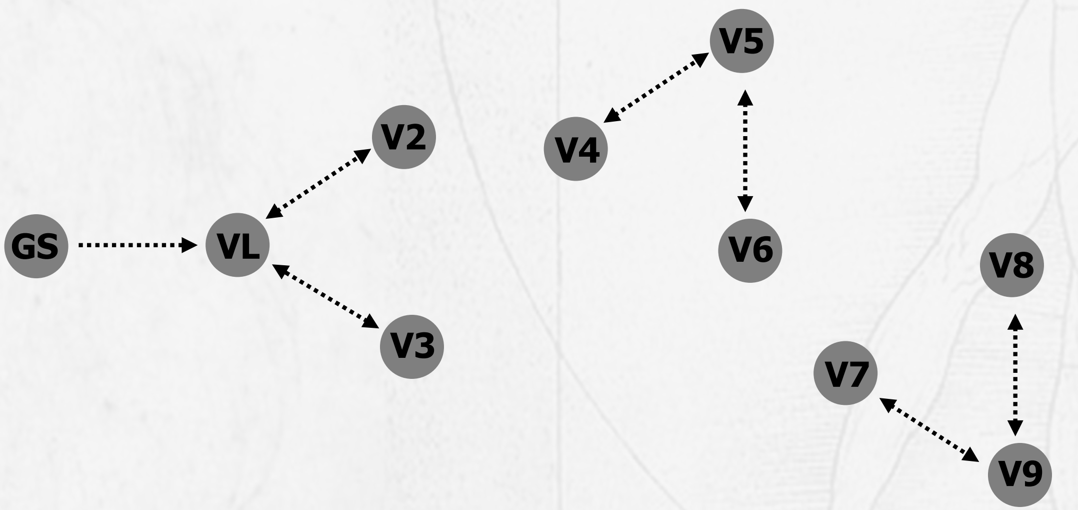
$$\gamma_i(t) - \gamma_j(t) \overset{t\to\infty}{\longrightarrow} 0\,, \qquad \forall\, i,j = 1,\dots,n$$
$$\dot{\gamma}_i(t) \overset{t\to\infty}{\longrightarrow} r(t)\,, \qquad \forall\, i = 1,\dots,n$$

$$\frac{1}{n}\frac{1}{T}\int_t^{t+T} \boldsymbol{Q}\,\boldsymbol{L}(\tau)\,\boldsymbol{Q}^\top d\tau \geq \mu\,\mathbb{I}_{\boldsymbol{n-1}}\,, \qquad \forall\, t \geq 0$$



- Future work

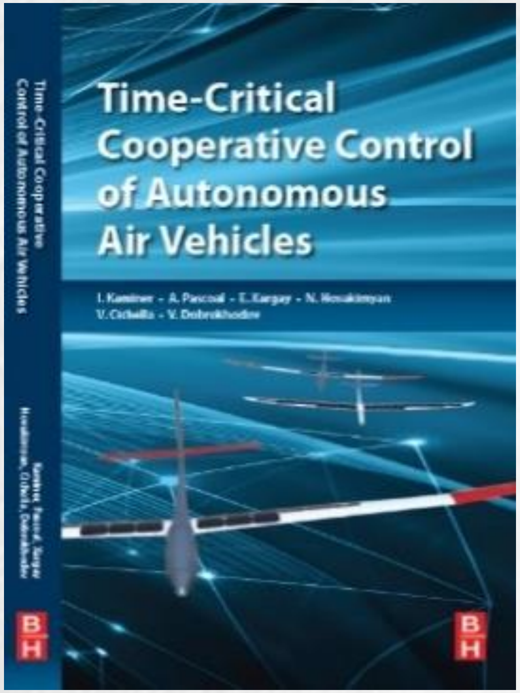




SUMMARY

- **Main objective: safe** use of **cooperative UxSs** in complex spaces

- Planning and coordinated tracking
  - Motion planning
  - VT tracking
  - Coordination control



(INCOMPLETE) LIST OF ACKNOWLEDGMENTS

- Naira Hovakimyan, UIUC
- Isaac Kaminer, NPS
- Dusan Stipanovic, UIUC
- Daniel Liberzon, UIUC
- Petros Voulgaris, UIUC
- Vladimir Dobrokhodov, NPS
- Claire Walton, NPS
- Hyung-Jin Yoon, UIUC
- Thiago de Souza Marinho, UIUC
- Syed Bilal Mehdi, GM
- Arun Lakshmanan, UIUC
- Ronald Choe, UIUC
- Javier Puig Navarro, UIUC
- Hanmin Lee, UIUC
- Enric Xargay, Barcelona, Spain
- Lorenzo Marconi, University of Bologna, Italy
- Roberto Naldi, University of Bologna, Italy
- Antonio Pascoal, IRS, IST, Lisbon, Portugal
- Anna Trujillo, NASA LaRC
- Alex Kirlik, UIUC
- Frances Wang, UIUC
- et al.

Thank you

UNIVERSIDADE
DE LISBOA

BACKUP SLIDES

WHAT IS AUTONOMY?

- **There is no formal definition of autonomy/autonomous system**

- **We say that "an autonomous system is a device (software or hardware) that performs some tasks or functions independently without human intervention."**
  - **Human-level decision making**

- **This implies that an autonomous system can have different levels of autonomy [1].**

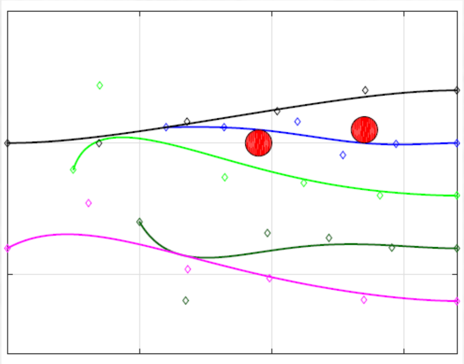
- ***Sensory/Motor* Autonomy:** Translate high-level human commands (e.g. reach desired altitude, cruise control, automated parallel parking, desired destination, etc.) and sensors (e.g. GPS, IMU, accelerometer) into platform dependent signals (e.g. roll, pitch, yaw angles, speed, angular speed, etc.) to achieve low-level tasks (waypoint navigation, follow pre-planned trajectories, etc.);
- ***Reactive* Autonomy:** sensory/motor autonomy + react to perturbations (wind, mechanical failure, etc.) coordinate with other objects/vehicles, sense and avoid, ...
- **Cognitive Autonomy:** reactive autonomy + recognize and obey to traffic signals, perform SLAM, plan/take decisions (for example based on battery level, road traffic and weather information, a set of desired destinations, etc.), learn, …

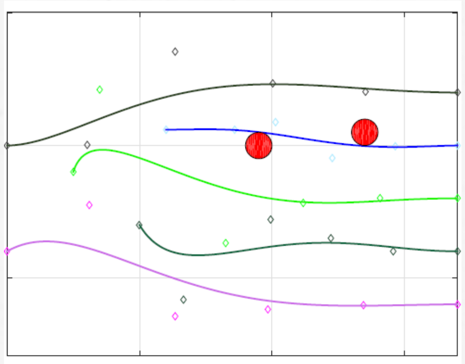
[1] Dario Floreano and Robert J. Wood. "Science, technology and the future of small autonomous drones." *Nature* 521.7553 (2015): 460-466.

OPTIMAL MOTION PLANNING

❖ Previous Work

- ❑ Trajectory Generation – Optimal control problem
- ❑ Bezier curves to efficiently generate trajectories
- ❑ Efficient and safe – multiple vehicles missions



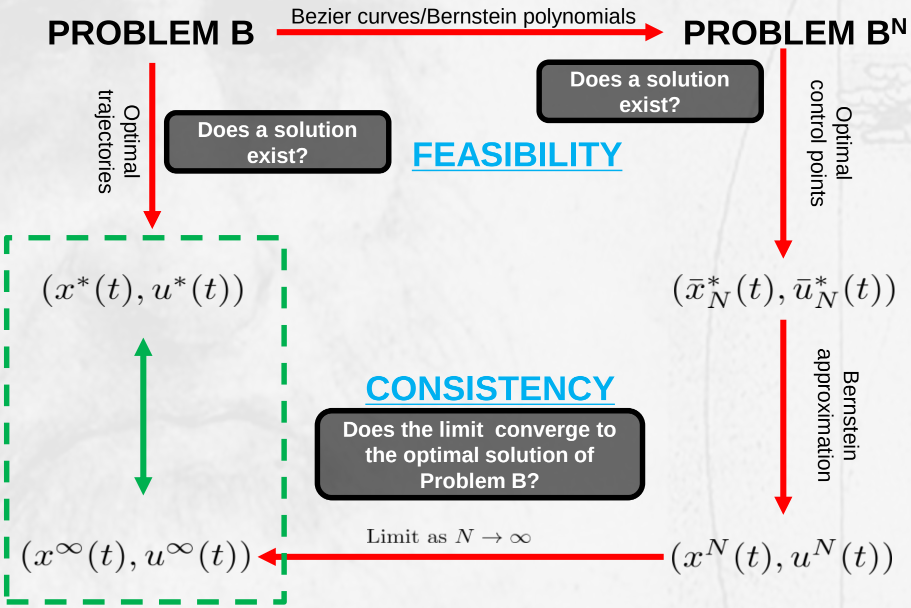


❖ Ongoing and Future work

- ❑ Theory – Feasibility/Consistency
- ❑ Implementation – Trajectory Generation toolbox

FUTURE WORK – OPTIMAL MOTION PLANNING

❖ Theory – feasibility/consistency

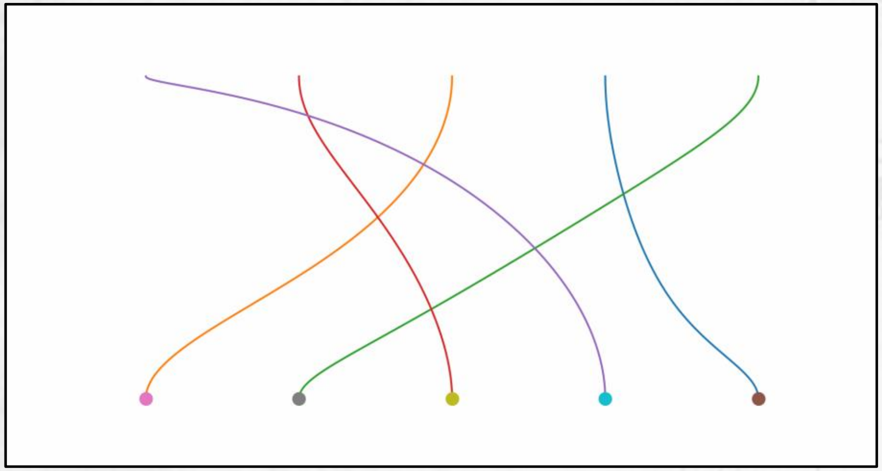


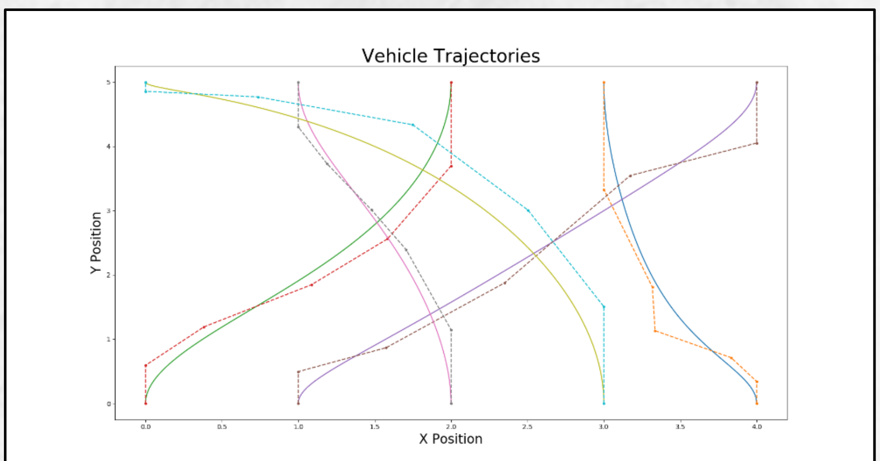
❖ Implementation – Trajectory generation toolbox

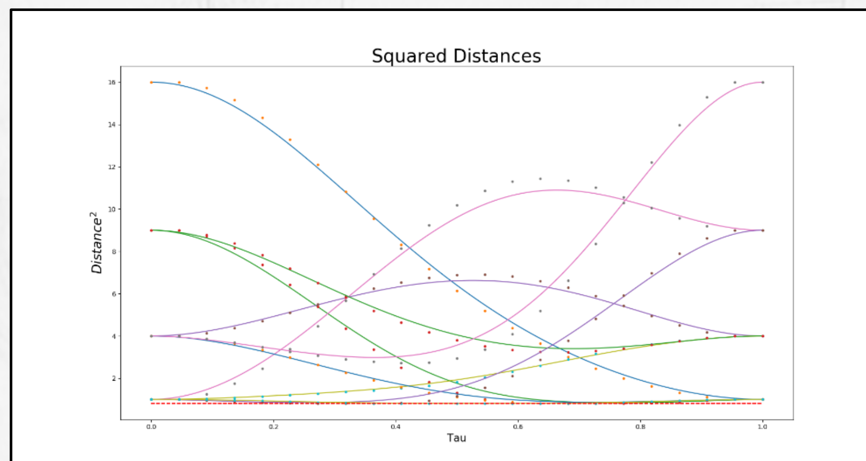
- ❑ Genetic algorithm – MATLAB, Julia, Python
- ❑ Distance between Bezier curves
- ❑ Flying and ground robots

Possible funding sources: NSF CMMI-DCSD.

ONGOING WORK

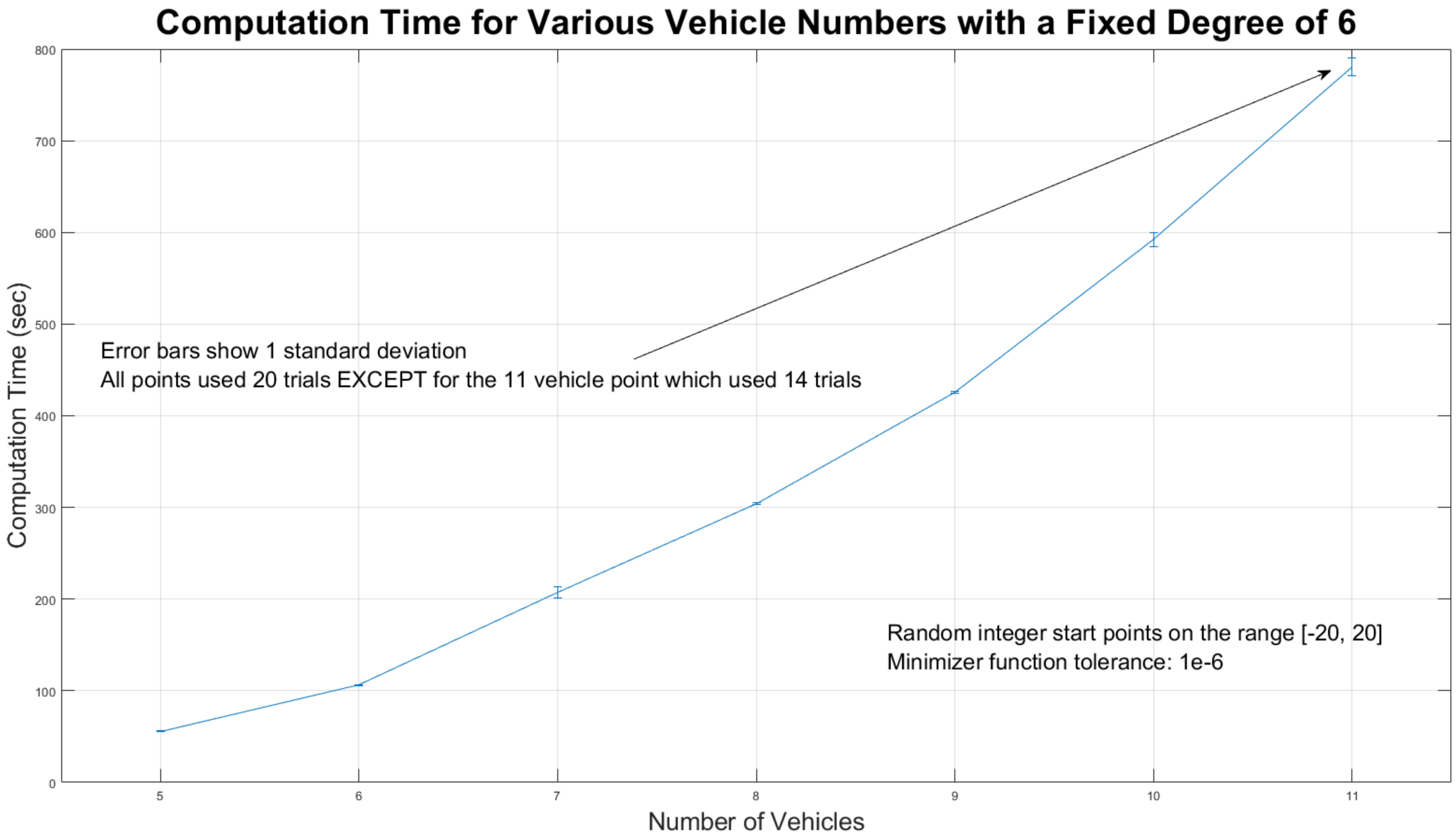






Computation time: 50 seconds

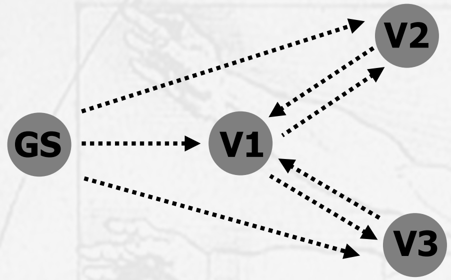
ONGOING WORK



COOPERATIVE CONTROL

❖ Previous Work

- ❑ Multi-agent coordination
- ❑ Switching topologies and dropouts
- ❑ Desired pace known to every vehicle



$$\gamma_i(t) - \gamma_j(t) \overset{t\to\infty}{\longrightarrow} 0\,, \qquad \forall\, i,j = 1,\ldots,n$$
$$\dot{\gamma}_i(t) \overset{t\to\infty}{\longrightarrow} r(t)\,, \qquad \forall\, i = 1,\ldots,n$$

$$\frac{1}{n}\frac{1}{T}\int_t^{t+T} \boldsymbol{Q}\,\boldsymbol{L}(\tau)\,\boldsymbol{Q}^\top d\tau \geq \mu\,\mathbb{I}_{\boldsymbol{n-1}}\,, \qquad \forall\, t \geq 0$$

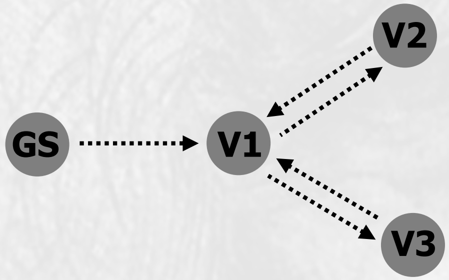
$$\ddot{\gamma}_i(t) = -k_D(\dot{\gamma}_i(t) - \underline{r(t)}) - k_P \sum_{j\in\mathcal{N}_i} (\gamma_i(t) - \gamma_j(t))$$

❖ Ongoing and Future work

- ❑ Leader-follower
- ❑ Low information – estimation
- ❑ Quantized information

FUTURE WORK – COOPERATIVE CONTROL

❖ Leader-Follower



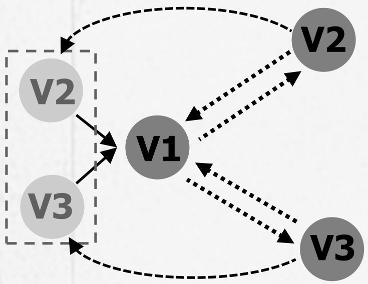
$$\ddot{\gamma}_1(t) = -k_D(\dot{\gamma}_1(t) - r(t)) - k_P \sum_{j \in \mathcal{N}_1} (\gamma_1(t) - \gamma_j(t))$$

$$\ddot{\gamma}_i(t) = -k_D(\dot{\gamma}_i(t) - \chi(t)) - k_P \sum_{j \in \mathcal{N}_i} (\gamma_i(t) - \gamma_j(t)), \qquad i = 2, 3$$

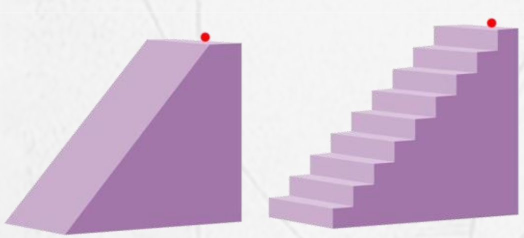
$$\dot{\chi}(t) = -k_I \sum_{j \in \mathcal{N}_i(t)} (\gamma_i(t) - \gamma_j(t)), \qquad i = 2, 3$$

❖ Low information

$$\frac{1}{n}\frac{1}{T}\int_t^{t+T} \boldsymbol{Q}\,\boldsymbol{L}(\tau)\,\boldsymbol{Q}^\top d\tau \geq \mu\,\mathbb{I}_{\boldsymbol{n-1}}, \qquad \forall\, t \geq 0$$

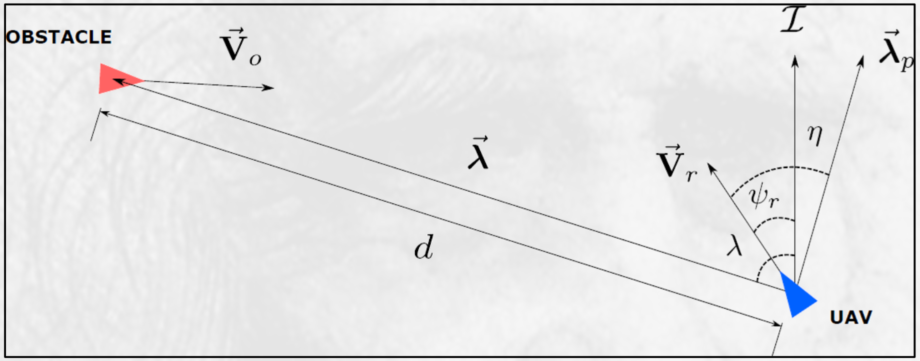


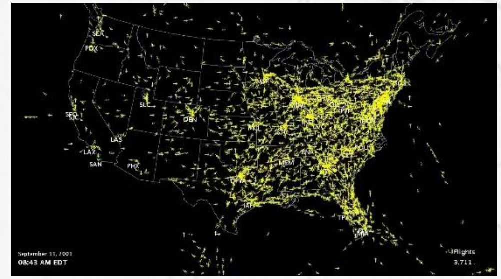
❖ Quantized information

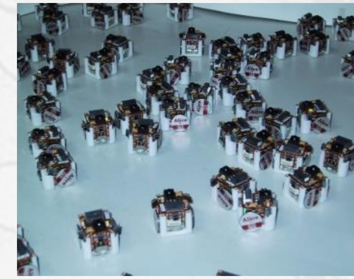


Possible funding sources: NSF CPS, AFOSR DURIP, ONR Science & Autonomy

CONTROL WITH LIMITED INFORMATION







- ❑ Problem:

$$\eta(t) = \psi_r(t) - (\lambda(t) - \tfrac{\pi}{2}) \to 0$$

$$d(t) \geq d_{sf}\,, \quad \forall t \geq t_0\,.$$

- ❑ Control law:

$$\dot{\psi}_{\text{ca}} = -k\eta(t)$$

- ❑ Main result:

$$d(t) \geq d_{sf}\,, \quad \forall t \geq t_0$$

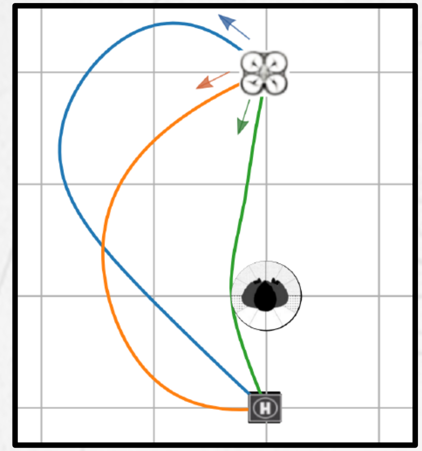
Future directions

- ❑ Collision detection with low amount of information (turn on/off)
- ❑ Can the same strategy be used to reach formation?
- ❑ Implementation – flying & ground vehicles

Possible funding sources: NSF CPS, AFOSR DURIP.

SOCIALLY AWARE ROBOTS

- ❑ Virtual Reality
- ❑ Psychology experiment design
- ❑ Machine learning



Package delivery robot in VR

Human

Physiological sensors

$\mathbf{x}$: robot's **position** and **velocity**

$\mathbf{y} = F(\mathbf{x}, \mathbf{z})$

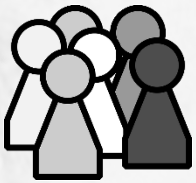
$\mathbf{y}$: **arousal state**

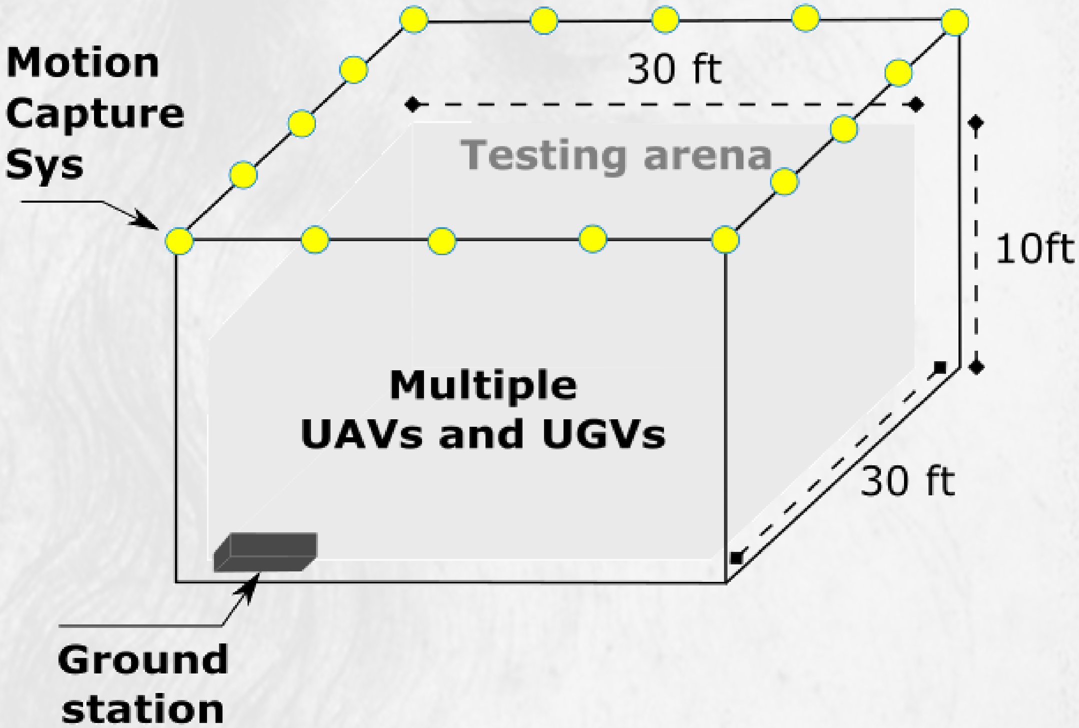
Experiments conducted on 62 human subjects for data collection

Possible funding sources: NSF NRI2.0, NSF Smart and Connected Communities, NASA STRG, DoD MURI, ONR Science of Autonomy

5 YEARS VISION

Co-OPerative Autonomy (COPA) Lab

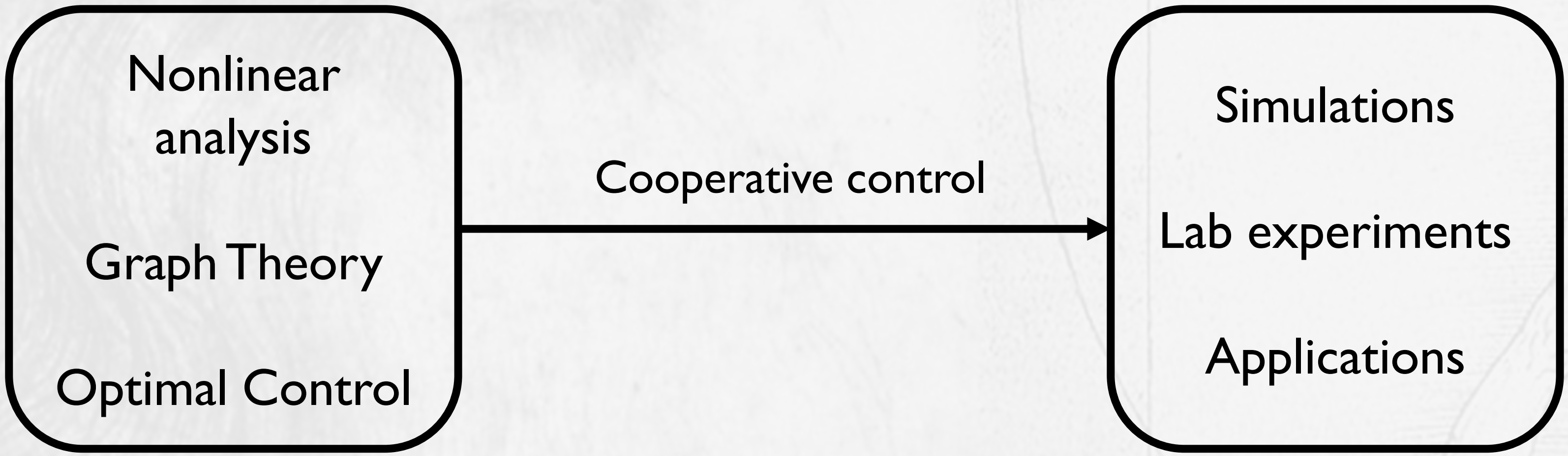




TEACHING PHILOSOPHY

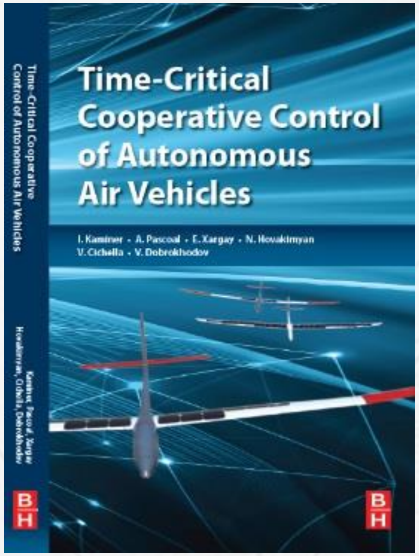
- Bridge solid theory with hands-on experience
  - Theory – Implementation – Applications/benefits

- Inspire curiosity
  - Share knowledge and understanding of the big picture
  - Emphasize the significance of the details that they need to work on
  - Connect them with the constantly evolving world



COURSES

- **Teaching Activities**
  - Introduction to Dynamics
  - Signal Processing
  - Control Theory
  - Robust and Adaptive Control
- **Mechanical and Aerospace Engineering – Missouri S&T**
  - Statics and Dynamics
  - Modelling and Analysis of Dynamic Systems
  - Automatic Control of Dynamic Systems
  - Flight Dynamics, Stability and Control
  - Dynamics of Mechanical and Aerospace Systems
  - Signal Processing
  - Mechanical and Aerospace Control Systems
  - …
- **Additional Courses**
  - Cooperative Autonomous Vehicles
  - Robust and Adaptive Control



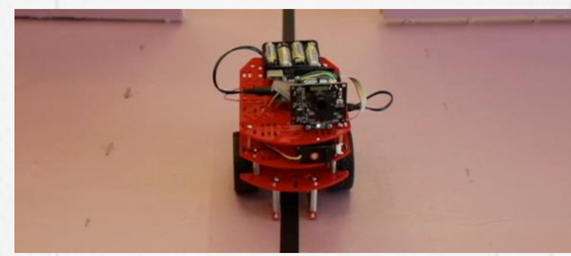
OUTREACH

the PYGMALION FESTIVAL
MUSIC TECH LIT MADE

Undergraduate students working on drones teleoperation

CLARK-LINDSEY

Interaction with seniors at eldercare facility



High-School students working on ground robots

"The most exciting phrase to hear in science is not 'Eureka!', but 'That's funny' " – Isaac Asimov (1929 – 1992)