A Time-Critical Cooperative Framework for Autonomous Mobility in Complex Scenarios

Javier Puig-Navarro

ACIT Institute Seminar Series

November 6th, 2020

University of Illinois
at Urbana-Champaign

NASA Langley Research Center
Autonomous Systems Integrated Research Branch

ILLINOIS

javier.puignavarro@nasa.gov

Outline

- **Introduction**
- **Trajectory planning**
  - Proximity queries
  - Silhouette-informed path-planning
- **Mission execution**
  - Path following for multirotors
  - Distributed time-critical coordination
- **Conclusions**
- **Future work**

Introduction

• • •

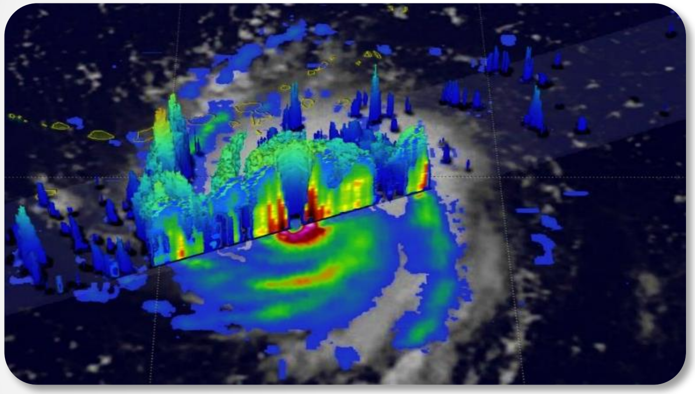
javier.puignavarro@nasa.gov

Coordinated Mobility

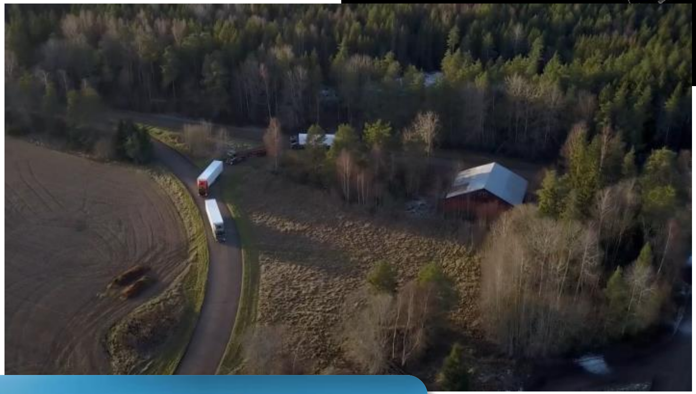
- **Ubiquity** of coordinated mobility solutions
- **Often resort to humans** to intuitively control

Air traffic control

Acquisition of correlated data



Package delivery

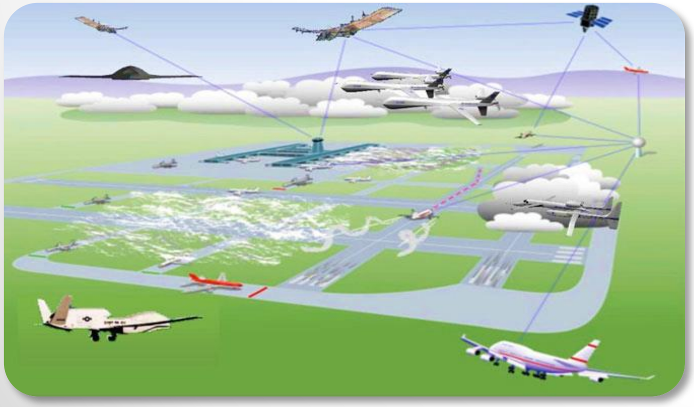


credit Scania Group

credit justbring.com

Ground traffic

credit senseable city lab, MIT

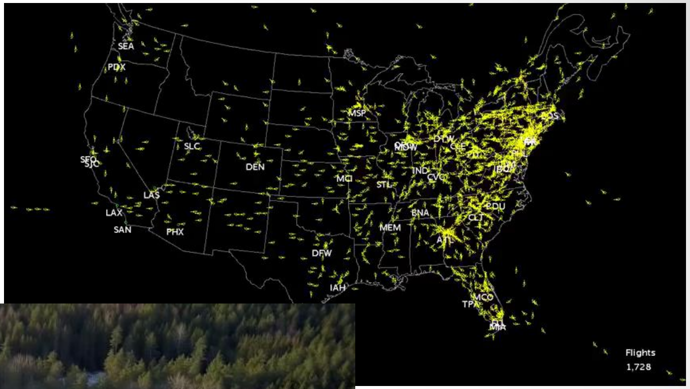


Integration of UAS in the NAS

Coordinated Mobility

- **Ubiquity** of coordinated mobility solutions
- **Often resort to humans** to intuitively control
  - Reduced capacity & efficiency
  - Human errors
- **Goal: develop algorithms that**
  - Automate **multi-agent mobility**
  - Provide **robust path planning** and **execution**
  - Capture all types of **coordination** and **time-varying temporal** constraints of interest

Air traffic control



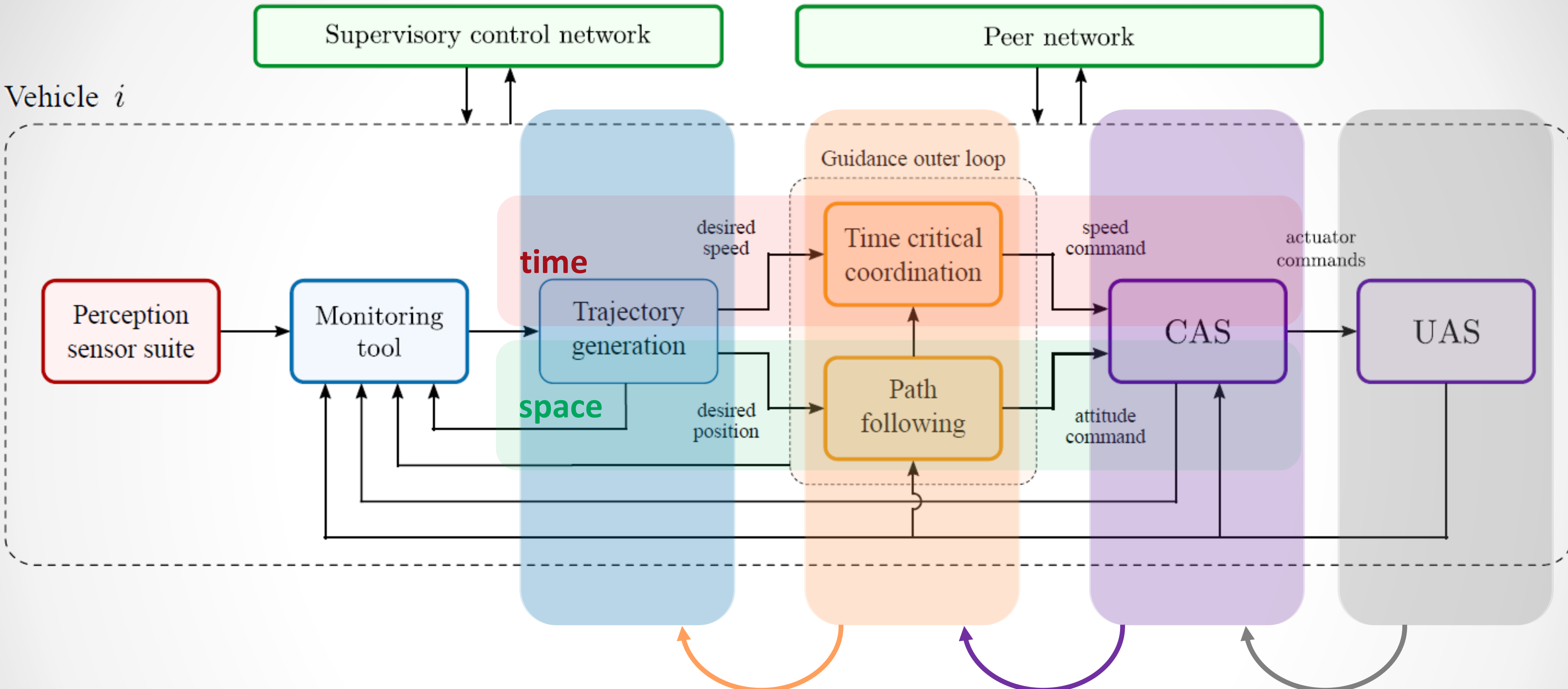
Package delivery

credit Scania Group

Ground traffic

credit senseable city lab, MIT

Architecture



1) Modular cascaded architecture
2) Decouple space & time when possible
3) Provide transient and steady-state guarantees

$$\|\epsilon_i(t)\| = k_o\|\epsilon_i(t_o)\|e^{-\lambda_i\,(t-t_o)} + \beta_i$$

Trajectory Planning

• • •

Proximity Queries

javier.puignavarro@nasa.gov

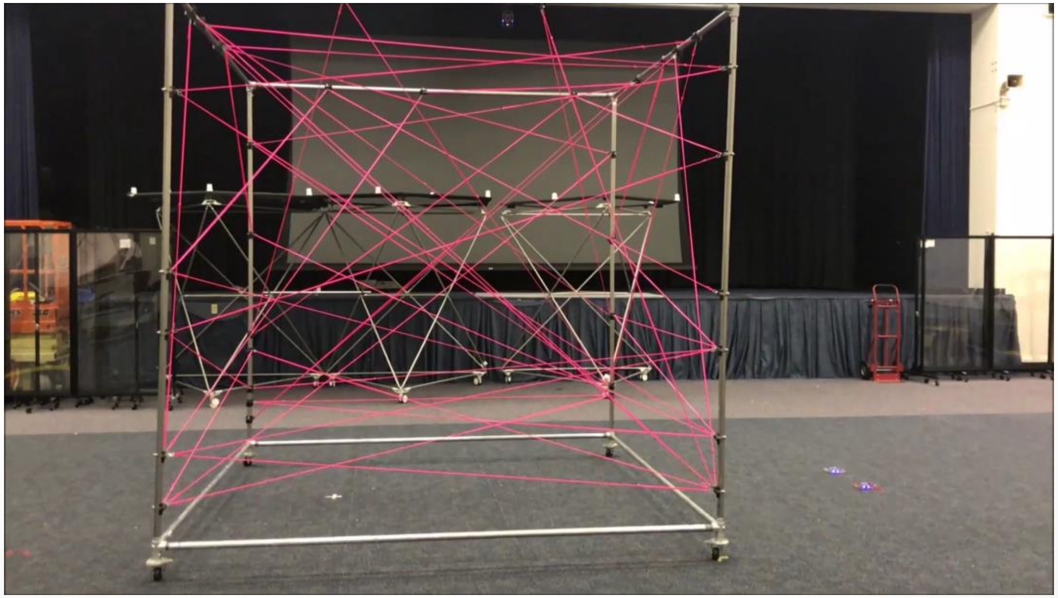
Problem Formulation

- Given a configuration space $\mathcal{X} \in \mathbb{R}^3$, an obstacle region $\mathcal{X}_{obs} = \cup_{i=1}^{n_o} \mathcal{O}_i$, and a safety distance d_s, we define the **unsafe region**

$$\mathcal{X}_{us} = \{\boldsymbol{x} \in \mathcal{X} \mid \|\boldsymbol{x} - \boldsymbol{y}\| < d_s, \forall \boldsymbol{y} \in \mathcal{X}_{obs}\}$$

- The trajectory generation algorithm aims to generate a path through the **flight-safe region** $\mathcal{X}_s = \mathcal{X} \setminus \mathcal{X}_{us}$ as a function of time

$$\boldsymbol{p_d} : t \mapsto \mathcal{X}_s$$



Cluttered environment – the wire maze

javier.puignavarro@nasa.gov

Problem Formulation

- Given a configuration space $\mathcal{X} \in \mathbb{R}^3$, an obstacle region $\mathcal{X}_{obs} = \cup_{i=1}^{n_o} \mathcal{O}_i$, and a safety distance d_s, we define the **unsafe region**

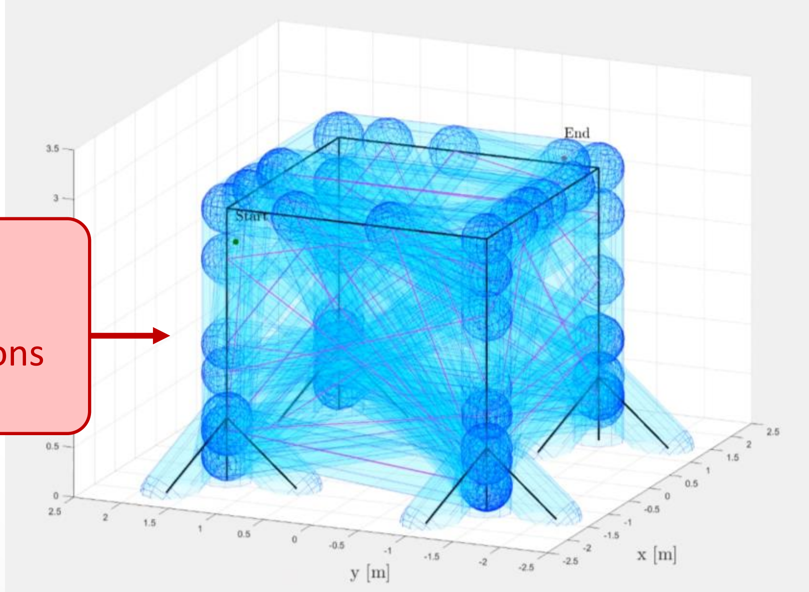
$$\mathcal{X}_{us} = \{\boldsymbol{x} \in \mathcal{X} \mid \|\boldsymbol{x} - \boldsymbol{y}\| < d_s, \forall \boldsymbol{y} \in \mathcal{X}_{obs}\}$$

- The trajectory generation algorithm aims to generate a path through the **flight-safe region** $\mathcal{X}_s = \mathcal{X} \setminus \mathcal{X}_{us}$ as a function of time

$$\boldsymbol{p_d} : t \mapsto \mathcal{X}_s$$

Safety distance

- Accounts for **uncertainty**
- **Trajectory-based** operations
- Very **narrow passages**



Unsafe regions around wires

Problem Formulation

- Given a configuration space $\mathcal{X} \subset \mathbb{R}^3$, an obstacle region $\mathcal{X}_{obs} = \cup_{i=1}^{n_o} \mathcal{O}_i$, and a safety distance d_s, we define the **unsafe region**

$$\mathcal{X}_{us} = \{\boldsymbol{x} \in \mathcal{X} \mid \|\boldsymbol{x} - \boldsymbol{y}\| < d_s, \forall \boldsymbol{y} \in \mathcal{X}_{obs}\}$$

- The trajectory generation algorithm aims to generate a path through the **flight-safe region** $\mathcal{X}_s = \mathcal{X} \setminus \mathcal{X}_{us}$ as a function of time

$$\boldsymbol{p_d} : t \mapsto \mathcal{X}_s$$

Proximity queries are the **backbone** of path-planning

subject to additional **constraints**:

1. Starts and finishes at designated points

$$\boldsymbol{p_d}(t_{init}) = \boldsymbol{x_{init}}\,, \qquad \boldsymbol{p_d}(t_{goal}) = \boldsymbol{x_{goal}}$$

2. Arbitrary speed profile with polynomial structure

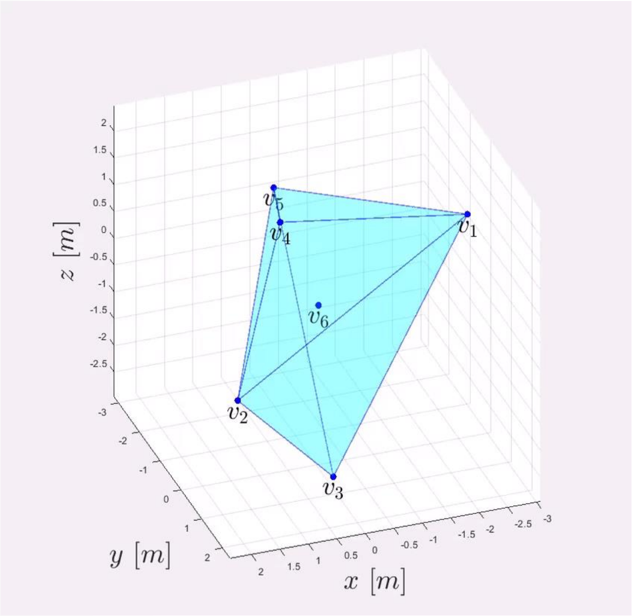
$$v_d(t) = \left\| \frac{d\,\boldsymbol{p_d}}{d\,t} \right\|$$

3. Twice continuous differentiable $\boldsymbol{p_d}(t) \in \mathcal{C}^2$

- Answer questions about the **relative position** between two geometric objects

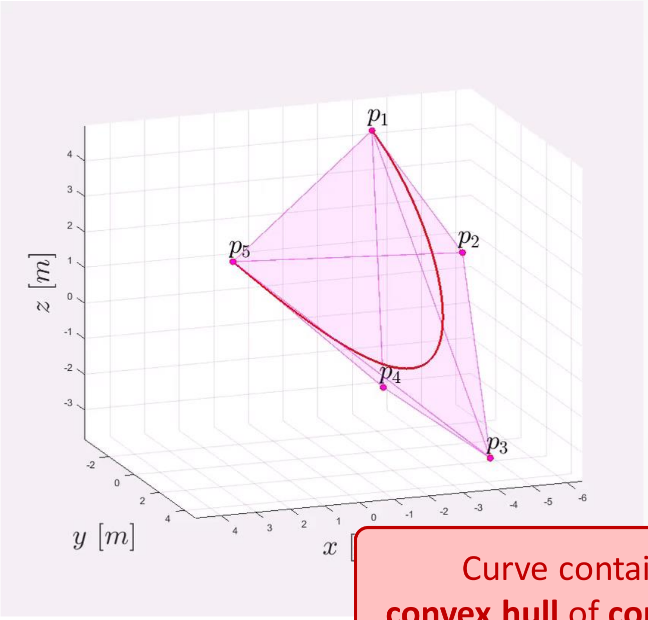
Polytopes

*Environmental **obstacles***



Polynomial curves

*Vehicle **trajectories***



Curve contained in **convex hull** of **control points**

- Polynomial curves of degree n expressed in a **Bernstein basis**

$$\boldsymbol{p_d}(\zeta) = \sum_{i=0}^{n} \boldsymbol{p_i} b_i^n(\zeta) \qquad b_i^n = \binom{n}{i}(1-\zeta)^{n-i}\zeta^i \qquad \zeta \in [0,1]$$

Control points

Proximity Queries - Algorithms

- Given the $\mathcal{V}$ representation of a convex polytope $\mathcal{O}$, and a polynomial parametric curve $\boldsymbol{p_d}(\zeta):[0,1]\rightarrow\mathbb{R}^3$ of degree n, we define the **distance** between $\mathcal{O}$ and $\boldsymbol{p_d}$ as

$$d\left(\mathcal{O},\boldsymbol{p_d}\right)=\min\left(\|\boldsymbol{y}-\boldsymbol{x}\|\mid\boldsymbol{y}\in\mathcal{O}\,,\,\boldsymbol{x}\in\boldsymbol{p_d}\right)$$

- We focus the study of proximity queries on **3 algorithms**

Collision	Tolerance Verification (TV)	Distance
Is $d>0$?	Is $d>d_s$?	What is the value of d?
Traditionally used in robotics • *No uncertainty* • *Slower than TV*	*Reduced computational cost* • ***Uncertainty*** • ***Faster than collision***	*Only used if distance needed* • *e.g. potential field* • *slowest algorithm*

- **1988 - Gilbert Johnson & Keerthi (GJK)** pioneered queries **between polytopes**
- **2011 - Chan** proposes a distance method for Bézier curves

New Tolerance Verification Algorithms

- The result of the **Tolerance Verification algorithm**, must return a binary answer

$$v_t = \begin{cases} 1, & \text{if } d(\mathcal{O}, \boldsymbol{p_d}) > d_s \\ 0, & \text{if } d(\mathcal{O}, \boldsymbol{p_d}) \leq d_s \end{cases}$$

Safe

Unsafe

where d_s is a **safety distance**

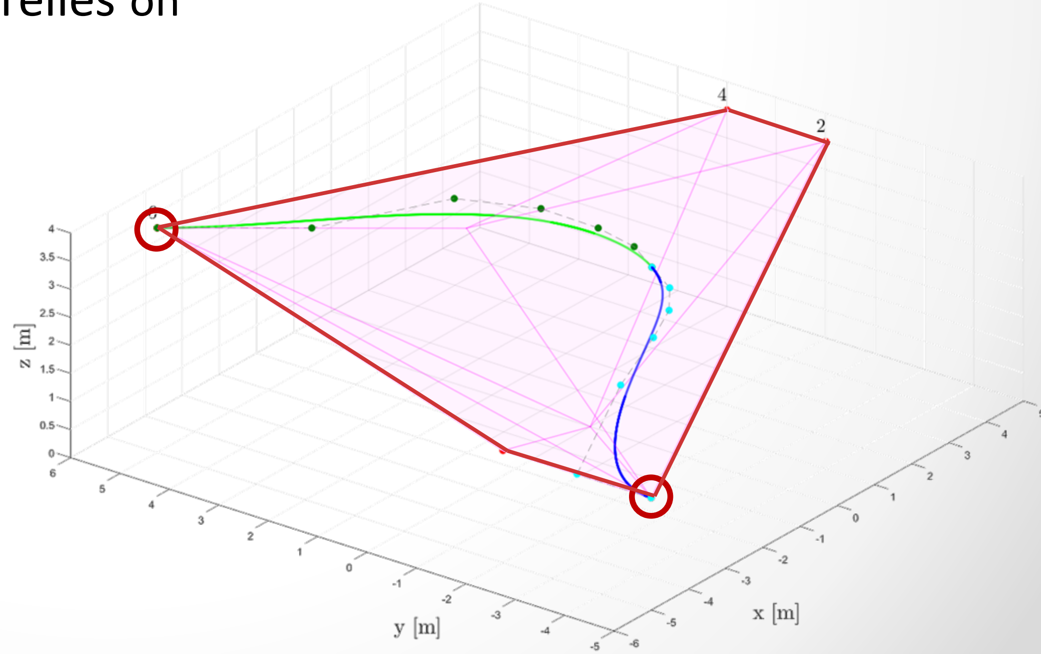
- Tolerance verification algorithm relies on

Bounding region of Bézier curve

1st and nth control points

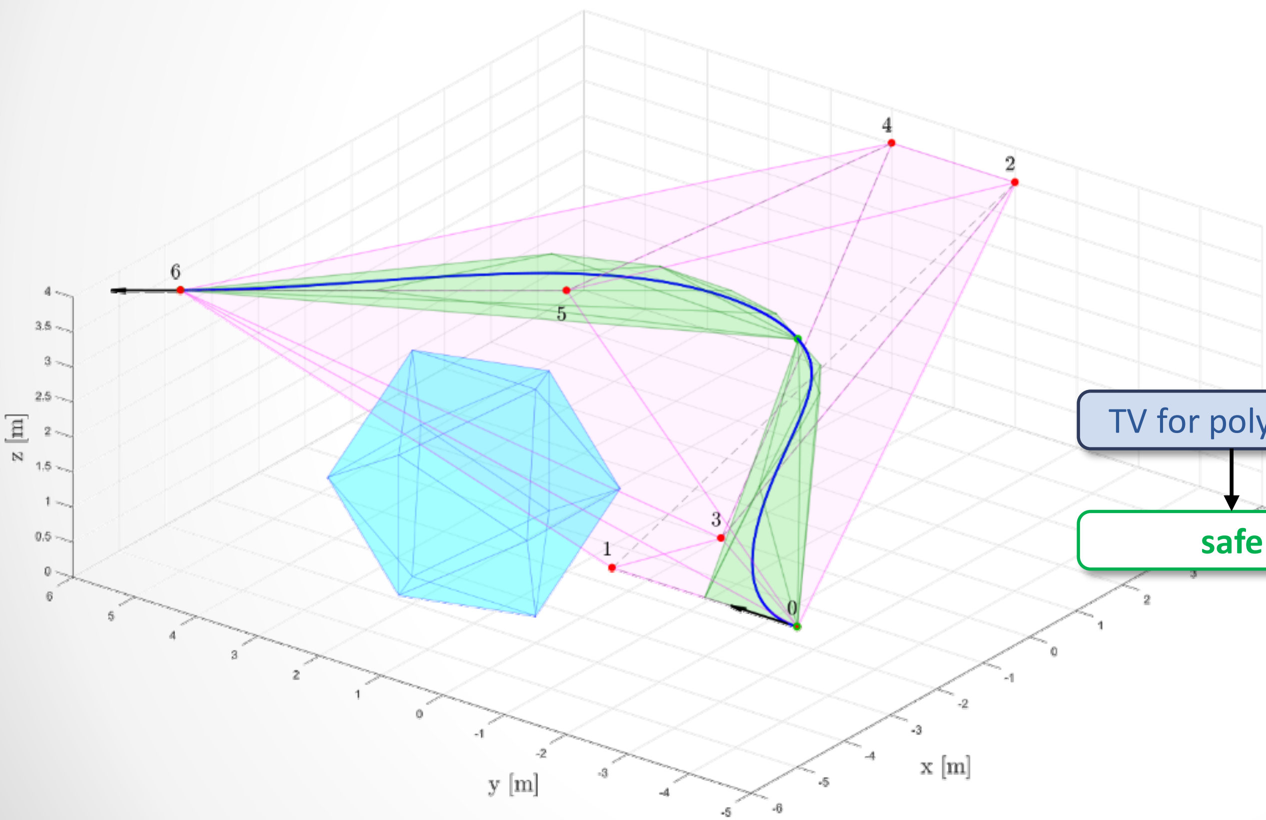
De Casteljau's algorithm

Modification of GJK for TV



Tolerance Verification - Safe Path

$$d(\mathcal{O}, \boldsymbol{p}) > d_s$$



TV for polytopes

↓

de Casteljau

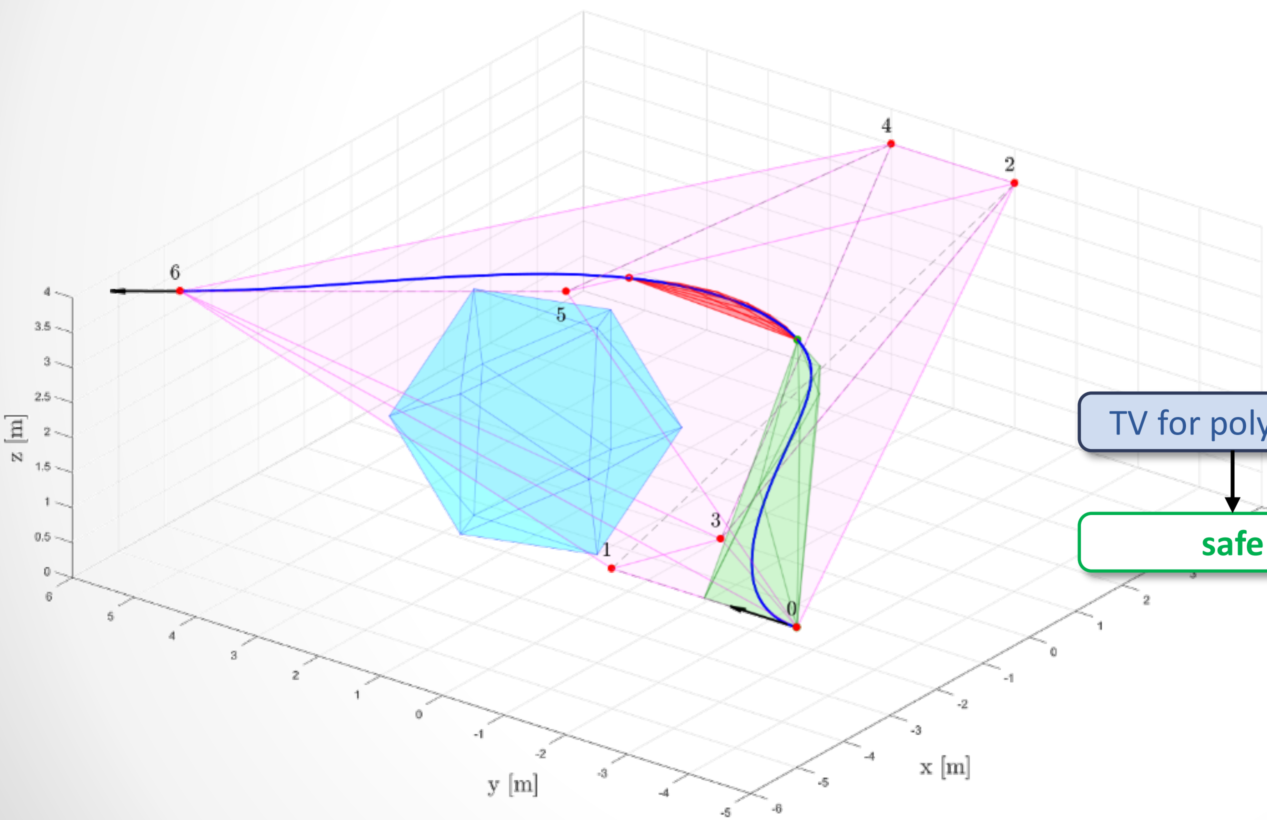
↙ ↘

TV for polytopes → **safe**

TV for polytopes → **safe**

Tolerance Verification - Unsafe Path

$$d(\mathcal{O}, \boldsymbol{p}) \leq d_s$$



TV for polytopes

de Casteljau

TV for polytopes

TV for polytopes

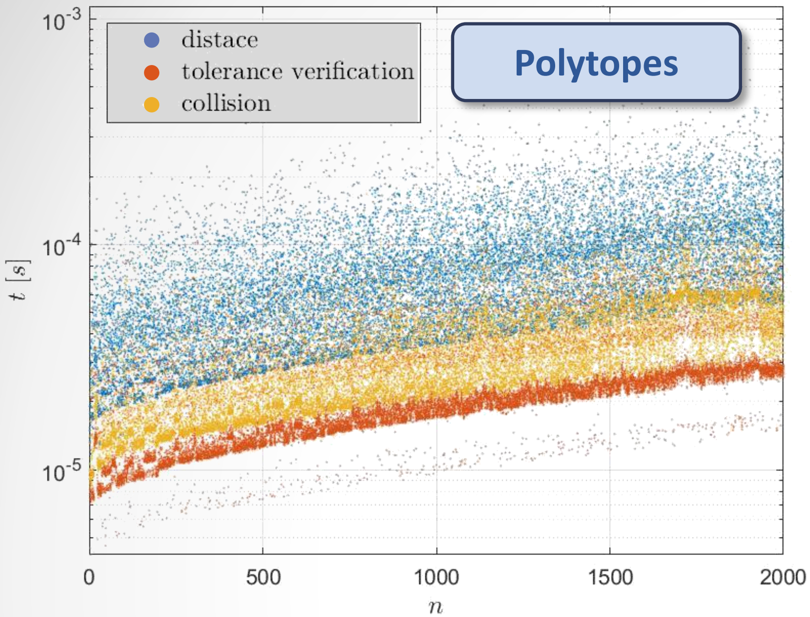
safe

de Casteljau

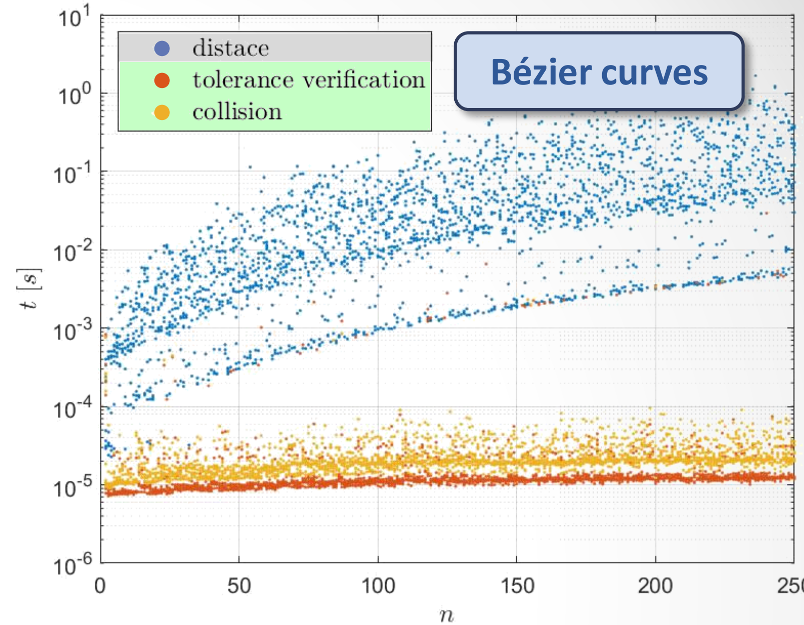
unsafe

ILLINOIS

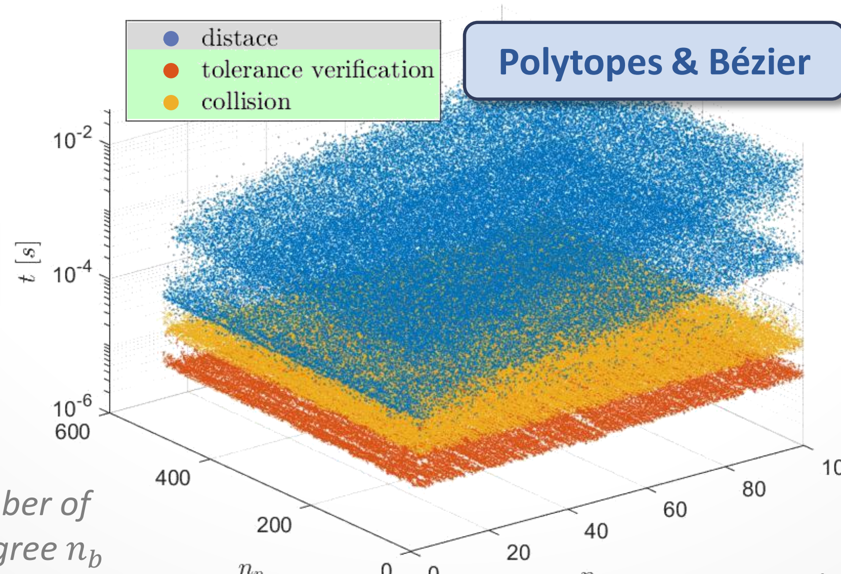
Speed Tests



Runtime of PQ with the number of vertices



Runtime of PQ with curve degree



Runtime PQ with number of vertices n_p & curve degree n_b

new algorithms
existing algorithms

Tolerance verification queries are **faster**

javier.puignavarro@nasa.gov

Trajectory Planning

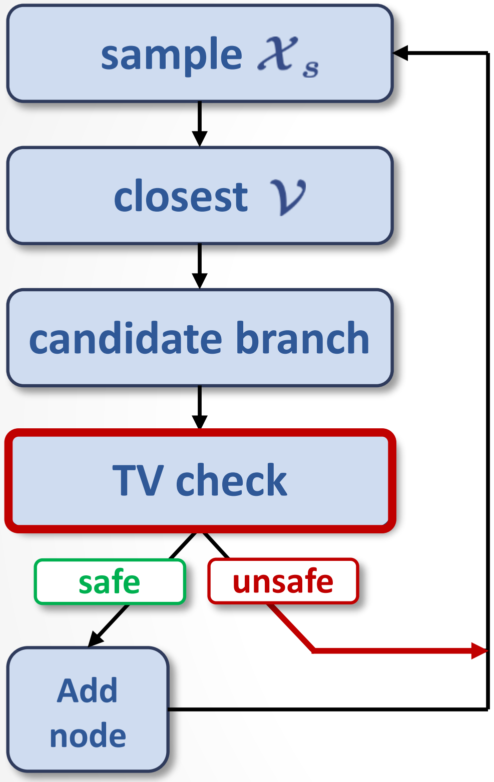
• • •

Silhouette-informed Trees

javier.puignavarro@nasa.gov

Rapidly-expanding Random Trees (RRT*)

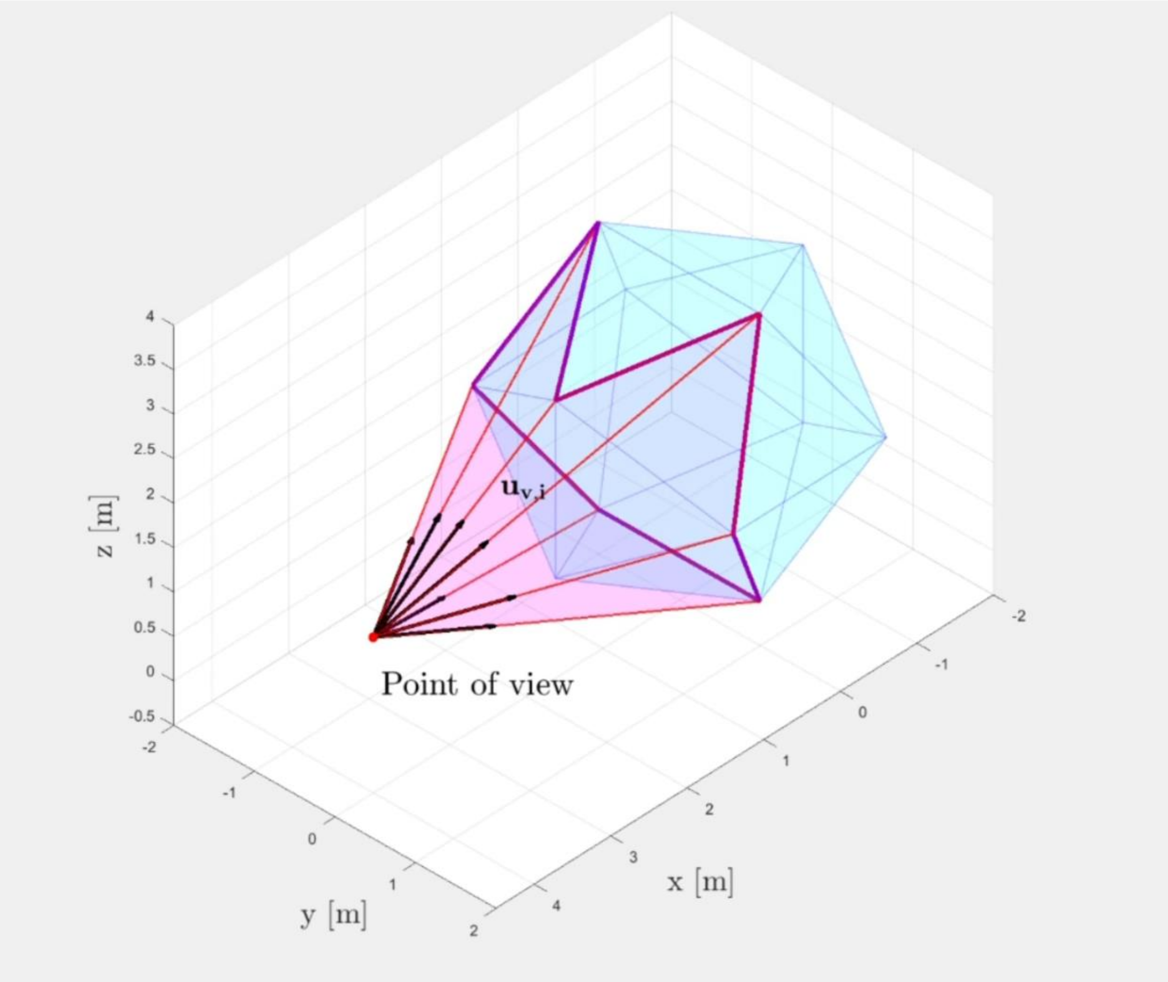
- Builds a **tree** $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ through the safe configuration space $\mathcal{X}_s$



- Probabilistic **completeness**
- Rewiring of tree ensures **asymptotic optimality**
- Replaced **collision** queries by **tolerance verification**
- If branch candidate **fails** → sample again

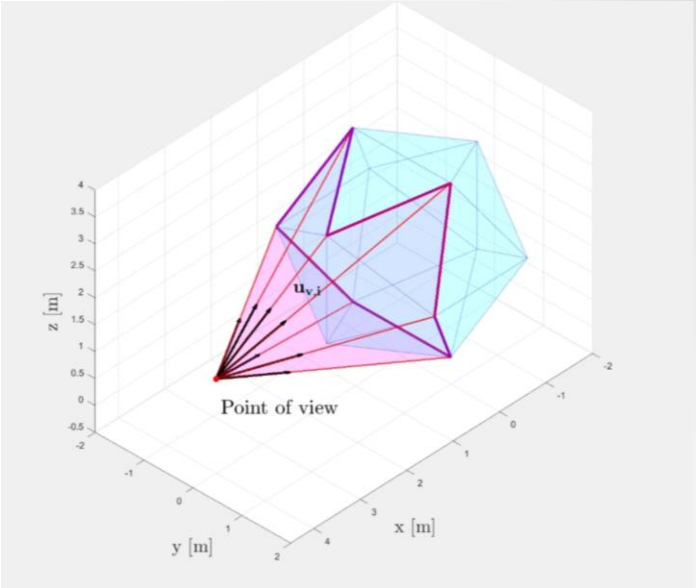
Silhouette

- The **silhouette** of a convex polyhedron $\mathcal{O}$ as seen from a point of view $\boldsymbol{p_v}$ is the closed sequence of edges that describes the boundary between the facets that are visible from $\boldsymbol{p_v}$ and the ones that are not

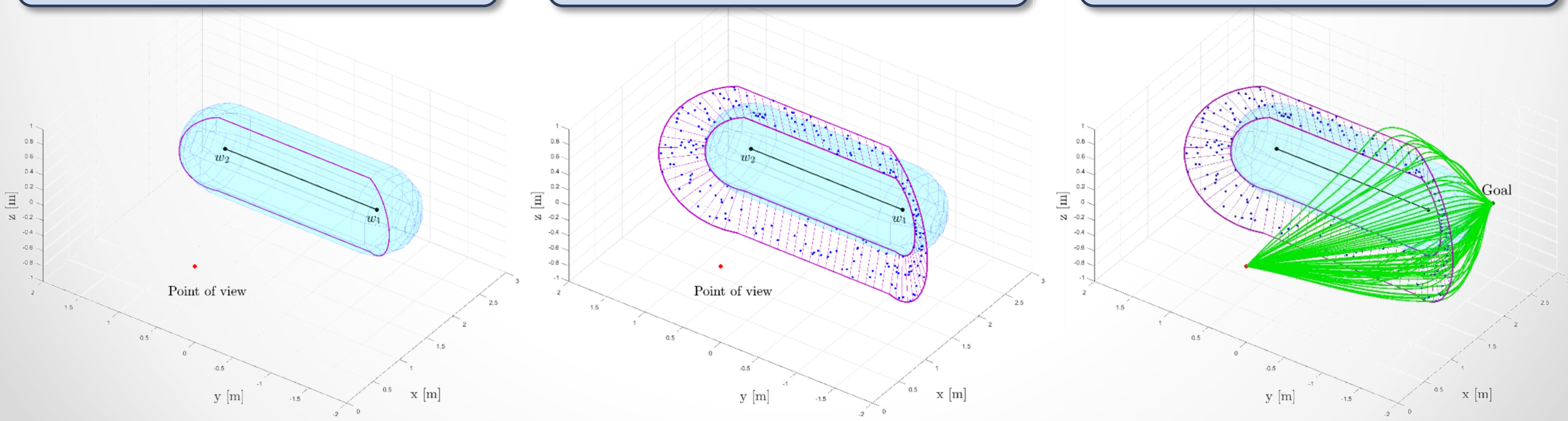


Silhouette

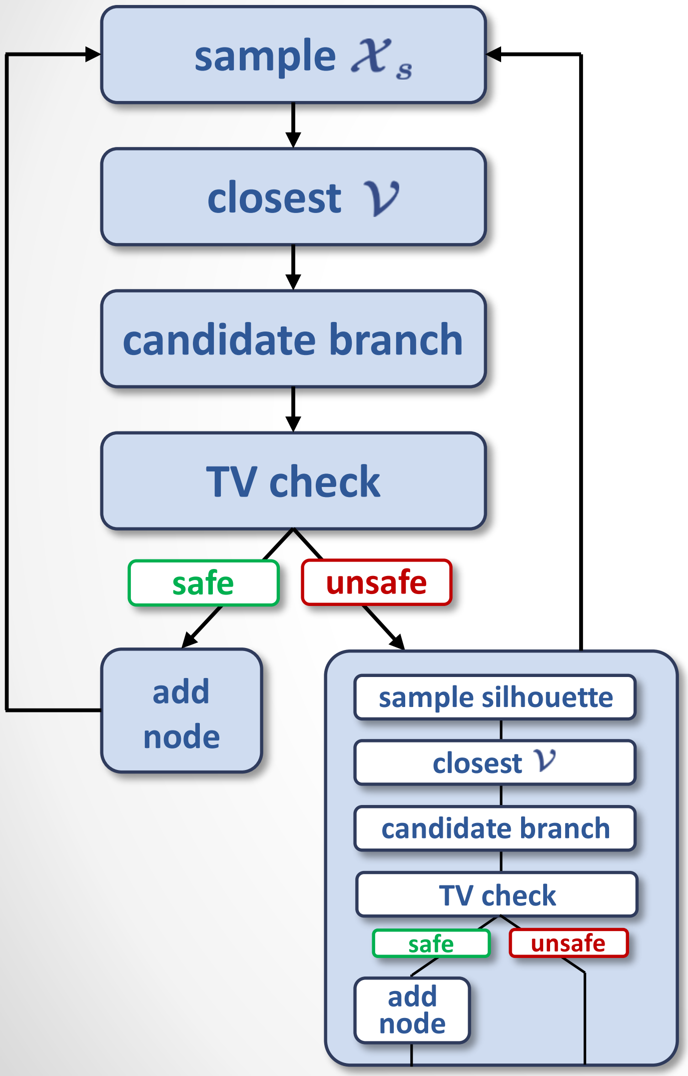
- Leverage the silhouette to **generate paths**
- Attention mechanism to focus on challenging areas around obstacles such as **narrow passages**
- Potential for planning with **partial information**
  - The silhouette is the **visible boundary** of an obstacle



Compute silhouette | Expand and sample | Generate paths



- Builds a **tree** $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ through the safe configuration space $\mathcal{X}_s$



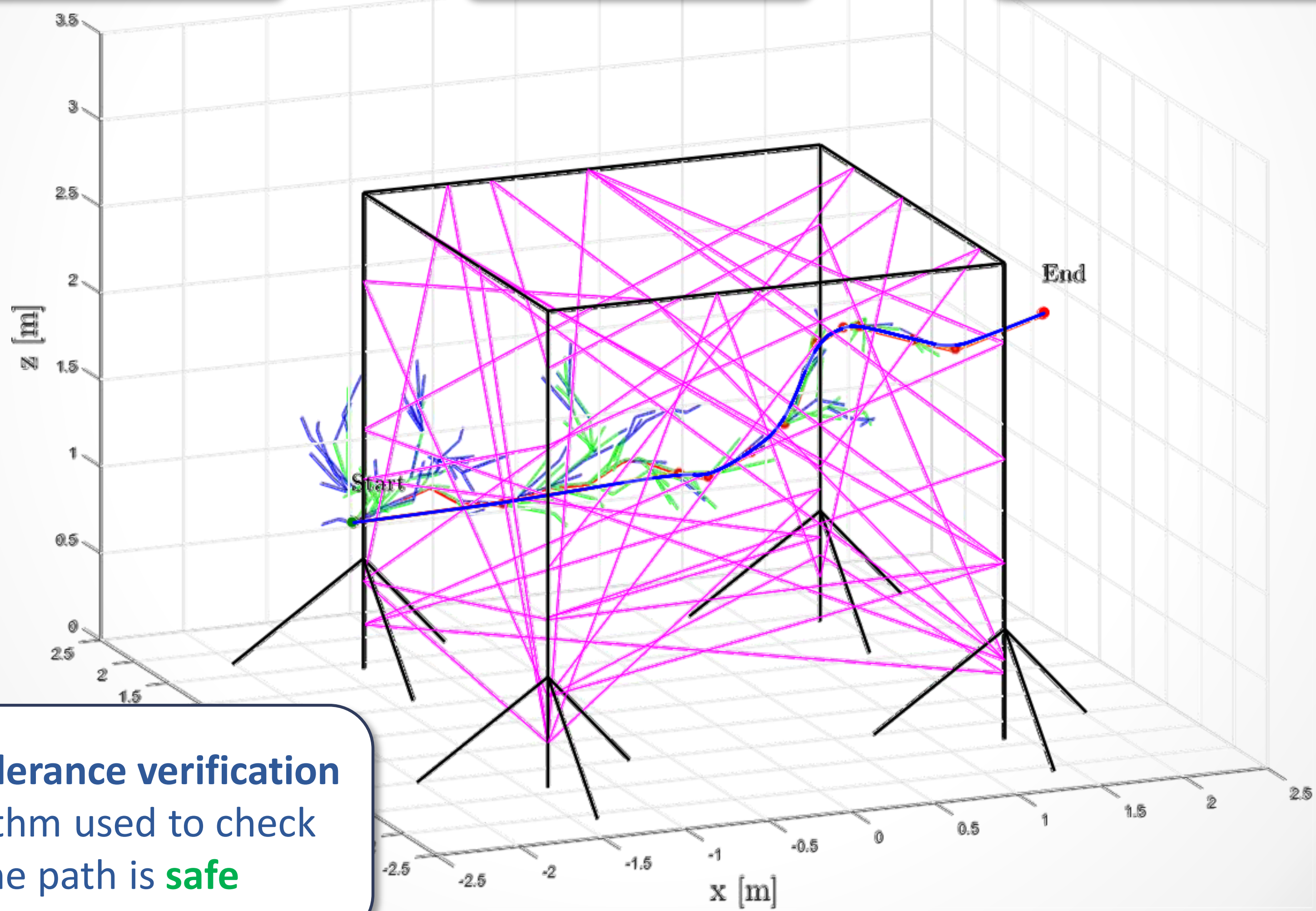
- Informs point sampling with silhouette if candidate branch fails a safety-distance check, capturing
  - Problematic obstacle
  - Local geometry of the obstacle
- **Improves narrow passage behavior**

javier.puignavarro@nasa.gov

Iterative Path Smoothing

- Given a tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ with a set of edges $\tilde{\mathcal{E}}$ that connects x_{init} and x_{goal}

Extract $\tilde{\mathcal{E}}$ → **Edge reduction** → **Smoothing**

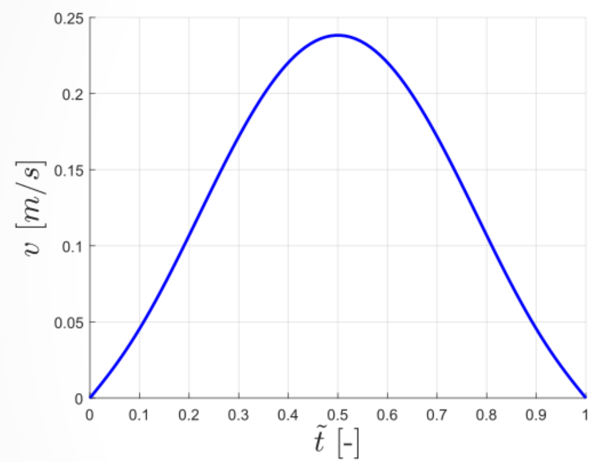


New **tolerance verification** algorithm used to check the path is **safe**

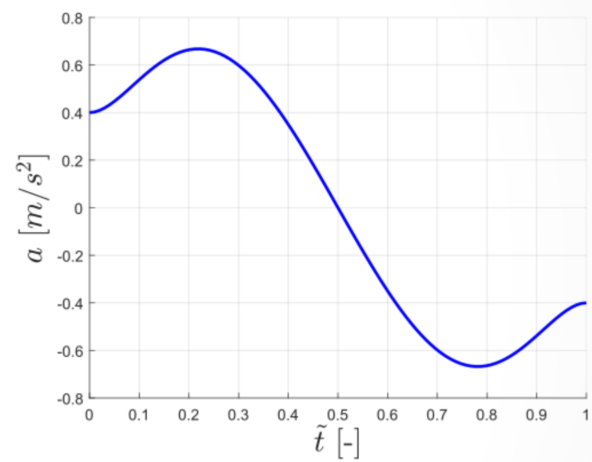
Temporal specifications

- Assume a polynomial speed profile $v(\tilde{t})$ is provided as a function of a normalized time $\tilde{t}$

$$v(\tilde{t}) = \sum_{i=0}^{n} v_k b_i^n(\tilde{t}) \,, \qquad \tilde{t} \in [0,1] \,, \qquad b_i^n(\tilde{t}) = \binom{n}{i} (1-\tilde{t})^{n-i} \, \tilde{t}^{\,i}$$



Speed



Acceleration

- Integrating the parametric speed $\sigma(\zeta)$ over the curve parameter ζ, or the speed profile $v(\tilde{t})$ over the normalized time $\tilde{t}$ provides the arc-length

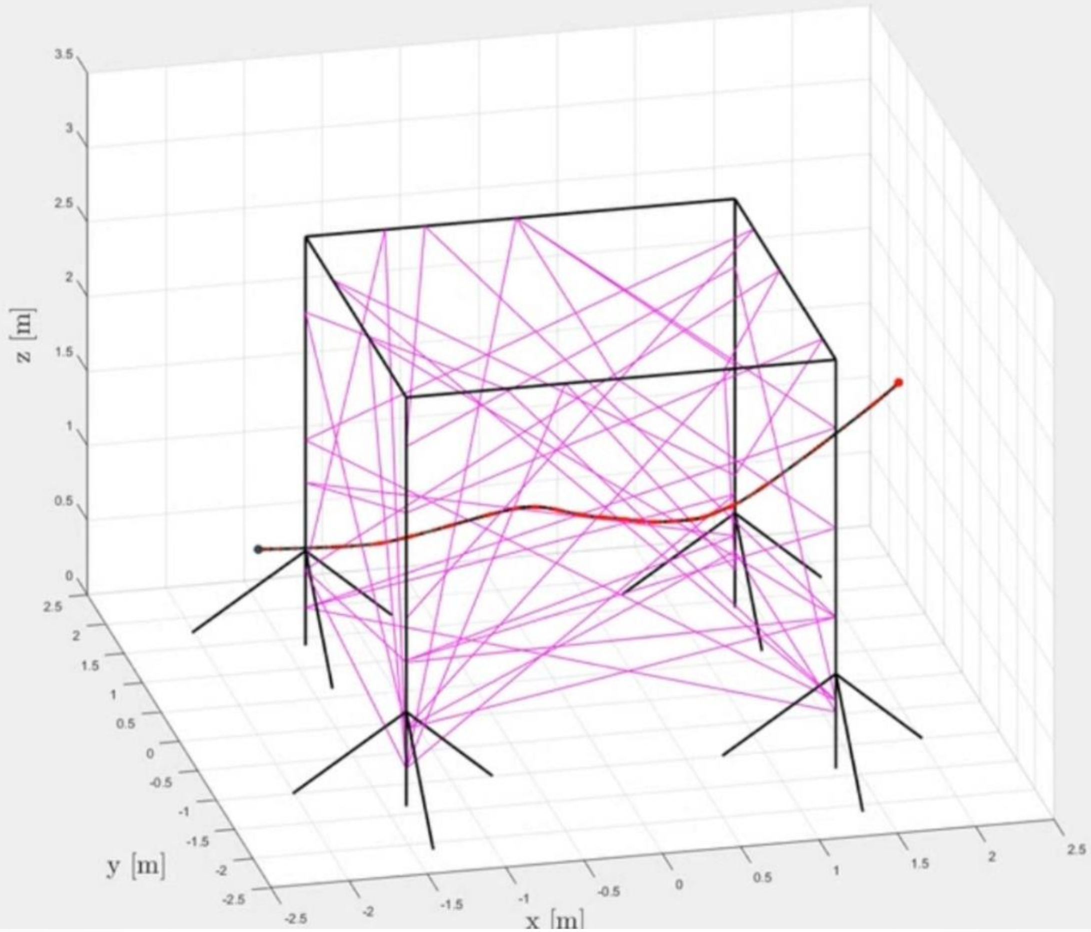
$$\int_0^{\zeta} \sigma(\epsilon)\, \mathrm{d}\epsilon = \int_0^{\tilde{t}} v(\tau)\, \mathrm{d}\tau$$

- $\sigma(\zeta) \in \mathbb{R}^+$
- $\int \sigma(\epsilon) \mathrm{d}\epsilon$ monotonically increasing
- Single root for every $\tilde{t} \in [0,1]$

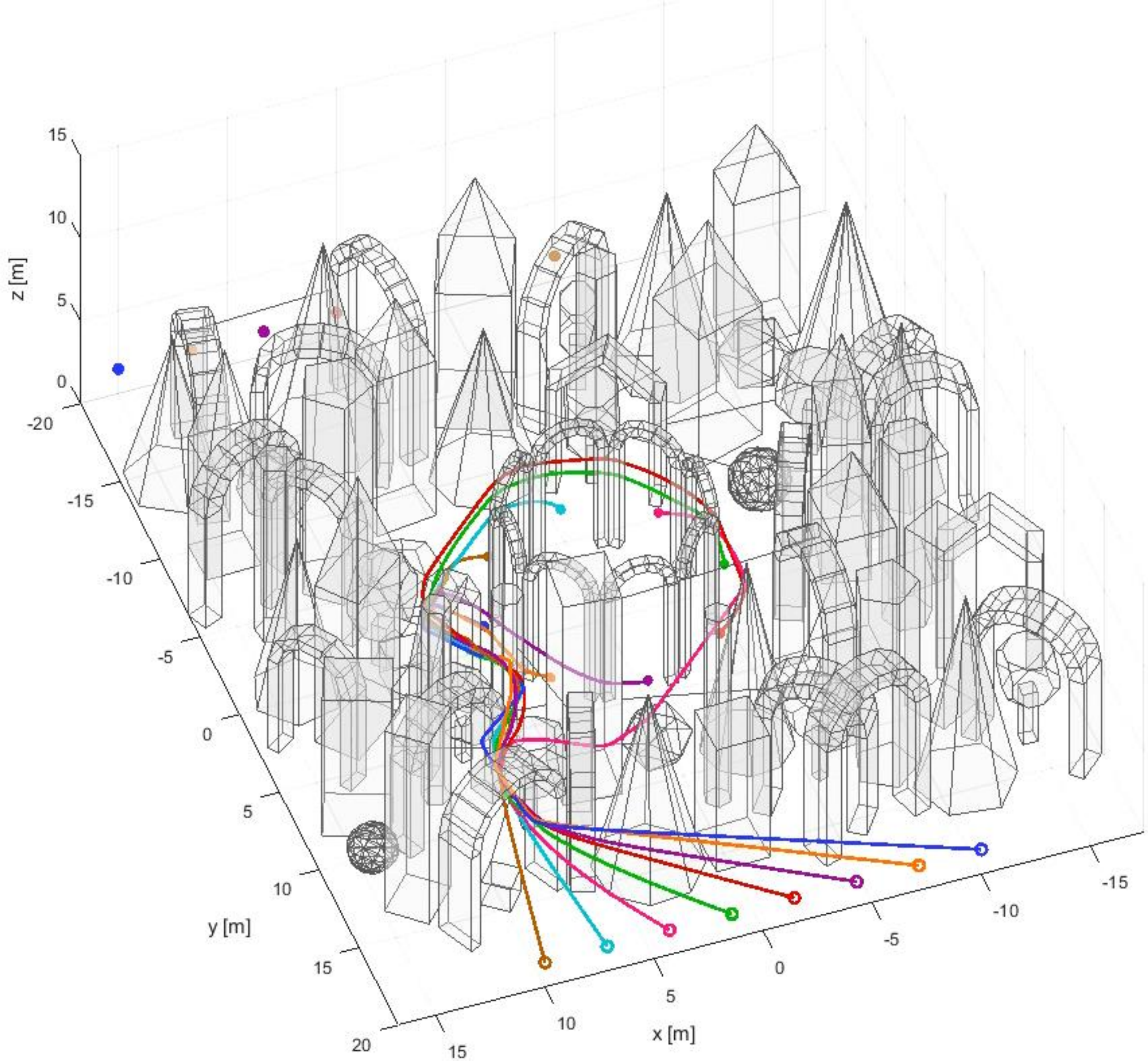
- This defines a relationship between ζ and $\tilde{t}$

javier.puignavarro@nasa.gov

Temporal Specifications



Multiple Vehicles



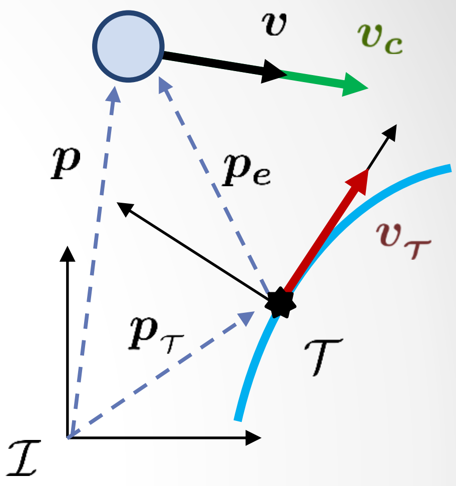
Time-coordinated execution

• • •

Path-following for multirotors

- **Position error dynamics** for a vehicle with holonomic constraints

$$\dot{\boldsymbol{p}}_{\boldsymbol{e}} = \boldsymbol{v} - \boldsymbol{v}_{\mathcal{T}}$$



- Then, the **speed command**

$$\boxed{\boldsymbol{v_c} = \boldsymbol{v}_{\mathcal{T}} - k_{p_e}\boldsymbol{p_e}}$$

ensures the **origin of the position error dynamics is GUES**

- **Non-ideal inner-loop controller**

1) Assume the speed tracking error $\boldsymbol{v_e} := \boldsymbol{v} - \boldsymbol{v_c}$ is a piecewise continuous, bounded function of t for all $t \geq 0$ with

$$\boxed{\delta_v(t) := \sup_{\tau\in[0,t]} \|\boldsymbol{v_e}(t)\|}$$

Time-coordinated execution

• • •

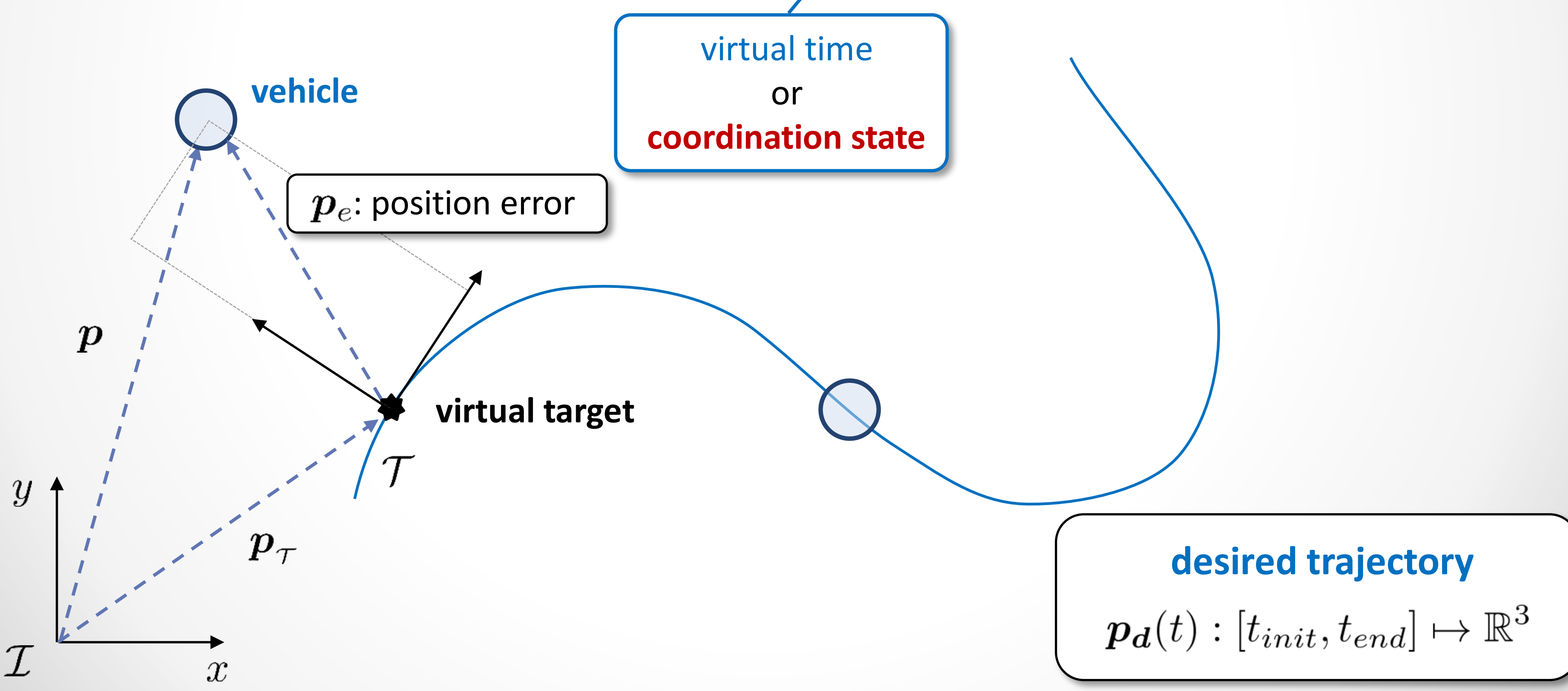
Distributed time-critical coordination

javier.puignavarro@nasa.gov

Virtual Target

- Defines the **desired location** on the path at time t
- Can **control speed** of virtual target as it slides along the path → **adds a DoF**
  - The location of the virtual target is given by the virtual time ξ

$$\boldsymbol{p}_{\mathcal{T}}(t) = \boldsymbol{p_d}(\xi(t)) \qquad \boxed{\xi} \in [t_{init}, t_{end}]$$

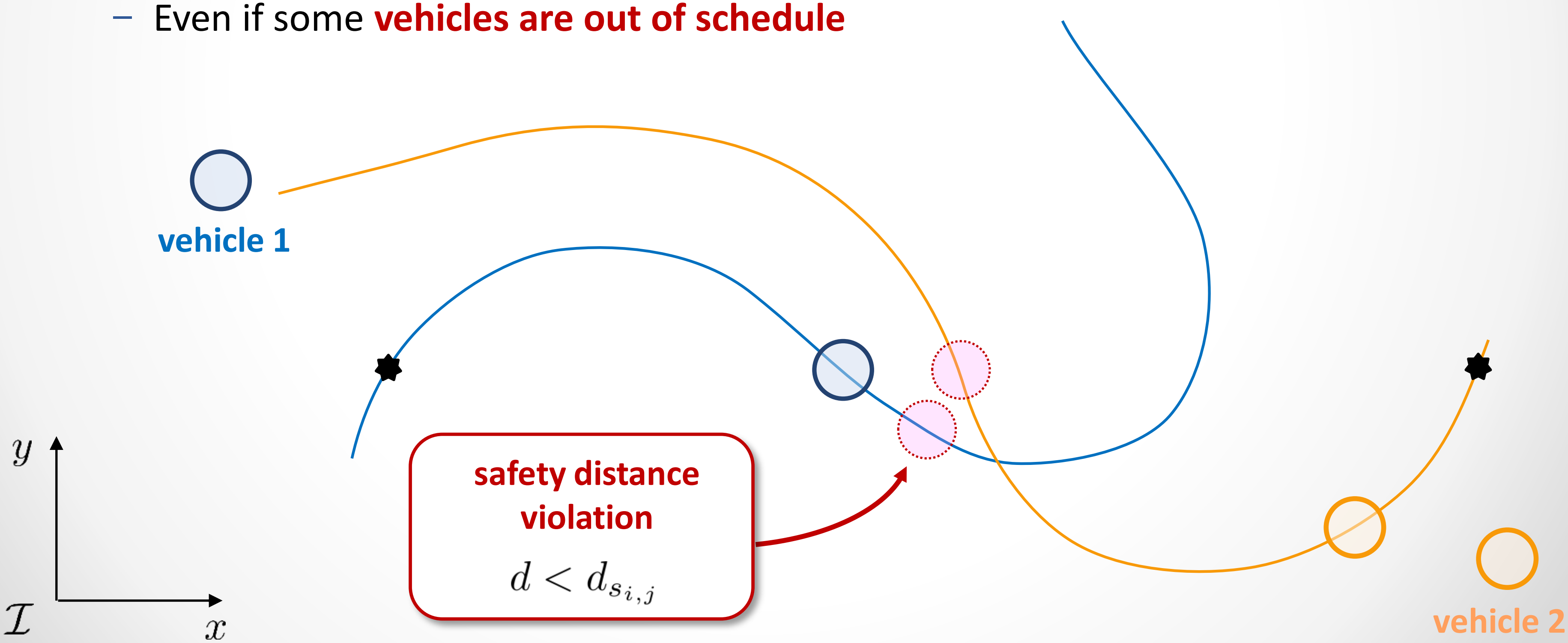


- Results in **1D coordination dynamics**
  - Given a fleet of n vehicles with coordinated trajectories $\boldsymbol{p_{d,i}}(t), \; i \in \{1, \ldots, n\}$
  - Deconflicted in time

safety distance

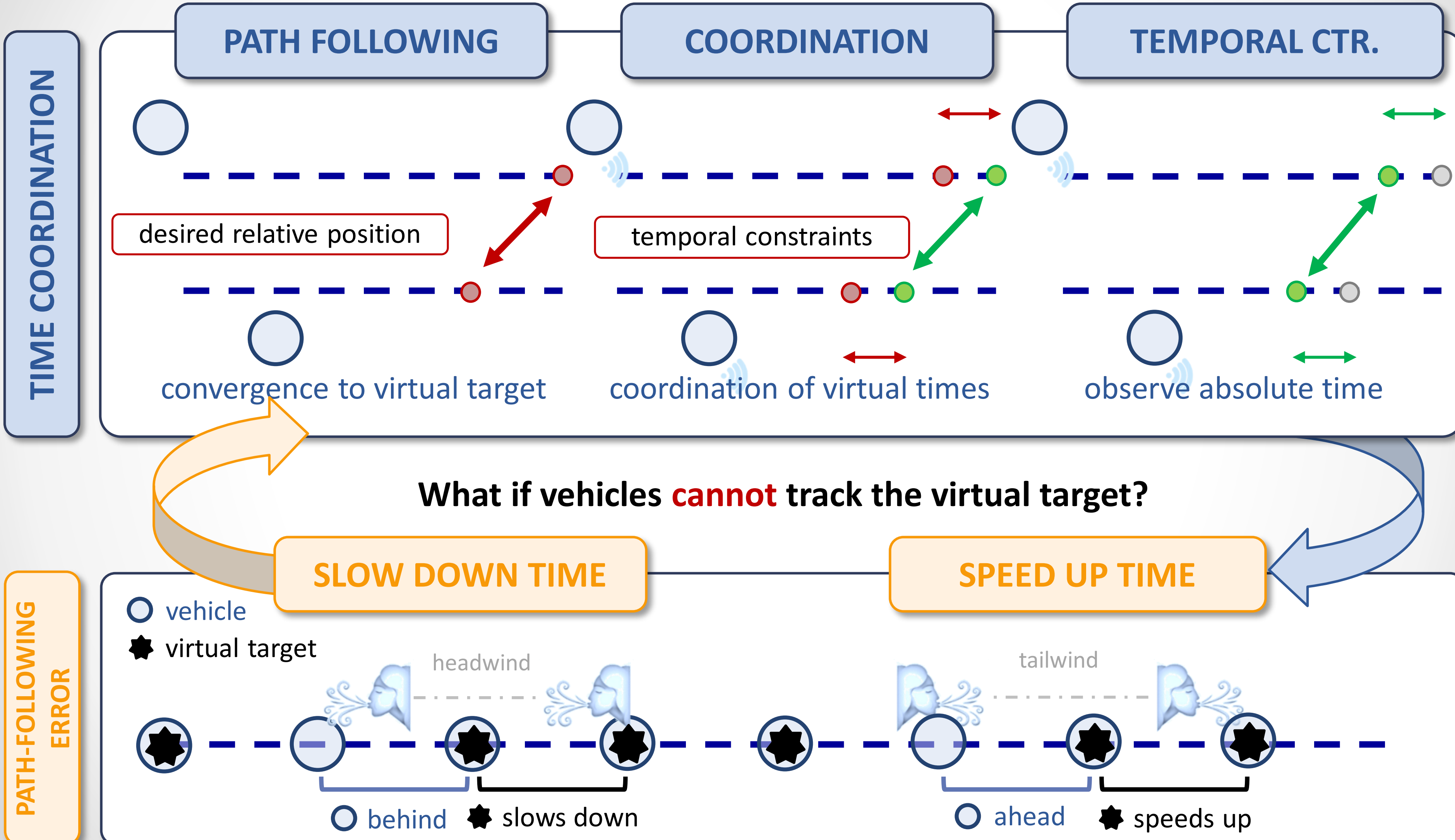
$$\|\boldsymbol{p_{d,i}}(t) - \boldsymbol{p_{d,j}}(t)\| \geq \boxed{d_{s_{i,j}}}, \quad \forall i \neq j$$

- Leverage the coordination states ξ_i to maintain coordination online
  - Even if some **vehicles are out of schedule**



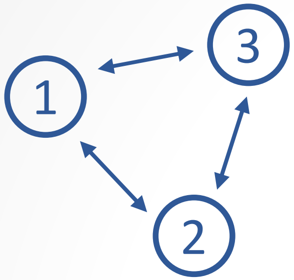
Decoupling Space & Time

- **Simplicity and flexibility → facilitates a wide range of temporal and coordination ctr.**



Problem Formulation

- The vehicles **exchange** their coordination states ξ_i **over a time-varying network**



$$\mathcal{G}(t) := (\mathcal{V}, \mathcal{E}(t)) \begin{cases} \mathcal{V}: & \text{vertices} \\ \mathcal{E}(t): & \text{time-varying edges} \end{cases}$$

- Introduce **reference coordination state** ξ_R that provides an absolute time

$$\boxed{\dot{\xi}_R = \rho}$$

and constant $\rho = 1$ if the mission is to be executed as planned

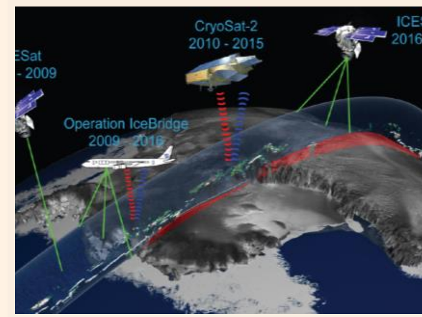
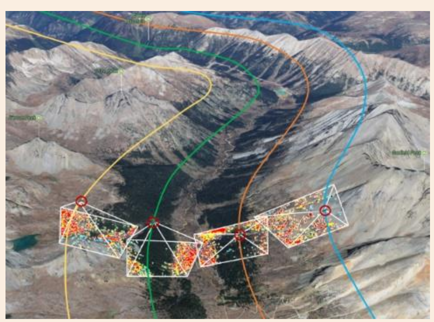
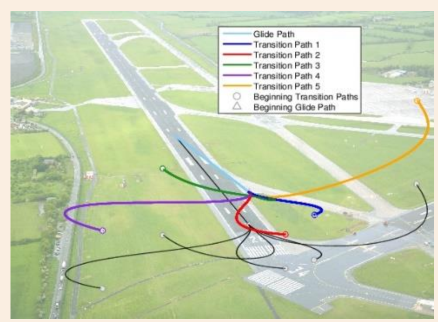
- **Control objective:** design a distributed control law that asymptotically solves

Relative position	$\xi_i(t) - \xi_j(t)$	$\xrightarrow{t \to \infty}$	$[-\Delta_c(t), \Delta_c(t)]$
Temporal constraints	$\xi_i(t) - \xi_R(t)$	$\xrightarrow{t \to \infty}$	$[-\Delta_a(t), \Delta_a(t)]$
Speed	$\dot{\xi}_i(t) - \dot{\xi}_R$	$\xrightarrow{t \to \infty}$	0

where Δ_c and Δ_a define the **coordination** and **temporal windows**, respectively.

Time-Critical Coordination Strategies

Temporal Constraints

Coordination	**Unenforced** $\Delta_a \to \infty$ **no arrival time**	**Relaxed** $0 < \Delta_a < \infty$ arrival **window** around **nominal time**	**Strict** $\Delta_a = 0$ **precise arrival time**
Tight $\Delta_c = 0$ **close proximity precision**			
Loose $\Delta_c > 0$ **mid-proximity operations**		 	

Distributed Control Law – Tight Coordination

Distributed consensus | **Path following Err.**

Link peers

Temporal link-weights

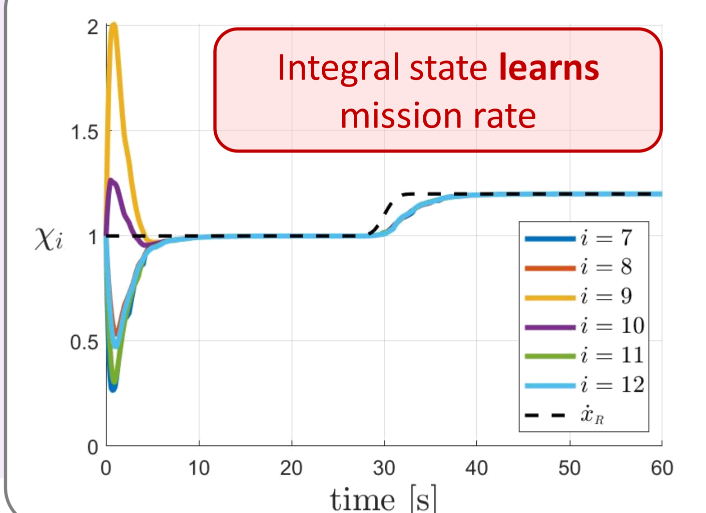
$$\dot{\xi}_i(t) = \underbrace{-k_P \sum_{j\in\mathcal{N}_i} (\xi_i - \xi_j)}_{\text{Coordination}} \underbrace{- k_R \boxed{\omega_{R_i}} (\xi_i - \xi_R)}_{\text{Temporal Constraints}} + \underbrace{\dot{\xi}_R}_{\text{Rate}} - k_{LT} \frac{\boldsymbol{p_{e,i}} \cdot \hat{\boldsymbol{t}}_i}{\|\boldsymbol{v_{d,i}}(\xi_i)\| + \epsilon}$$

End peers

$$\dot{\xi}_i(t) = \underbrace{-k_P \sum_{j\in\mathcal{N}_i} (\xi_i - \xi_j)}_{\text{Coordination}} + \underbrace{\chi_i}_{\text{Integral state}}$$

$$\dot{\chi}_i(t) = -k_I \sum_{j\in\mathcal{N}_i} (\xi_i - \xi_j)$$

$$\chi_i(0) = \chi_{i0}$$



javier.puignavarro@nasa.gov

Distributed Control Law – Tight Coordination

Distributed consensus

Path following Err.

Link peers

$$\dot{\xi}_i(t) = -\underbrace{k_P \sum_{j\in\mathcal{N}_i}(\xi_i-\xi_j)}_{\text{Coordination}} - \underbrace{k_R\,\omega_{R_i}(\xi_i-\xi_R)}_{\text{Temporal Constraints}} + \underbrace{\dot{\xi}_R}_{\text{Rate}} - k_{LT}\frac{\boldsymbol{p}_{e,i}\cdot\hat{\boldsymbol{t}}_i}{\|\boldsymbol{v}_{d,i}(\xi_i)\|+\epsilon}$$

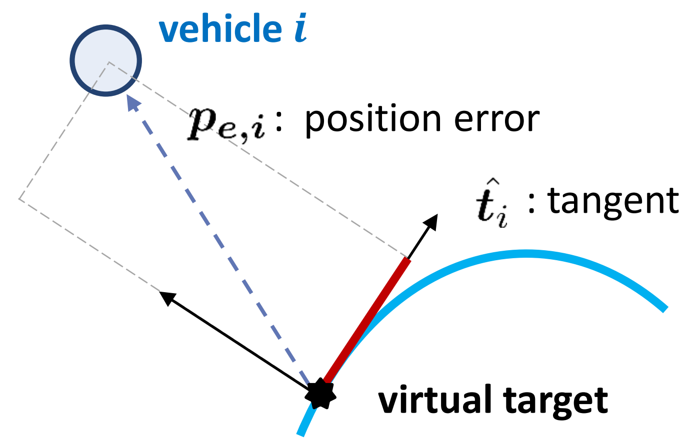
End peers

$$\dot{\xi}_i(t) = -\underbrace{k_P \sum_{j\in\mathcal{N}_i}(\xi_i-\xi_j)}_{\text{Coordination}} + \underbrace{\chi_i}_{\text{Integral state}}$$

$$\dot{\chi}_i(t) = -k_I \sum_{j\in\mathcal{N}_i}(\xi_i-\xi_j)$$

$$\chi_i(0) = \chi_{i0}$$

Long-track error normalized by desired speed (units of s)



Theoretical Results

• • •

Tight Coordination

Network Assumptions

2) Each agent can **only** exchange information with a set of **neighboring agents** $\mathcal{N}_i(t)$

3) Communications among agents are **bidirectional**, and information is transmitted **continuously with no delays**

4) **Inter-vehicle communications** network at time t satisfies:

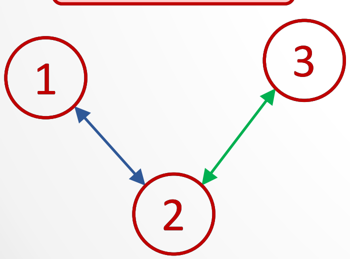
- Parameters $\boldsymbol{T}$ and $\boldsymbol{\mu}$ characterize the **Quality of Service (QoS)** of the network
- Graph **connected in an integral sense**, not piecewise in time

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

$\boldsymbol{L}(t)$: Laplacian matrix of $\mathcal{G}(t)$

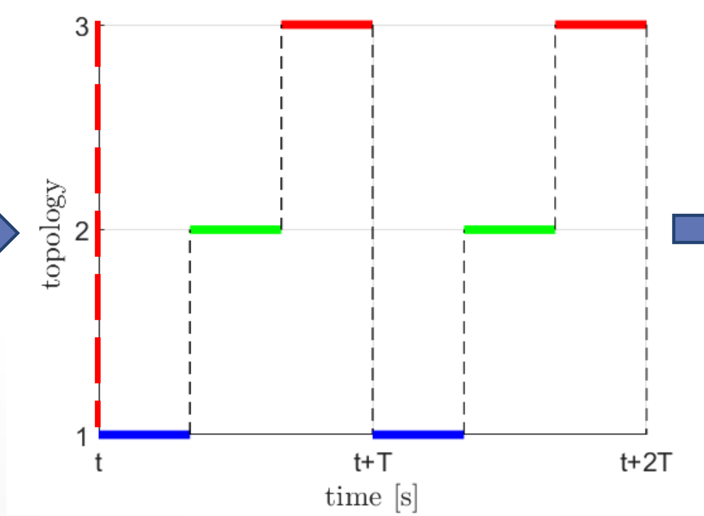
$\boldsymbol{Q}$ eliminates the 0 eigenvalue of $\boldsymbol{L}(t)$

Topology 3



Graph **fully** connected

Graph disconnected



$$\frac{1}{n}\frac{1}{T}\int_t^{t+T} \boldsymbol{Q}\boldsymbol{L}(\tau)\boldsymbol{Q}^\top d\tau \geq 0.111\ \mathbb{I}_2$$

$$T = 1\,s$$

$$\boldsymbol{Q} = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

javier.puignavarro@nasa.gov

Unenforced Temporal Constraints

Theorem 1 *Assume $\mathcal{G}(t)$ and the inner-loop controller satisfies Assumptions 1 through 4. Then, there* ... k_P*,* k_I*,* k_{LT} *and* k_{p_e} *such that:*

Deviation from planned time will keep increasing

Temporal ctr	$\lvert\xi_i - \xi_R\rvert \leq \kappa_{\xi_{R_0}} \lVert\zeta(0)\rVert$	$+\ \kappa_{\xi_{R_1}}\, \delta_v\, t,$
Coordination ctr	$\lvert\xi_i - \xi_j\rvert \leq \kappa_{\xi_0} \lVert\zeta(0)\rVert\, e^{-\lambda_u t}$	$+\ \kappa_{\xi_1}\, \delta_v,$
Speed	$\lvert\dot{\xi}_i - \dot{\xi}_R\rvert \leq \kappa_{\dot{\xi}_0} \lVert\zeta(0)\rVert\, e^{-\lambda_u t}$	$+\ \kappa_{\dot{\xi}_1}\, \delta_v,$

for known constants $\kappa_{\xi_{R_0}}$, $\kappa_{\xi_{R_1}}$, κ_{ξ_0}, ... *teed rate of convergence*

Effects of **path-following error** and **speed tracking error**

$$\lambda_u \geq \frac{k_P n \mu}{(1 + k_P n T)^2} \left(1 + \beta \frac{n}{n_\ell}\right)^{-1}, \qquad \beta \geq 2.$$

Performance improves with $\begin{cases} \uparrow \mu & - \ \textit{integral connectivity} \\ \downarrow T & - \ \textit{temporal window} \end{cases}$

Relaxed Temporal Constraints

Theorem 2 *Assume $\mathcal{G}(t)$ and the inner-loop controller satisfies Assumptions 1 through 4. Then, there exist known control gains* k_P, k_R, k_I, k_{LT}, k_{p_e} τ_1 *such that:*

Provides **performance requirements** for inner-loop algorithms $\kappa_{\xi_{R1}} \delta_v \leq \Delta_a$

Temporal ctr	$\lvert \xi_i - \xi_R \rvert \leq \max\left(\Delta_a, k_{\xi_{R1}}\,\delta_v\right), \ \forall t \geq \bar{t},$
Coordination ctr	$\lvert \xi_i - \xi_j \rvert \leq \kappa_{\xi_0} \lVert \zeta(0) \rVert\, e^{-\lambda_r t} + \kappa_{\xi_1}\,\delta_v,$
Speed	$\lvert \dot{\xi}_i - \dot{\xi}_R \rvert \leq \kappa_{\dot{\xi}_0} \lVert \zeta(0) \rVert\, e^{-\lambda_r t} + \kappa_{\dot{\xi}_1}\,\delta_v,$

for known constants $\bar{t}$, $\kappa_{\xi_{R_1}}$, κ_{ξ_0}, κ_{ξ_1}, *rate of convergence*

Effects of **path-following error** and **speed tracking error**

$$\lambda_r \geq \frac{k_P n \mu}{(1 + (k_P n + k_R) T)^2} \left(1 + \beta\left(1 + \frac{n_\ell}{n} f_R\right)\right)^{-1}, \ \beta \geq 2,$$

where

Consistent results: - **performance** improves with $\uparrow \mu, \ \downarrow T$
- effect of the reference and time-varying link weighs included through k_R, f_R

Strict Temporal Constraints

Theorem 3 *Assume $\mathcal{G}(t)$ and the inner-loop controller satisfies Assumptions 1 through 4. Then, there exist known control gains k_P, k_R, k_I, k_{LT}, and k_{p_e} such that:*

Temporal ctr	$\lvert x_i - x_R \rvert \leq \kappa_{\xi_{R_0}} \lVert \boldsymbol{\zeta}(0) \rVert \, e^{-\lambda_s t} + \kappa_{\xi_{R_1}} \delta_v,$
Coordination ctr	$\lvert x_i - x_j \rvert \leq \kappa_{\xi_0} \lVert \boldsymbol{\zeta}(0) \rVert \, e^{-\lambda_s t} + \kappa_{\xi_1} \delta_v,$
Speed	$\lvert \dot{x}_i - \rho \rvert \leq \kappa_{\dot{\xi}_0} \lVert \boldsymbol{\zeta}(0) \rVert \, e^{-\lambda_s t} + \kappa_{\dot{\xi}_1} \delta_v,$

for known constants $\kappa_{\xi_{R_0}}$, $\kappa_{\xi_{R_1}}$, κ_{ξ_0}
teed rate of convergence

Effects of **path-following error** and **speed tracking error**

$$\lambda_s \geq \frac{k_P n \mu}{(1 + (k_P n + k_R)T)^2} \left(1 + \beta \left(1 + \frac{n_\ell}{n} f_s\right)\right)^{-1}, \quad \beta \geq 2,$$

Consistent results: - **performance** improves with $\uparrow \mu, \ \downarrow T$
- effect of the reference and constant link weighs included through k_R, f_S

Simulation Results

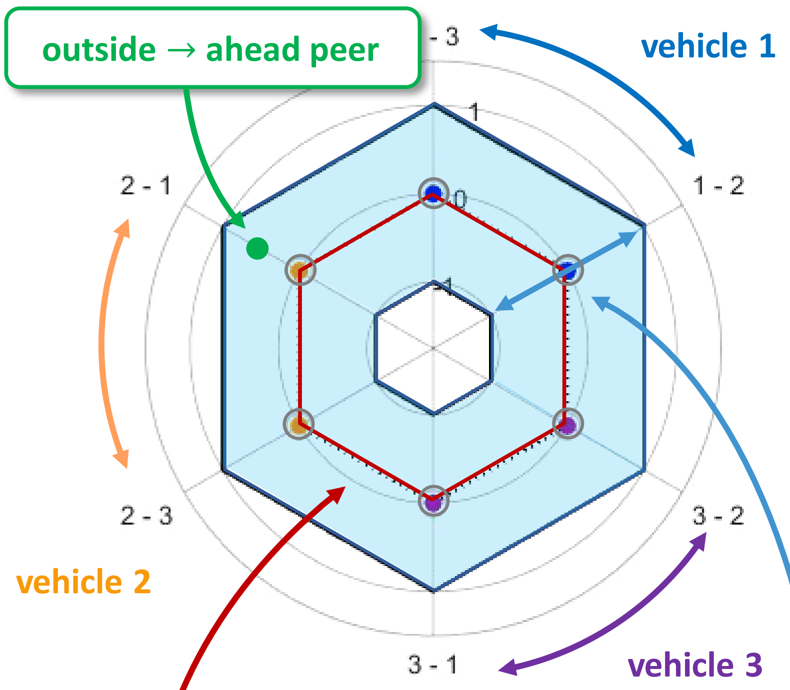
• • •

Loose Coordination

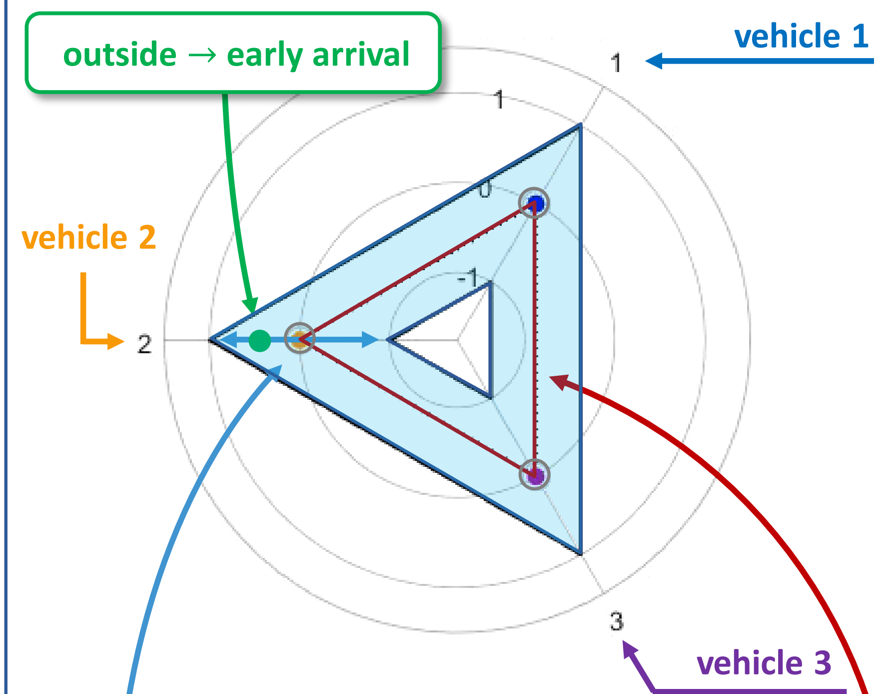
javier.puignavarro@nasa.gov

Visualization of Time-Critical Objectives

Coordination Constraints



Temporal Constraints



Zero polygon

$$\xi_i - \xi_j = 0, \;\; \forall i \neq j$$

Coord. window

$$|\xi_i - \xi_j| < \Delta_c$$

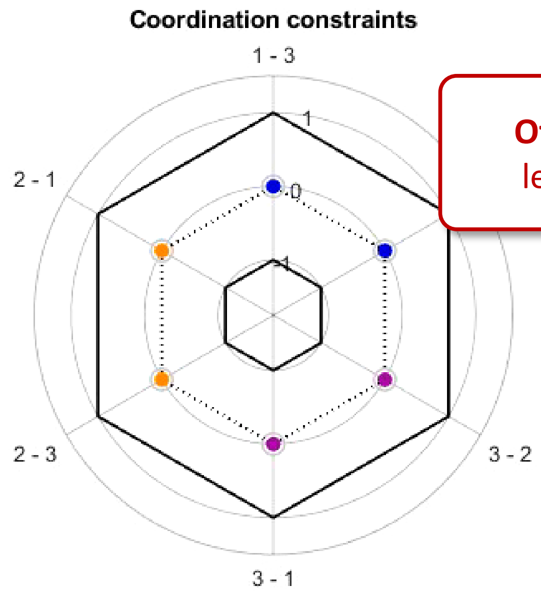
Temporal window

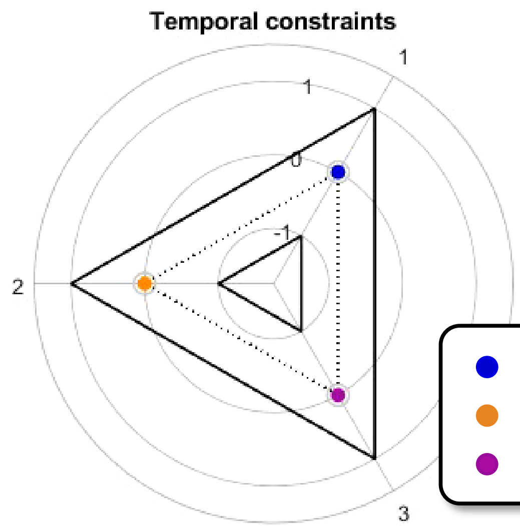
$$|\xi_i - \xi_R| < \Delta_a$$

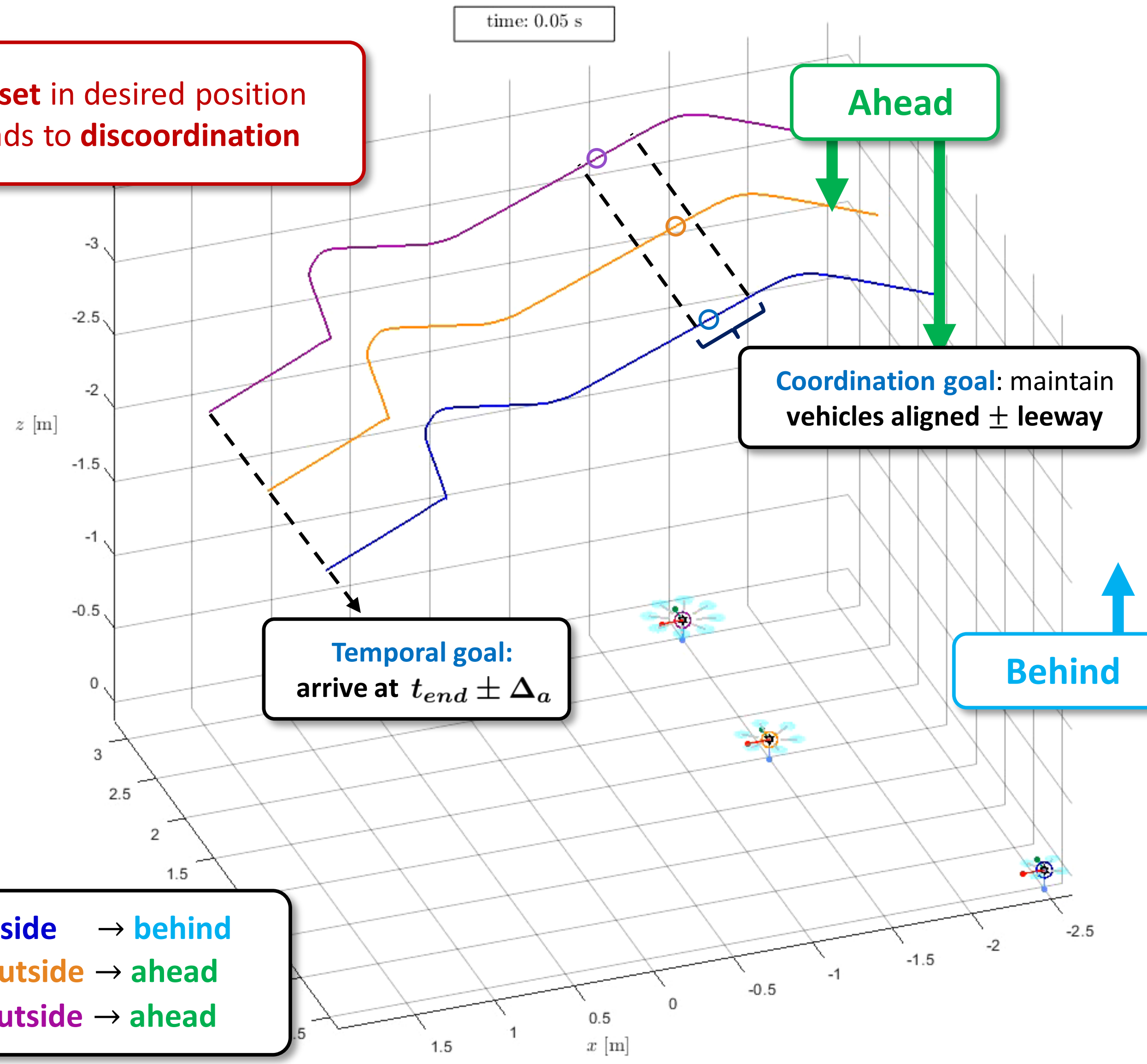
Zero polygon

$$\xi_i - \xi_R = 0, \;\; \forall i$$

Time-Critical Coordination







Offset in desired position leads to **discoordination**

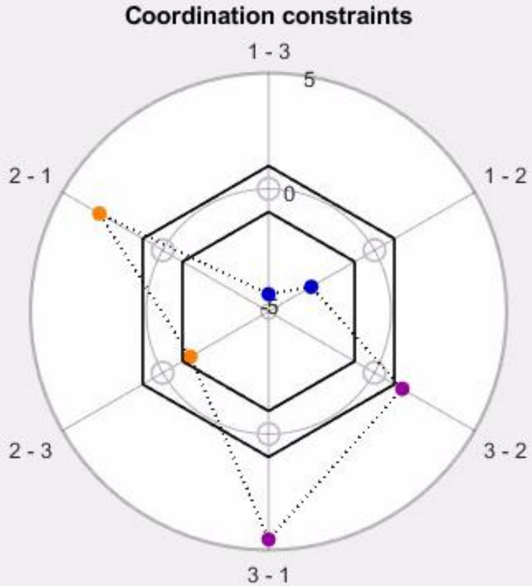
Ahead

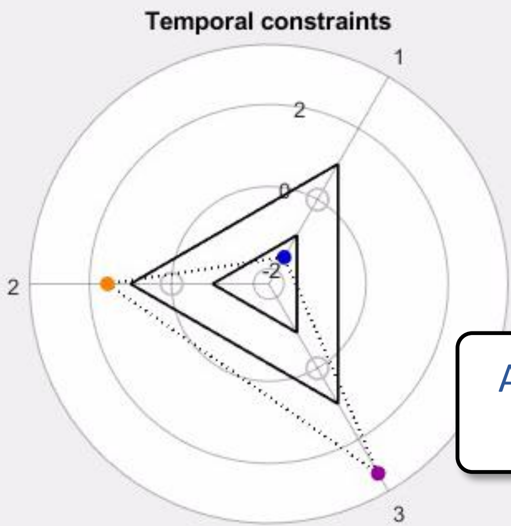
Coordination goal: maintain **vehicles aligned $\pm$ leeway**

Temporal goal: **arrive at $t_{end} \pm \Delta_a$**

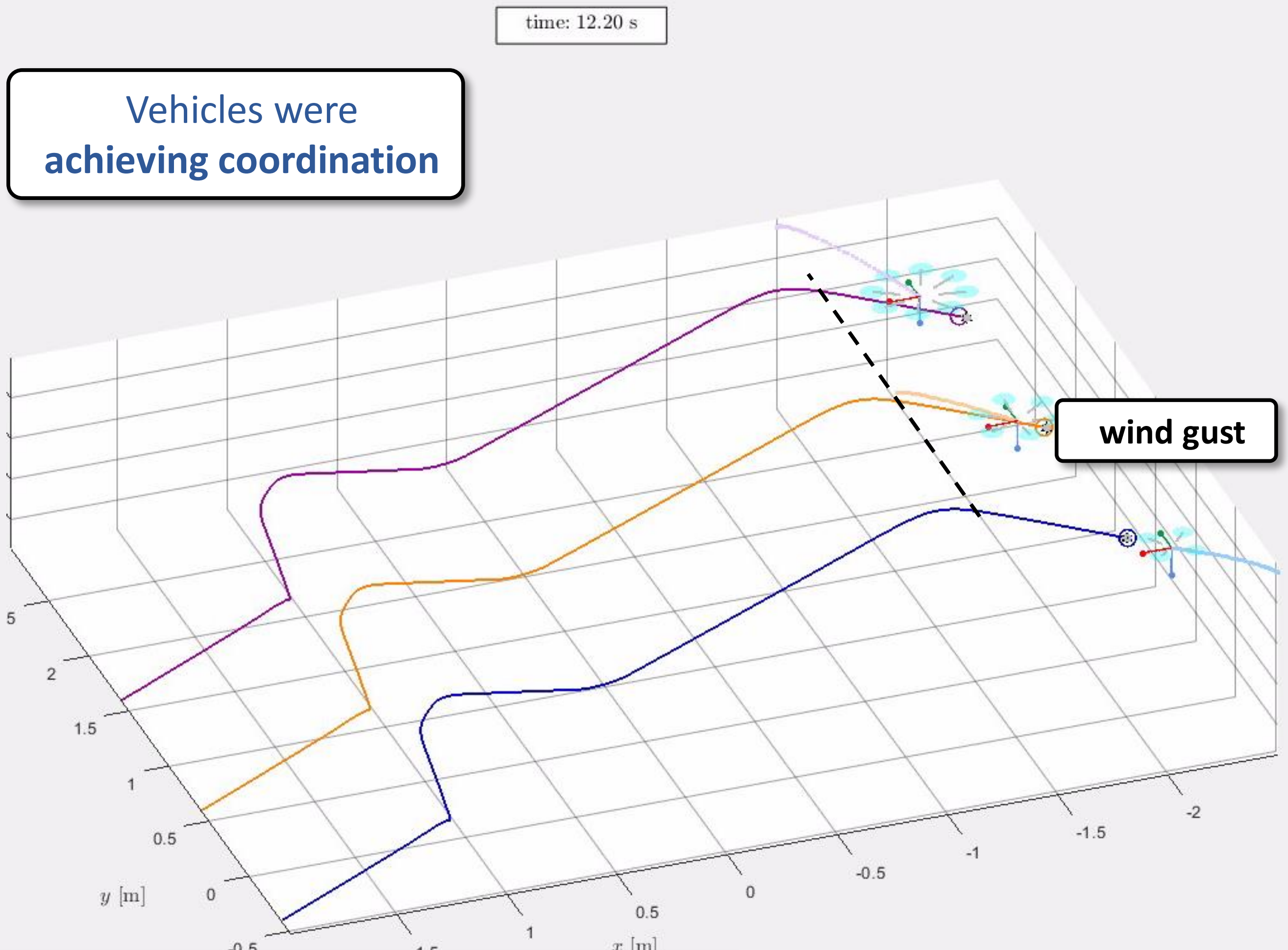
Behind

- **Inside** → **behind**
- **Outside** → **ahead**
- **Outside** → **ahead**



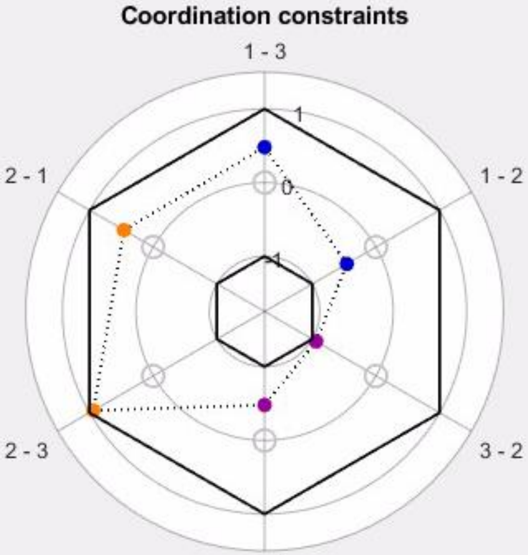


Vehicles were **achieving coordination**

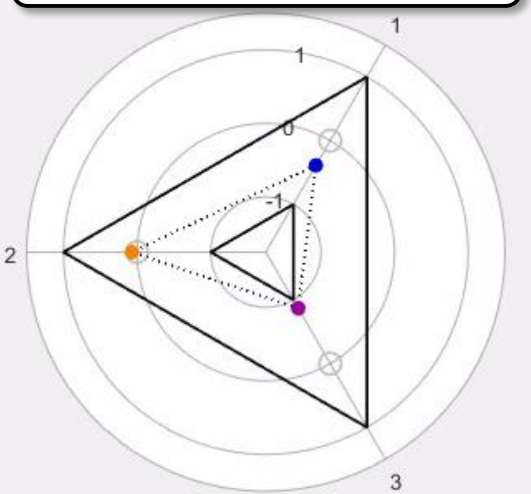


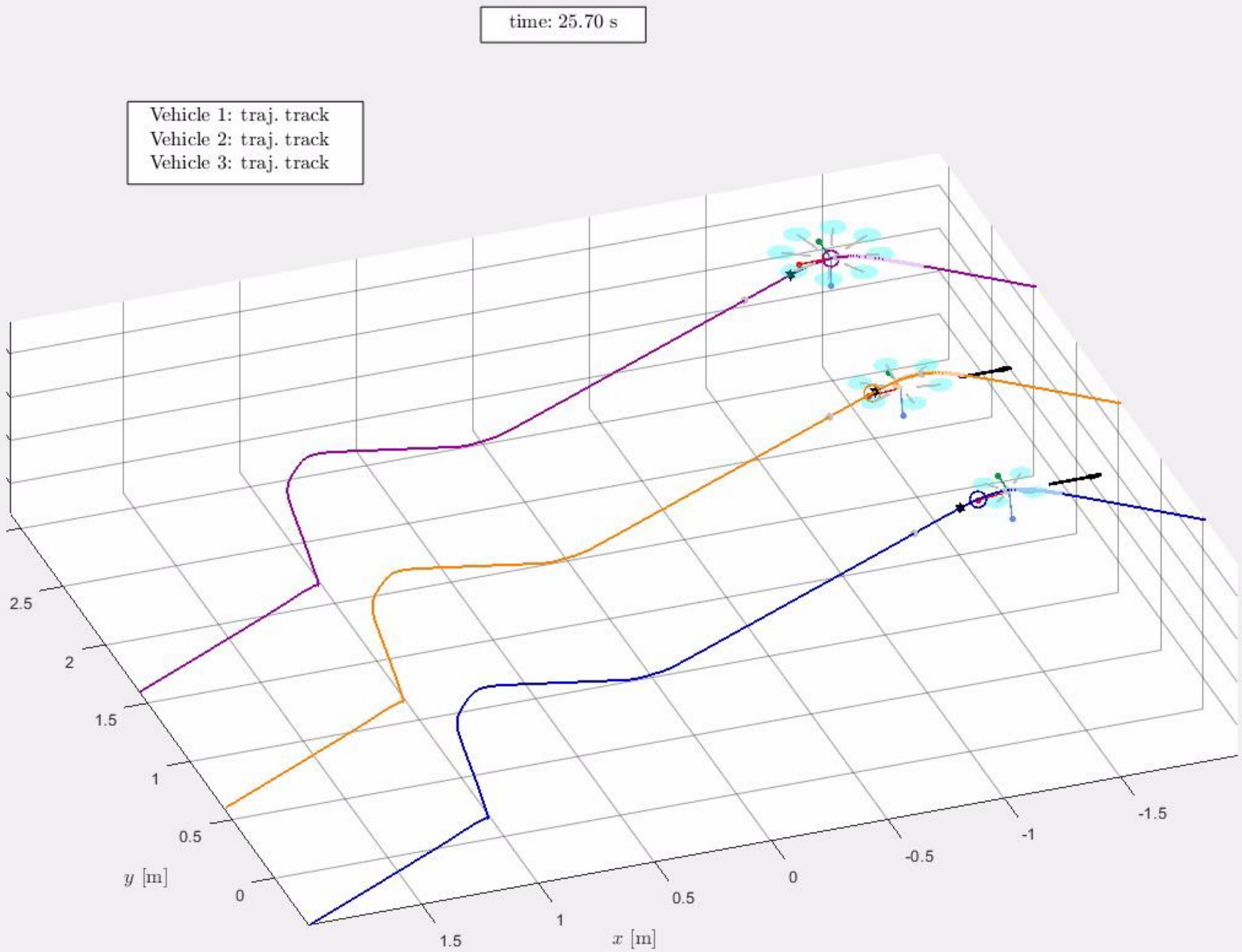
As ● and ● are **pushed behind** by the **wind**, they will move **inward** in temporal plot

Time-Critical Coordination

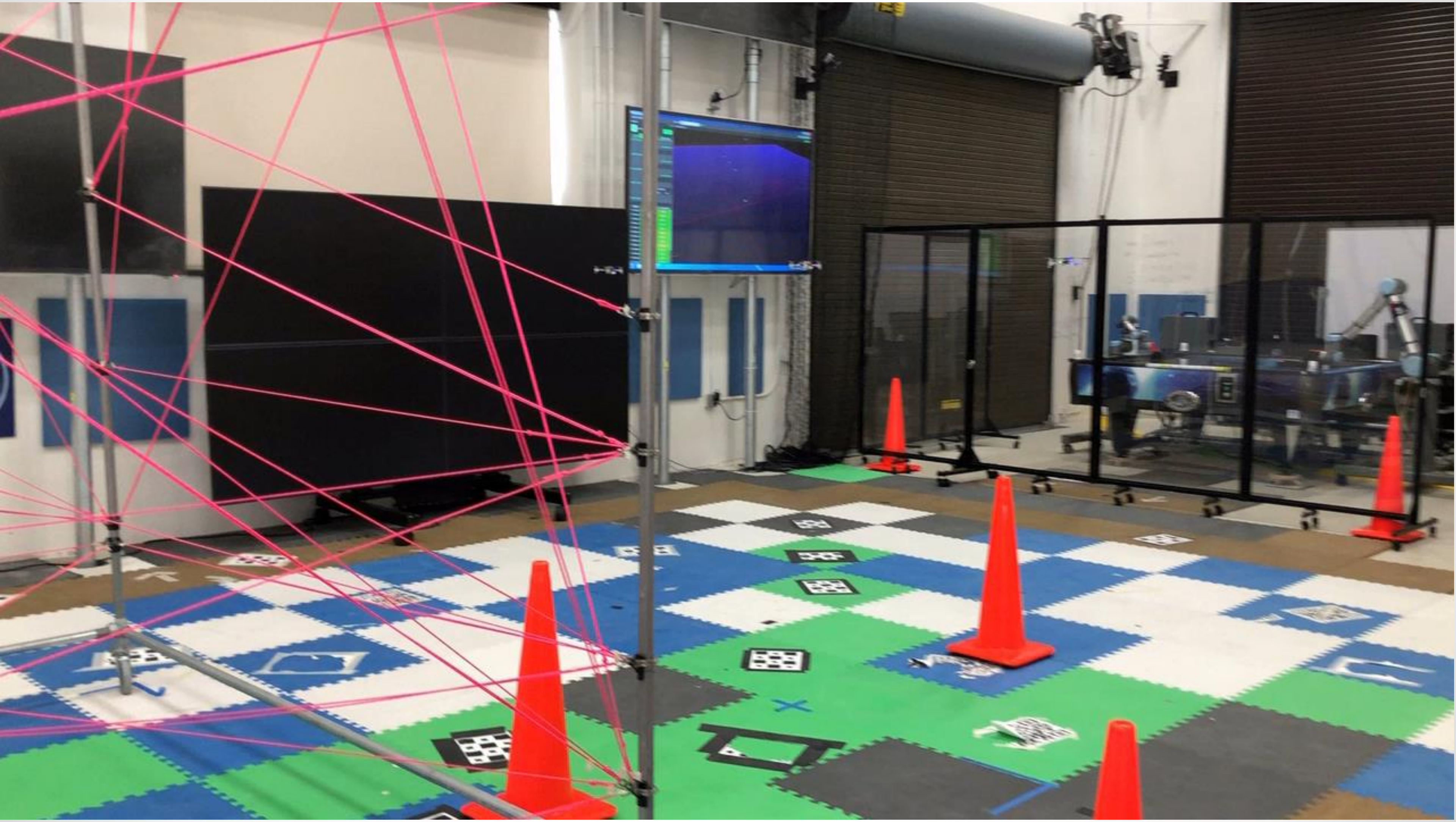


Converge to coordination and temporal **windows**



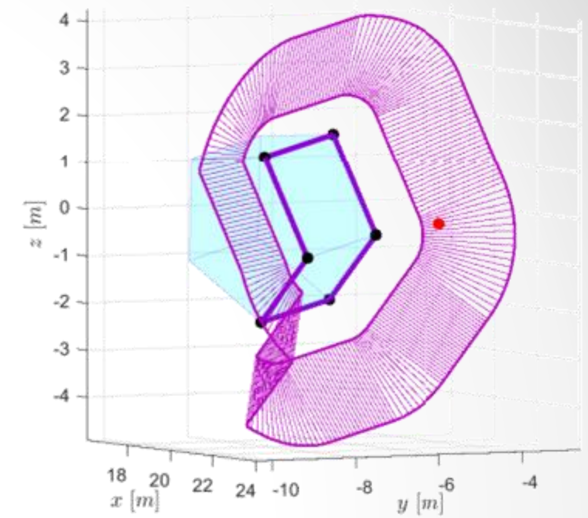


Flight Test



Conclusions

- **Trajectory planning**
  - Algorithms for **tolerance verification**
  - **Silhouette** computation algorithms inspired by the Expanding Polytope Algorithm (EPA)
  - Leveraged silhouette information to improve **narrow passage behavior**



- **Coordinated mission execution**
  - Non-linear **path-following** algorithm for multirotors
    - Transient and steady-state guarantees under some assumptions
  - Distributed coordination protocol that imposes a spectrum of time-critical strategies

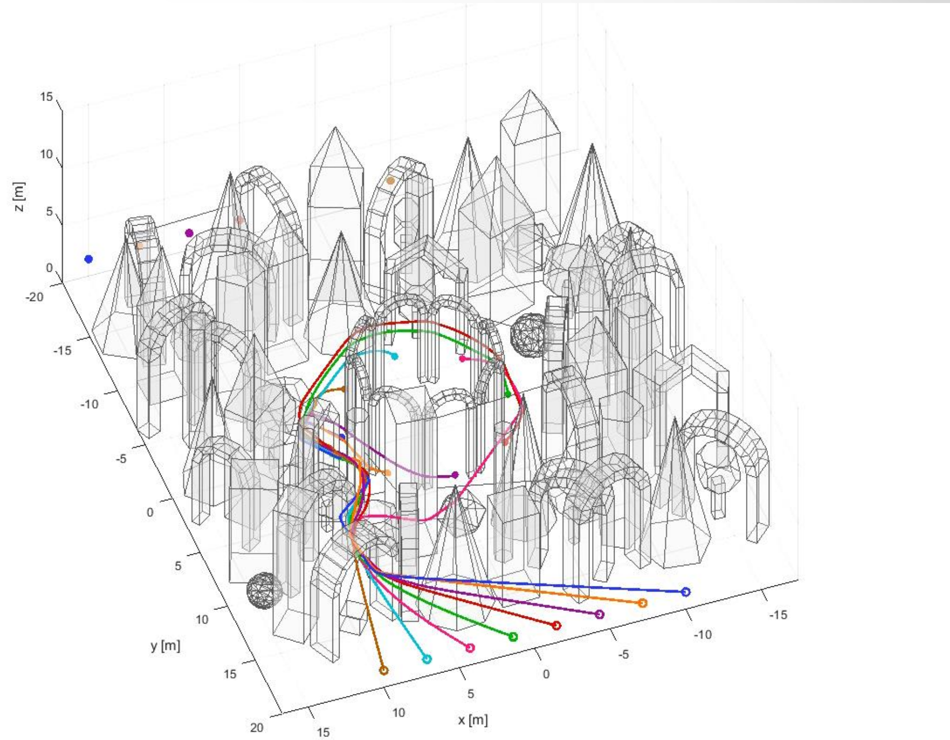
	Temporal constraints		
	Unenforced	**Relaxed**	**Strict**
Coordination constraints	Tight	Tight	Tight
	Loose	Loose	Loose

  - Transient and steady-state guarantees for tight coordination

javier.puignavarro@nasa.gov

Future Work

- **Autonomous time deconfliction** in urban-like scenario
  - challenging design of speed profiles
  - All vehicles must fly through narrow passage
- Use of **silhouette** information for online trajectory planning
  - Partial obstacle information
  - Fusion with perception



- **Relaxation of assumptions for path-following algorithms**
  - Switched system theory to achieve global exponential stability
  - Consider flight envelope
- **Relaxation of networking assumptions for coordination protocol**
  - Directed networks

javier.puignavarro@nasa.gov

Acknowledgements

- **Adviser: prof. Naira Hovakimyan**
- **ATTRACTOR team**
  - **Dr. Danette Allen**
  - **Dr. Natalia Alexandrov**
  - **Dr. Meghan Chandarana**
  - **Ben Kelley**
  - **Matt Vaughan**
  - **Ralph Williams**
  - **Thea Avila**
  - **Kyle Mcquarry**
  - **Keith Arthur**

ATTRACTOR
DOCENDO DISCIMUS

University of Illinois
at Urbana-Champaign

NASA Langley Research Center
Autonomous Systems Integrated Research Branch

javier.puignavarro@nasa.gov