Distributed Coordinated Path Following Using Guiding Vector Fields

Weijia Yao, Hector Garcia de Marina, Zhiyong Sun, Ming Cao

the Netherlands

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

the Netherlands
In the Netherlands, when you fly a drone, you need to be careful, because...
Chimps $\rightarrow$ singularity areas
in
zoo

multiple drones

guiding vector fields

troubles for drones
Related work

• Single-robot path-following is a classic problem.
• Guiding vector field based algorithms have been widely studied.
• These algorithms achieve high path-following accuracy.

Lawrence, et al., AIAA GNCC, 2007; Lawrence, et al., JGCD, 2008; Nelson, et al., ACC, 2006; Nelson, et al., T-RO, 2007; Goncalves, et al., T-RO, 2010; Kapitanyuk, et al., T-CST, 2017; Sujit et al., CSM, 2014.
A singularity-free guiding vector field that enables distributed multi-robot coordinated motion along general desired paths in n-dimensional spaces.

For mobile robots, $n = 2$ or $3$;
For robotic arms, $n$ is the number of joints.

We achieve this with rigorous mathematical guarantees.
Equations of guiding vector fields

**Mathematics**

Desired path $\mathcal{P}$ in 2D

$$\phi(x, y) = 0$$

2D guiding vector field

$$\chi(x, y) = \text{Rot}(90) \nabla \phi + (-k \phi \nabla \phi)$$

**Example**

Desired path $\mathcal{P}$: Circle

$$\phi(x, y) = x^2 + y^2 - 1 = 0$$
Visualization of guiding vector fields

\[
\text{Rot}(90) \nabla \phi + (-k \phi \nabla \phi) = \chi(x, y)
\]

\[
\begin{bmatrix}
\dot{x} \\
\dot{y}
\end{bmatrix} = \chi(x, y)
\]
Generalization to $n$-dimension

Given a desired path living in an $n$-dimensional space, we can similarly derive an $n$–dimensional guiding vector field:

$$n\text{-dimensional guiding vector field} = \text{Tangential component} + \text{Orthogonal component}$$
Singularity points

- **Singularity point** is a point where a vector field becomes **zero**.
- They may be **isolated** or **connected** to form an area.
- They are **undesirable!**
  - Vehicles can get stuck (no guidance).
  - No global convergence to desired paths.
Singularity points always exist!

Topological (inherent) property

If the desired paths are closed, or self-intersecting, there are always singularity points in the guiding vector field!

Question 1: Can we remove singularity points while maintaining the continuity of the vector field?

YES!
Solution: “topological surgery”

Cut and lift the desired path to a higher-dimensional space


- \((\cos w, \sin w)\) guiding point

**Actual** robot motion in 2D

**Virtual** robot motion in 3D
We have created a *singularity-free* guiding vector field for a *single robot*.

**Question 2:** How to extend it for *multi-robot coordination*?
An example of six robots

Objective: robots keep equal distances on a circle

without coordination  

with coordination
An example of six robots

The coordination is achieved via local communication on neighboring robots’ virtual coordinates.

Arrows of different colors belong to vector fields for different robots. For clarity, only parts of the vector fields around robots are shown.
Mathematics for coordination

• Distributed coordination via the virtual coordinate $w$

$$\begin{bmatrix} \chi_x \\ \chi_y \\ \chi_w \end{bmatrix} + k_c \begin{bmatrix} 0 \\ 0 \\ \text{virtual control} \end{bmatrix} = \chi_i$$

guiding vector field $\chi$

coordination term

coordination vector field

classic consensus algorithm on the virtual coordinate $w$
Mathematics for coordination

- Distributed coordination via the virtual coordinate $w$

$$
\begin{bmatrix}
\chi_x \\
\chi_y \\
\chi_w
\end{bmatrix}
+ k_c
\begin{bmatrix}
0 \\
0 \\
\text{virtual control}
\end{bmatrix}
= \chi_i
$$

$\chi$ guiding vector field
$\chi_w$ virtual control

$\chi_i$ coordination term
$\chi_w$ vector field

$$
- \sum_{\text{neighbors } j} \left[
\left(
\chi_w \text{ my virtual coordinate } w_i
- \chi_w \text{ my neighbor’s virtual coordinate } w_j
\right)
- \chi_w \text{ difference } \Delta_{ij}
\right]
$$

Previous example: $\Delta_{ij} = \frac{2\pi}{6}$
Mathematics for coordination

- Distributed coordination via the virtual coordinate $w$

$$
\begin{bmatrix}
\chi_x \\
\chi_y \\
\chi_w
\end{bmatrix} + k_c
\begin{bmatrix}
0 \\
0 \\
\text{virtual control}
\end{bmatrix} = \chi_i
$$

- The coordination vector field is not a gradient of any potential function.
- Global convergence is guaranteed.
Fixed-wing: Dubins car model and control

\[
\begin{align*}
\dot{x}_i &= v \cos \theta \\
\dot{y}_i &= v \sin \theta \\
\dot{z}_i &= u^z_i \\
\dot{\theta}_i &= u^\theta_i
\end{align*}
\]

\(v\) is a constant speed
\(\theta\) is the yaw angle
\(u^z_i\) and \(u^\theta_i\) are control inputs

Control inputs design principle: the fixed-wing orientation becomes aligned with the arrows given by the coordination guiding vector field.
Fixed-wing: Dubins car model and control

\[ \dot{x}_i = v \cos \theta \]
\[ \dot{y}_i = v \sin \theta \]
\[ \dot{z}_i = u_i^z \]
\[ \dot{\theta}_i = u_i^\theta \]

\( v \) is a constant speed
\( \theta \) is the yaw angle
\( u_i^z \) and \( u_i^\theta \) are control inputs

\[ u_i^z = v \mathbf{x}_{i3} / \sqrt{x_{i1}^2 + x_{i2}^2} \]
\[ u_i^\theta = \text{Sat}_a^b \left( -\mathbf{X}_i^P \mathbf{E} \hat{\mathbf{x}}_i^P / \| \mathbf{X}_i^P \| - k_\theta \mathbf{h}_i \mathbf{E} \mathbf{X}_i^P \right) \]
Coordinated flight of fixed-wings
Coordinated flight of fixed-wings

All codes are open source in the autopilot Paparazzi website.

wiki.paparazziuav.org
Other applications

3D volume coverage (with XY, YZ sideview)  Coordinated motion on different paths  Coordinated motion on surfaces
Path following with collision avoidance

Without Safety Barrier Certificate (collision happens)

With Safety Barrier Certificate (no collision)

Wang, Ames, Egerstedt, T-RO, 2017
Conclusion

• We propose a coordination guiding vector field for multi-robot distributed path following.

• Our approach:
  • Distributed and scalable
  • Enables following of complex paths
  • Low communication & computational cost
  • Singularity-free & has global convergence guarantees

• Experiments with fixed-wing aircraft (saturated Dubins car model)
Thank you!

Feel free to ask questions and contact us!

All codes are available here: wiki.paparazziuav.org/wiki/Module/guidance_vector_field
Appendix
Appendix 1: 3D guiding vector field

3D vector field:
\[ \chi(x, y, z) = \nabla \phi_1 \times \nabla \phi_2 - k_1 \phi_1 \nabla \phi_1 - k_2 \phi_2 \nabla \phi_2 \]

Tangential
Orthogonal

The construction can be extended to \( n \)-dimensions.
Appendix 2: topological surgery

Mathematically, it is simple, relying on a parametric equation

Circle: \[
\begin{align*}
x &= \cos w \\
y &= \sin w
\end{align*}
\]

Helix: \[
\begin{align*}
\phi_1(x, y, w) &= x - \cos w = 0 \\
\phi_2(x, y, w) &= y - \sin w = 0
\end{align*}
\]
Appendix 3: guiding point vs trajectory point

- \((\cos w, \sin w)\) guiding point

Actual robot motion in 2D

Virtual robot motion in 3D