Co-Operative Control of Two-Robot Teams for Two-Dimensional Level Curve Tracking

2017 ASME Summer School on Multibody Systems and Nonlinear Dynamics

Sarthak Chatterjee
Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy NY
About me...(from)

- Bachelor in Electronics and Telecommunication Engineering, Jadavpur University, Kolkata, India. (2015)
About me...(to)

- 2nd year PhD student, Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy NY
Acknowledgments

Wencen Wu

Dan Negrut

Radu Serban
Outline

Part I: Co-Operative Curve Tracking Without Explicitly Estimating the Field Gradient

Part II: Curve Tracking With Two Nonholonomic Wheeled Robots (Ongoing)
Part I: Co-Operative Curve Tracking Without Explicitly Estimating the Field Gradient
Mobile Sensor Networks and what can be done with them...

- In recent years, a lot of focus has been on environmental monitoring by the use of mobile sensor networks.
- Mobile Sensor Networks offer the advantage over Static Sensor Networks in that comparatively fewer agents can be used to monitor complex landscapes.
- Motivating Example: Tracking level curves of oil spills!

Khepera Robots: Examples of Mobile Sensor Networks
Using vs. Not using the field gradient, and the number of agents: Critical Issues

- Wu and Zhang in [1] elegantly achieve excellent rates of convergence using multiple (6) agents.
- However, this comes at the cost of increased computational power!

- On the other hand, Matveev et al. in [2] use a gradient-free controller with a single agent.
- Lesser computational power comes at the cost of unsatisfactory convergences as the robot “circulates” along the level curve.

Figures from [1] and [2]
Goals and Objectives

- **Goal:** Develop a controller for 2 mobile agents in the plane.
- **Objective:** Given a particular level value, can we steer the mobile agents fast enough to the desired level value and keep their formation moving along a level curve in the presence of noisy measurement values?
- Consider: \( f(x, y) = x^2 + y^2 \)

*Function values stay constant on each circle*
Development of the Control Law

- Key aspects:
  1. Decomposition of the velocities of each mobile agent along two mutually perpendicular directions.
  2. Development of separate control laws along each direction.
The control law

- Velocity decomposition: along two mutual directions
  \[ v_i = v_{i,q} + v_{i,n} = v_{i,q}q_i + v_{i,n}n_i \]

  - Along the ‘q’ direction:
    \[ v_{i,q} = k_1((r_j - r_i) \cdot q - d_{i,j}^0) + k_2 \text{sgn}((y_c - z_d)(y_1 - y_2)) \]

  - Along the ‘n’ direction:
    \[ v_{i,n} = \begin{cases} 
      C + ay_i, & \text{if } |y_c - z_d| < \varepsilon \\
      0, & \text{otherwise}
    \end{cases} \]
A brief summary of analysis techniques

• We use Lyapunov functions to show convergence.
• Lyapunov functions: Continuously differentiable functions such that:

\[ V(\text{state}=\text{where I want my state to ultimately converge}) = 0 \]
\[ V(\text{state} \neq \text{where I want my state to ultimately converge}) > 0 \]
A brief summary of analysis techniques

- Our Lyapunov function is the simple quadratic:

\[ V = \frac{1}{2} (z_c - z_d)^2 \]

- Convergence under certain conditions, that is, will have:

\[ \dot{V} < 0 \]

\[ \dot{V} = \text{grad } V \cdot \frac{dX}{dt} = \|\text{grad } V\| \left\| \frac{dX}{dt} \right\| \cos \varphi \]
Simulation Results

- First trial: Level curves having a value of 500 of the ellipse:

\[ f(x, y) = (x - 20)^2 + 8(y - 20)^2 \]
Simulation Results (Continued)

- First trial: Evolution of tracked level value at formation center versus time for the ellipse:
  \[ f(x, y) = (x - 20)^2 + 8(y - 20)^2 \]
Simulation Results (Continued)

• Second trial: Level curves having a value of 2 of the Matyas Function:
  \[ f(x, y) = 0.26(x^2 + y^2) - 0.48xy \]
Simulation Results (Continued)

- Second trial: Evolution of tracked level value at formation center versus time for the Matyas Function:

$$f(x, y) = 0.26(x^2 + y^2) - 0.48xy$$
The control law

- Velocity decomposition: along two mutual directions

\[ \mathbf{v}_i = \mathbf{v}_{i,q} + \mathbf{v}_{i,n} = v_{i,q}\mathbf{q}_i + v_{i,n}\mathbf{n}_i \]

- Along the ‘\( q \)’ direction:

\[ v_{i,q} = k_1((\mathbf{r}_j - \mathbf{r}_i) \cdot \mathbf{q} - d_{i,j}^0) + k_2 \text{sgn}((y_c - z_d)(y_1 - y_2)) \]

- Along the ‘\( n \)’ direction:

\[ v_{i,n} = \begin{cases} C + ay_i, & \text{if } |y_c - z_d| < \varepsilon \\ 0, & \text{otherwise} \end{cases} \]
Issues and Additions

- Interpretation of the controller as a sliding mode control law!
- **Reason:** Discontinuity in the control law.
- Gradient-free control law for an $N$-agent system!
References


Part II: Curve Tracking With Two Wheeled Robots (Unicycles) (Ongoing)
Issues with the particle model and unicycles

- **Particle model**: Two orthogonal design directions, ‘q’ and ‘n’

- **Reality**: Hardware implementations of the algorithm mostly allow us to follow the ‘unicycle’ model.

- **Two options**: Move forward or turn on axis.
‘The Unicycle Model’

- A common model found in introductory robotics courses.

- Motions possible:
  1. Forward direction
  2. Turning on axis

- More realistically catered to the implementation of curve-tracking algorithms in hardware.

\[
\begin{align*}
\dot{x} &= v \cos \phi \\
\dot{y} &= v \sin \phi \\
\dot{\phi} &= \omega
\end{align*}
\]

Figure from: https://sejoker.github.io/quickbot-slides/img/unicicle-model.png
Inspiration

• We will adopt a modification of the scheme in [1] (Lu, You, and Wu) in which unicycles are used for quickly seeking the ‘source’ of a two-dimensional field.

• Realistic constraints on the linear and angular velocities are considered.

\[ |v_i| \leq v_{\text{max}}, \quad i = 1, 2 \]
\[ |\omega_i| \leq \omega_{\text{max}}, \quad i = 1, 2 \]
What We Adopt from [1]

- Dual – module control structure.
Benefits of the dual-control-module structure

• Formation control and curve tracking are independent problems.

• Formation control module: design only angular velocity to help maintain formation!

• Curve tracking module: design only linear velocity for curve tracking!

• Also gradient-free, uses only instantaneous field values and not gradient information.
Formation Control Module Design

- Want: \( d = d^0, \varphi_1 = \pi/2, \varphi_2 = -\pi/2 \)
Formation Control Module Design - Continued

• Summary of design:
  1. Get System Dynamics:

\[
\begin{align*}
\dot{d} &= -v_1 \cos \varphi_1 - v_2 \cos \varphi_2, \\
\dot{\varphi}_1 &= \frac{v_1 \sin \varphi_1 + v_2 \sin \varphi_2}{d} + \omega_1, \\
\dot{\varphi}_2 &= \frac{v_1 \sin \varphi_1 + v_2 \sin \varphi_2}{d} + \omega_2.
\end{align*}
\]

2. Define backstepping tracking error variables:

\[
\begin{align*}
\mathcal{D}_d &= d - d^0, \\
\mathcal{D}_1 &= \cos \varphi_1 - \frac{k_d}{2} \left( d - d^0 \right), \\
\mathcal{D}_2 &= \cos \varphi_2 - \frac{k_d}{2} \left( d - d^0 \right),
\end{align*}
\]
Formation Control Module Design - Continued

3. Design:

\[
\begin{align*}
\omega_1 &= \frac{k_d}{2 \sin \varphi_1} \left( v_1 \cos \varphi_1 + v_2 \cos \varphi_2 \right) - \frac{v_1 \sin \varphi_1 + v_2 \sin \varphi_2}{d} + k_{D_1} D_1, \\
\omega_2 &= \frac{k_d}{2 \sin \varphi_2} \left( v_1 \cos \varphi_1 + v_2 \cos \varphi_2 \right) - \frac{v_1 \sin \varphi_1 + v_2 \sin \varphi_2}{d} + k_{D_2} D_2,
\end{align*}
\]

to give the closed-loop system:

\[
\begin{bmatrix}
\dot{D}_d \\
\dot{D}_1 \\
\dot{D}_2
\end{bmatrix} =
\begin{bmatrix}
-k_d \frac{v_1 + v_2}{2} & -v_1 & -v_2 \\
0 & -k_{D_1} & 0 \\
0 & 0 & -k_{D_2}
\end{bmatrix}
\begin{bmatrix}
D_d \\
D_1 \\
D_2
\end{bmatrix}
\]

\( \mathcal{A} \)
Overview of Main Theoretical Contributions

• Conditionally exponentially stable (!) closed-loop dynamics!

• Easy to show, Lyapunov theory for linear systems is well-defined. Just take

\[ V = (\text{State})^T \mathcal{P}(\text{State}), \quad \mathcal{P} \succ 0 \implies A^T \mathcal{P} + \mathcal{P} A = -Q, \quad Q \succ 0 \]

• Study of the input uncertainty rejection capability of said controller.

\[
\begin{bmatrix}
\dot{D}_d \\
\dot{D}_1 \\
\dot{D}_2
\end{bmatrix} = A \begin{bmatrix}
D_d \\
D_1 \\
D_2
\end{bmatrix} + B \begin{bmatrix}
\Delta v_1 \\
\Delta v_2 \\
\Delta \omega_1 \\
\Delta \omega_2
\end{bmatrix}
\]

• Closed-loop system input-to-state stable in presence of input uncertainties.
So what’s left?

Field Values $z(r_1), z(r_2)$

Forward Velocities $v_1, v_2$

- Intuitions: Once again, sliding mode controller having many sliding manifolds!
THANK YOU!
QUESTIONS?