Autonomous Mobile Robot Design

Topic: Wheeled Mobile Robot Motion Control

Dr. Kostas Alexis (CSE)
Overview

- The objective of a kinematic controller is to follow a trajectory described by its position and/or velocity profiles as function of time.
- Motion control is not straightforward because mobile robots are typically nonholonomic and MIMO systems.
- Most controllers (including the one presented here) are not considering the dynamics of the system.

Based on the book “Autonomous Mobile Robots” by R. Siegwart
Motion Control: Open Loop

- Trajectory (path) divided in motion segments of clearly defined shape:
  - Straight lines and segments of a circle
  - Dubins car, and Reeds-Shepp car

- Control problem:
  - Pre-compute a smooth trajectory
    - Based on line, circle (and clothoid) segments

- Disadvantages:
  - Pre-computation of feasible trajectories is not an easy task
  - Limitations and constraints of the robots velocities and accelerations.
  - Does not adapt or correct the trajectory if dynamic changes of the environment occur.
  - The resulting trajectories are usually not smooth (in acceleration, jerk, etc)
Motion Control: Feedback Control

- Find a control matrix $K$, if one exists:

\[
K = \begin{bmatrix}
  k_{11} & k_{12} & k_{13} \\
  k_{21} & k_{22} & k_{23}
\end{bmatrix}
\]

- With $k_{ij} = k(t, e)$

- such that the control of $v(t)$ and $\omega(t)$

\[
\begin{bmatrix}
  v(t) \\
  \omega(t)
\end{bmatrix} = K \cdot e = K \cdot \begin{bmatrix}
  x \\
  y \\
  \theta
\end{bmatrix} ^R
\]

drives the error $e$ to zero: $\lim_{t \to \infty} e(t) = 0$

- MIMO state feedback control
The kinematics of a differential drive mobile robot described in the inertial frame \([x_I, y_I, \theta]\) is given by:

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{bmatrix}
=egin{bmatrix}
\cos \theta & 0 & v \\
\sin \theta & 0 & \omega \\
0 & 1 & 0
\end{bmatrix}
\]

where \(\dot{x}\) and \(\dot{y}\) are the linear velocities expressed on the inertial frame.

Let \(\alpha\) denote the angle between the axis \(x_R\) of the robot's reference frame and the vector connecting the center of the axle of the wheels with the final position.
Kinematic Position Control: Coordinates Transformation

- Coordinates transformation into polar coordinates with its origin at goal position:
  \[
  \rho = \sqrt{\Delta x^2 + \Delta y^2}
  \]
  \[
  \alpha = -\theta + \text{atan2}(\Delta y, \Delta x)
  \]
  \[
  \beta = -\theta - \alpha
  \]

- System description, in the new polar coordinates:

\[
\begin{bmatrix}
\dot{\rho} \\
\dot{\alpha} \\
\dot{\beta}
\end{bmatrix} =
\begin{bmatrix}
-\cos \alpha & 0 \\
\frac{\sin \alpha}{\rho} & -1 \\
-\frac{\sin \alpha}{\rho} & 0
\end{bmatrix}
\begin{bmatrix}
\nu \\
\omega
\end{bmatrix}
\]

\[
\begin{bmatrix}
\dot{\rho} \\
\dot{\alpha} \\
\dot{\beta}
\end{bmatrix} =
\begin{bmatrix}
\cos \alpha & 0 \\
\frac{\sin \alpha}{\rho} & 1 \\
\frac{\sin \alpha}{\rho} & 0
\end{bmatrix}
\begin{bmatrix}
\nu \\
\omega
\end{bmatrix}
\]

\[
I_1 = \begin{pmatrix}
-\frac{\pi}{2} & \frac{\pi}{2}
\end{pmatrix}
\]

\[
I_2 = (-\pi, -\pi/2] \cup (\pi/2, \pi]
\]
Kinematic Position Control: Remarks

- The coordinates transformation is not defined at \( x = y = 0 \)
- For \( a \in I_1 \) the forward direction of the robot point toward the goal, for \( a \in I_2 \) it is the backward direction.

\[
\alpha \in I_1 = \left( -\frac{\pi}{2}, \frac{\pi}{2} \right)
\]

- By properly defining the forward direction of the robot at its initial configuration, it is always possible to have \( a \in I_1 \) at \( t = 0 \). However, this does not mean that \( a \) remains in \( I_1 \) for all time \( t \).
Kinematic Position Control: The Control Law

- It can be shown that with:
  \[ v = k_\rho \dot{\rho} \]
  \[ \omega = k_\alpha \alpha + k_\beta \beta \]
  \[ \begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_\rho \rho \cos \alpha \\ k_\rho \sin \alpha - k_\alpha \alpha - k_\beta \beta \\ -k_\rho \sin \alpha \end{bmatrix} \]

- The feedback control system

- Will drive the robot to \([\rho, \alpha, \beta] = [0,0,0]\)

- The control signal \(v\) has always constant sign
  - The direction of movement is kept positive or negative during the movement
  - A parking maneuver is always performed in the most natural way, and without even inverting its motion.
Kinematic Position Control: Resulting Path
Autonomous Mobile Robot Design

Topic: L1-Guidance

Dr. Kostas Alexis (CSE)
L1 Guidance Law

- The L1 Guidance law enables following of curved or straight trajectories from a fixed-wing UAV.
- It uses the UAV’s velocity, as well as its change in velocity due to “external disturbances” such as wind.
- It tracks its instantaneous speed to quickly adapt to velocity changes.
Atlantik Solar

Test Flight Day #24
(May 18th 2015)

Aircraft: AtlantikSolar UAV Prototype (AS-P)
Location: Rothenburg, Switzerland
Flights performed: 3
Mission: High-fidelity 3-D mapping of area using pre-computed paths with guaranteed coverage
L1 Guidance Law

- The L1 guidance selects a forward reference point along the UAV’s reference trajectory for every step.
- The distance between the UAV and any reference point is L1.
- The centripetal acceleration to keep the UAV on course comes from the basic equation for centripetal acceleration:
  \[ a_c = \frac{v^2}{r} \]
- Where \( v \) is the velocity of the UAV, and \( r \) is the radius of the circle of the instantaneous curve in the trajectory. In this case, the centripetal acceleration \( a_{s,cmd} \) takes the form:
  \[ a_{s,cmd} = \frac{2v^2}{L1} \sin(\eta) \]
L1 Guidance Law

- The L1 guidance selects a forward reference point along the UAV’s reference trajectory for every step.
- The distance between the UAV and any reference point is L1.
- The centripetal acceleration to keep the UAV on course comes from the basic equation for centripetal acceleration:

\[ a_c = \frac{v^2}{r} \]

- Where \( v \) is the velocity of the UAV, and \( r \) is the radius of the circle of the instantaneous curve in the trajectory. In this case, the centripetal acceleration \( a_{s,cmd} \) takes the form:

\[ a_{s,cmd} = \frac{2v^2}{L_1} \sin(\eta) \]
L1 Guidance Law

- The L1 guidance selects a forward reference point along the UAV's reference trajectory for every step.
- The distance between the UAV and any reference point is L1.
- The centripetal acceleration to keep the UAV on course comes from the basic equation for centripetal acceleration:

\[ a_c = \frac{v^2}{r} \]

- Where \( v \) is the velocity of the UAV, and \( r \) is the radius of the circle of the instantaneous curve in the trajectory. In this case, the centripetal acceleration \( a_{s,cmd} \) takes the form:

\[ a_{s,cmd} = \frac{2v^2}{L_1} \sin(\eta) \]

Knowing the direction of the L1 vector, the UAV will try to align its velocity vector with the L1 vector.
L1 Guidance Law

- Knowing the direction of the L1 vector, the UAV will try to align its velocity vector with the L1 vector.
- If the current path of the UAV is far off from its planned trajectory, the guidance logic turns the UAV a great amount to assist it to come back on track.
L1 Guidance: Tracking straight lines

- Centripetal acceleration required to bring the UAV on track is approximated by:
  \[ \sin(\eta) \approx \eta \]

- As \( \eta \) is considered to be small, so are \( \eta_1 \) and \( \eta_2 \).
  \[ \sin \eta \approx \eta_1 + \eta_2 \]

- Let \( d \) denote the cross-track error – the distance perpendicular to the desired flight path to the UAV. Then:
  \[ \eta_1 \approx \sin(\eta_1) = \frac{d}{L_1} \quad \eta_2 \approx \sin(\eta_2) = \frac{d}{V} \]

- Combining the above with the guidance formula:
  \[ a_{cmd} = \frac{2V^2}{L_1} \sin \eta \approx 2\frac{V}{L_1} \left( d + \frac{V}{L_1}d \right) \]
L1 Guidance: Tracking straight lines

- Let us also assume no-inner-loop dynamics. Then as long as $\eta_2$ is small:

  $$ a_{s_{cmd}} \approx -\ddot{d} $$

- Subsequently equation:

  $$ a_{s_{cmd}} = 2 \frac{V^2}{L_1} \sin \eta \approx 2 \frac{V}{L_1} \left( \dot{d} + \frac{V}{L_1} d \right) $$

- Can be reformulated as:

  $$ \dddot{d} + 2\zeta \omega_n \dot{d} + \omega_n^2 d = 0 \quad \zeta = \frac{1}{\sqrt{2}}, \quad \omega_n = \sqrt{2}V/L_1 $$
L1 Guidance: Tracking non-straight lines

- Assuming again small angles-analysis:
  \[ \sin \eta \approx \eta = \eta_1 + \eta_2 \]

- Then since:
  \[ \eta_1 \approx \frac{d - d^*_{ref.pt.}}{L_1}, \quad \eta_2 \approx \frac{\ddot{d}}{V} \]

- And \( a_{s cmd} \approx -\ddot{d} \): the guidance law reduces to:
  \[ \ddot{d} + \frac{2V}{L_1} \dot{d} + \frac{2V^2}{L_1^2} d = \frac{2V^2}{L_1^2} d^*_{ref.pt.} \]
L1 Guidance: Tracking non-straight lines

- In general, considering the vehicle speed and the length for L1, and assuming small angle for $\eta$ there is a time difference of approximately $L1/V$ between $d^*$ and $d_{ref,pt}^*$.

  \[ \frac{d_{ref,pt}^*(s)}{d^*(s)} \approx e^{\tau s}, \quad \tau \approx L1/V \]

- Therefore:

  \[ \frac{d(s)}{d^*(s)} = \frac{\omega_n^2 e^{\tau s}}{s^2 + 2\zeta\omega_n s + \omega_n^2} \]

  where $\zeta = 0.707$, $\omega_n = \frac{\sqrt{2V}}{L_1}$, $\tau \approx L1/V$
L1 Guidance: Tracking circular paths

- When a UAV is following a circular path, the centripetal acceleration is aiming to be normal to the tangential acceleration it would have if it were following the circle perfectly.

- With simplifications, the circular path model also produces a second order linear system:

$$0 \approx \ddot{d} + 2\zeta \omega_n \dot{d} + \omega_n^2 d$$
Atlantik Solar

Test Flight Days #9&10
(May 5th & 6th 2014)

Aircraft: AtlantikSolar UAV Prototype
Location: Tuggen, Switzerland
Flights performed: 5
Tests: Autopilot Waypoint-following
Code Examples and Tasks

How does this apply to my project?

- To control the state of the robot and make it follow the desired trajectory.
Find out more

- http://www.kostasalexis.com/lqr-control.html
Thank you!

Please ask your question!