Smooth Motion Generation for the Unicycle Robot
via Dynamic Path Inversion

Aurelio PIAZZI
DII, Università di Parma

31 January 2006
Outline

• Introduction
• The problem and a first result
• An algorithm for the path inversion extended problem
• An example
• Iterative steering and supervision
• Summary
Introduction

Consider a wheeled mobile robot governed by the unicycle model

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

Within a supervisor motion control architecture we want to steer the WMR on assigned planned paths using iterative steering with smooth controls (continuous accelerations).

\( \nu(t), \omega(t) \) are \( C^1 \)-functions
Introduction

At the instants of replanning we have to guarantee continuity of the WMR state \((x, y, \theta)\) and continuity of the velocities \((v, \omega)\) and accelerations \((\dot{v}, \dot{\omega})\).

So, we introduce the extended state of the WMR:

\[
\{x(t), y(t), \theta(t), v(t), \omega(t), \dot{v}(t), \dot{\omega}(t)\}
\]

The extended state is required be overall continuous regardless of any issued replanning.
The problem and a first result

The Path Inversion Extended Problem for WMRs:
Given any traveling time $t_1 > 0$, find control inputs $\nu(\cdot), \omega(\cdot) \in C^1([0, t_1])$ such that the WMR starting from an arbitrary initial extended state $A$

$$p_A = [x_A \ y_A]^T = [x(0) \ y(0)]^T \quad \theta_A = \theta(0)$$
$$\nu_A = \nu(0) \quad \omega_A = \omega(0)$$
$$\dot{\nu}_A = \dot{\nu}(0) \quad \dot{\omega}_A = \dot{\omega}(0)$$

reaches, following a given path $\Gamma$, the arbitrary final extended state $B$

$$p_B = [x_B \ y_B]^T = [x(t_1) \ y(t_1)]^T \quad \theta_B = \theta(t_1)$$
$$\nu_B = \nu(t_1) \quad \omega_B = \omega(t_1)$$
$$\dot{\nu}_B = \dot{\nu}(t_1) \quad \dot{\omega}_B = \dot{\omega}(t_1)$$
The problem and a first result

• The PIE problem is both a reachability problem in the extended state space and a smooth path inversion problem.

**Proposition 1** (Guarino, Piazza, Romano 2004 TR)
Assign any $t_f > 0$. If a Cartesian path $\Gamma$ is generated by the unicycle model with inputs $v(t), \omega(t) \in C^1([0,t_1])$ and $v(t) \neq 0 \ \forall \ t \in [0,t_1]$, then $\Gamma$ is a $G^3$-path. Conversely, given any $G^3$-path $\Gamma$ there exist inputs $v(t), \omega(t) \in C^1([0,t_1])$ with $v(t) \neq 0 \ \forall \ t \in [0,t_1]$ and initial conditions such that the path generated by the unicycle model coincides with the given $\Gamma$. 
The problem and a first result

Proof (necessity)

The trajectory $\mathbf{p}(t) \triangleq [x(t) \, y(t)]^T$ can be interpreted as a curve parameterization of the path $\Gamma$.

Considering that $v(t) > 0 \lor v(t) < 0 \ \forall t \in [0, t_1]$ curve $\mathbf{p}(t)$ is regular.

Moreover from the unicycle equations $\Rightarrow \ddot{\mathbf{p}}(\cdot) \in C^0([0, t_1]) \Rightarrow \mathbf{p}(t)$ is a $G^2$-curve

$\Rightarrow \Gamma$ is $G^2$-path

However, we cannot claim $\ddot{\mathbf{p}}(\cdot) \in C^0([0, t_1])$ ...

Substituting the second derivatives

$$\dot{x}(t) = \dot{v}(t) \cos \theta(t) - v(t) \omega(t) \sin \theta(t)$$

$$\dot{y}(t) = \dot{v}(t) \sin \theta(t) + v(t) \omega(t) \cos \theta(t)$$

in the curvature expression

$$k_c(t) = \frac{\dot{x}(t)\dot{y}(t) - \ddot{x}(t)\dot{y}(t)}{\left(\dot{x}^2(t) + \dot{y}^2(t)\right)^{3/2}} \Rightarrow k_c(t) = \frac{\omega(t)}{v(t)} \text{ if } v(t) > 0 \quad k_c(t) = -\frac{\omega(t)}{v(t)} \text{ if } v(t) < 0$$
The problem and a first result

Compute \( \frac{dk_c}{ds}(t) \) where

\[
  s = \begin{cases} 
    \int_0^t v(\xi)d\xi & \text{if } v > 0 \\
    -\int_0^t v(\xi)d\xi & \text{if } v < 0 
  \end{cases}
\]

\[
\frac{dk_c}{ds}(t) = \frac{dk_c(t)}{dt} = \frac{\dot{\omega}(t)v(t) - \omega(t)v(t)}{v^3(t)}
\]

(holds for both \( v(t) > 0 \) and \( v(t) < 0 \))

\[
\Rightarrow \quad \frac{dk_c}{ds}(t) \in C^0([0,t_1])
\]

\[
\frac{d\kappa}{ds}(s) = \frac{dk_c}{ds}(f^{-1}(s)) \in C^0([0,s_1])
\]

because \( f^{-1}(s) \) is continuous too.

\[
\Rightarrow \quad p(t) \text{ is then a } G^3\text{-curve} \quad \Rightarrow \quad \Gamma \text{ is a } G^3\text{-path}.
\]
The problem and a first result

Proof of sufficiency (constructive)

\( \Gamma \), the \( G^3 \)-path, be defined by the \( G^3 \)-curve \( \mathbf{p}(u) = \begin{bmatrix} \alpha(u) \\ \beta(u) \end{bmatrix}, \ u \in [u_0, u_1]. \)

Initial conditions:
\( x(0) = \alpha(u_0), \ y(0) = \beta(u_0), \ \theta(0) = \tau(u_0) \)

Freely choose any \( v(\cdot) \in C^1([0, t_1]) \)
with \( v(t) > 0 \ \forall t \in [0, t_1] \)
and \( \int_{0}^{t_1} v(\xi) \, d\xi = s_1 \)

Then, the angular velocity be given by
\[
\omega(t) \triangleq v(t) \kappa(s) \bigg|_{s=\int_{0}^{t} v(\xi) \, d\xi}
\]
The problem and a first result

\[ \dot{\omega}(t) = \dot{v}(t) \kappa \left( \int_0^t v(\xi) d\xi \right) + v(t) \frac{d}{dt} \left[ \kappa \left( \int_0^t v(\xi) d\xi \right) \right] \]

\[ = \dot{v}(t) \kappa \left( \int_0^t v(\xi) d\xi \right) + v^2(t) \frac{d\kappa}{ds} \left( \int_0^t v(\xi) d\xi \right) \]

\[ \Rightarrow \omega(\cdot) \in C^1([0,t_1)) \]

To complete the proof show that the following time functions are a solution of the unicycle model...

\[ x(t) = \alpha(u) \bigg|_{u=f^{-1}\left( \int_0^t v(\xi) d\xi \right)} \quad y(t) = \beta(u) \bigg|_{u=f^{-1}\left( \int_0^t v(\xi) d\xi \right)} \]

\[ \theta(t) = \arg\left\{ \tau(u) \right\} \bigg|_{u=f^{-1}\left( \int_0^t v(\xi) d\xi \right)} \]
The problem and a first result

• Proposition 1 makes evident that to solve the PIE problem with $v(t) \neq 0$ the given $\Gamma$ must be a $G^3$-path.

• To generate a $G^3$-path it is not necessary to guarantee $v(t) \neq 0$ for any $t$ …
An algorithm for the path inversion extended problem

The PIE problem: Find $\nu(\cdot), \omega(\cdot) \in C^1\left([0, t_1]\right)$ to steer along $\Gamma$ the WMR from $A = \{p_A, \theta_A, \nu_A, \omega_A, \dot{\nu}_A, \dot{\omega}_A\}$ to $B = \{p_B, \theta_B, \nu_B, \omega_B, \dot{\nu}_B, \dot{\omega}_B\}$.

Solution proposed:
I. Path $\Gamma$ connecting $p_A$ with $p_B$ is a $G^3$-path. [almost necessary]
II. $\nu(t) \neq 0 \ \forall t \in (0, t_1)$. [advisable]

Restrictions on $A$ and $B$ by requirement I:
If $\nu_A = 0$ then $\omega_A = 0$; If $\nu_A = 0 \land \dot{\nu}_A = 0$ then $\omega_A = 0 \land \dot{\omega}_A = 0$;
If $\nu_B = 0$ then $\omega_B = 0$; If $\nu_B = 0 \land \dot{\nu}_B = 0$ then $\omega_B = 0 \land \dot{\omega}_B = 0$. 
An algorithm for the path inversion extended problem

Restrictions on $A$ and $B$ by requirement II:
If $v_A > 0$ then $v_B \geq 0$; If $v_A > 0 \land v_B = 0$ then $\dot{v}_B \leq 0$;
If $v_A < 0$ then $v_B \leq 0$; If $v_A < 0 \land v_B = 0$ then $\dot{v}_B \geq 0$;
If $v_A = 0 \land \dot{v}_A > 0$ then $v_B \geq 0$; If $v_A = 0 \land \dot{v}_A > 0 \land v_B = 0$ then $\dot{v}_B \leq 0$;
If $v_A = 0 \land \dot{v}_A < 0$ then $v_B \leq 0$; If $v_A = 0 \land \dot{v}_A < 0 \land v_B = 0$ then $\dot{v}_B \geq 0$.

Illustrating the second statement:
An algorithm for the path inversion extended problem

A four step algorithm

**Step 1:** Decide for a forward or backward movement \((v > 0 \text{ or } v < 0)\) of the WMR.

If \(v_A = v_B = 0\) and \(\dot{v}_A = \dot{v}_B = 0\) then the extended state \(\mathbf{B}\) can be reached with a FM or a BM.
An algorithm for the path inversion extended problem

The supervisor can choose according to the following selection criteria:

<table>
<thead>
<tr>
<th></th>
<th>FM</th>
<th>BM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(v_A &gt; 0) \land (v_B &gt; 0)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(v_A &gt; 0) \land (v_B = 0) \land (\dot{v}_B \leq 0)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(v_A = 0) \land (v_B &gt; 0) \land (\dot{v}_A \geq 0)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(v_A = 0) \land (v_B = 0) \land (\dot{v}_A \geq 0) \land (\dot{v}_B &lt; 0)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(v_A = 0) \land (v_B = 0) \land (\dot{v}_A &gt; 0) \land (\dot{v}_B \leq 0)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(v_A = 0) \land (v_B &lt; 0)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(v_A &lt; 0) \land (v_B = 0) \land (\dot{v}_B \geq 0)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(v_A = 0) \land (v_B &lt; 0) \land (\dot{v}_A \leq 0)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(v_A = 0) \land (v_B = 0) \land (\dot{v}_A \leq 0) \land (\dot{v}_B &gt; 0)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(v_A = 0) \land (v_B = 0) \land (\dot{v}_A &lt; 0) \land (\dot{v}_B \geq 0)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(v_A = v_B = 0) \land (\dot{v}_A = \dot{v}_B = 0)$</td>
<td>FM or BM</td>
<td>BM</td>
</tr>
</tbody>
</table>
An algorithm for the path inversion extended problem

Step 2: (Path Planning) Choose a G³-path \( \Gamma \) connecting \( p_A \) with \( p_B \). \( \Gamma \) must satisfy geometric interpolating conditions at the endpoints that depend on the extended states \( A \) and \( B \).

Path \( \Gamma \) is to be defined by the G³-curve \( p(u), \ u \in [u_0, u_1] \)

\[
\begin{align*}
p(u_0) &= p_A, \quad p(u_1) = p_B \\
\tau(u_0) &= \begin{cases} 
[\cos \theta_A \sin \theta_A]^T & \text{if FM} \\
- [\cos \theta_A \sin \theta_A]^T & \text{if BM} 
\end{cases} \\
\tau(u_1) &= \begin{cases} 
[\cos \theta_B \sin \theta_B]^T & \text{if FM} \\
- [\cos \theta_B \sin \theta_B]^T & \text{if BM} 
\end{cases}
\end{align*}
\]

To assign endpoint curvatures and curvature derivatives two cases are to be considered: \( v_A \neq 0 \) and \( v_B \neq 0 \) (general case); \( v_A = 0 \) and/or \( v_B = 0 \) (critical case).
An algorithm for the path inversion extended problem

general case:

\[ \kappa_A \triangleq \kappa(0) = \begin{cases} \frac{\omega_A}{v_A} & \text{if FM} \\ -\frac{\omega_A}{v_A} & \text{if BM} \end{cases} \]

\[ \dot{\kappa}_A \triangleq \dot{\kappa}(0) = \frac{\ddot{\omega}_A v_A - \omega_A \dot{v}_A}{v_A^3} \]

\[ \kappa_B \triangleq \kappa(s_1) = \begin{cases} \frac{\omega_B}{v_B} & \text{if FM} \\ -\frac{\omega_B}{v_B} & \text{if BM} \end{cases} \]

\[ \dot{\kappa}_B \triangleq \dot{\kappa}(s_1) = \frac{\ddot{\omega}_B v_B - \omega_B \dot{v}_B}{v_B^3} \]

critical cases:

If \( v_A = 0 \wedge \dot{v}_A = 0 \wedge \omega_A = 0 \wedge \dot{\omega}_A = 0 \) then choose any \( \kappa_A \) and \( \dot{\kappa}_A \).

If \( v_B = 0 \wedge \dot{v}_B = 0 \wedge \omega_B = 0 \wedge \dot{\omega}_B = 0 \) then choose any \( \kappa_B \) and \( \dot{\kappa}_B \).
An algorithm for the path inversion extended problem

critical cases:

If \( v_A = 0 \land \omega_A = 0 \land \dot{v}_A \neq 0 \) then \( \kappa_A \triangleq \begin{cases} \frac{\dot{\omega}_A}{\dot{v}_A} & \text{if } \text{FM} (\dot{v}_A > 0) \\ \frac{\dot{\omega}_A}{\dot{v}_A} & \text{if } \text{BM} (\dot{v}_A < 0) \end{cases} \), choose any \( \dot{\kappa}_A \).

If \( v_B = 0 \land \omega_B = 0 \land \dot{v}_B \neq 0 \) then \( \kappa_B \triangleq \begin{cases} \frac{\dot{\omega}_B}{\dot{v}_B} & \text{if } \text{FM} (\dot{v}_B > 0) \\ \frac{\dot{\omega}_B}{\dot{v}_B} & \text{if } \text{BM} (\dot{v}_B < 0) \end{cases} \), choose any \( \dot{\kappa}_B \).

The supervisor has still plenty of freedom in shaping the path \( \Gamma \): use \( \eta^3\)-splines (also called \( G^3\)-splines) [Piazzi, Romano, Guarino 2003 ECC], a straightforward and very flexible tool (next lesson…).
An algorithm for the path inversion extended problem

Step 3: (Velocity Planning)

Choose \( v(\cdot) \in C^1([0, t_1]) \) with \( v(t) \neq 0 \ \forall t \in (0, t_1) \)
such that

\[
\begin{align*}
v(0) &= v_A, \quad v(t_1) = v_B, \\
v'(0) &= v'_A, \quad v'(t_1) = v'_B,
\end{align*}
\]

\[
\int_0^{t_1} v(\xi) d\xi = \begin{cases} 
  s_1 & \text{if FM} \\
  -s_1 & \text{if BM}
\end{cases}
\]

A solution with minimum jerk is provided by the five cubic spline planning of Guarino, Piazzi, Romano (2004 IV).
An algorithm for the path inversion extended problem

Step 4: (Computing the angular velocity)

The angular velocity $\omega(\cdot) \in C^1([0, t_1])$ is computed according to (for both FM and BM):

$$\omega(t) \triangleq \left| v(t) \right| \cdot \kappa(s) \left| s = \int_0^t v(\xi) d\xi \right.$$
An algorithm for the path inversion extended problem

**Proposition 2**

Consider any traveling time $t_1 > 0$ and any extended states $A$ and $B$ satisfying the restrictions imposed by requirements I and II. Then, the velocities $v(\cdot), \omega(\cdot) \in C^1([0, t_1])$ synthesized by the four step algorithm steer the WMR from $A$ at time 0 to $B$ at time $t_1$ in such a way that the Cartesian path of the WMR exactly matches the $G^3$-path $\Gamma$ planned at step 2 of the algorithm.

**Proof:** Basically, it is a slight extension of the proof provided for Proposition 1 (sufficiency part). For details Guarino, Piazzi, Romano 2004 TR.
An example

Extended states (expressed in m, rad, m/s, rad/s, m/s², rad/s²):

\[
\mathbf{A} = \begin{bmatrix} \mathbf{2} \\ \mathbf{1} \end{bmatrix}, \quad \frac{\pi}{4}, \quad 0, \quad 0, \quad 0, \quad 0
\]

\[
\mathbf{B} = \begin{bmatrix} \mathbf{4} \\ \mathbf{3} \end{bmatrix}, \quad -\frac{\pi}{6}, \quad 0.5, \quad 0.5, \quad 0, \quad 0.05
\]

Γ is planned using the η³-spline, a seven order polynomial curve that interpolates two Cartesian points with arbitrary assigned unit tangent vectors, curvatures and curvature derivatives.
An example

The planned linear and angular velocities used as feedforward control in the time interval [0, 4] \( (t_1 = 4 \text{ s}) \) (piecewise cubic spline planning with minimization of the maximum jerk).
Iterative steering and supervision

Iterative steering updating:

At time $t_1$ the actual state of the WMR is $\tilde{p}_B, \tilde{\theta}_B$.

So, the extended state at time $t_1$ is

$$\tilde{B} = \{\tilde{p}_B, \tilde{\theta}_B, v_B, \omega_B, \dot{v}_B, \dot{\omega}_B\}$$

to issue the next steering the supervisor does:

- $A \leftarrow \tilde{B}$;
- Choose a new future extended state $B$ to be reached at the new time $t_1$;
- Apply the four step algorithm (a new $\Gamma$ is planned to join $p_A$ with $p_A$).

The updating can be done before time $t_1$ …

A latency time to accommodate the computational time for the updating must be considered …
Iterative steering and supervision

A supervisory architecture for iterative steering (and autonomous navigation)
Iterative steering and supervision

Ideas for motion control supervision and autonomous navigation

• Iterative learning
• Fault detection
• Obstacle avoidance
• Artificial intelligence
• Sensor fusion
Summary

- We have introduced the concept of extended state of WMR (state + inputs and their derivatives)
- We have solved an extended problem of dynamic path inversion for WMRs
- We have discussed iterative steering and supervision
- To complete the picture (don’t miss Corrado’s lessons!):
  1. path generation and shaping using $\eta^3$-splines;
  2. velocity planning with jerk minimization.
References