Symmetry-Preserving Control System for Trajectory Tracking of a Nonholonomic Mobile Robot

Marcin Nowicki  Krzysztof Kozlowski  Dariusz Pazderski

Poznan University of Technology  Chair of Control and Systems Engineering

The Israeli Conference on Robotics 2013
Motivation

Practical:

We simply want autonomous vehicles!

Theoretical point of view:

- nonlinear equation
- underactuated
- nonholonomic

Still an open issue!
Motivation

Practical:

We simply want autonomous vehicles!

Theoretical point of view:

• nonlinear equation

• underactuated

• nonholonomic

Still an open issue!
Motivation

Practical:

We simply want autonomous vehicles!

Theoretical point of view:

- nonlinear equation
- underactuated
- nonholonomic

Still an open issue!
Motivation

Practical:

We simply want autonomous vehicles!

Theoretical point of view:

- nonlinear equation
- underactuated
- nonholonomic

Still an open issue!
Motivation

Practical:

We simply want autonomous vehicles!

Theoretical point of view:

- nonlinear equation
- underactuated
- nonholonomic

Still an open issue!
Problem statement

Task - Trajectory Tracking

Task: pursue the trajectory given by the reference robot in time regime

In the best case: global convergence and robustness

Configuration: \( q = (x, y, \theta)^T \)
- translation and rotation on a plane

Our approach: use a special property of the system, namely symmetry.

\( (x, y) \) - relatively easy to measure (e.g. GPS)
\( \theta \) - hard and noisy measurements
Problem statement

Task - Trajectory Tracking

Task: pursue the trajectory given by the reference robot in time regime

In the best case: global convergence and robustness

Configuration: \( q = (x, y, \theta)^T \)

- translation and rotation on a plane
- \((x, y)\) - relatively easy to measure (e.g. GPS)
- \(\theta\) - hard and noisy measurements

Our approach: use a special property of the system, namely symmetry.
Problem statement

Task - Trajectory Tracking

Task: pursue the trajectory given by the reference robot in time regime

In the best case: global convergence and robustness

Configuration: $q = (x, y, \theta)^T$
translation and rotation on a plane

$(x, y)$ - relatively easy to measure (e.g. GPS)
$\theta$ - hard and noisy measurements

Our approach: use a special property of the system, namely symmetry.
Problem statement

Task - Trajectory Tracking

Task: pursue the trajectory given by the reference robot in time regime

In the best case: global convergence and robustness

Configuration: $q = (x, y, \theta)^T$

$\theta$ - hard and noisy measurements

Our approach: use a special property of the system, namely symmetry.

$\theta$ - relatively easy to measure (e.g. GPS)
Problem statement

Task - Trajectory Tracking

**Task**: pursue the trajectory given by the reference robot in time regime

In the best case: global convergence and robustness

Configuration: \( q = (x, y, \theta)^T \)
- translation and rotation on a plane

\((x, y)\) - relatively easy to measure (e.g. GPS)
- \(\theta\) - hard and noisy measurements

Our approach: use a special property of the system, namely symmetry.
**Problem statement**

Why that approach?

curved configuration space, not flat

usually: \( q \in \mathbb{R}^2 \)

result: push off the configuration space

use angle \( \theta \) to encode; \( q \in SO(2) \)

result: always stays on the manifold

Use Lie group to represent configuration space

For this we need invariance!
Problem statement

Why that approach?

curved configuration space, not flat

usually: \( q \in \mathbb{R}^2 \)

result: push off the configuration space

use angle \( \theta \) to encode; \( q \in SO(2) \)

result: always stays on the manifold

Use Lie group to represent configuration space

For this we need invariance!
Problem statement

Why that approach?

curved configuration space, not flat

usually: \( q \in \mathbb{R}^2 \)

result: push off the configuration space

use angle \( \theta \) to encode; \( q \in SO(2) \)

result: always stays on the manifold

Use Lie group to represent configuration space

For this we need invariance!
Problem statement

Why that approach?

curved configuration space, not flat

usually: \( q \in \mathbb{R}^2 \)

result: push off the configuration space

use angle \( \theta \) to encode; \( q \in SO(2) \)

result: always stays on the manifold

For this we need invariance!
Problem statement

Why that approach?

curved configuration space, not flat

usually: \( q \in \mathbb{R}^2 \)

result: push off the configuration space

use angle \( \theta \) to encode; \( q \in SO(2) \)

result: always stays on the manifold

Use Lie group to represent configuration space

For this we need invariance!
Problem statement

Why that approach?

curved configuration space, not flat

usually: \( q \in \mathbb{R}^2 \)

result: push off the configuration space

use angle \( \theta \) to encode; \( q \in SO(2) \)

result: always stays on the manifold

Use Lie group to represent configuration space

For this we need invariance!
Problem statement

Why that approach?

curved configuration space, not flat

usually: $q \in \mathbb{R}^2$

result: push off the configuration space

use angle $\theta$ to encode; $q \in SO(2)$

result: always stays on the manifold

Use Lie group to represent configuration space

For this we need invariance!
Lie groups

Lie group: a group that is also a manifold with an action (group multiplication).

A dynamic system:
\[ \dot{q} = f(q, u) \]
\[ z = h(q, u) \]

A Lie group \( G \) with an element: \( g \in G \)

Lie group transformation:
- state: \( \varphi_g(q) \)
- input: \( \psi_g(u) \)
- output: \( \rho_g(z) \)

The symmetry group maps solution to solution.
Lie groups

Lie group: a group that is also a manifold with an action (group multiplication).

\[
\dot{q} = f(q, u) \\
z = h(q, u)
\]

A Lie group \( G \) with an element: \( g \in G \)

The symmetry group maps solution to solution

A dynamic system:

- state: \( \varphi_g(q) \)
- input: \( \psi_g(u) \)
- output: \( \rho_g(z) \)
Lie groups

Lie group: a group that is also a manifold with an action (group multiplication).

A dynamic system:
\[
\begin{align*}
\dot{q} &= f(q, u) \\
z &= h(q, u)
\end{align*}
\]
A Lie group $G$ with an element: $g \in G$

Lie group transformation:
- state: $\varphi_g(q)$
- input: $\psi_g(u)$
- output: $\rho_g(z)$

The symmetry group maps solution to solution
Lie groups

Lie group: a group that is also a manifold with an action (group multiplication).

A dynamic system:
\[
\begin{align*}
\dot{q} &= f(q, u) \\
z &= h(q, u)
\end{align*}
\]

A Lie group \( G \) with an element: \( g \in G \)

The symmetry group maps solution to solution

Lie group transformation:
- state: \( \varphi_g(q) \)
- input: \( \psi_g(u) \)
- output: \( \rho_g(z) \)
### G-invariant control system

**Definition**

System $\dot{q} = f(q, u)$ is $G$-invariant if:

$$f(\phi_g(q), \psi_g(u)) = \frac{\partial}{\partial q} \phi_g(q) f(q, u)$$

$$h(\phi_g(q), \psi_g(u)) = \rho_g(h(q, u))$$

action on the system remains the dynamics unchanged.

In our case (a mobile robot) we have:

- a group: $G = SE(2)$
- a state space: $Q : SO(2) \times \mathbb{R}^2 = SE(2)$
- $\phi_g(q)$ - same as group multiplication

Invariance expressed by the independence of the coordinate changes.
**G-invariant control system**

**Definition**

System $\dot{q} = f(q, u)$ is $G$-invariant if:

$$f(\varphi_g(q), \psi_g(u)) = \frac{\partial}{\partial q} \varphi_g(q) f(q, u)$$

$$h(\varphi_g(q), \psi_g(u)) = \rho_g(h(q, u))$$

action on the system remains the dynamics unchanged.

In our case (a mobile robot) we have:

- a group: $G = SE(2)$
- a state space: $Q : SO(2) \rtimes \mathbb{R}^2 = SE(2)$
- $\varphi_g(q)$ - same as group multiplication

Invariance expressed by the independence of the coordinate changes.
**G-invariant control system**

**Definition**

System \( \dot{q} = f(q, u) \) is \( G \)-invariant if:

\[
\begin{align*}
    f(\varphi_g(q), \psi_g(u)) &= \frac{\partial}{\partial q} \varphi_g(q)f(q, u) \\
    h(\varphi_g(q), \psi_g(u)) &= \rho_g(h(q, u))
\end{align*}
\]

action on the system remains the dynamics unchanged.

In our case (a mobile robot) we have:

- a group: \( G = SE(2) \)
- a state space: \( Q : SO(2) \rtimes \mathbb{R}^2 = SE(2) \)
- \( \varphi_g(q) \) - same as group multiplication

Invariance expressed by the independence of the coordinate changes.
G-invariant control system

Invariant tracking error

• Typical form $q_r - q$ brakes symmetry.

• using Cartan moving frame method we construct new one:
  $$\eta = \varphi_{\gamma(q_r)}(q) - \varphi_{\gamma(q_r)}(q_r)$$

• the same for the estimation output error:
  $$E = \rho_{\gamma(\hat{q})}({\hat{q}}) - \rho_{\gamma(\hat{q})}(q)$$
G-invariant control system

Invariant tracking error

• Typical form $q_r - q$ brakes symmetry.

• using Cartan moving frame method we construct new one:

$$\eta = \varphi_{\gamma(q_r)}(q) - \varphi_{\gamma(q_r)}(q_r)$$

• the same for the estimation output error:

$$E = \rho_{\gamma(\hat{q})}(\hat{q}) - \rho_{\gamma(\hat{q})}(q)$$
**G-invariant control system**

**Invariant tracking error**

- Typical form $q_r - q$ brakes symmetry.

- Using Cartan moving frame method we construct new one:
  \[
  \eta = \varphi_{\gamma(q_r)}(q) - \varphi_{\gamma(q_r)}(q_r)
  \]

- The same for the estimation output error:
  \[
  E = \rho_{\gamma(q)}(\hat{q}) - \rho_{\gamma(q)}(q)
  \]
G-invariant control system

Invariant tracking error

• Typical form $q_r - q$ brakes symmetry.

• using Cartan moving frame method we construct new one:

$$\eta = \varphi_{\gamma(q_r)}(q) - \varphi_{\gamma(q_r)}(q_r)$$

• the same for the estimation output error:

$$E = \rho_{\gamma(\hat{q})}(\hat{q}) - \rho_{\gamma(\hat{q})}(q)$$
**Definition**

Controller $u = k(q, q_r, u_r)$ is $G$-invariant if:

$$k(\varphi_g(q), \varphi_g(q_r), \psi_g(u)) = \psi_g(q, k(q, q_r, u_r))$$

- stabilizes the system on reference (admissible) trajectory $q_r$.
- we use simple linear state feedback (it works!).
$G$-invariant controller

**Definition**

Controller $u = k(q, q_r, u_r)$ is $G$-invariant if:

$$k(\varphi_g(q), \varphi_g(q_r), \psi_g(u)) = \psi_g(q, k(q, q_r, u_r))$$

- stabilizes the system on reference (admissible) trajectory $q_r$.
- we use simple linear state feedback (it works!).
**G-invariant controller**

**Definition**

Controller \( u = k(q, q_r, u_r) \) is G-invariant if:

\[
k(\varphi_g(q), \varphi_g(q_r), \psi_g(u)) = \psi_g(q, k(q, q_r, u_r))
\]

- stabilizes the system on reference (admissible) trajectory \( q_r \).
- we use simple linear state feedback (it works!).
G-invariant observer

**Definition**

G-invariant observer is a system in the form:

\[
\dot{\hat{q}} = f(\hat{q}, u) + W(\hat{q}) \mathcal{L}(\mathcal{I}, E) E
\]

- \(W(\hat{q})\) - invariant moving frame,
- \(\mathcal{L}(\mathcal{I}, E)\) - “gain matrix”

it is something like a nonlinear observer:

\[
\dot{\hat{q}} = f(\hat{q}, u) + L(\cdot) (h(\hat{q}, u) - z)
\]

(however any other: Kalman, HGO, Sliding-mode)

- respects symmetry
- the vector field is oriented by moving frame.
- deals with the lack of measurement.
# G-invariant observer

## Definition

G-invariant observer is a system in the form:

\[
\dot{\hat{q}} = f(\hat{q}, u) + W(\hat{q}) \mathcal{L}(\mathcal{I}, E) E
\]

- \( W(\hat{q}) \) - invariant moving frame,
- \( \mathcal{L}(\mathcal{I}, E) \) - “gain matrix”

It is something like a nonlinear observer:

\[
\dot{\hat{q}} = f(\hat{q}, u) + L(\cdot) (h(\hat{q}, u) - z)
\]

(However any other: Kalman, HGO, Sliding-mode)

- respects symmetry
- the vector field is oriented by moving frame.
- deals with the lack of measurement.
**G-invariant observer**

**Definition**

G-invariant observer is a system in the form:

\[ \dot{\hat{q}} = f(\hat{q}, u) + W(\hat{q}) \mathcal{L}(\mathcal{I}, E) E \]

- \( W(\hat{q}) \) - invariant moving frame,
- \( \mathcal{L}(\mathcal{I}, E) \)- “gain matrix”

It is something like a nonlinear observer:

\[ \dot{\hat{q}} = f(\hat{q}, u) + L(\cdot) (h(\hat{q}, u) - z) \]

(however any other: Kalman, HGO, Sliding-mode)

- respects symmetry
- the vector field is oriented by moving frame.
- deals with the lack of measurement.
**G-invariant observer**

**Definition**

G-invariant observer is a system in the form:

\[ \dot{\hat{q}} = f(\hat{q}, u) + W(\hat{q}) \mathcal{L}(I, E) E \]

- \(W(\hat{q})\) - invariant moving frame,
- \(\mathcal{L}(I, E)\) - “gain matrix”

it is something like a nonlinear observer:

\[ \dot{\hat{q}} = f(\hat{q}, u) + L(\cdot) (h(\hat{q}, u) - z) \]

(however any other: Kalman, HGO, Sliding-mode)

- respects symmetry

- the vector field is oriented by moving frame.

- deals with the lack of measurement.
**G-invariant observer**

**Definition**

G-invariant observer is a system in the form:

\[
\dot{\hat{q}} = f(\hat{q}, u) + W(\hat{q}) \mathcal{L}(\mathcal{I}, E) E
\]

- \( W(\hat{q}) \) - invariant moving frame,
- \( \mathcal{L}(\mathcal{I}, E) \) - "gain matrix"

It is something like a nonlinear observer:

\[
\dot{\hat{q}} = f(\hat{q}, u) + L(\cdot) (h(\hat{q}, u) - z)
\]

(however any other: Kalman, HGO, Sliding-mode)

- respects symmetry
- the vector field is oriented by moving frame.
- deals with the lack of measurement.
Case study

Mobile robot model:
\[
\begin{align*}
\dot{x} &= v \cos \theta \\
\dot{y} &= v \sin \theta \\
\dot{\theta} &= \omega
\end{align*}
\]
\[
z = (x, y)^T
\]

Lie group Transformations:
\[
\begin{align*}
\varphi_g(q) &= \begin{pmatrix} \text{Rot}(\theta_g) & (x) \\ (y) & \theta + \theta_g \end{pmatrix} \\
\psi_g(u) &= \begin{pmatrix} v \\ \omega \end{pmatrix} \\
\rho_g(z) &= \text{Rot}(\theta_g) \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_g \\ y_g \end{pmatrix}
\end{align*}
\]

Invariant tracking error:
\[
\eta = \begin{pmatrix} \text{Rot}^T(\theta_r) \begin{pmatrix} x - x_r \\ y - y_r \end{pmatrix} \\ \theta - \theta_r \end{pmatrix}
\]

Invariant estimation output error:
\[
E = \text{Rot}^T(\hat{\theta}) \begin{pmatrix} \hat{x} - x \\ \hat{y} - y \end{pmatrix}
\]
Case study

Controller:
\[ v = v_r - k_1 \eta_x \]
\[ \omega = \omega_r - k_2 \eta_y - k_3 \eta_\omega \]

Observer:
\[
\begin{pmatrix}
\dot{\hat{x}} \\
\dot{\hat{y}} \\
\dot{\hat{\theta}}
\end{pmatrix}
= \begin{pmatrix}
v \cos \hat{\theta} \\
v \sin \hat{\theta} \\
\omega
\end{pmatrix}
+ \begin{pmatrix}
\text{Rot}(\hat{\theta}) & 0 \\
0 & 1
\end{pmatrix}
\mathcal{L}(I, E) \text{Rot}^T(\hat{\theta})
\begin{pmatrix}
\hat{x} - x \\
\hat{y} - y
\end{pmatrix},
\]

with:
\[
\mathcal{L}(I, E) = \begin{pmatrix}
-|v|a & vbE_y - \omega \\
\omega - vbE_y & -|v|c \\
0 & -vb
\end{pmatrix}
\]
Results

E1. Nominal.
Results

E2. Noise (an aleatory vector field).
Conclusions

- symmetry-preserving control system has been presented - all details in the paper

- uses geometric property to get more precise result

- utilitarian approach to a variety of dynamic systems

- stable and robust
Conclusions

• symmetry-preserving control system has been presented - all details in the paper

• uses geometric property to get more precise result

• utilitarian approach to a variety of dynamic systems

• stable and robust
Conclusions

• symmetry-preserving control system has been presented - all details in the paper

• uses geometric property to get more precise result

• utilitarian approach to a variety of dynamic systems

• stable and robust
Conclusions

• symmetry-preserving control system has been presented - all details in the paper

• uses geometric property to get more precise result

• utilitarian approach to a variety of dynamic systems

• stable and robust
Thank you
Thank you.
Thank you..
I do not know...