Trajectory correction algorithms for a 3D underwater vehicle using affine transformations

Quang-Cuong Pham and Yoshihiko Nakamura

Nakamura-Takano Laboratory
Department of Mechano-Informatics
University of Tokyo

Trajectory deformation

- Planning trajectories for nonholonomic robots (e.g. cars, submarines, quadrotors, satellites,...) is difficult and time-consuming

Trajectory deformation

- Planning trajectories for nonholonomic robots (e.g. cars, submarines, quadrotors, satellites,...) is difficult and time-consuming
- Better to deform a previously planned trajectory than re-plan anew

Trajectory deformation

- Planning trajectories for nonholonomic robots (e.g. cars, submarines, quadrotors, satellites,...) is difficult and time-consuming
- Better to deform a previously planned trajectory than re-plan anew
- Existing methods: e.g.
  - inputs perturbation (e.g. Lamiraux et al, IEEE T Rob 2004)
  - Euclidean transformations (e.g. Cheng et al, IEEE T Rob 2008; Seiler et al, WAFR 2010)

Trajectory deformation

- Planning trajectories for nonholonomic robots (e.g. cars, submarines, quadrotors, satellites,...) is difficult and time-consuming
- Better to deform a previously planned trajectory than re-plan anew
- Existing methods: e.g.
  - inputs perturbation (e.g. Lamiraux et al, IEEE T Rob 2004)
  - Euclidean transformations (e.g. Cheng et al, IEEE T Rob 2008; Seiler et al, WAFR 2010)
- Drawbacks of these methods:

Trajectory deformation

- Planning trajectories for nonholonomic robots (e.g. cars, submarines, quadrotors, satellites,...) is difficult and time-consuming
- Better to deform a previously planned trajectory than re-plan anew
- Existing methods: e.g.
  - inputs perturbation (e.g. Lamiraux et al, IEEE T Rob 2004)
  - Euclidean transformations (e.g. Cheng et al, IEEE T Rob 2008; Seiler et al, WAFR 2010)
- Drawbacks of these methods:
  - iterative search/gradient descent to find the appropriate deformation

Trajectory deformation

- Planning trajectories for nonholonomic robots (e.g. cars, submarines, quadrotors, satellites,...) is difficult and time-consuming
- Better to deform a previously planned trajectory than re-plan anew
- Existing methods: e.g.
  - inputs perturbation (e.g. Lamiraux et al, IEEE T Rob 2004)
  - Euclidean transformations (e.g. Cheng et al, IEEE T Rob 2008; Seiler et al, WAFR 2010)
- Drawbacks of these methods:
  - iterative search/gradient descent to find the appropriate deformation
  - require trajectory re-integration at each step

Trajectory deformation

- Planning trajectories for nonholonomic robots (e.g. cars, submarines, quadrotors, satellites,...) is difficult and time-consuming
- Better to deform a previously planned trajectory than re-plan anew
- Existing methods: e.g.
  - inputs perturbation (e.g. Lamiraux et al, IEEE T Rob 2004)
  - Euclidean transformations (e.g. Cheng et al, IEEE T Rob 2008; Seiler et al, WAFR 2010)
- Drawbacks of these methods:
  - iterative search/gradient descent to find the appropriate deformation
  - require trajectory re-integration at each step
  - approximate corrections

Trajectory deformation

- Planning trajectories for nonholonomic robots (e.g. cars, submarines, quadrotors, satellites,...) is difficult and time-consuming
- Better to deform a previously planned trajectory than re-plan anew
- Existing methods: e.g.
  - inputs perturbation (e.g. Lamiraux et al, IEEE T Rob 2004)
  - Euclidean transformations (e.g. Cheng et al, IEEE T Rob 2008; Seiler et al, WAFR 2010)
- Drawbacks of these methods:
  - iterative search/gradient descent to find the appropriate deformation
  - require trajectory re-integration at each step
  - approximate corrections
- Advantages of the proposed method based on affine transformations (Pham, RSS 2011):

Trajectory deformation

- Planning trajectories for nonholonomic robots (e.g. cars, submarines, quadrotors, satellites,...) is difficult and time-consuming
- Better to deform a previously planned trajectory than re-plan anew
- Existing methods: e.g.
  - inputs perturbation (e.g. Lamiraux et al, IEEE T Rob 2004)
  - Euclidean transformations (e.g. Cheng et al, IEEE T Rob 2008; Seiler et al, WAFR 2010)
- Drawbacks of these methods:
  - iterative search/gradient descent to find the appropriate deformation
  - require trajectory re-integration at each step
  - approximate corrections
- Advantages of the proposed method based on affine transformations (Pham, RSS 2011):
  - single step (no iterative search/gradient descent)

Trajectory deformation

- Planning trajectories for nonholonomic robots (e.g. cars, submarines, quadrotors, satellites,...) is difficult and time-consuming
- Better to deform a previously planned trajectory than re-plan anew
- Existing methods: e.g.
  - inputs perturbation (e.g. Lamiraux et al, IEEE T Rob 2004)
  - Euclidean transformations (e.g. Cheng et al, IEEE T Rob 2008; Seiler et al, WAFR 2010)
- Drawbacks of these methods:
  - iterative search/gradient descent to find the appropriate deformation
  - require trajectory re-integration at each step
  - approximate corrections
- Advantages of the proposed method based on affine transformations (Pham, RSS 2011):
  - single step (no iterative search/gradient descent)
  - no trajectory re-integration

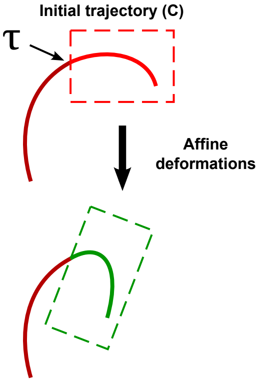
Trajectory deformation

- Planning trajectories for nonholonomic robots (e.g. cars, submarines, quadrotors, satellites,...) is difficult and time-consuming
- Better to deform a previously planned trajectory than re-plan anew
- Existing methods: e.g.
  - inputs perturbation (e.g. Lamiraux et al, IEEE T Rob 2004)
  - Euclidean transformations (e.g. Cheng et al, IEEE T Rob 2008; Seiler et al, WAFR 2010)
- Drawbacks of these methods:
  - iterative search/gradient descent to find the appropriate deformation
  - require trajectory re-integration at each step
  - approximate corrections
- Advantages of the proposed method based on affine transformations (Pham, RSS 2011):
  - single step (no iterative search/gradient descent)
  - no trajectory re-integration
  - exact, algbraic, corrections

Affine trajectory deformation

- A transformation $\mathcal{F}$ deforms a trajectory $\mathcal{C} = (x(t), y(t))_{t \in [0,T]}$ into $\mathcal{C}'$ at a time instant τ by

$$\begin{array}{ll} \forall t < \tau & \mathcal{C}'(t) = \mathcal{C}(t) \\ \forall t \geq \tau & \mathcal{C}'(t) = \mathcal{F}(\mathcal{C}(t)) \end{array}$$

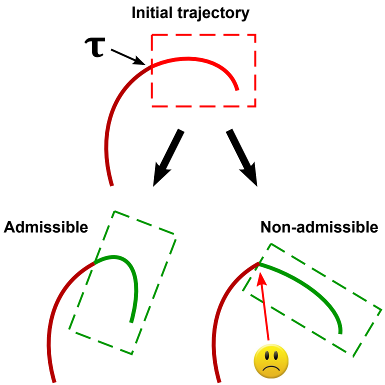


Affine trajectory deformation

- A transformation $\mathcal{F}$ deforms a trajectory $\mathcal{C} = (x(t), y(t))_{t \in [0,T]}$ into $\mathcal{C}'$ at a time instant τ by

$$\begin{array}{ll} \forall t < \tau & \mathcal{C}'(t) = \mathcal{C}(t) \\ \forall t \geq \tau & \mathcal{C}'(t) = \mathcal{F}(\mathcal{C}(t)) \end{array}$$

- Not all affine transformations deform $\mathcal{C}$ into an admissible $\mathcal{C}'$

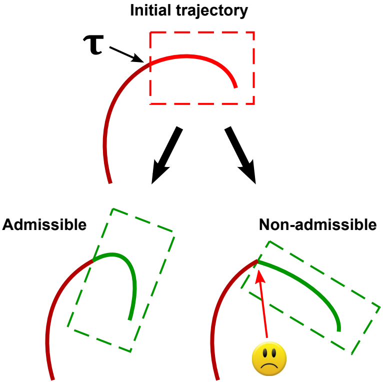


Affine trajectory deformation

- A transformation $\mathcal{F}$ deforms a trajectory $\mathcal{C} = (x(t), y(t))_{t\in[0,T]}$ into $\mathcal{C}'$ at a time instant τ by

$$\begin{array}{ll} \forall t < \tau & \mathcal{C}'(t) = \mathcal{C}(t) \\ \forall t \geq \tau & \mathcal{C}'(t) = \mathcal{F}(\mathcal{C}(t)) \end{array}$$

- Not all affine transformations deform $\mathcal{C}$ into an admissible $\mathcal{C}'$
- How to characterize the set of admissible affine transformations?



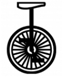
Admissible affine transformations for some systems

Surprisingly, the set of admissible affine transformations can be shown to be a Lie subgroup of the General Affine group (GA_2 or GA_3) of dimension...

-   2 for the unicycle and omni-directional mobile robots (out of the 6 dimensions of GA_2)

Admissible affine transformations for some systems

Surprisingly, the set of admissible affine transformations can be shown to be a Lie subgroup of the General Affine group (GA_2 or GA_3) of dimension...

-   2 for the unicycle and omni-directional mobile robots (out of the 6 dimensions of GA_2)

-   1 for the bicycle or kinematic car (out of the 6 dimensions of GA_2)

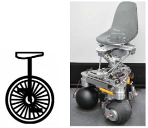
Admissible affine transformations for some systems

Surprisingly, the set of admissible affine transformations can be shown to be a Lie subgroup of the General Affine group (GA_2 or GA_3) of dimension...

-   2 for the unicycle and omni-directional mobile robots (out of the 6 dimensions of GA_2)
-   1 for the bicycle or kinematic car (out of the 6 dimensions of GA_2)
-  4 for the 3D underwater vehicle (out of the 12 dimensions of GA_3)

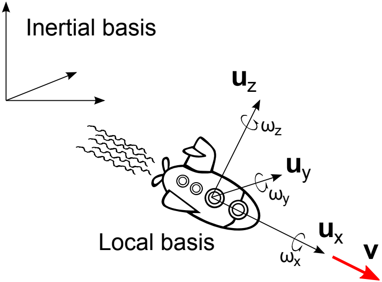
Admissible affine transformations for some systems

Surprisingly, the set of admissible affine transformations can be shown to be a Lie subgroup of the General Affine group (GA_2 or GA_3) of dimension...

-  2 for the unicycle and omni-directional mobile robots (out of the 6 dimensions of GA_2)
-  1 for the bicycle or kinematic car (out of the 6 dimensions of GA_2)
-  4 for the 3D underwater vehicle (out of the 12 dimensions of GA_3)
-  1 for the 3D bevel needle (out of the 12 dimensions of GA_3)

Trajectory correction for a 3D underwater vehicle

Model description:



Position of the robot: (x, y, z)
Orientation of the robot: (ϕ, θ, ψ)

Kinematic equations:

$$\left\{\begin{array}{ccc} \dot{v} & = & a \\ \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} & = & \mathbf{R}(\phi, \theta) \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \\ \dot{x} & = & v \cos\psi \cos\theta \\ \dot{y} & = & v \sin\psi \cos\theta \\ \dot{z} & = & -v \sin\theta \end{array}\right.$$

Trajectory correction for a 3D underwater vehicle (II)

Conditions for a trajectory to be admissible

- The position (x, y, z) must be continuous

Trajectory correction for a 3D underwater vehicle (II)

Conditions for a trajectory to be admissible

- The position (x, y, z) must be continuous
- The orientation (ϕ, θ, ψ) must be continuous

Trajectory correction for a 3D underwater vehicle (II)

Conditions for a trajectory to be admissible

- The position (x, y, z) must be continuous
- The orientation (ϕ, θ, ψ) must be continuous
- The linear velocity v must be continuous (to avoid infinite linear accelerations)

Trajectory correction for a 3D underwater vehicle (II)

Conditions for a trajectory to be admissible

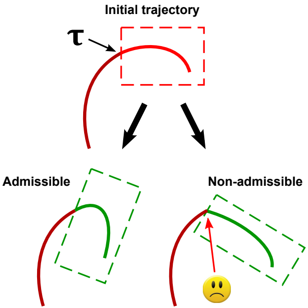
- The position (x, y, z) must be continuous
- The orientation (ϕ, θ, ψ) must be continuous
- The linear velocity v must be continuous (to avoid infinite linear accelerations)
- The angular velocities $(\omega_x, \omega_y, \omega_z)$ must be continuous (e.g. if using rudders)

Trajectory correction for a 3D underwater vehicle (II)

Conditions for a trajectory to be admissible

- The position (x, y, z) must be continuous
- The orientation (ϕ, θ, ψ) must be continuous
- The linear velocity v must be continuous (to avoid infinite linear accelerations)
- The angular velocities $(\omega_x, \omega_y, \omega_z)$ must be continuous (e.g. if using rudders)

Recall:



Trajectory correction for a 3D underwater vehicle (II)

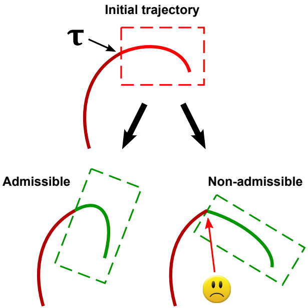
Conditions for a trajectory to be admissible

- The position (x, y, z) must be continuous
- The orientation (ϕ, θ, ψ) must be continuous
- The linear velocity v must be continuous (to avoid infinite linear accelerations)
- The angular velocities $(\omega_x, \omega_y, \omega_z)$ must be continuous (e.g. if using rudders)

In contrast,

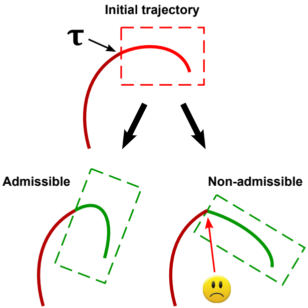
- The linear acceleration a is not required to be continuous

Recall:



Trajectory correction for a 3D underwater vehicle (II)

Recall:



Conditions for a trajectory to be admissible

- The position (x, y, z) must be continuous
- The orientation (ϕ, θ, ψ) must be continuous
- The linear velocity v must be continuous (to avoid infinite linear accelerations)
- The angular velocities $(\omega_x, \omega_y, \omega_z)$ must be continuous (e.g. if using rudders)

In contrast,

- The linear acceleration a is not required to be continuous
- The angular accelerations are not required to be continuous

Trajectory correction for a 3D underwater vehicle (III)

- One can show that, at time τ, the space of affine transformations that guarantee the previous conditions is a subgroup of dimension 4 of GA_3

Trajectory correction for a 3D underwater vehicle (III)

- One can show that, at time τ, the space of affine transformations that guarantee the previous conditions is a subgroup of dimension 4 of GA_3
- These corrections are of the form

$$\forall t \geq \tau \quad C'(t) = C(\tau) + \mathbf{M}(C(t) - C(\tau)),$$

where $C(t) = (x(t), y(t), z(t))$ and the matrix of $\mathbf{M}$ in some well-defined basis has 4 free coefficients (out of 9)

Trajectory correction for a 3D underwater vehicle (III)

- One can show that, at time τ, the space of affine transformations that guarantee the previous conditions is a subgroup of dimension 4 of GA_3
- These corrections are of the form

$$\forall t \geq \tau \quad C'(t) = C(\tau) + \mathbf{M}(C(t) - C(\tau)),$$

  where $C(t) = (x(t), y(t), z(t))$ and the matrix of $\mathbf{M}$ in some well-defined basis has 4 free coefficients (out of 9)
- To correct the final position, one only needs 3 free coefficients $\Rightarrow$ "redundancy"

Trajectory correction for a 3D underwater vehicle (III)

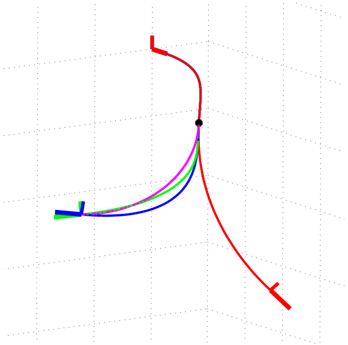
- One can show that, at time τ, the space of affine transformations that guarantee the previous conditions is a subgroup of dimension 4 of GA_3
- These corrections are of the form

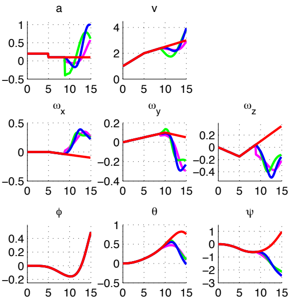
$$\forall t \geq \tau \quad C'(t) = C(\tau) + \mathbf{M}(C(t) - C(\tau)),$$

  where $C(t) = (x(t), y(t), z(t))$ and the matrix of $\mathbf{M}$ in some well-defined basis has 4 free coefficients (out of 9)
- To correct the final position, one only needs 3 free coefficients $\Rightarrow$ "redundancy"
- Note: after obtaining a deformed trajectory $C'(t)$ by the above formula, one can recover the commands $(a, \omega_x, \omega_y, \omega_z)$ by some differentiations and elementary operations

Trajectory correction for a 3D underwater vehicle (IV)

Three examples of final position corrections

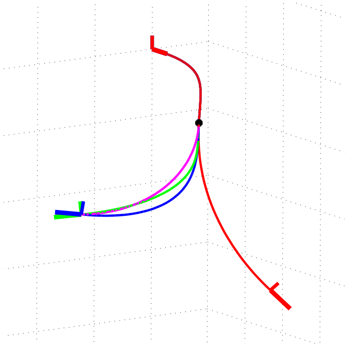


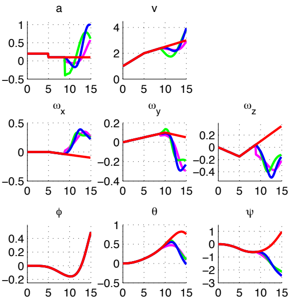


- The original trajectory is in red
- The three corrections respect the continuities of x, y, z, v, ϕ, θ, ψ

Trajectory correction for a 3D underwater vehicle (IV)

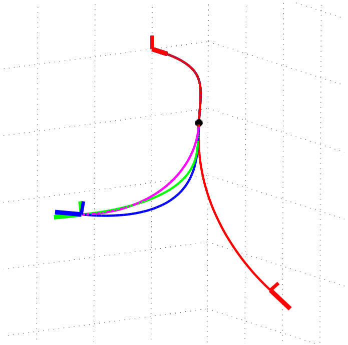
Three examples of final position corrections

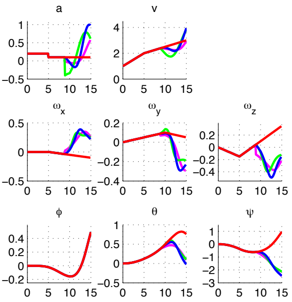




- The original trajectory is in red
- The three corrections respect the continuities of x, y, z, v, ϕ, θ, ψ
- The magenta correction results from an affine transformation that does not belong to the admissible group $\Rightarrow$ it does not respect the continuity of the angular velocities (cf ω_z)

Three examples of final position corrections





- The original trajectory is in red
- The three corrections respect the continuities of x, y, z, v, ϕ, θ, ψ
- The magenta correction results from an affine transformation that does not belong to the admissible group $\Rightarrow$ it does not respect the continuity of the angular velocities (cf ω_z)
- The green and blue corrections correct towards a same final position, but with different final orientations ("redundancy")

Conclusion and ongoing research

- Other systems that can benefit from affine corrections:

Conclusion and ongoing research

- Other systems that can benefit from affine corrections:
  - All 2D wheeled robots without slip

Conclusion and ongoing research

- Other systems that can benefit from affine corrections:
  - All 2D wheeled robots without slip
  - Bevel needle (used in surgery)

Conclusion and ongoing research

- Other systems that can benefit from affine corrections:
  - All 2D wheeled robots without slip
  - Bevel needle (used in surgery)
  - Ongoing research on other 3D mobile robots (quadrotor, satellites, space robots,...)

Conclusion and ongoing research

- Other systems that can benefit from affine corrections:
  - All 2D wheeled robots without slip
  - Bevel needle (used in surgery)
  - Ongoing research on other 3D mobile robots (quadrotor, satellites, space robots,...)
  - Redundant manipulators (holonomic)

Conclusion and ongoing research

- Other systems that can benefit from affine corrections:
  - All 2D wheeled robots without slip
  - Bevel needle (used in surgery)
  - Ongoing research on other 3D mobile robots (quadrotor, satellites, space robots,...)
  - Redundant manipulators (holonomic)
- Advantages of affine trajectory corrections (reminder):

Conclusion and ongoing research

- Other systems that can benefit from affine corrections:
  - All 2D wheeled robots without slip
  - Bevel needle (used in surgery)
  - Ongoing research on other 3D mobile robots (quadrotor, satellites, space robots,...)
  - Redundant manipulators (holonomic)
- Advantages of affine trajectory corrections (reminder):
  - single step

Conclusion and ongoing research

- Other systems that can benefit from affine corrections:
  - All 2D wheeled robots without slip
  - Bevel needle (used in surgery)
  - Ongoing research on other 3D mobile robots (quadrotor, satellites, space robots,...)
  - Redundant manipulators (holonomic)
- Advantages of affine trajectory corrections (reminder):
  - single step
  - no trajectory re-integration

Conclusion and ongoing research

- Other systems that can benefit from affine corrections:
  - All 2D wheeled robots without slip
  - Bevel needle (used in surgery)
  - Ongoing research on other 3D mobile robots (quadrotor, satellites, space robots,...)
  - Redundant manipulators (holonomic)
- Advantages of affine trajectory corrections (reminder):
  - single step
  - no trajectory re-integration
  - exact corrections

Conclusion and ongoing research

- Other systems that can benefit from affine corrections:
  - All 2D wheeled robots without slip
  - Bevel needle (used in surgery)
  - Ongoing research on other 3D mobile robots (quadrotor, satellites, space robots,...)
  - Redundant manipulators (holonomic)
- Advantages of affine trajectory corrections (reminder):
  - single step
  - no trajectory re-integration
  - exact corrections
- More theoretical questions

Conclusion and ongoing research

- Other systems that can benefit from affine corrections:
  - All 2D wheeled robots without slip
  - Bevel needle (used in surgery)
  - Ongoing research on other 3D mobile robots (quadrotor, satellites, space robots,...)
  - Redundant manipulators (holonomic)
- Advantages of affine trajectory corrections (reminder):
  - single step
  - no trajectory re-integration
  - exact corrections
- More theoretical questions
  - How to compute systematically the set of admissible affine deformations for a given system?

Conclusion and ongoing research

- Other systems that can benefit from affine corrections:
  - All 2D wheeled robots without slip
  - Bevel needle (used in surgery)
  - Ongoing research on other 3D mobile robots (quadrotor, satellites, space robots,...)
  - Redundant manipulators (holonomic)
- Advantages of affine trajectory corrections (reminder):
  - single step
  - no trajectory re-integration
  - exact corrections
- More theoretical questions
  - How to compute systematically the set of admissible affine deformations for a given system?
  - Is it always a Lie group?

Conclusion and ongoing research

- Other systems that can benefit from affine corrections:
  - All 2D wheeled robots without slip
  - Bevel needle (used in surgery)
  - Ongoing research on other 3D mobile robots (quadrotor, satellites, space robots,...)
  - Redundant manipulators (holonomic)
- Advantages of affine trajectory corrections (reminder):
  - single step
  - no trajectory re-integration
  - exact corrections
- More theoretical questions
  - How to compute systematically the set of admissible affine deformations for a given system?
  - Is it always a Lie group?
  - How about using more general groups (e.g. projective)?