cartesian_controllers: Motion, Force and Compliance Control for Robotic Manipulators

FZI Forschungszentrum Informatik, Karlsruhe

Arne Roennau, Stefan Scherzinger
Agenda

1. Motivation: Why Cartesian Control?
2. Control Approach
3. Applications
4. Outlook
5. Conclusion
Why Cartesian Control in Task Space?

- Closed loop force control
- Direct teaching
- Contact-rich manipulation
- Manual guidance
- Dynamic target tracking

Use ROS Control with the `cartesian_controllers`
Challenges with Industrial Robots

Often contact dominated tasks

- **Autonomous skills**
  - Dealing with uncertainty
  - Force-sensitive interaction

- **Goals**

- **Robot independant**
  - Formulation in task space
Let’s go for more… Active Cartesian Compliance

Active compliance

\[ \mathbf{F} = c(x^d - x) + k(\dot{x}^d - \dot{x}) + I(\ddot{x}^d - \ddot{x}) \]

Impedance:
\[ x(t) \rightarrow \mathbf{F}(t) \quad \tau(t) = \mathbf{J}^T \mathbf{F}(t) \]
- Torque-actuated robot joints
- Joint-torque sensors

Admittance:
\[ \mathbf{F}(t) \rightarrow x(t) \quad \Delta q = \mathbf{J}^{-1} \Delta x \]
- Motion-actuated robot joints
- End effector force-torque sensor
Agenda

Motivation: Why Cartesian Control?

Control Approach

Applications

Outlook

Conclusion
The ROS framework… reuse it! make it available!

Documented ROS Robots

Source: Ken Conley, Tully Foote, wiki.ros.org/Robots, 2017 changed over to robots.ros.org

willowgarage.com

wiki.ros.org/Metrics
The setting within ⚑ROS Control

You have:

- Joint position/velocity streaming interface
- ROS control HW abstraction
- Library of flexible controllers
- Applications with intuitive end effector control

**Robot**

**ROS control HW abstraction**

- VelocityJointInterface
- PositionJointInterface
- EffortJointInterface

**cartesian_controllers**

- cartesian_motion_controller
- cartesian_force_controller
- cartesian_compliance_controller
- cartesian_controller_handle

**User application**
Three main controllers in \textit{cartesian\_controllers}

- \texttt{cartesian\_motion\_controller}
  - You want to follow a moving target
  - The targets might be sparsely sampled
  - You prefer smoothness over accuracy

- \texttt{cartesian\_force\_controller}
  - You want to control the robot with a wrench in contacts
  - You have a wrist FT sensor

- \texttt{cartesian\_compliance\_controller}
  - You want to follow a moving target
  - You want to react to external disturbances
  - You have a wrist FT sensor
Our Approach

- Instantaneous joint motion from Cartesian error

$$\tau = H(q)\ddot{q} + C(q, \dot{q}) + G(q)$$

$$\ddot{q} = H^{-1}(J^T f - C(q, \dot{q}) - G(q))$$

$$\ddot{q} = H^{-1} J^T f$$

Admittance:
- Directly to joint space!
- Common wrench interface
Our Approach

- Instantaneous joint motion from Cartesian error

\[ \ddot{q} = H^{-1}J^Tf \]

\[ \epsilon = \text{cartesian_motion_controller} \]
\[ \text{Pose error} \]
\[ \text{cartesian_force_controller} \]
\[ \text{Net force} \]
\[ \text{cartesian_compliance_controller} \]
\[ \text{Both} \]

\[ f = K_p \epsilon + K_d \dot{\epsilon} \]

\[ \ddot{q} = H^{-1}J^T(K_p \epsilon + K_d \dot{\epsilon}) \]
The control loop

- **cartesian_motion_controller**
  - $x^d$
  - $f^d$
  - $f^s$
  - $\dot{K}_d$
  - $H^{-1}J^T$
  - $\ddot{q}$
  - $\dot{q}$
  - $q$
  - $q^d$

- **cartesian_force_controller**
  - Net force
  - Pose error
  - Both

- **Both**
  - $g(q)$
The control loop

- Hard to obtain
- No benefit on motion-actuated robots!

Virtual (unrealistic) dynamics
- As long as it solves IK ...
- Is there something better than realistic values?
Task space linearization

Closed-Loop control scheme
(on dynamics-conditioned twin)

Dynamics-conditioned twin

Scherzinger et al., *Inverse Kinematics with Forward Dynamics Solvers for Sampled Motion Tracking*, IEEE ICAR 2019 (to appear)
cartesian force controller
- no singularities / no IK
- no dynamic parameters
- smooth robot motion
- FT sensor needed
cartesian compliance controller

- no singularities / no IK
- no dynamic parameters
- variable stiffness
- smooth robot motion
- FT sensor needed
cartesian motion controller
- no singularities / no IK
- no dynamic parameters
- linear cartesian motions
- dynamic target tracking
- no FT sensor needed
cartesian motion controller
• no singularities / no IK
• no dynamic parameters
• dynamic target tracking
• no FT sensor needed
cartesian compliance controller

- no singularities / no IK
- no dynamic parameters
- variable stiffness
- advanced assembly strategies
- FT sensor needed
cartesian compliance controller

- no singularities / no IK
- no dynamic parameters
- variable stiffness
- smooth robot motion
- dynamic target tracking
- FT sensor needed
cartesian compliance controller
- no singularities / no IK
- no dynamic parameters
- variable stiffness
- linear motions in task space
- compliant screw assembly
- FT sensor needed
Agenda

Motivation: Why Cartesian Control? 1
Control Approach 2
Applications 3
1 4 5
Outlook
Conclusion
Satellite assembly

INDUSTRIAL SETTINGS

- Contact-dominated assembly tasks
- Object poses with uncertainty
- More relevant for complex insertion

GOALS

- Error correcting contact skills for autonomous execution that are transferable to different robots
Approach and Methods

▶ Contact skill extraction in simulation

Let humans solve tilting and jamming for challenging configurations

Scherzinger et al, Contact Skill Imitation Learning for Robot-Independent Assembly Programming, IEEE IROS 2019
Agenda

Motivation: Why Cartesian Control?  
Control Approach  
Applications  
Outlook  
Conclusion
Summary

Control your robots in Cartesian task space!!!

<table>
<thead>
<tr>
<th>Baseline</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint position/velocity streaming interface</td>
<td>Application with fast, direct, task space control</td>
</tr>
</tbody>
</table>

Baseline
- Task space control
- Add-on compliance
- Closed loop force control
- Object manipulation

Goal
- cartesian_controllers
  - cartesian_motion_controller
  - cartesian_force_controller
  - cartesian_compliance_controller

Available as OpenSource!
github.com/fzi-forschungszentrum-informatik/cartesian_controllers

More details here:

Further questions?...Ask us!... roennau@fzi.de  scherzin@fzi.de