Rigid vs. Compliant Contacts: Modelling, Identification and Control for Whole Body Motion

Michael Mistry
Reader in Robotics
School of Informatics
University of Edinburgh

15th November 2016
Rigid vs. Compliant Contact Force

![Graph showing rigid and compliant forces vs. position.](image-url)
Rigid vs. Compliant Contacts

Rigid:

Difficult to predict (for simulation, planning, etc.) but easier when controlling robots

Compliant:

Easier for simulation, planning (has smooth derivatives, etc.) but more difficult when controlling robots

Often we get away with assuming “compliance” is an unmodeled disturbance
Rigid Body Dynamics

Robot acceleration

\[ M(q) \ddot{q} + h(q, \dot{q}) = S^T \tau + J_C^T(q) \lambda \]

Contact Jacobian

Contact forces

Rigid Contact

\[ J_C \dot{q} = 0 \]

\[ J_C \ddot{q} + \dot{J}_C \dot{q} = 0 \]

Compliant Contact

\[ J_C \dot{q} = \dot{p} \]

\[ J_C \ddot{q} + \dot{J}_C \dot{q} = \ddot{p} \]
Rigid Contacts

Compliant Contacts

Control

Identification
Rigid Contacts are simpler for control, because the contact forces may often be ignored.

\[ M\ddot{q} + h = S^T \tau + J_C^T \lambda \]

Construct the Orthogonal Projection, \( P \), such that:

\[ PJ_c^T = 0 \]

e.g.: \( P = I - J_C^+ J_C \)

Then control motion in the subspace independent of constraint forces:

\[ PM\ddot{q} + Ph = PS^T \tau \]
Control and optimisation of tasks with contact using Projected Operational Space Control

Control equation is divided into three mutually orthogonal components:

\[ \tau = PJ^T F + PN \tau_0 + (I - P) \tau_C \]

- Task space motion, under constraints
- Null space motion, under constraints
- Constraint Forces, no motion

Mistry and Righetti, Operational Space Control of Constrained and Underactuated Systems, RSS 2011
Underactuated Operational Space Control

- Rigid Body Dynamics
- Constraint Forces
- Task Space Dynamics
- Null-Space Dynamics

Constraints:
- \( \tau_{\text{constraint}} \)
- \( \tau_{\text{null}} \)
- \( F \)

Tasks:
- \( \tau_{\text{task}} \)

Robot

Mistry and Righetti, Operational Space Control of Constrained and Underactuated Systems, RSS 2011
Valerio Ortenzi, Maxime Adjigble, Kuo Jeffrey, Rustam Stolkin, Michael Mistry, *An Experimental Study of Robot Control During Environmental Contacts Based on Projected Operational Space Dynamics*, Humanoids 2014
Identifying the Contact Jacobian (without Force/Tactile sensing)

\[ J_C(q) = \Lambda J(q) \]

Direction of Constraints \quad Jacobian of Reference Frame in Contact

\[ \Lambda \dot{p} = 0 \quad \dot{p}^T \Lambda^T = 0 \]

For example, on a horizontal table, z axis is constrained:

\[ \Lambda = [0 \quad 0 \quad 1] \]

\[ \Lambda \dot{p} = \Lambda \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \dot{z} = 0 \]

V Ortenzi, H Lin, M Azad, R Stolkin, J Kuo, M Mistry Kinematics-based estimation of contact constraints using only proprioception, Humanoids 2016
\[ \Lambda \dot{\mathbf{p}} = 0 \quad \dot{\mathbf{p}}^T \Lambda^T = 0 \]

Stack of observed velocities (in contact):
\[
\begin{bmatrix}
\dot{\mathbf{p}}_1^T \\
\dot{\mathbf{p}}_2^T \\
\vdots \\
\dot{\mathbf{p}}_n^T
\end{bmatrix}
\]
\[ B \dot{\mathbf{p}}^T \]

We want to find the right null space of \( B \dot{\mathbf{p}}^T \)

Compute the SVD:
\[ B \dot{\mathbf{p}}^T = U S V^T \]

\( \Lambda \) is then constructed using the columns of \( V \) which correspond to very small singular values
Desired Velocities (during exploration)  

Actual Velocities

Red vectors differ significantly from desired, use these for surface estimation
V Ortenzi, H Lin, M Azad, R Stolkin, J Kuo, M Mistry Kinematics-based estimation of contact constraints using only proprioception, Humanoids 2016
Observed velocities during exploration

Reconstructed normal vector (for 10 trials)
Observed velocities during exploration

Reconstructed normal vector (for 10 trials)

Perhaps extend this method to identify (rigid) curved surfaces with a piece-wise linear approximation:
Rigid Body Dynamics

Robot acceleration

\[ M(q) \ddot{q} + h(q, \dot{q}) = S^T \tau + J^T_C(q) \lambda \]

Contact Jacobian

Compliant Contact

\[
\begin{align*}
J_C \dot{q} &= \dot{p} \\
J_C \ddot{q} + J_C \dot{q} &= \ddot{p}
\end{align*}
\]

Contact forces

Acceleration at contact points will influence robot’s momentum
To control the robot’s momentum (for balance) we need to control contact point accelerations.

The idea is to compute desired values for contact point accelerations (as well as joint accelerations) to achieve the desired rate of change of robot’s momentum:

\[
\dot{h}_d = \begin{bmatrix}
    mk_p (c_d - c) + mk_v (\dot{c}_d - \dot{c}) \\
    k_l (l_d - l)
\end{bmatrix}
\]

com position

angular momentum
To control the robot’s momentum (for balance) we need to control contact point accelerations.

The idea is to compute desired values for contact point accelerations (as well as joint accelerations) to achieve the desired rate of change of robot’s momentum:

\[
\dot{h}_d = \begin{bmatrix}
    mk_p (c_d - c) + mk_v (\dot{c}_d - \dot{c}) \\
    k_l (l_d - l)
\end{bmatrix}
\]

given a model of compliant contact force:

\[
f_n = \max(0, k \delta \frac{3}{2} + \lambda \delta \frac{1}{2} \dot{\delta})
\]

Azad-Featherstone Model

we can derive the rate of change of momentum due to motion at contacts:

\[
\dot{h}_i = -\delta_i \frac{1}{2} \begin{bmatrix}
    k_i D_i (p_i - r_i) + \lambda_i D_i \dot{p}_i \\
    k_i (p_i - r_i) + \lambda_i \dot{p}_i
\end{bmatrix}
\]
Fig. 1. graphical demonstration of the control algorithm
Model unaware of compliant contact at hand (assumes it is rigid)  
Model compensates for the compliant contact at hand

Control Identification

Rigid Contacts

Compliant Contacts
General Model of Normal Force at Compliant Contact:

\[ f_z = \begin{cases} 
  k z^a + \lambda z^b \dot{z}^c & z \geq 0 \\
  0 & z < 0 
\end{cases} \]

\[
\begin{align*}
  a &= c = 1, b = 0 & \text{Kelvin-Voigt (linear)} \\
  a &= b = \frac{3}{2}, c = 1 & \text{Hunt-Crossley} \\
  a &= \frac{3}{2}, b = \frac{1}{2}, c = 1 & \text{Azad-Featherstone}
\end{align*}
\]

Training Trajectory:

force prediction:

\[ \mathbf{\sigma} = \mathbf{A} [k \quad \lambda]^T \]

\[ \mathbf{A} = \begin{bmatrix} \zeta & \nu \\ \zeta^{\frac{3}{2}} & \zeta^{\frac{3}{2}} \nu \\ \zeta^{\frac{1}{2}} & \zeta^{\frac{1}{2}} \nu \end{bmatrix} \]

Kelvin-Voigt (linear)

Hunt-Crossley

Azad-Featherstone

Fit stiffness/damping parameters via Least Squares:

\[ [\hat{k} \quad \hat{\lambda}]^T = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{\sigma} \]
M Azad, V Ortenzi, H Lin, E Rueckert, M Mistry, Model Estimation and Control of Compliant Contact Normal Force, Humanoids 2016
Fitness values for each model (higher is better):

<table>
<thead>
<tr>
<th></th>
<th>linear K/V</th>
<th>non-linear H/C</th>
<th>non-linear A/F</th>
</tr>
</thead>
<tbody>
<tr>
<td>sponge</td>
<td>0.965 ± 0.001</td>
<td>0.981 ± 0.001</td>
<td>0.986 ± 0.001</td>
</tr>
<tr>
<td>dice</td>
<td>0.976 ± 0.001</td>
<td>0.845 ± 0.002</td>
<td>0.853 ± 0.002</td>
</tr>
</tbody>
</table>
Force Control task:

The graph shows the force control task with two different materials: a dice (linear) and a sponge (non-linear). The graphs display the force over time, with markers indicating the command and the actual and predicted forces. The linear material shows a consistent response to the command, while the non-linear material exhibits a more complex and varying response.
Rigid vs. Compliant Contacts

Rigid:

Difficult to predict (for simulation, planning, etc.) but easier when controlling robots

Compliant:

Easier for simulation, planning (has smooth derivatives, etc.) but more difficult when controlling robots

Controllers should ideally be aware of the dynamics of compliant contacts:

Compliant models are a generalisation of rigid ones. A compliant model can still approximate a rigid contact (but not the other way around).
Thanks to:

Morteza Azad
Hsiu-Chin Lin
Vijaykumar Rajasekaran
Chie Takahashi
Joshua Smith
Valerio Ortenzi
Keyhan Kouhkiloui
Maxime Adjigble
Michael Jacob Mathews