Safe Motion Control in physical Human-Robot Interaction
Workshop Safety for Human-Robot Interaction in Industrial Settings
2015 IEEE/RSJ International Conference on Intelligent Robots and Systems
Hamburg, Germany

Sami Haddadin

Institut für Regelungstechnik
Leibniz Universität Hannover
Appelstraße 11
30167 Hannover
www.irt.uni-hannover.de
haddadin@irt.uni-hannover.de
Interaction

Cognitive HRI

Physical HRI
Physical Interaction

Design
- Force/Torque Sensing
- Lightweight Design
- Mechatronics Design

Control
- Force & Impedance Control
- Interaction Control & Learning
- Collision/Contact Handling

Planning
- Interaction Planning
- Real-Time Motion Planning
- Reflex Planning

Human Safety

1. Act compliant
2. Feel and understand contacts
3. Move safely
Safe Real-Time Motion Planning

- Real-Time Motion Algorithm
- 3D Sensing
- Human Safety
Depth Sensors: From Stereo Vision to Kinect

- Stereo Vision
- Time of Flight
- Structured Light

→ More later on…
Algorithm Classes

- Safe Brachistochrone Learning safe behaviors
- Optimization Based
- C-/Task-Space Planning
- Trajectory Planning
- Dynamical Systems
- Danger Index Based Planning
- Human Aware Motion Planning
- Online Trajectory Generation
- Potential Fields
- Variable Attractor Dynamics
- Dynamic Movement Primitives
- Circular Fields

Work of Andrea Maria Zanchettin and Paolo Rocco in Safe Motion Planning!
C-Space Planning: Kulić & Croft 2006-2006

\[ J = w_G f_G (d_G) + w_0 f_0 (d_0) + w_D K \cdot DC \]

- „Extension“ of (Nokata, Ikuta et al., 2002)
- C-space planner based on best-first planning approach, meaning following steepest decent of cost function (Latombe, 1991)

- Danger Criterion:

\[ DC = \frac{I_s}{I_{\text{max}}} f_{CoM} (D_{CoM}) \]

- Reflected inertia based safety coefficient
- Distance based safety coefficient
Optimization: Bicchi et. al. 2003-2005

\[
\begin{align*}
\min_T \int_0^T 1 \, dt \\
M_{\text{rot}} \dddot{x}_{\text{rot}} + K_{\text{transm}} (x_{\text{rot}} - x_{\text{link}}) = u \\
M_{\text{link}} \dddot{x}_{\text{link}} + K_{\text{transm}} (x_{\text{link}} - x_{\text{rot}}) = 0 \\
|\dddot{x}_{\text{link}}| \leq v_{\text{safe}} (K_{\text{transm}}) \\
|u| \leq U_{\text{max}},
\end{align*}
\]

with initial and terminal conditions:

\[
\begin{align*}
\dot{x}_{\text{rot}}(0) &= 1, \quad \dot{x}_{\text{rot}}(0) = 0 \\
\dot{x}_{\text{link}}(0) &= 1, \quad \dot{x}_{\text{link}}(0) = 0 \\
\dot{x}_{\text{rot}}(T) &= 0, \quad \dot{x}_{\text{rot}}(T) = 0 \\
\dot{x}_{\text{link}}(T) &= 0, \quad \dot{x}_{\text{link}}(T) = 0,
\end{align*}
\]
Real-Time Motion Planning
Potential Fields: Khatib 1986

*The manipulator moves in a field of forces.*
*The position to be reached is an attractive pole for the end effector and obstacles are repulsive surfaces for the manipulator parts.*

„Borrowed slide“ from Howie Choset
Potential Fields: Khatib 1986
Dynamical Systems for Collision Avoidance

2\textsuperscript{nd} order dynamical system: \[ M\ddot{x}_d + K(x_d - x_g) + D\dot{x}_d = 0 \]

Potential Fields \[ M\ddot{x}_d + K(x_d - x_g) + D\dot{x}_d = \sum F_i(x_{o,i}, x_d, \ldots) \]

Potential Fields + Task feed-forward, like DMP \[ M\ddot{x}_d + K(x_d - x_g) + D\dot{x}_d = \sum F_i(x_{o,i}, x_d, \ldots) + F_{task} \]

Variable Attractor Dynamics \[ \dot{M}\ddot{x}_d + \dot{K}(x_d - x_g) + \dot{D}\dot{x}_d = \sum F_i(x_{o,i}, x_d, \ldots) \]

Variable Attractor Dynamics + Velocity scaling \[ \dot{M}\ddot{x}_d + \dot{K}(x_d - x_g) + \dot{D}\dot{x}_d = \sum F_i(x_{o,i}, x_d, \ldots) \]
Variable Attractor Dynamics: Haddadin et al. 2009
Variable Attractor Dynamics: Haddadin et al. 2009
Dynamic Movement Primitives: Park et al. 2008

\[ \tau \dot{v} = K(g - x) - Dv - K(g - x_0)\theta + Kf(\theta) + \varphi(x, v) \]
\[ \tau \dot{x} = v. \]

\[ \varphi(x, v) = -\nabla_x U_{\text{dyn}}(x, v) \]
\[ = \lambda(-\cos \theta)^{\beta-1} \frac{||v||}{p} \left( \beta \nabla_x \cos \theta - \frac{\cos \theta}{p} \nabla_x p \right) \]

\[ U_{\text{static}}(x) = \begin{cases} \frac{\eta}{2} \left( \frac{1}{p(x)} - \frac{1}{p_0} \right)^2 & : p(x) \leq p_0 \\ 0 & : p(x) > p_0 \end{cases} \]

\[ U_{\text{dyn}}(x, v) = \begin{cases} \lambda(-\cos \theta)^{\beta} \frac{||v||}{p(x)} & : \frac{\pi}{2} < \theta \leq \pi \\ 0 & : 0 \leq \theta \leq \frac{\pi}{2} \end{cases} \]
Dynamic Movement Primitives: Park et al. 2008
GPU-Enhanced Variable Attractor Dynamics: Kaldestadt et al. 2014

Environment Observer (Kinect)

Point Cloud Volume Filter (CUDA)

Calculate Force / Forces (CUDA)

Send to Robot Controller

Robot Position and Robot Joint Angles

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Voxel map resolution</th>
<th>Robot vertices</th>
<th>Algorithm calculation time</th>
<th>Performance factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>10 mm</td>
<td>4701</td>
<td>1.426 s</td>
<td>129.63x</td>
</tr>
<tr>
<td>GPU</td>
<td>10 mm</td>
<td>4701</td>
<td>0.011 s</td>
<td>402.07x</td>
</tr>
<tr>
<td>CPU</td>
<td>5 mm</td>
<td>4701</td>
<td>5.692 s</td>
<td>68.63x</td>
</tr>
<tr>
<td>GPU</td>
<td>5 mm</td>
<td>4701</td>
<td>0.014 s</td>
<td>273.54x</td>
</tr>
<tr>
<td>CPU</td>
<td>10 mm</td>
<td>2468</td>
<td>0.755 s</td>
<td></td>
</tr>
<tr>
<td>GPU</td>
<td>10 mm</td>
<td>2468</td>
<td>0.011 s</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>5 mm</td>
<td>2468</td>
<td>3.009 s</td>
<td></td>
</tr>
<tr>
<td>GPU</td>
<td>5 mm</td>
<td>2468</td>
<td>0.011 s</td>
<td></td>
</tr>
</tbody>
</table>
GPU-Enhanced Variable Attractor Dynamics: Kaldestadt et al.
2014
Circular Fields: Haddadin et. al. 2011

\[ F = \dot{x}_r \times B = \dot{x}_r \times \sum_i I \frac{c_i \times \dot{x}_r}{l_i^2} da_i \]
Circular Fields: Haddadin et. al. 2011
Multi-Agent Based Prediction: Haddadin et al. 2013
Cost Based Multi-Agents + Dynamical Systems
How can we ensure that a motion is safe for a human?
Concretely: Safety Analysis of Human Injury & Pain

Is this safe?
Crash-Testing in Robotics

Haddadin et. al. RSS 2007
How to Connect Injury Data with Robot Motion?

But the robot configuration changes...

It may carry a tool...

Is it maximum forces, velocities, energies, stress,...?

And there are so many contact situations...
Embedding Injury Knowledge into Control: Haddadin et al.
2010-2012

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau_J + \tau_{ext} \]

\[ m_u = [u^T \Lambda_v(q)^{-1} u]^{-1} \]

with \[ \Lambda(q) = (J(q)M(q)^{-1}J(q)^T)^{-1} \]

Cannot change \( m_u \) anyways.

\( v_u \) in impact direction \( u \) governs the impact.

Determine \( v_{u,\text{safe}} \) of point of interest given \( m_u \) and \( c_u \).
Embedding Injury Knowledge into Control: Haddadin et al. 2010-2012

\[ m_u(q) \]

Safety curve

\[ S_c(m_u, c_u, \text{bodypart}) \]

\[ m \]

\[ v_{u,\text{safe}} \]
Injury data

(mass, velocity, geometry, bodypart) → observed injury/pain

Pain data
Embedding Injury Knowledge into Control: Haddadin et al. 2010-2012
Example Usecase: Safe Motion Unit

Haddadin et. al. IJRR2012, Haddadin et. al. at 2014, Haddadin et. al. IROS2012, Haddadin & Haddadin Biomecanica Hungaria 2012
Embedding Injury Knowledge into Control: Haddadin et al. 2010-2012

1. Assume biomechanical injury/pain data required your application
2. Need to know your robot dynamics.
   • Solutions:
     1. Ask your robot manufacturer for the data.
     2. Do it yourself (e.g. Oussama's PUMA paper, Alessandro's LWR paper).
3. Attach to the safety relevant surface locations of your robot (Points of Interest, POIs) surface primitives that can be associated to your basic injury/pain data
4. Run through the set of POIs and calculate their respectively safe velocity.
5. Scale the commanded velocity to the most conservative POI.

Result:

\[ v_{\text{max}} \; \text{s.t.} \; \text{safety!!} \]
Thanks to ... and many others!
Thanks!