Model-driven design of interaction robot control

Yury Brodskiy, Robert Wilterdink, Jan Broenink, Stefano Stramigioli
Content

• Intrinsically Passive Control (IPC)
  – Control paradigm
  – Control language
  – "Tools of the trade"

• Model Driven Development (MDD)
  – Two sides of model driven design
  – Functional decomposition

• Component Based Software Engineering (CBSE)
  – Desired result
  – From simulation to implementation
  – "Tools of the trade"
Intrinsically Passive Controller

- The only way to ensure stability of interaction with environment. (Anderson, 1989, Hogan 1985)
- A way for intuitive interaction with people (Sheridan 1992)

Interaction: relation of $F$ and $x$

\[
\begin{pmatrix}
R(s) \\
R_E(s)
\end{pmatrix}
\begin{pmatrix}
F(s) \\
x(s)
\end{pmatrix}
= \begin{pmatrix}
0 \\
0
\end{pmatrix}
\]
Stiffness control

• Interaction control
  – Passive
  – Robust
  – Intuitive
  – Compliant

Thursday, 15, March 12
Control “language”

Data flow

Conjugated variables interconnection

- Explicit notation for power interconnection
- Re-usabilitiy
  - Non-causal models
  - Physical concepts
“Tools of the trade”

- Dynamic models
- Controller simulation

Thursday, 15, March 12
An overview

Meta-model
Signal-flow graphs, Bond-graphs, Equations

Tools
MatLab, LabView
20Sim, OpenModelica ...

Conforms
Provides

Models
Controllers, Dynamic models

Languages
Modelica, MatLab, Block-diagrams, Bond-graph
Model driven development

- Two sides of the same coin
  - Functional architecture
  - Component architecture
From simulation to implementation

Coordination (FSM, planner) -> Controller -> Kinematics

Code generation

Orocus RTT

ROS
  rviz -> Controller
  rxplot

Controller -> Kinematics

youBot driver

Monitor

Firmware / youBot

KUKA youBot

Data flow

Asynchronous operations

Outputs events

Input events
Components language
“Tools of the trade”
Framework independent code generation

- Computation Model
  - Implementation Template
  - Framework component model
- Code generation
- Code
- Component deployment
- Running Component
- Framework

- Activity
- Input
- Artifact
Conclusion

- Combined “Tools of the trade” shorten development cycle
- Control language increase insight on control laws
- MDD & CBSE promote reuse