A constraint-based approach for multi-modal robot control

Joris De Schutter
KU Leuven
Department of Mechanical Engineering
Division of Production Engineering,
Machine Design and Automation (PMA)

See and Touch: 1st Workshop on multimodal sensor-based robot control for HRI and soft manipulation, *IROS 2015, Hamburg, September 28th, 2015*
Outline

• Brief overview of previous work @KU Leuven

• Constraint-based approach

• Task modeling and constraint specification

• Control and estimation

• Recent applications

• Discussion
Brief overview previous work @KU Leuven

These videos and many more @

- youtube channel: “kuleuven robotics and mechatronics”
- my home page: http://people.mech.kuleuven.be/~jdeschut/

Overview of relevant papers at these webpages.
Brief overview previous work @KU Leuven

- 1978: (Jos Simons)
  - first force control results
  - close-tolerance peg-in-hole assembly
  - custom-built 5-axis robot
  - force/torque measurements based on motor currents

- ~ 1984
  - several experiments (peg-in-hole, 2D/3D contour following, palletizing, opening/closing a door)
  - bulky hydraulic Cincinnati-T3 robot
  - custom-built 6 axis force/torque sensor (Rik (Henri) Beliën)
  - task frame at robot end effector
  - velocity-resolved control
Brief overview previous work @KU Leuven

• Early nineties
  o add model-based feedforward (Herman Bruyninckx)
  o 2D contour tracking – peg-on-hole
  o extend to cooperating robots

• ~ 2000
  o combination of force control with vision
  o multiple ‘task frames’

• 2000-2003
  o identification of contact geometry (Tine Lefebvre)

• 2005
  o human-robot interaction
  o multi-modal sensing
  o multiple task frames
Lessons learnt

- similarities between different types of ‘geometric’ sensors
- soft contacts can be modeled as geometric deformations using compliance model (hard contacts are modeled by hard constraints)
- task defined by various ‘constraints’, specified in different spaces (“frames”)
- feedback is always too late:
  - (task execution speed) ~ (control bandwidth) x (geometric task execution error)
  - for contact: (geometric error) = (force error) x (compliance)
  - proportionality in this eq. depends on error in geometric model:
  - task execution speed and accuracy are improved by feedforward control based on geometric models
- contact geometry can be identified by observing motion (and contact forces) during task execution or active sensing motions
Constraint-based control

• Consider every robot system as a set of degrees of freedom
• Formulate every robot task as an optimization problem
• Optimization variables
  o velocity, acceleration or torque for each actuated dof at every time instant
• Objective function
  o conflicting constraints: minimize constraint violations
  o task redundancy: e.g. minimize kinetic energy
  o local (instantaneous) vs. global (over time), e.g. minimum time, minimum energy
Constraint-based control

- **Constraints** (not exhaustive...)
  - **task-related**
    - follow predefined trajectory in some direction
    - track object in some direction based on sensor information (e.g. vision, force, distance)
    - apply force or torque in some direction
    - impose impedance or admittance in some direction
  - **robot system-related**
    - avoid joint limits: position, velocity, acceleration, torque
    - avoid self-collision
  - **environment-related**
    - avoid collisions, define forbidden regions
  - **human interaction-related**
    - impose impedance or admittance in some direction
    - provide haptic feedback for teleoperation
Constraint-based control

- Roots
  - Ambler, Popplestone (Artificial Intelligence 75)
    - specifying the goal position in assembly tasks using geometric relations between objects
  - O. Khatib (IJRR87)
    - operational space formulation
  - Y. Nakamura (IJRR86, Book91)
    - optimization and redundancy resolution
  - Samson, Le Borgne, Espiau (Book91)
    - task function approach
  - ...
Constraint-based control

• Other, parallel developments:
  o ‘Stack-of-Tasks’
    • Inspired by/applied to whole-body manipulation for humanoid robots
    • CNRS-LAAS, CNRS-LIRMM, CNRS-AIST JRL
    • Mansard, Lamiraux, Stasse, Kheddar, Khatib, Chaumette et al.
  o Architecture for whole-body manipulation for humanoid robots
    • Khatib, Sentis, Park (Stanford University), e.g. ICRA2006
Constraint-based control

- basic approach (De Schutter et al. IJRR2007)

\[ P: \text{`plant': robot + environment}; \quad C: \text{controller}; \quad M+E: \text{model update + estimator} \]

control input \( u \): desired joint velocities

system output \( y \): controlled variables

\( \Rightarrow \text{Task specification = Imposing constraints} \ y_d \ \text{on} \ y \)

measurements \( z \): observe the plant.

geometric disturbances, \( \chi_u \)

control input \( u \): may also be desired accelerations or torques
Task modeling and constraint specification

- systematic approach for deriving expressions for the task constraints
  - → low-level constraint controllers are derived automatically
  - → location of frames is updated automatically
Task modeling and constraint specification

Laser-plane feature:

\[ \chi_{fI}^a = (x^a, y^a)^T, \quad (2) \]
\[ \chi_{fII}^a = (\phi^a, \theta^a, \psi^a)^T \quad (3) \]
\[ \chi_{fIII}^a = (z^a). \quad (4) \]
Task modeling and constraint specification

- systematic approach for modelling geometric uncertainty
  - → estimators for off-line calibration or on-line adaptation are derived automatically

\[ q \quad o_1' \xrightarrow{X_u I} \quad o_1 \xrightarrow{X_f I} \quad f_1' \xrightarrow{X_u II} \quad f_1 \]

\[ w \quad o_2' \xleftarrow{X_u IV} \quad o_2 \xleftarrow{X_f III} \quad f_2' \xleftarrow{X_u III} \quad f_2 \]

- \( w \): world
- \( o \): object
- \( f \): feature
- \( q \): robot coordinates
- \( X_u \): uncertainty coordinates (e.g. calibration values)
- \( X_f \): feature coordinates

**J. De Schutter, A constraint-based approach for multi-modal robot control**
Control and estimation

- **lowel-level control: velocity-resolved (IJRR2007)**
  - constraint controller:
    \[
    \dot{y}_d^\circ = \dot{y}_d + K_p (y_d - y)
    \]
    where \( y \) is measured or estimated and \( d \) refers to ‘desired’
  - task controller (‘generalized inverse kinematics’):
    \[
    A \dot{q}_d = \dot{y}_d^\circ + B \dot{X}_u
    \]
    where \( A(q,X_f,X_u) \), \( B(q,X_f,X_u) \), and \( X_u \) is estimated
  - generalized inverse yields desired joint velocities:
    - conflicting constraints are handled by constraint priorities or constraint weights
    - kinematic redundancy is solved by using weights in joint space
Control and estimation

• lowel-level control: acceleration-resolved
  (De Laet et al., KU Leuven internal report as addendum to IJRR2007)
  - constraint controller:
    \[ \ddot{y}_d = \dot{y}_d + K_v (\dot{y}_d - \dot{y}) + K_p (y_d - y) \]
  - task controller: generalized inverse kinematics is now solved at acceleration level yielding desired joint accelerations
  - using the desired joint accelerations and the dynamic model of the robot platform control inputs \( u \) are obtained at torque level
  - solution both for soft contact and hard contact
Control and estimation

• extension: dealing with inequality constraints
  (... example solution...)
  o define ‘safety zone’ before the true constraint
  o if the safety zone is entered, start a constraint controller with \( y_d \) equal to the border of the safety zone
  o increase the weight of the constraint (e.g. exponentially) if the distance to the true constraint becomes smaller
Control and estimation

• medium-level task controller
  o finite state machine
    • activates/deactivates constraints
    • assigns constraint priorities and/or constraint weights
    • sets desired constraint values
    • monitors the task execution based on the sensor readings and the measured or estimated task coordinates
Control and estimation

- estimator
  - updates all feature coordinates $X_f$ and estimates all constraint values $y$ and uncertainty coordinates $X_u$ (+ time derivatives)
  - process model for the estimator follows from the task modeling, e.g. for the velocity-resolved case and for geometric uncertainties at position/velocity/acceleration level:

$$\frac{d}{dt} \begin{pmatrix} q \\ \dot{X}_f \\ X_u \\ \dot{X}_u \\ \ddot{X}_u \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -J_f^{-1}J_u & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} q \\ \dot{X}_f \\ X_u \\ \dot{X}_u \\ \ddot{X}_u \end{pmatrix}$$

$$+ \begin{pmatrix} 1 \\ -J_f^{-1} J_q \\ 0 \\ 0 \\ 0 \end{pmatrix} \dot{q}_d.$$
Recent applications

• 2007
  o simultaneous visual servoing of multiple objects
  o adding robot and collision constraints

• 2008
  o multi-modal sensor-based control: force sensor, laser sensor, camera, laser scanner

• 2011
  o human-robot co-manipulation using PR2
  o > 30 constraints
  o force controller without force sensor
Recent applications

• 2015
  • quadrotor with shared control:
    • remotely controlled (forward/backward) with visual feedback to operator
    • local collision control (US and infrared sensors)
    • addition of local yaw control to point camera to target

• 2015
  • exoskeleton with compliant actuators
    • bilateral-lower limb exoskeleton to assist sit-to-stand
Recent applications

- Constraint-based approach is applied in several projects:
  - ROBOHOW.COG. EU-FP7-288533
  - Factory-in-a-Day EU-FP7-609206
  - MIRAD, sponsored by IWT (Flanders)
Discussion

- limitations of the approach in IJRR2007
  - constraint specification is uniquely based on (full 6D) kinematic loop equations
    - coordinate singularities if minimal coordinates are used
    - more general task modeling approach allowing general expressions and based on expression graphs and automatic differentiation is presented in Aertbeliën et al. (IROS2014)
  - low-level task controller is velocity-resolved
    - acceleration-resolved and torque-based approach was presented in De Laet et al. (internal report)
- limitations of existing software implementations
  - see e.g. Aertbeliën et al. (IROS2014) for a comparison
Discussion

- challenges for (general-purpose) software implementations
  - large variety of systems and tasks (‘skills’)
    - robot platforms (topology, #dofs, kinematics, dynamics)
    - sensor systems
    - environments (e.g. soft/hard contact)
    - tasks (constraints, objectives, priorities, etc.)
    - low-level control approaches
  - need for flexible software environments to
    - formulate (different types of) constrained optimization problems
    - choose appropriate solvers: numerical + symbolic (reasoning)
    - monitor task execution using finite state machine
    - compose complex systems (incl. ‘skills’) from subsystems
    - make Domain Specific Languages with well-defined and simple semantics for coherent sets of applications to avoid the need for (re)writing code for every application
Thanks ...

- KU Leuven GOA/10/11-project: *Global real-time optimal control of robots and mechatronic systems*
- Many colleagues and collaborators: Herman Bruyninckx, Erwin Aertbeliën, Wilm Decré, Nick Vanthienen, Gianni Borghesan, Enea Scioni, Tine Lefebvre, Ruben Smits, Tinne De Laet, Johan Rutgeerts, Rik Beliën, et al.