iTaSC concepts and tutorial

Robohow

Wilm Decré Tinne De Laet
Dominick Vanthienen
Herman Bruyninckx
Joris De Schutter

Katholieke Universiteit Leuven
Department of Mechanical Engineering
Division PMA
Robotics Research Group

March 12, 2012

problem statement

challenge

programming general sensor-based robot systems for complex tasks

complex tasks:

- combination of subtasks
- sensor feedback
- large variety of robot systems
- uncertain environments

problem statement

current state

- reprogramming for every task
- specialist
- time consuming + expensive

our goal

development of programming support:

- non-specialists
- less time consuming

problem statement

programming support

SYSTEMATIC approach of specification of tasks using constraints
'iTaSC': instantaneous Task Specification using Constraints

our contribution

framework with:

- systematic approach and
- estimation support for uncertain environments

aim of presentation

aim of presentation

- to show, by means of an example application, how framework for 'Constraint-based task specification and Estimation for Sensor-Based Robot Systems in the Presence of Geometric Uncertainty' works and what its advantages are
- explain generic control and estimation scheme
- show application to other example tasks
- give status, extensions, and outlook

laser tracing task

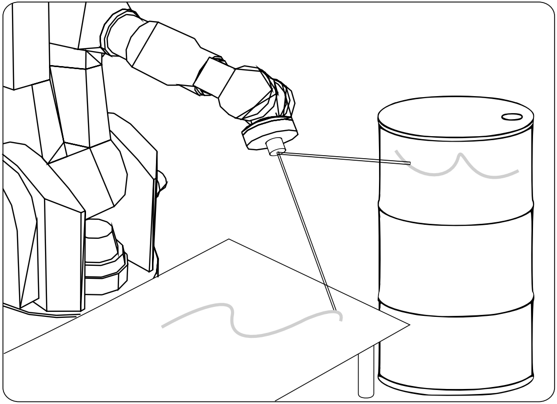


Figure: simultaneous laser tracing on a plane and a barrel

overview

general principle

- robot task: accomplishing relative motion and/or controlled dynamic interaction between objects
- specify task by imposing constraints
  $\Rightarrow$ *task function approach* or *constraint-based task programming*

application independent versus application dependent

- application independent: control and estimation scheme
- application dependent - but systematic: task modeling procedure

control and estimation scheme

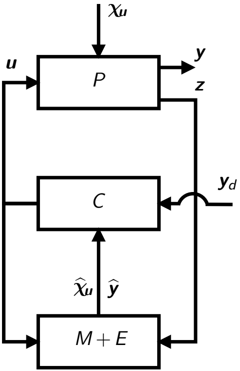


Figure: general control scheme

- plant P:
  - robot system ($\boldsymbol{q}$)
  - environment
- controller C
- model update and estimation $M+E$

control and estimation scheme

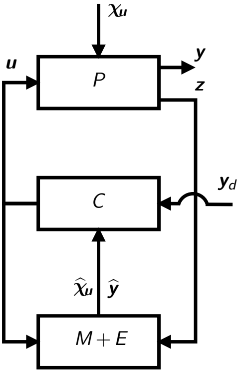


Figure: general control scheme

nomenclature:

- *control input* $\boldsymbol{u}$: desired joint velocities
- *system output* $\boldsymbol{y}$: controlled variables $\Rightarrow$ task specification $=$ imposing constraints $\boldsymbol{y}_d$ on $\boldsymbol{y}$
- *measurements* $\boldsymbol{z}$: observe the plant
- *geometric disturbances*, $\chi_{\boldsymbol{u}}$

control and estimation scheme

conclusion

task independent derivation of
controller block and model update and estimation block
IF
specific *task modeling* procedure is used

task modeling

- task modeling uses TASK COORDINATES:
- two types of task coordinates:
  - *feature coordinates*, χ_f
  - *uncertainty coordinates*, χ_u
- task coordinates defined in user-defined frames

goal

choose frames and task coordinates in a way the task specification becomes intuitive

framework presents procedure to do this

task modeling procedure

four steps:

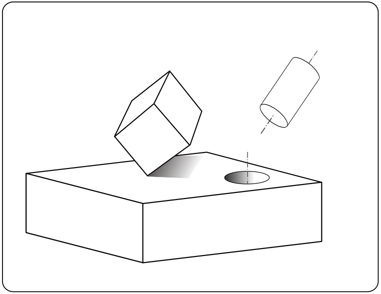
1. identify objects and features and assign reference frames
2. choose feature coordinates χ_f
3. choose uncertainty coordinates χ_u
4. specify task

task modeling procedure

four steps:

1. identify objects and features and assign reference frames
2. choose feature coordinates χ_f
3. choose uncertainty coordinates χ_u
4. specify task

STEP 1: object and feature frames



a feature is linked to an object

- physical entity
  (vertex, edge, face, surface. . .)
- abstract geometric property
  (symmetry axis, reference frame of a sensor,. . .)

STEP 1: object and feature frames

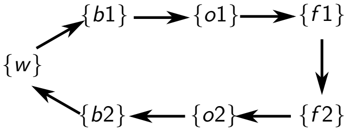


Figure: object and feature frames

each constraint needs four frames:

- two object frames: $o1$ and $o2$
- two feature frames: $f1$ and $f2$

STEP 1: object and feature frames

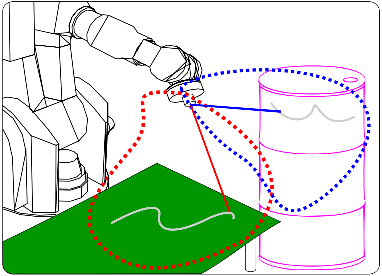
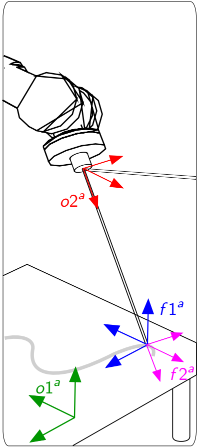


Figure: object and feature frames laser tracing

- natural task description imposes two motion constraints:
  1. trace figure on plane
  2. trace figure on barrel
- ⇒two feature relationships:
  1. feature *a*: for the laser-plane
  2. feature *b*: for the laser-barrel
- the objects are:
  1. laser *a* and laser *b*
  2. the plane
  3. the barrel

STEP 1: object and feature frames

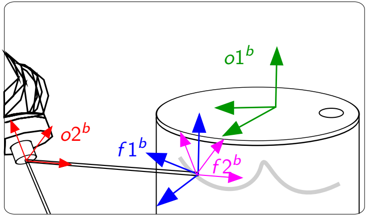


object and feature frames

- for laser-plane feature:
  - frame $o1^a$ fixed to plane
  - frame $o2^a$ fixed to first laser, z-axis along laser beam
  - frame $f1^a$ same orientation as $o1^a$, at intersection of laser with plane
  - frame $f2^a$ same position as $f1^a$ and same orientation as $o2^a$
- for laser-barrel feature:

robohow KATHOLIEKE UNIVERSITEIT LEUVEN

STEP 1: object and feature frames



object and feature frames

- for laser-plane feature:
- for laser-barrel feature:
  - frame $o1^b$ fixed to barrel, x-axis along axis of barrel
  - frame $o2^b$ fixed to second laser, z-axis along the laser beam
  - frame $f1^b$ at intersection of laser with barrel, z-axis perpendicular to barrel surface, x-axis parallel to barrel axis
  - frame $f2^b$ same position as $f1^b$, same orientation as $o2^b$

task modeling procedure

four steps:

1. identify objects and features and assign reference frames
2. choose feature coordinates χ_f
3. choose uncertainty coordinates χ_u
4. specify task

STEP 2: feature coordinates

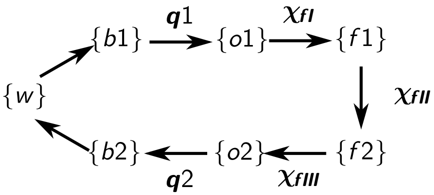
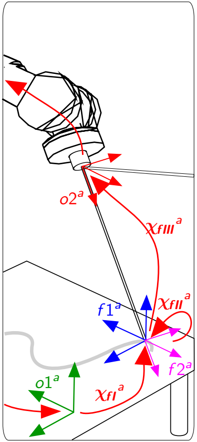


Figure: object and feature frames and feature coordinates

- in general six degrees of freedom between $o1$ and $o2$
- $o1 \rightarrow f1 \rightarrow f2 \rightarrow o2 =$ virtual kinematic chain
- for every feature $\boldsymbol{\chi_f}$ can be partitioned

$$\boldsymbol{\chi_f} = \begin{pmatrix} \boldsymbol{\chi_{fI}}^T & \boldsymbol{\chi_{fII}}^T & \boldsymbol{\chi_{fIII}}^T \end{pmatrix}^T$$

STEP 2: feature coordinates



- laser-plane feature:

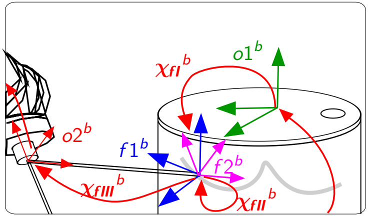
$$\chi_{fI}{}^a = \begin{pmatrix} x^a & y^a \end{pmatrix}^T \quad (1)$$

$$\chi_{fII}{}^a = \begin{pmatrix} \phi^a & \theta^a & \psi^a \end{pmatrix}^T \quad (2)$$

$$\chi_{fIII}{}^a = \begin{pmatrix} z^a \end{pmatrix} \quad (3)$$

- laser-barrel feature

STEP 2: feature coordinates



- laser-plane feature
- laser-barrel feature:

$$\chi_{fI}{}^b = \left(x^b \quad \alpha^b \right)^T \qquad (1)$$

$$\chi_{fII}{}^b = \left(\phi^b \quad \theta^b \quad \psi^b \right)^T \qquad (2)$$

$$\chi_{fIII}{}^b = \left(z^b \right) \qquad (3)$$

task modeling procedure

four steps:

1. identify objects and features and assign reference frames
2. choose feature coordinates χ_f
3. choose uncertainty coordinates χ_u
4. specify task

STEP 3: uncertainty coordinates

focus on two types of geometric uncertainty:

1. uncertainty pose of object, and
2. uncertainty pose of feature wrt corresponding object

uncertainty *coordinates represent* pose uncertainty of real frame wrt modeled frame:

$$\chi_u = \begin{pmatrix} \chi_{uI}{}^T & \chi_{uII}{}^T & \chi_{uIII}{}^T & \chi_{uIV}{}^T \end{pmatrix}^T \qquad (4)$$

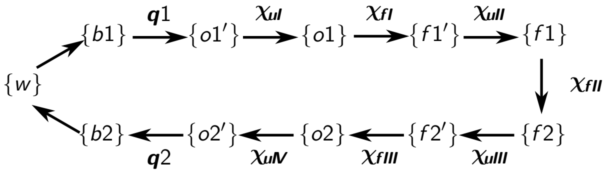
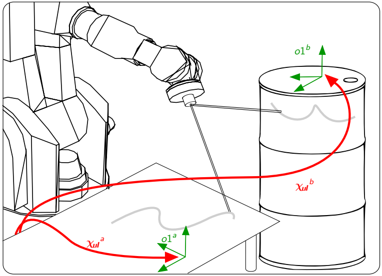


Figure: feature and uncertainty coordinates

STEP 3: uncertainty coordinates



- unknown position and orientation plane :

$$\chi_{ul}{}^{a} = \begin{pmatrix} h^a & \alpha^a & \beta^a \end{pmatrix}^T$$

- unknown position barrel:

$$\chi_{ul}{}^{b} = \begin{pmatrix} x_u^b & y_u^b \end{pmatrix}^T$$

task modeling procedure

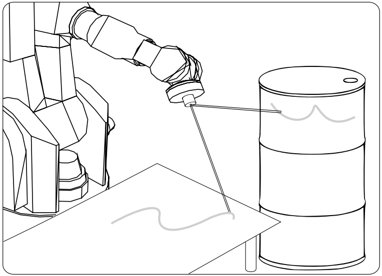
four steps:

1. identify objects and features and assign reference frames
2. choose feature coordinates χ_f
3. choose uncertainty coordinates χ_u
4. specify task

STEP 4: task specification

observation

task is easily specified using task coordinates χ_f and χ_u



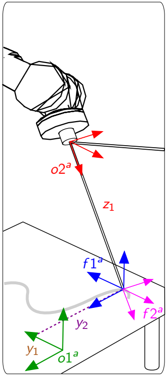
remember: task objective is twofold:

1. trace desired figure on plane
2. trace desired figure on barrel

STEP 4: task specification

observation

task is easily specified using task coordinates χ_f and χ_u



- **output equations:**
  - for the plane:

$$y_1 = x^a \quad \text{and} \quad y_2 = y^a$$

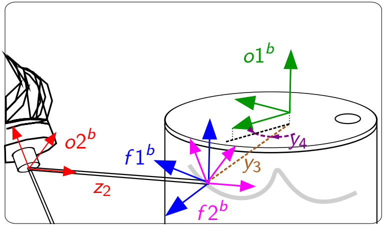
  - for the barrel
- **constraint equations:**
  in this example the desired paths are circles: $y_{id}(t)$, for $i = 1, \ldots, 4$
- **measurement equations:**

$$z_1 = z^a \quad \text{and} \quad z_2 = z^b$$

STEP 4: task specification

observation

task is easily specified using task coordinates χ_f and χ_u



- **output equations:**
  - for the plane
  - for the barrel:

$$y_3 = x^b \quad \text{and} \quad y_4 = \alpha^b$$

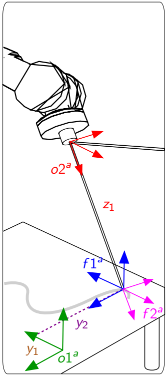
- **constraint equations:**
  in this example the desired paths are circles: $y_{id}(t)$, for $i = 1, \ldots, 4$
- **measurement equations:**

$$z_1 = z^a \quad \text{and} \quad z_2 = z^b$$

STEP 4: task specification

observation

task is easily specified using task coordinates χ_f and χ_u



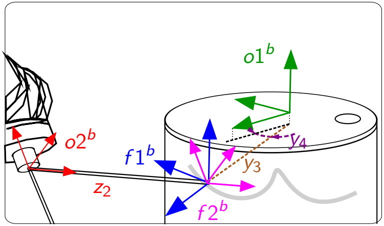
- **output equations:**
  - for the plane
  - for the barrel
- **constraint equations:**
  in this example the desired paths are circles: $y_{id}(t)$, for $i = 1, \ldots, 4$
- **measurement equations:**

$$z_1 = z^a \quad \text{and} \quad z_2 = z^b$$

STEP 4: task specification

observation

task is easily specified using task coordinates χ_f and χ_u



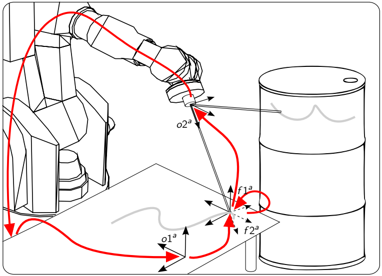
- **output equations:**
  - for the plane
  - for the barrel
- **constraint equations:**
  in this example the desired paths are circles: $y_{id}(t)$, for $i = 1, \ldots, 4$
- **measurement equations:**

$$z_1 = z^a \quad \text{and} \quad z_2 = z^b$$

STEP 4: task specification

observation

task is easily specified using task coordinates χ_f and χ_u



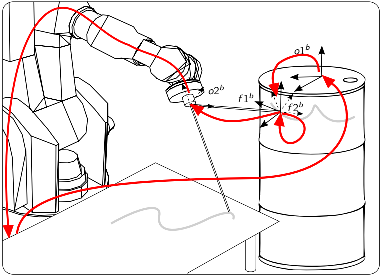
position loop constraints:
two position loop constraints, one for each feature relationship

- laser-plane feature *a*
- laser-barrel feature *b*

STEP 4: task specification

observation

task is easily specified using task coordinates χ_f and χ_u



position loop constraints:
two position loop constraints, one for each feature relationship

- laser-plane feature *a*
- laser-barrel feature *b*

task modeling

conclusion

- application dependent - but systematic modeling procedure provided easy task specification and uncertainty modeling
- application independent controller and model update and estimation block automatically derived

$\Rightarrow$ overall fast and easy task specification

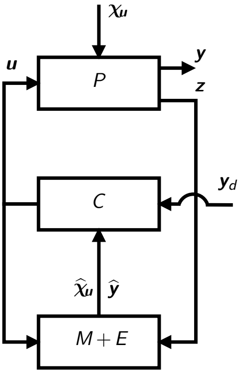


Figure: general control scheme

overview

Equations (1)

- *robot system equation:* relates the control input $\boldsymbol{u}$ to the rate of change of the robot system state:

$$\frac{d}{dt}\begin{pmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{pmatrix} = \boldsymbol{s}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{u}) \quad (5)$$

- *output equation:* relates the position based outputs $\boldsymbol{y}$ to the joint and feature coordinates:

$$\boldsymbol{f}(\boldsymbol{q}, \chi_{\boldsymbol{f}}) = \boldsymbol{y} \quad (6)$$

Equations (2)

- *measurement equation:* relates the position based measurements $\boldsymbol{z}$ to the joint and feature coordinates:

$$\boldsymbol{h}(\boldsymbol{q}, \chi_{\boldsymbol{f}}) = \boldsymbol{z} \tag{7}$$

- *artificial constraints:* used to specify the task:

$$\boldsymbol{y} = \boldsymbol{y}_d \tag{8}$$

- *natural constraints:* for rigid environments:

$$\boldsymbol{g}(\boldsymbol{q}, \chi_{\boldsymbol{f}}) = \mathbf{0} \tag{9}$$

$\longrightarrow$ special case of the artificial constraints with $\boldsymbol{y}_d = 0$

Equations (3)

- dependency relation between $\boldsymbol{q}$ and $\chi_{\boldsymbol{f}}$, perturbed by uncertainty coordinates $\chi_{\boldsymbol{u}}$:

$$\boldsymbol{l}(\boldsymbol{q}, \chi_{\boldsymbol{f}}, \chi_{\boldsymbol{u}}) = \boldsymbol{0} \tag{10}$$

$\longrightarrow$ nonholonomic systems: replace $\boldsymbol{q}$ by operational coordinates $\chi_{\boldsymbol{q}}$
$\longrightarrow$ derived using position closure equations $\Rightarrow$ *loop constraints*

auxiliary coordinates

the benefit of introducing feature coordinates $\chi_{\boldsymbol{f}}$ is that they can be chosen according to the specific task at hand, such that equations (6)–(9) can much be simplified. A similar freedom of choice exists for the uncertainty coordinates in equation (10)

control law

goal

1. provide system input $\boldsymbol{u}$ at each time step

- here: assume a velocity-controlled robot ($\boldsymbol{u} = \dot{\boldsymbol{q}}_d$)
- control law is based on system linearization, resulting in an equation of the form (details in appendix):

$$\boldsymbol{A}\dot{\boldsymbol{q}}_d = \dot{\boldsymbol{y}}_d^\circ + \boldsymbol{B}\widehat{\dot{\chi}}_{\boldsymbol{u}}, \tag{11}$$

with

$$\dot{\boldsymbol{y}}_d^\circ = \dot{\boldsymbol{y}}_d + \boldsymbol{K}_p(\boldsymbol{y}_d - \boldsymbol{y}) \tag{12}$$

- *weighted pseudo-inverse solving approach* can handle over- and/or underconstrained cases next to constraint weighting: levels of constraints based on nullspace projections

model update and estimation

goal

1. provide estimate for system outputs $\boldsymbol{y}$ used in feedback terms of constraint equations (12)
2. provide estimate for the uncertainty coordinates $\boldsymbol{\chi_u}$ used in control input (**??**)
3. maintain consistency between joint and feature coordinates $\boldsymbol{q}$ and $\boldsymbol{\chi_f}$ based on the loop constraints

model update and estimation

model update and estimation is based on an extended system model:

$$\frac{d}{dt}\begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{\chi_f} \\ \boldsymbol{\chi_u} \\ \dot{\boldsymbol{\chi}}_u \\ \ddot{\boldsymbol{\chi}}_u \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{J_f}^{-1}\boldsymbol{J_u} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{q} \\ \boldsymbol{\chi_f} \\ \boldsymbol{\chi_u} \\ \dot{\boldsymbol{\chi}}_u \\ \ddot{\boldsymbol{\chi}}_u \end{pmatrix} + \begin{pmatrix} \boldsymbol{1} \\ -\boldsymbol{J_f}^{-1}\boldsymbol{J_q} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{pmatrix} \dot{\boldsymbol{q}}_d \qquad (13)$$

explanation:

1. first row: system equation
2. second row: time-derivative of loop closure $\boldsymbol{l}(\boldsymbol{q}, \boldsymbol{\chi_f}, \boldsymbol{\chi_u}) = \boldsymbol{0}$
3. further rows: 'motion models' for uncertainty coordinates $\boldsymbol{\chi_u}$ (in this example: constant acceleration model)

this model is used in an estimator, e.g. Kalman filter or particle filter

model update and estimation

prediction-correction procedure

- **prediction**
  1. generate prediction based on extended system model
  2. eliminate inconsistencies between predicted estimates
- **correction**
  1. generate updated estimated based on predicted estimates and information from sensor measurements
  2. eliminate inconsistencies between predicted estimates

overview

example applications

rehabilitation robot (LWR)

- shared control between robot and human (conflicting constraints)
- constraint weighting (hence impedance) varies during therapy
- trajectory constraints imposed by robot are collected from demonstration by healthy human

example applications

human-robot comanipulation with PR2-robot

- robot head tracks head of human
- grippers are kept parallel and at constant distance
- end effector wrenches are controlled to zero
- joint limits are avoided (inequality constraints)
- obstacle in environment is avoided (inequality constraint)

overview

status, extensions & outlook

lowest level (constraint level)

- both equality and inequality constraints
- control input at velocity, acceleration or torque level
- constraint weighting in constraint space (overconstrained case), joints space (underconstrained case) or constraint priorities based on null-spaces
- constraint values or trajectories can be obtained from (human) demonstrations

status, extensions & outlook

intermediate level (skill level)

- controlled by Finite State Machine
- activates/deactivates constraints
- changes priorities/weights
- changes desired constraint values

status, extensions & outlook

robot systems: holonomic/nonholonomic

- fixed arm
- mobile platforms
- mobile platforms with two arms
- quadrotor helicopter
- multiple robots
- ...

status, extensions & outlook

software support available

- constraint & skill level
- specification & control of constraints
- TODO: estimation of geometric uncertainties

from instantaneous optimal control to globally optimal control

- every robot task is formulated as a global constrained optimization problem (e.g. to plan optimal trajectory)
- fast numerical solver (ACADO) developed at KU Leuven (OPTEC) (OPTEC: Centre of Excellence 'Optimization in Engineering')

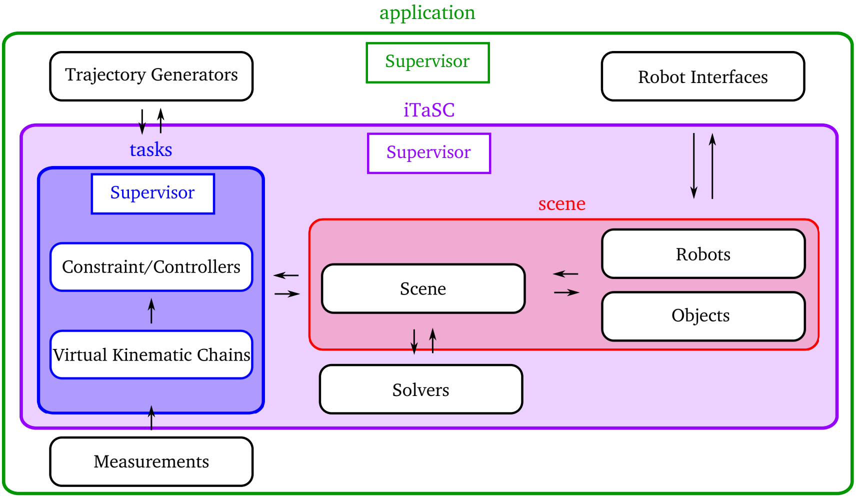
overview

software support

- ✓ modular design
- ✓ flexible user interface: add/remove constraints, change weights...
- ✓ modular task specification: share and reuse tasks
- ✓ separation of concerns: communication, computation, coordination, configuration, and connectivity

- implementation with Orocos
- code available under LGPL/BSD license
- www.orocos.org/itasc

software support



overview

conclusion (1)

conclusion

- motion specification and estimation in unified framework
- automatic application independent derivation of control and model update and estimation
- application dependent - but systematic - task modeling

further reading

framework journal paper

- Constraint-Based Task Specification and Estimation for Sensor-Based Robot Systems in the Presence of Geometric Uncertainty
- Joris De Schutter, Tinne De Laet, Johan Rutgeerts, Wilm Decré, Ruben Smits, Erwin Aertbeliën, Kasper Claes, and Herman Bruyninckx
- Journal of Robotics Research, May 2007, vol. 26, no. 5, pages 433–455

extended framework conference paper

- Extending iTaSC to Support Inequality Constraints and Non-Instantaneous Task Specification
- Wilm Decré, Ruben Smits, Herman Bruyninckx, and Joris De Schutter
- Proceedings of the International Conference on Robotics and Automation, 2009, pages 964–971

THANKS FOR YOUR ATTENTION!