# iTaSC concepts and tutorial

European Robotics Forum 2012, Odense, Denmark

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

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

KATHOLIEKE UNIVERSITEIT LEUVEN

# 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

## 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

# laser tracing task

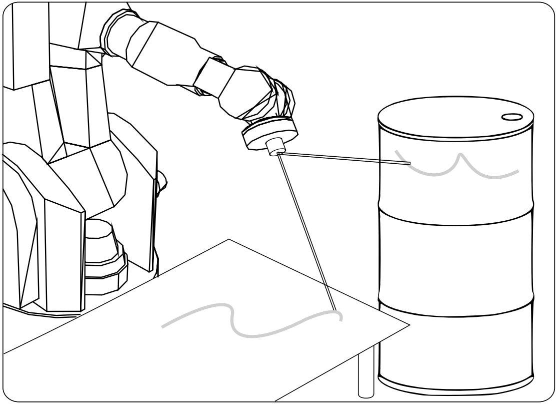


Figure: simultaneous laser tracing on a plane and a barrel

# 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

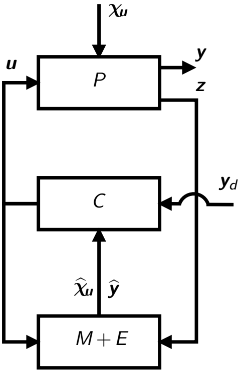



Figure: general control scheme

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

# control and estimation scheme

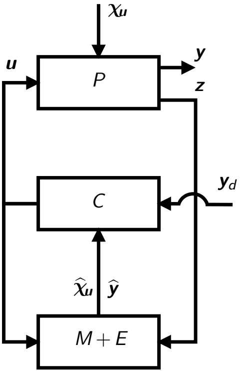



Figure: general control scheme

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*, $\chi_f$
  - *uncertainty coordinates*, $\chi_u$
- 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:

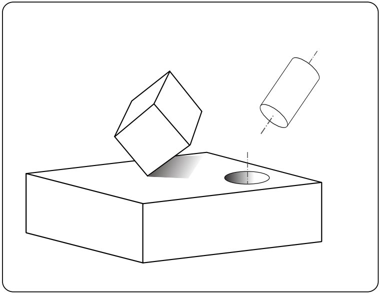
1. identify objects and features and assign reference frames
2. choose feature coordinates $\chi_f$
3. choose uncertainty coordinates $\chi_u$
4. specify task

# task modeling procedure

four steps:

1. identify objects and features and assign reference frames
2. choose feature coordinates $\chi_f$
3. choose uncertainty coordinates $\chi_u$
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

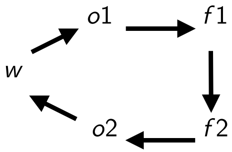



Figure: object and feature frames and feature coordinates

each constraint needs four frames:

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

# STEP 1: object and feature frames

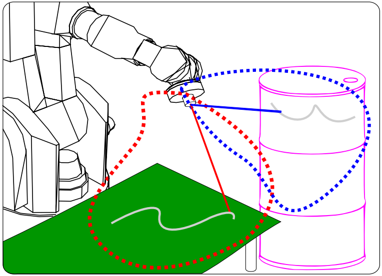
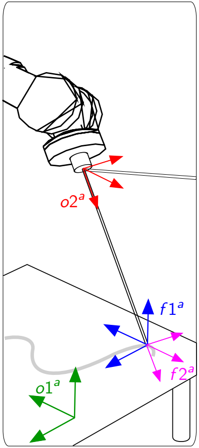


Figure: object and feature frames laser tracing

- 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

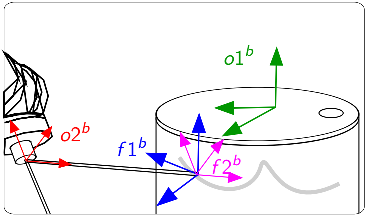
# STEP 1: object and feature frames




**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:

# 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 $\boldsymbol{\chi_f}$
3. choose uncertainty coordinates $\boldsymbol{\chi_u}$
4. specify task

# STEP 2: feature coordinates

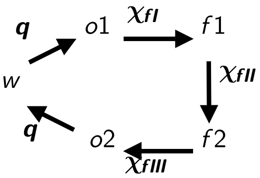
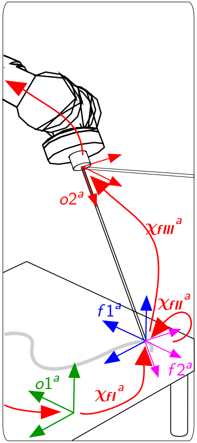



Figure: object and feature frames and feature coordinates

- 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

$$\boldsymbol{\chi_f} = \begin{pmatrix} \boldsymbol{\chi_{fI}}^T & \boldsymbol{\chi_{fII}}^T & \boldsymbol{\chi_{fIII}}^T \end{pmatrix}^T \quad (1)$$

# STEP 2: feature coordinates

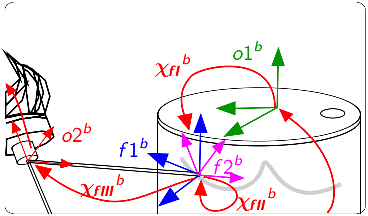



- laser-plane feature:

$$\chi_{fI}{}^a = \begin{pmatrix} x^a & y^a \end{pmatrix}^T \quad (2)$$
$$\chi_{fII}{}^a = \begin{pmatrix} \phi^a & \theta^a & \psi^a \end{pmatrix}^T \quad (3)$$
$$\chi_{fIII}{}^a = \begin{pmatrix} z^a \end{pmatrix} \quad (4)$$

- laser-barrel feature

# STEP 2: feature coordinates




- laser-plane feature
- laser-barrel feature:

$$\chi_{fI}{}^b = \begin{pmatrix} x^b & \alpha^b \end{pmatrix}^T \quad (2)$$

$$\chi_{fII}{}^b = \begin{pmatrix} \phi^b & \theta^b & \psi^b \end{pmatrix}^T \quad (3)$$

$$\chi_{fIII}{}^b = \begin{pmatrix} z^b \end{pmatrix} \quad (4)$$

# task modeling procedure

four steps:

1. identify objects and features and assign reference frames
2. choose feature coordinates $\boldsymbol{\chi_f}$
3. choose uncertainty coordinates $\boldsymbol{\chi_u}$
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:

$$\chi_u = \begin{pmatrix} \chi_{uI}^T & \chi_{uII}^T & \chi_{uIII}^T & \chi_{uIV}^T \end{pmatrix}^T \quad (5)$$

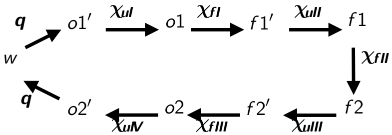
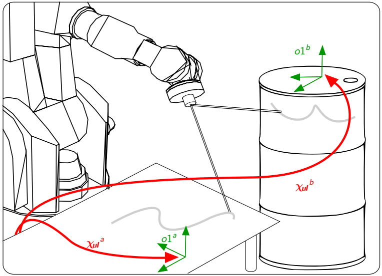



Figure: feature and uncertainty coordinates

# 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:

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



KATHOLIEKE UNIVERSITEIT LEUVEN

# task modeling procedure

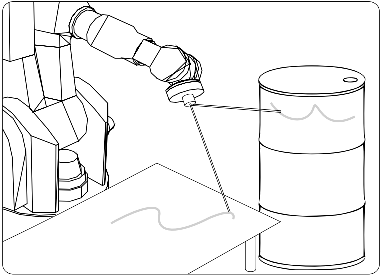
four steps:

1. identify objects and features and assign reference frames
2. choose feature coordinates $\boldsymbol{\chi_f}$
3. choose uncertainty coordinates $\boldsymbol{\chi_u}$
4. specify task

# STEP 4: task specification

**observation**

task is easily specified using task coordinates $\chi_f$ and $\chi_u$



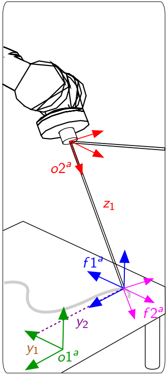
remember: task objective is twofold:

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

# STEP 4: task specification

## observation

task is easily specified using task coordinates $\chi_f$ and $\chi_u$




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

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

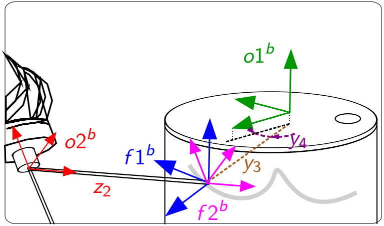
  - for the barrel
- **constraint equations:**
  in this example the desired paths are circles: $y_{id}(t)$, for $i = 1, \ldots, 4$
- **measurement equations:**

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

# STEP 4: task specification

## observation

task is easily specified using task coordinates $\chi_f$ and $\chi_u$




- **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:**

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

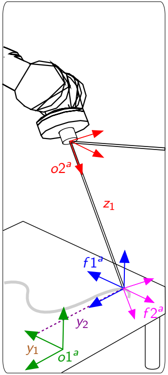


KATHOLIEKE UNIVERSITEIT LEUVEN

# STEP 4: task specification

## observation

task is easily specified using task coordinates $\chi_f$ and $\chi_u$




- **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:**

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

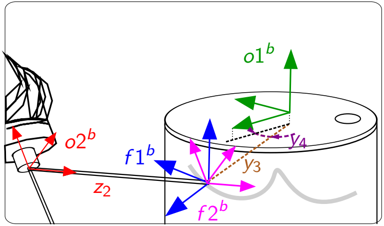


KATHOLIEKE UNIVERSITEIT LEUVEN

# STEP 4: task specification

## observation

task is easily specified using task coordinates $\chi_f$ and $\chi_u$



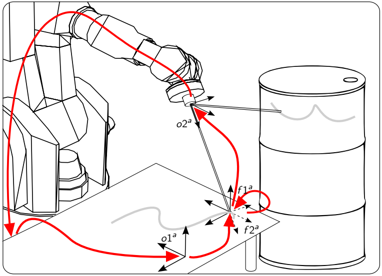

- **output equations:**
  - for the plane
  - for the barrel
- **constraint equations:**
  in this example the desired paths are circles: $y_{id}(t)$, for $i = 1, \ldots, 4$
- **measurement equations:**

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

# STEP 4: task specification

## observation

task is easily specified using task coordinates $\chi_f$ and $\chi_u$



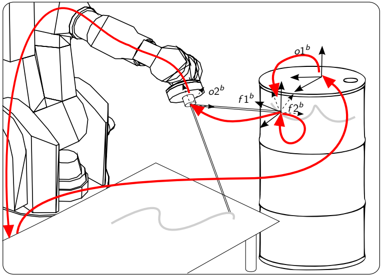

**position loop constraints:**
two position loop constraints, one for each feature relationship

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

# STEP 4: task specification

## observation

task is easily specified using task coordinates $\chi_f$ and $\chi_u$




**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

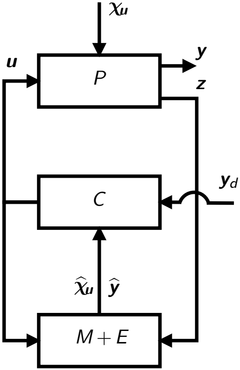



Figure: general control scheme

# overview

# Equations (1)

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

$$\frac{d}{dt}\begin{pmatrix}\boldsymbol{q}\\ \dot{\boldsymbol{q}}\end{pmatrix} = \boldsymbol{s}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{u}) \tag{6}$$

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

$$\boldsymbol{f}\left(\boldsymbol{q}, \chi_{\boldsymbol{f}}\right) = \boldsymbol{y} \tag{7}$$

# Equations (2)

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

$$\boldsymbol{h}(\boldsymbol{q}, \chi_{\boldsymbol{f}}) = \boldsymbol{z} \tag{8}$$

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

$$\boldsymbol{y} = \boldsymbol{y}_d \tag{9}$$

- *natural constraints:* for rigid environments:

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

$\rightarrow$ 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 $\chi_{\boldsymbol{u}}$:

$$\boldsymbol{l}(\boldsymbol{q}, \chi_{\boldsymbol{f}}, \chi_{\boldsymbol{u}}) = \boldsymbol{0} \tag{11}$$

  $\rightarrow$ nonholonomic systems: replace $\boldsymbol{q}$ by operational coordinates $\chi_{\boldsymbol{q}}$
  $\rightarrow$ 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 (7)–(10) can much be simplified. A similar freedom of choice exists for the uncertainty coordinates in equation (11)

# 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:

$$\boldsymbol{A}\dot{\boldsymbol{q}}_d = \dot{\boldsymbol{y}}_d^{\circ} + \boldsymbol{B}\widehat{\dot{\boldsymbol{\chi}}}_{\boldsymbol{u}} \tag{12}$$

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

# model update and estimation

## goal

1. provide estimate for system outputs $\boldsymbol{y}$ used in feedback terms of constraint equations (24)
2. provide estimate for the uncertainty coordinates $\boldsymbol{\chi_u}$ used in control input (26)
3. maintain consistency between joint and feature coordinates $\boldsymbol{q}$ and $\boldsymbol{\chi_f}$ based on the loop constraints

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

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

# 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

# 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

## remark

this presentation focused on the *basic* functionality of the framework
further generalizations include inequality constraints and motion planning

# 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!

# overview

# human-robot co-manipulation

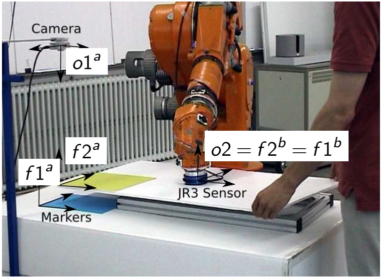



Figure: the experimental setup for the human-robot co-manipulation task

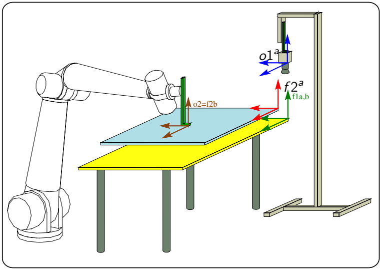



Figure: the object and feature frames for a human-robot co-manipulation task

# object and feature frames

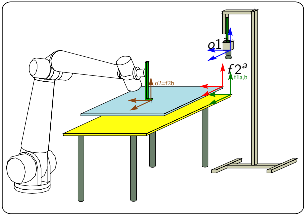



Figure: the object and feature frames for a human-robot co-manipulation task

- natural task description imposes three motion constraints:
  - align one side of the object according to the camera
  - carry the weight and generate downward motion to realize desired contact force
  - follow human intent
- ⇒two feature relationships:
  - feature *a* : visual servoing
  - feature *b*: force control
- the objects are:
  1. the environment (or camera)
  2. the object

# object and feature frames

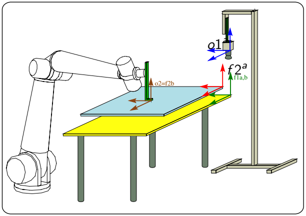



Figure: the object and feature frames for a human-robot co-manipulation task

- frame $o1^a$ fixed to robot environment (camera)
- frame $o2$ at center of object
- $o1b$ fixed to $o2$ by a compliance
- frame $f1^a$ at reference pose on support
- frame $f2^a$ fixed to the object
- no force $\Rightarrow$ frames $f1^b$ and $f2^b$ coincide with $o2$,
  forces $\Rightarrow$ $f1^b$ and $f2^b$ deviate from each other

# feature coordinates

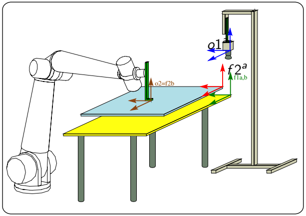



Figure: the object and feature frames for a human-robot co-manipulation task

- for feature $a$:

$$\chi_{fI}{}^{a} = \begin{pmatrix} - \end{pmatrix} \quad (14)$$

$$\chi_{fII}{}^{a} = \begin{pmatrix} x^a & y^a & z^a & \phi^a & \theta^a & \psi^a \end{pmatrix}^T \quad (15)$$

$$\chi_{fIII}{}^{a} = \begin{pmatrix} - \end{pmatrix} \quad (16)$$

- for feature $b$:

$$\chi_{fI}{}^{b} = \begin{pmatrix} - \end{pmatrix} \quad (17)$$

$$\chi_{fII}{}^{b} = \begin{pmatrix} x^b & y^b & z^b & \phi^b & \theta^b & \psi^b \end{pmatrix}^T \quad (18)$$

$$\chi_{fIII}{}^{b} = \begin{pmatrix} - \end{pmatrix} \quad (19)$$

# task specification

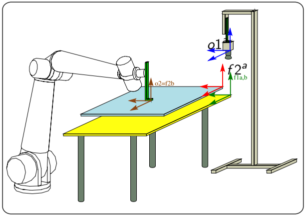



Figure: the object and feature frames for a human-robot co-manipulation task

- **output equations:**
  - camera:
  $$y_1 = x^a, \quad y_2 = y^a \tag{14}$$
  - contact force with support:
  $$y_3 = F_z = k_z x^b, \quad y_4 = T_x = k_{\alpha x}\phi^b, \quad y_5 = T_y = k_{\alpha y}\theta^b \tag{15}$$
  - human interaction:
  $$y_6 = F_x = k_x x^b, \quad y_7 = F_y = k_y y^b, \quad y_8 = T_z = k_{\alpha z}\psi^b \tag{16}$$
- **constraint equations:**
- **measurement equations:**

# task specification

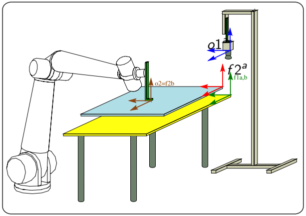



Figure: the object and feature frames for a human-robot co-manipulation task

- **output equations:**
- **constraint equations:**

$$y_{1d} = 0\text{mm}, \quad y_{2d} = 60\text{mm}$$
$$y_{3d} = F_{zd}, \quad y_{4d} = 0, \quad y_{5d} = 0$$
$$y_{6d} = y_{7d} = y_{8d} = 0 \tag{14}$$

- **measurement equations:** in this example all the outputs can be measured:

$$z_i = y_i \quad \text{for } i = 1, \ldots, 8 \tag{15}$$

# results

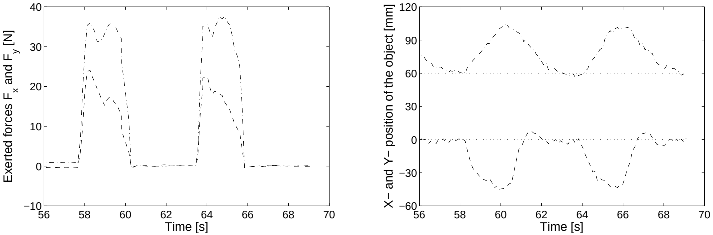



Figure: the left plot shows the forces $F_x$ and $F_y$, exerted by the operator during the co-manipulation task. the right plot shows the alignment errors $x^a$ and $y^a$ as measured by the camera.

# mobile robot

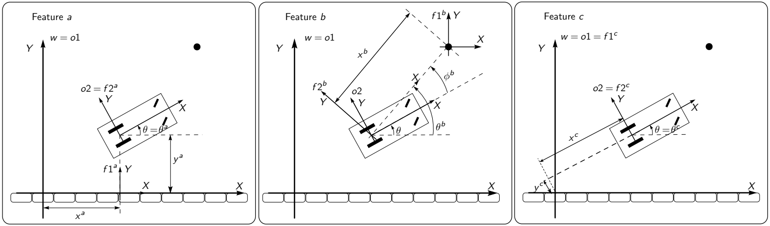



Figure: left for feature $a$, ultrasonic sensor; middle for feature $b$, range finder; right for feature $c$, robot trajectory

# object and feature frames

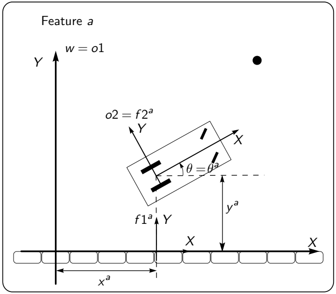



Figure: feature *a*

- task description: move robot along a trajectory with respect to the world while measuring distance to a wall with ultrasonic sensor and measuring the distance and angle to a beacon
- ⇒three feature relationships:
  1. feature *a*: ultrasonic sensor
  2. feature *b*: range finder
  3. feature *c*: motion specification
- the objects are:
  1. mobile robot
  2. environment (wall, beacon)

# object and feature frames

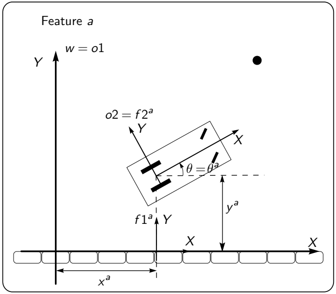



Figure: feature $a$

- frame $o1$, fixed to wall, its $x$-axis along the wall
- frame $o2$, fixed to mobile robot
- for feature $a$ (ultrasonic sensor):
  - frame $f1^a$, same orientation as $o1$ and able to move in $x$ direction of $o1$
  - frame $f2^a$, fixed to frame $o2$

# object and feature frames

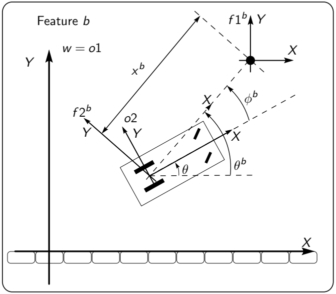



Figure: feature $b$

- frame $o1$, fixed to wall, its $x$-axis along the wall
- frame $o2$, fixed to mobile robot
- for feature $b$ (range finder):
  - frame $f1^b$, at the beacon location, fixed to frame $o1$
  - frame $f2^b$, $x$-axis is beam of range finder hitting the beacon

# object and feature frames

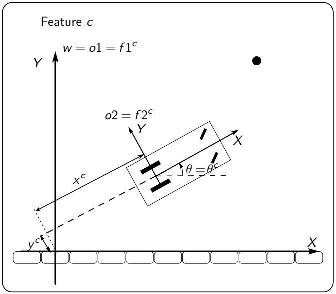



Figure: feature $c$

- frame $o1$, fixed to wall, its $x$-axis along the wall
- frame $o2$, fixed to mobile robot
- for feature $c$ (path tracking):
  - frame $f1^c$, coinciding with $o1$
  - frame $f2^c$, coinciding with $o2$

# feature coordinates

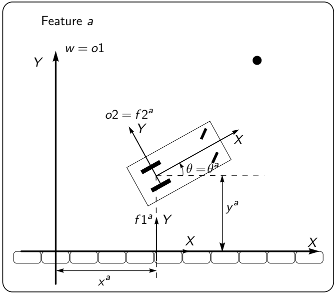



Figure: feature $a$

for each of the three features a minimal set of position coordinates exists representing the 3DOF between $o1$ and $o2$:

- for feature $a$ (ultrasonic sensor):

$$\chi_{fI}{}^a = \begin{pmatrix} x^a \end{pmatrix} \quad (16)$$

$$\chi_{fII}{}^a = \begin{pmatrix} y^a & \theta^a \end{pmatrix}^T \quad (17)$$

$$\chi_{fIII}{}^a = \begin{pmatrix} - \end{pmatrix} \quad (18)$$

# feature coordinates

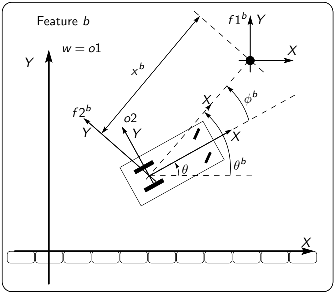



Figure: feature $b$

for each of the three features a minimal set of position coordinates exists representing the 3DOF between $o1$ and $o2$:

- for feature $b$ (range finder):

$$\chi_{fI}{}^{b} = \begin{pmatrix} - \end{pmatrix} \quad (16)$$

$$\chi_{fII}{}^{b} = \begin{pmatrix} x^b & \theta^b \end{pmatrix}^T \quad (17)$$

$$\chi_{fIII}{}^{b} = \begin{pmatrix} \phi^b \end{pmatrix} \quad (18)$$

# feature coordinates

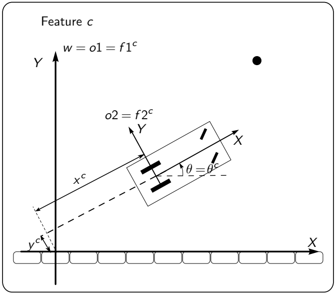



Figure: feature $c$

for each of the three features a minimal set of position coordinates exists representing the 3DOF between $o1$ and $o2$:

- for feature $c$ (path tracking):

$$\chi_{fI}{}^{c} = \left( \; - \; \right) \quad (16)$$

$$\chi_{fII}{}^{c} = \left( \; x^c \quad y^c \quad \theta^c \; \right)^T \quad (17)$$

$$\chi_{fIII}{}^{c} = \left( \; - \; \right) \quad (18)$$

# operational space robot coordinates

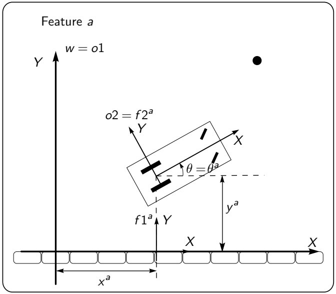



Figure: feature $a$

Nonholonomic robot:

- position loop constraints cannot be written in terms of $\boldsymbol{q}$
- $\Rightarrow$ define operational space robot coordinates $\boldsymbol{\chi_q}$
- natural choice: $\boldsymbol{\chi_q} = \boldsymbol{\chi_f}^c$
- dependency relation between $\dot{\boldsymbol{\chi}}_{\boldsymbol{q}}$ and $\dot{\boldsymbol{q}}$ is very simple: (*nonholonomic constraint*)

$$\dot{\boldsymbol{\chi}}_{\boldsymbol{q}} = \begin{pmatrix} \dot{x}^c \\ \dot{y}^c \\ \dot{\theta}^c \end{pmatrix} = \boldsymbol{J_r}\dot{\boldsymbol{q}} \qquad (16)$$

# uncertainty coordinates

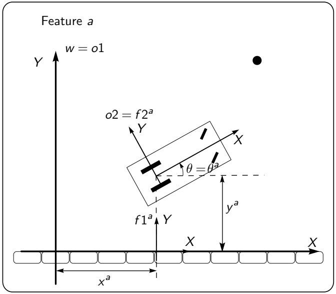



Figure: feature *a*

Nonholonomic robot:

- dependency relation between $\dot{\chi}_{\boldsymbol{q}}$ and $\dot{\boldsymbol{q}}$ is very simple: (*nonholonomic constraint*)

$$\dot{\chi}_{\boldsymbol{q}} = \begin{pmatrix} \dot{x}^c \\ \dot{y}^c \\ \dot{\theta}^c \end{pmatrix} = \boldsymbol{J_r}\dot{\boldsymbol{q}} \qquad (16)$$

- replace $\boldsymbol{q}$ in (7) and (11) by $\chi_{\boldsymbol{q}}$results in:

$$\boldsymbol{C_q} = \frac{\partial \boldsymbol{f}}{\partial \chi_{\boldsymbol{q}}}\boldsymbol{J_r} \quad \boldsymbol{J_q} = \frac{\partial \boldsymbol{I}}{\partial \chi_{\boldsymbol{q}}}\boldsymbol{J_r} \qquad (17)$$

# uncertainty coordinates

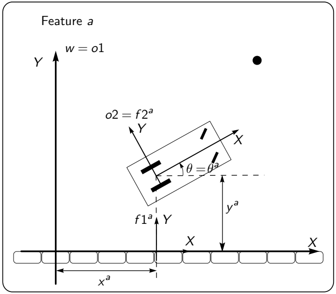



Figure: feature $a$

- the nonholonomic constraint which may be disturbed by wheel slip:

$$\dot{\chi}_{\boldsymbol{q}} = \boldsymbol{J_r}\left(\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}_{slip}\right) \qquad (16)$$

- $\dot{\boldsymbol{q}}_{slip} = s\dot{\boldsymbol{q}}$, with $s$ the estimated slip rate
- $\Rightarrow \chi_{\boldsymbol{uN}} = \boldsymbol{q}_{slip}$ and from (20): $\boldsymbol{J_u} = \boldsymbol{J_q}$

# task specification

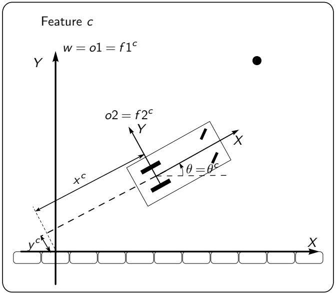



Figure: feature $c$

- **output equations**

$$y_1 = x^c, \quad y_2 = y^c, \quad y_3 = \theta^c.(16)$$

- **constraint equations:**
  from the desired path in terms of $x^a$, $y^a$ and $\theta^a$, the desired values $y_{1d}(t)$, $y_{2d}(t)$ and $y_{3d}(t)$ can be obtained

- **measurement equations:**

$$z_1 = y^a, \quad z_2 = x^b, \quad z_3 = \theta^b \,(17)$$

# feedback control

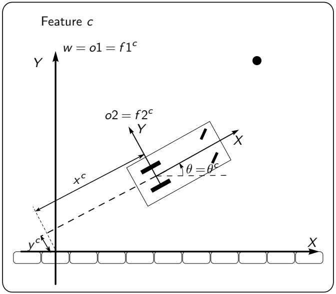



Figure: feature $c$

the path controller is implemented in operation space, by applying constraints (24) with

$$\boldsymbol{K_p} = \begin{pmatrix} k_p & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{{k_p}^2}{2sign(\dot{x}_c)} & k_p \end{pmatrix}, \qquad (16)$$

and $k_p$ a feedback constant

# results

without slip:

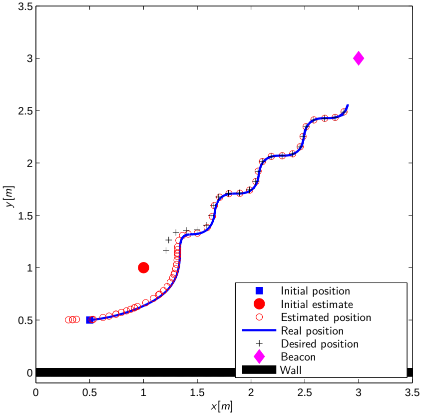
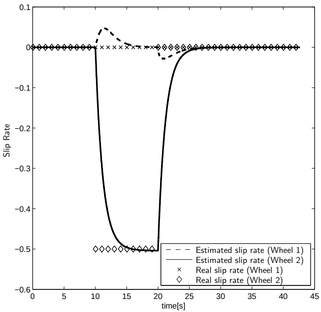


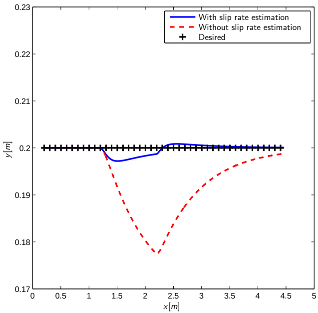

Figure: localization and path tracking control of a mobile robot

# results

with slip:




(a) estimation of the slip rate on both wheels




(b) trajectory of mobile robot

Figure: localization and path tracking control of a mobile robot with slip

# multiple robots with simultaneous tasks

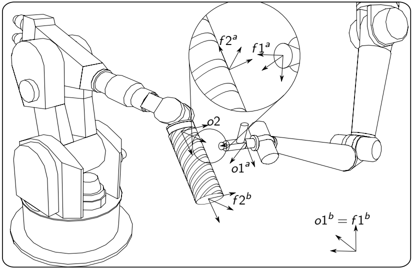



Figure: two robots performing simultaneous pick-and-place and painting operations on a single work piece

# overview

control details

- control law
- closed loop behavior
- invariant constraint weighting

# control law (1)

- differentiate output equation (7) to obtain an output equation at velocity level:

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} + \frac{\partial \boldsymbol{f}}{\partial \chi_{\boldsymbol{f}}}\dot{\chi}_{\boldsymbol{f}} = \dot{\boldsymbol{y}}, \tag{17}$$

written as:

$$\boldsymbol{C_q}\dot{\boldsymbol{q}} + \boldsymbol{C_f}\dot{\chi}_{\boldsymbol{f}} = \dot{\boldsymbol{y}}. \tag{18}$$

- differentiate position loop constraint (11):

$$\frac{\partial \boldsymbol{l}}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} + \frac{\partial \boldsymbol{l}}{\partial \chi_{\boldsymbol{f}}}\dot{\chi}_{\boldsymbol{f}} + \frac{\partial \boldsymbol{l}}{\partial \chi_{\boldsymbol{u}}}\dot{\chi}_{\boldsymbol{u}} = \boldsymbol{0} \tag{19}$$

or:

$$\boldsymbol{J_q}\dot{\boldsymbol{q}} + \boldsymbol{J_f}\dot{\chi}_{\boldsymbol{f}} + \boldsymbol{J_u}\dot{\chi}_{\boldsymbol{u}} = \boldsymbol{0} \tag{20}$$

# control law (2)

- $\dot{\chi}_f$ solved from (20):

$$\dot{\chi}_f = -\boldsymbol{J_f}^{-1}\left(\boldsymbol{J_q}\dot{\boldsymbol{q}} + \boldsymbol{J_u}\dot{\chi}_u\right) \tag{21}$$

- substituting (21) into (18) yields the modified output equation:

$$\boldsymbol{A}\dot{\boldsymbol{q}} = \dot{\boldsymbol{y}} + \boldsymbol{B}\dot{\chi}_u \tag{22}$$

  where $\boldsymbol{A} = \boldsymbol{C_q} - \boldsymbol{C_f}\boldsymbol{J_f}^{-1}\boldsymbol{J_q}$ and $\boldsymbol{B} = \boldsymbol{C_f}\boldsymbol{J_f}^{-1}\boldsymbol{J_u}$.

- plant assumed to be ideal velocity controlled system:

$$\dot{\boldsymbol{q}} = \boldsymbol{u} = \dot{\boldsymbol{q}}_d. \tag{23}$$

# control law (3)

- Constraint equation (9) expressed at velocity level and include feedback:

$$\dot{\boldsymbol{y}} = \underbrace{\dot{\boldsymbol{y}}_d + \boldsymbol{K}_p \left( \boldsymbol{y}_d - \boldsymbol{y} \right)}_{\dot{\boldsymbol{y}}_d^{\circ}} \tag{24}$$

- Applying constraint (24) to (22), and substituting system equation (23):

$$\boldsymbol{A} \dot{\boldsymbol{q}}_d = \dot{\boldsymbol{y}}_d^{\circ} + \boldsymbol{B} \widehat{\dot{\boldsymbol{\chi}}}_{\boldsymbol{u}} \tag{25}$$

Solving for the control input $\dot{\boldsymbol{q}}_d$:

$$\dot{\boldsymbol{q}}_d = \boldsymbol{A}_{\boldsymbol{W}}^{\#} \left( \dot{\boldsymbol{y}}_d^{\circ} + \boldsymbol{B} \widehat{\dot{\boldsymbol{\chi}}}_{\boldsymbol{u}} \right) \tag{26}$$

# closed loop behavior

substituting control input (26) in system equation (23) and then in output equation (22), and solving for $\dot{\boldsymbol{y}}$:

$$\dot{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{A}^{\#}_{\boldsymbol{W}}\dot{\boldsymbol{y}}^{\circ}_{d} + \left(\boldsymbol{A}\boldsymbol{A}^{\#}_{\boldsymbol{W}} - \boldsymbol{1}\right)\boldsymbol{B}\dot{\boldsymbol{\chi}}_{\boldsymbol{u}} + \boldsymbol{A}\boldsymbol{A}^{\#}_{\boldsymbol{W}}\boldsymbol{B}\left(\widehat{\dot{\boldsymbol{\chi}}}_{\boldsymbol{u}} - \dot{\boldsymbol{\chi}}_{\boldsymbol{u}}\right) \qquad (27)$$

# invariant constraint weighting

- *pseudo-inverse approach* to handle over- and/or underconstrained cases
- in joint space: mass matrix of robot
- in Cartesian space, $\boldsymbol{W} = diag(w_i^2)$, with:

$$w_i = \alpha \frac{1}{\Delta_{pi} k_{pi}} \quad \text{or} \quad w_i = \alpha \frac{1}{\Delta_{vi}} \tag{28}$$

- next to weighting: levels of constraints based on nullspace projections