iTASC: a Tool for Multi-Sensor Integration in Robot Manipulation

Ruben Smits, Tinne De Laet, Kasper Claes, Herman Bruyninckx, Joris De Schutter

***Abstract*— iTASC (acronym for ‘instantaneous task specification and control’) [1] is a systematic constraint-based approach to specify complex tasks of general sensor-based robot systems. iTASC integrates both instantaneous task specification and estimation of geometric uncertainty in a unified framework. Automatic derivation of controller and estimator equations follows from a geometric task model that is obtained using a *systematic task modeling* procedure. The approach applies to a large variety of robot systems (mobile robots, multiple robot systems, dynamic human-robot interaction, etc.), various sensor systems, and different robot tasks.**

Using an example task, this paper shows that iTASC is a powerful tool for multi-sensor integration in robot manipulation. The example task includes multiple sensors: encoders, a force sensor, cameras, a laser distance sensor and a laser scanner. The paper details the systematic modeling procedure for the example task and elaborates on the task specific choice of two types of task coordinates: feature coordinates, defined with respect to object and feature frames, which facilitate the task specification, and uncertainty coordinates to model geometric uncertainty. Experimental results for the example task are presented.

I. INTRODUCTION

The goal of our research is to develop programming support for the implementation of complex, sensor-based robotic tasks in the presence of geometric uncertainty. Examples of complex tasks include sensor-based navigation and 3D manipulation in partially or completely unknown environments, using redundant robotic systems such as mobile manipulator arms, cooperating robots, robotic hands or humanoid robots, and using multiple sensors such as vision, force, torque, tactile and distance sensors.

The foundation for this programming support is iTASC, a generic and systematic approach [1] to specify and control a task while dealing properly with geometric uncertainty.

Previous work on specification of sensor-based robot tasks, such as force controlled manipulation [2]–[5] or force controlled compliant motion combined with visual servoing [6], was based on the concept of the *compliance frame* [7] or *task frame* [8]. In this frame, different control modes, such as trajectory following, force control, visual servoing or distance control, are assigned to each of the translational directions along the frame axes and to each of the rotational directions about the frame axes. The drawback of the task frame approach is that it only applies to task geometries for which separate control modes can be assigned independently to three pure translational and three pure rotational directions along the axes of a *single* frame.

A more systematic approach is to assign control modes and corresponding constraints to *arbitrary* directions in the six dimensional manipulation space. This approach, known as *constraint-based programming*, opens up new applications involving a much more complex geometry and/or involving multiple sensors that control different directions in space simultaneously.

Seminal theoretical work on constraint-based programming of robot tasks was done by Ambler and Popplestone [9] and by Samson and coworkers [10]. Our own preliminary work on iTASC was presented in [11], while iTASC, the mature framework of which this paper shows an application, is thoroughly discussed in [1]. Other applications of iTASC were presented in [12] and [13].

This paper is organized as follows. Section II introduces the example task and states the contribution of the paper. Section III provides a brief overview of the generic control and estimation approach. Section IV applies the task modeling procedure of [1] to the example task. Sections V and VI provide details on the used control and estimation equations. Finally, Section VII and Section VIII present experimental results and state the conclusions.

II. EXAMPLE TASK

This paper shows the application of iTASC to a multi-robot manipulation task involving multiple sensors, under-constrained specification as well as estimation of uncertain geometric parameters.

A. Robot system

The robot system consists of two 6 degrees of freedom robots.

B. Robot task

The task consists of three subtasks which are executed simultaneously:

1) the contour of an unknown $2.5D$ workpiece, held by robot 1, is tracked by a probe mounted on robot 2, as shown in Fig. 2(a). Both the contact force between probe and contour and the tangential velocity along the contour are controlled. In addition, the orientation of the probe with respect to the contour tangent and the distance of the probe with respect to the front

Manuscript received April 30, 2008.

This work was supported by the K.U.Leuven’s Concerted Research Action GOA/05/10. Tinne De Laet is a Doctoral Fellow of the Fund for Scientific Research–Flanders (F.W.O.) in Belgium.

All authors are with the Department of Mechanical Engineering, Katholieke Universiteit Leuven, Celestijnenlaan 300B, B-3001 Leuven, Belgium.

Corresponding author: Ruben Smits (phone: +32-16322480; fax: +32-16322987; e-mail: ruben.smits@mech.kuleuven.be).

plane of the workpiece are kept constant, while the probe axis remains perpendicular to the front plane of the workpiece. This subtask specifies six (equality) constraints.

2) the end effector of robot 1 keeps a specified minimum distance to the closest person as shown in Fig. 2(b). This subtask specifies one (inequality) constraint.
3) a camera (camera 1) attached to robot 1, shown in Fig. 2(c), keeps the closest person in vertical position in the middle of the image. This subtask specifies three (equality) constraints.

All together the contour following task in the human populated environment specifies 10 (or 9)[1] constraints for the 12 dof robot system. The two (or three) remaining dofs of the robot system are used to satisfy 12 secondary constraints: to keep the robot as close as possible to its nominal working position.

C. Geometric uncertainty

In this application following sources of geometric uncertainty are explicitly modeled: *(1)* the shape of the describing contour of the $2.5D$ workpiece; *(2)* the height and orientation of the front plane of the workpiece relative to the end effector of robot 1, and *(3)* the position of the closest person, expressed in x, y-coordinates of the world frame. All together these sources of geometric uncertainty are modeled by 7 coordinates. All other geometric transformations are supposed to be known, even though there may be other sources of geometric uncertainty. For example, the relative position between both robots is only poorly calibrated, which introduces important geometric errors.

D. Sensors

Robot 1 is equipped with a camera (camera 1, Fig. 2(a)) mounted on its end effector. This camera provides an image of the closest person, but this image is not used for visual servoing. Robot 2 is equipped with three sensors mounted on its end effector (Fig. 2(b)): *(1)* a six axis force/torque sensor is used to control the contact force between the probe and the contour, and to estimate the orientation of the contour tangent and normal. Only two force components of the sensor are used in this application; *(2)* a camera (camera 2) directed towards the contact point provides a second estimate of the contour tangent and normal; *(3)* (not shown in Fig. 2(b):) a laser distance sensor measures the distance of the end effector to the front plane of the workpiece. This information is used to control the distance of the probe with respect to the front plane of the workpiece and to keep the probe axis perpendicular to this plane. Additionally, the robot system contains a horizontal laser scanner (Fig. 2(b)-2(c)), fixed to the environment, which is used to estimate the motion of people in the robot environment. All together 6 scalar signals are measured and fed back to the robot system.

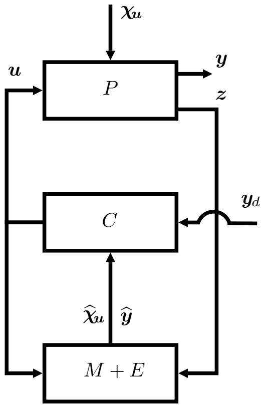


Fig. 1. General control and estimation scheme including plant *P*, controller *C*, and model update and estimation block *M*+*E*.

E. Contribution of the paper

The main contributions of the paper are: *(1)* to show how to derive the control and estimation equations for this task involving 10 primary constraints, 7 uncertainty coordinates, 6 scalar measurements and 12 secondary constraints, *(2)* to present experimental results for this integrated task, *(3)* the development of a laser scanner based people tracker, Section VI-A, *(4)* extension of the $2D$ contour tracking approach [1] to a $3D$ context, and *(5)* presentation of experimental results for this contour tracking approach ([1] only contains a $2D$ simulation).

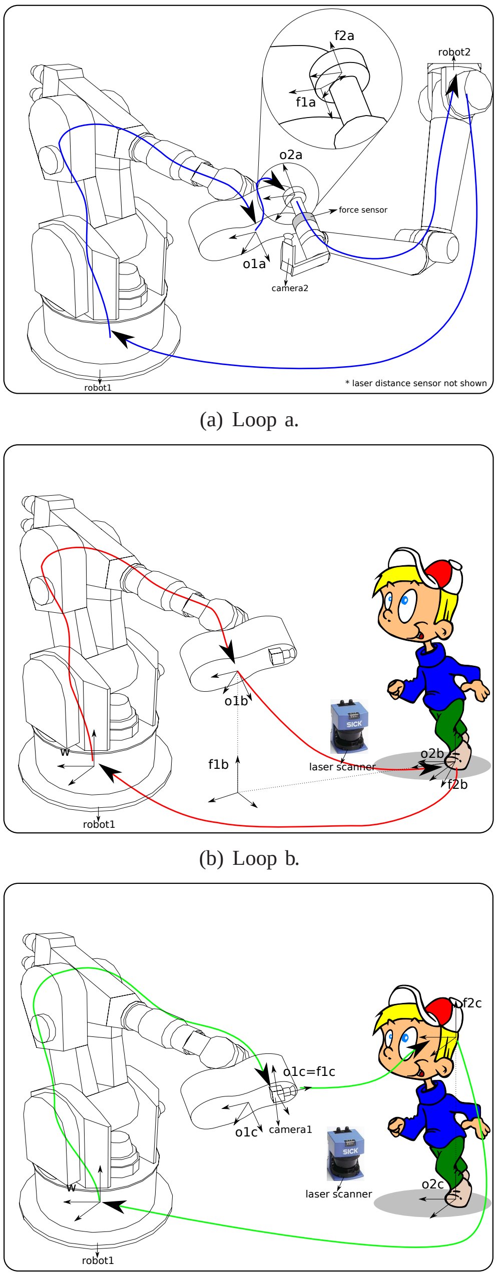
III. CONTROL AND ESTIMATION SCHEME

Fig. 1 shows the general control and estimation scheme presented in [1] and used throughout this paper. This scheme includes the Plant *P*, the Controller *C*, and the Model Update and Estimation block *M*+*E*. The Plant *P* represents both the robot system (where $\boldsymbol{q}$ reprents the *internal state*) and the environment.

The *control input* to the plant is $\boldsymbol{u}$, in the case of a velocity-based control scheme, this input corresponds to the set of desired joint velocities. The *system output* is $\boldsymbol{y}$, which represents the controlled variables. Task specification consists of imposing constraints to the system output $\boldsymbol{y}$. These constraints take the form of desired values $\boldsymbol{y}_d(t)^2$. The plant is observed through *measurements* $\boldsymbol{z}$. Not all system outputs are directly measured, and an estimator is needed to generate estimates $\widehat{\boldsymbol{y}}$. These estimates are needed in the control law *C*.

In general the plant is disturbed by various disturbance inputs. Here we focus on *geometric disturbances*, represented by coordinates $\boldsymbol{\chi_u}$. These coordinates represent modeling errors, uncontrolled degrees of freedom in the robot system or geometric disturbances in the robot environment. As with the system outputs, not all these disturbances can be measured directly, but they can be estimated by including a disturbance observer in the estimator block $M + E$.

[1] If the inequality constraint is not active.



(a) Loop a.

(b) Loop b.

(c) Loop c.

Fig. 2. Loops.

IV. Task Modeling

A typical robot task accomplishes relative motion between objects. The relative motion between two objects is *specified by imposing constraints* on the relative motion between one feature on the first object and a corresponding feature on the second object. Each such constraint needs four frames: *two object frames* (called $o1$ and $o2$, each attached to one of the objects), and *two feature frames* (called $f1$ and $f2$, each attached to one of the corresponding features of the objects). For an application in $3D$ space, there are in general six degrees of freedom between $o1$ and $o2$. The connection $o1 \rightarrow f1 \rightarrow f2 \rightarrow o2$ forms a *kinematic chain*, that is, the degrees of freedom between $o1$ and $o2$ are distributed over three submotions: the relative motion of $f1$ with respect to $o1$ (I), the relative motion of $f2$ with respect to $f1$ (II), and the relative motion of $o2$ with respect to $f2$ (III). The three submotions are modeled using *feature coordinates* χ_{fI}, χ_{fII} and χ_{fIII}, respectively, with

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

$$\begin{tikzcd}
 & o1' \arrow[r, "\chi_{uI}"] & o1 \arrow[r, "\chi_{fI}"] & f1' \arrow[r, "\chi_{uII}"] & f1 \arrow[dd, "\chi_{fII}"] \\
w \arrow[ur, "q"] & & & & \\
 & o2' \arrow[ul, "q"] & o2 \arrow[l, "\chi_{uIV}"] & f2' \arrow[l, "\chi_{fIII}"] & f2 \arrow[l, "\chi_{uIII}"]
\end{tikzcd}$$

Fig. 3. Feature and uncertainty coordinates. The primed frames represent the modelled frame poses while the others are the actual ones.

We consider two kinds of geometric uncertainty: *(1)* uncertainty in the pose of an object with respect to the world, and *(2)* uncertainty in the pose of a feature with respect to its corresponding object. *Uncertainty coordinates* χ_u are introduced to represent the pose uncertainty of a real frame with respect to a modeled frame:

$$\chi_u = \begin{pmatrix} \chi_{uI}{}^T & \chi_{uII}{}^T & \chi_{uIII}{}^T & \chi_{uIV}{}^T \end{pmatrix}^T , \qquad (2)$$

in which:

- χ_{uI} represents the pose uncertainty of $o1$,
- χ_{uII} represents the pose uncertainty of $f1$ with respect to $o1$,
- χ_{uIII} represents the pose uncertainty of $f2$ with respect to $o2$, and
- χ_{uIV} represents the pose uncertainty of $o2$.

Fig. 3 summarizes the definitions of the object and feature frames, and of the feature and uncertainty coordinates. In this figure w represents the world frame.

For the example task four kinematic loops as defined in Fig. 3 are recognized, one for following the contour of the unknown object (a), one for keeping a minimum distance to the closest person (b), one for pointing the camera to the head of the closest person (c), and one for measuring the distance of the end effector of robot 2 to the front plane of the workpiece (d). Each of these loops is further detailed below.

A. Contour following

The contour following task was described in [1] for the two-dimensional case. This paper extends the contour following to the three-dimensional context of the example task. The object and feature frames are defined as follows (Fig. 2(a)):

- frame $o1^a$ is fixed to the workpiece held by robot 1, with the z-axis perpendicular to the front plane of the workpiece;

[2] Further on we omit time dependency in the notation.

- frame $o2^a$ is fixed to the probe held by the robot 2, with its z-axis along the probe's symmetry axis;
- frame $f1^a$ is located at the current contact point between the contour and the probe. The frame's z-axis is parallel to the z-axis of $o1^a$, its x-axis is parallel to the tangent of the contour.
- frame $f2^a$ has the same position and orientation as $o2^a$.

In the case of a *known* contour a minimal set of feature position coordinates exists representing the six degrees of freedom between $o1^a$ and $o2^a$:

$$\begin{aligned} \boldsymbol{\chi_{fI}}^a &= \begin{pmatrix} z^a & s^a \end{pmatrix}, & (3)\\ \boldsymbol{\chi_{fII}}^a &= \begin{pmatrix} y^a & \theta^a & \phi^a & \psi^a \end{pmatrix}^T, & (4)\\ \boldsymbol{\chi_{fIII}}^a &= \begin{pmatrix} - \end{pmatrix}, & (5) \end{aligned}$$

where z^a is expressed in $f1^a$ and represents the distance from the contact point to the front plane of the workpiece, s^a is the arc length along the contour, y^a is expressed in $f1^a$ and represents the distance of the robot end effector to the contour perpendicular to the contour, and θ^a, ϕ^a and ψ^a are ZXY-Euler angles expressed in $f1^a$ and represent the orientation of the probe with respect to the contour. Using s^a and the planar contour model, the pose of frame $f1^a$ with respect to $o1^a$ in a plane parallel to the front plane of the workpiece can be determined.

In the case of an *unknown* contour no set of minimal position coordinates exists to model the relative position of $f1^a$ with respect to $o1^a$. A minimal set of coordinates however exists at velocity level: $\dot{\boldsymbol{\chi_{fI}}}^a = \begin{pmatrix} \dot{z}^a & \dot{s}^a \end{pmatrix}$. Instead of integrating $\dot{s}^a$, the homogeneous transformation matrix between $f1^a$ and $o1^a$ has to be updated at each time step using $\dot{\boldsymbol{\chi_{fI}}}^a$.

Since the real contour is not known, the modeled contour frame $f1^a$ may deviate from the real contour frame. Therefore, uncertainty coordinates are introduced:

$$\boldsymbol{\chi_{uII}}^a = \begin{pmatrix} y_u^a & \theta_u^a \end{pmatrix}^T, \qquad (6)$$

with y_u^a the distance between the modeled and the real contour, and θ_u the orientation error between the tangents of the modeled and the real contour.

For this subtask constraints are specified on the following outputs, and are easily expressed using the feature coordinates:

$$\begin{aligned} &y_1 = z^a, \quad \dot{y_2} = \dot{s}^a, \quad y_3 = y^a - R,\\ &y_4 = \theta^a, \quad y_5 = \phi^a, \text{ and } \quad y_6 = \psi^a, \end{aligned} \qquad (7)$$

where R is the radius of the probe following the contour.

The measurement equations for the magnitude of the contact force and the orientation of the tangent to the contour are easily expressed using the feature coordinates:

$$z_1 = K(y^a - R), \quad z_2 = \theta^a, \text{ and } \quad z_3 = \theta^a, \qquad (8)$$

where K represents the contact stiffness, z_2 represents the orientation of the tangent obtained from the force measurement, while z_3 represents the orientation of the tangent obtained using the image of camera 2.

B. Minimum distance to closest person

For the minimum distance loop (Fig. 2(b)):

- frame $o1^b = o1^a$;
- frame $o2^b$ is at the position of the closest person and with the same orientation as w;
- frame $f1^b$ is located on the ground, just below $o1^b$ and with its z-axis perpendicular to the floor;
- frame $f2^b$ has the same position as $o2^b$, its z-axis is perpendicular to the floor and its x-axis pointing towards the origin of $f2^b$.

$$\begin{aligned} \boldsymbol{\chi_{fI}}^b &= \begin{pmatrix} \alpha^b & \beta^b & \gamma^b & z^b \end{pmatrix}, & (9)\\ \boldsymbol{\chi_{fII}}^b &= \begin{pmatrix} x^b \end{pmatrix}^T, & (10)\\ \boldsymbol{\chi_{fIII}}^b &= \begin{pmatrix} \theta^b \end{pmatrix}, & (11) \end{aligned}$$

where α^b, β^b and γ^b are the ZXZ-Euler angles expressed in $f1^b$ that describe the orientation of $o1^b$ with respect to $f1^b$. z^b, expressed in $f1^b$, represents the height of the end effector with respect to the ground. x^b, expressed in $f2^b$, represents the distance of the closest person to the robot end effector. θ^b, expressed in $o2^b$, represents the direction in which the closest person is located.

The position of the closest person is unknown. Therefore, uncertainty coordinates are introduced:

$$\boldsymbol{\chi_{uIV}}^{b,c} = \begin{pmatrix} x_u^{b,c} & y_u^{b,c} \end{pmatrix}^T, \qquad (12)$$

with $x^{b,c}$ and $y^{b,c}$ the $x-$ and $y-$position of the closest person in the world reference frame.

An inequality constraint is specified on:

$$y_7 = x^b. \qquad (13)$$

The measurement equations for the position of the closest person as determined by the laser scanner based people tracker are easily specified using the uncertainty coordinates:

$$z_4 = x_u^{b,c} \quad \text{and} \quad z_5 = y_u^{b,c}. \qquad (14)$$

C. Camera pointing to closest person

For the camera pointing loop (Fig. 2(c)):

- frame $o1^c = o1^a$;
- frame $o2^c$ is at the position of the closest person and with the same orientation as w,
- frame $f1^c$ is fixed to camera 1 held by the robot 1 and with the z-axis along the camera's principal ray (the line through the principal point and the origin of the pinhole model);
- frame $f2^c$ is located at the closest person's head and is rotated $\frac{\pi}{2}$ around x with respect to $o2^c$ (to avoid singularities).

The feature coordinates expressing the submotions are:

$$\begin{aligned} \boldsymbol{\chi_{fI}}^c &= \begin{pmatrix} - \end{pmatrix}, & (15)\\ \boldsymbol{\chi_{fII}}^c &= \begin{pmatrix} x^c & y^c & z^c & \phi^c & \theta^c & \psi^c \end{pmatrix}^T, & (16)\\ \boldsymbol{\chi_{fIII}}^c &= \begin{pmatrix} - \end{pmatrix}, & (17) \end{aligned}$$

where x^c, y^c and z^c are expressed in $f1^c$ and x^c and y^c represent the position of the closest person's head in the

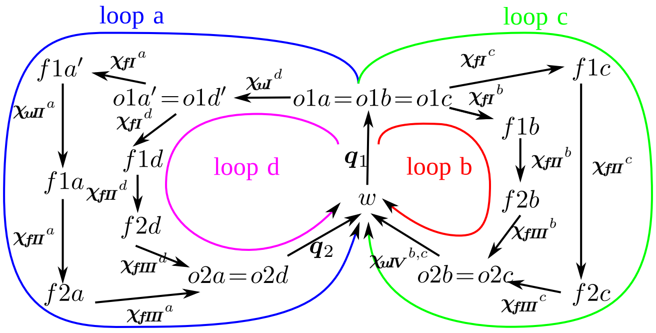


Fig. 4. Object and feature frames and feature coordinates.

camera image and z^c the distance of the person's head to the camera. ϕ^c, θ^c and ψ^c are XYZ-Euler angles expressed in $f1^c$ and represent the pan and tilt of the camera and the orientation of the closest person's head in the camera image respectively.

The position of the closest person is modeled as in Section IV-B, while constraints are specified on:

$$y_8 = x^c, \quad y_9 = y^c, \text{ and } \quad y_{10} = \psi^c. \tag{18}$$

D. Laser distance sensor

This kinematic loop is modeled in detail in [1], and is only briefly described here. Three uncertainty coordinates are introduced corresponding to the errors on the height and the orientation of the front plane of the workpiece with respect to their modeled values: $\boldsymbol{\chi_{uI}}{}^{d}$. The laser distance measurement is denoted by z_6. If the laser distance measurement is not available, the uncertainty coordinates are set equal to zero.

E. Nominal working position

As secondary constraints the robot has to keep the robot as close as possible to its nominal working position, see Section II-B. To this end, constraints can be set to the robot joint positions:

$$y_{si} = q_i, \tag{19}$$

where s indicates the secondary nature of the constraints and with $i = 1, \ldots, 12$ and q_i the i'th joint of the robot system.

F. Overview

Fig. 4 provides an overview of the four kinematic loops showing the object and feature frames, the feature coordinates and the uncertainty coordinates.

All feature coordinates are collected into a single vector $\boldsymbol{\chi_f}$, all uncertainty coordinates are collected into a vector $\boldsymbol{\chi_u}$, while all measurements are collected into a vector $\boldsymbol{z}$.

V. CONTROL BLOCK

The equations for the control block follow automatically from the task model of Section IV. While [1], [12], [13] show the derivation of a velocity based control scheme, other control schemes are discussed in detail in [14].

Since we use a velocity based control scheme, the control input for the robot corresponds to:

$$\dot{\boldsymbol{q}}_d = \boldsymbol{A}^{\#}_{\boldsymbol{W}} \left(\dot{\boldsymbol{y}}^{\circ}_d + \boldsymbol{B} \widehat{\dot{\boldsymbol{\chi}}}_{\boldsymbol{u}} \right), \tag{20}$$

where $\#_{\boldsymbol{W}}$ denotes the weighted pseudoinverse [**?**], [15] with weighting matrix $\boldsymbol{W}$ and

$$\dot{\boldsymbol{y}}^{\circ}_d = \dot{\boldsymbol{y}}_d + \boldsymbol{K}_p \left(\boldsymbol{y}_d - \boldsymbol{y} \right). \tag{21}$$

The first term of the desired joint velocities (20) controls the system outputs to their desired values, while the second term corresponds to a feedforward term accounting for the rate of change of the (estimated) uncertainty coordinates. The first term of control equation (21) corresponds to a feedforward of the time derivative of the desired constraint values (7,13,18), while the second term is a feedback term to compensate for drift, modeling errors and disturbances[3].

The expressions for $\boldsymbol{A}$ and $\boldsymbol{B}$ are given in [1]. We use the Orocos Kinematics and Dynamics Library (KDL, `http://www.orocos.org/kdl`) to derive the jacobians necessary to build $\boldsymbol{A}$ and $\boldsymbol{B}$ based on the kinematic loops defined in Section IV.

Secondary constraints are only realized to the extent that they do not conflict with the primary constraints [16]. The inclusion of secondary constraints (19) modifies the control input (20) to:

$$\begin{aligned}\dot{\boldsymbol{q}}_d \;=\; & \boldsymbol{A_p}^{\#}_{\boldsymbol{W}_p} \left(\dot{\boldsymbol{y}}^{\circ}_{d,p} + \boldsymbol{B}_p \widehat{\dot{\boldsymbol{\chi}}}_{\boldsymbol{u}} \right) + \\ & \left(\boldsymbol{1} - \boldsymbol{A_p}^{\#}_{\boldsymbol{W}_p} \boldsymbol{A}_p \right) \left(\boldsymbol{A_s} \left(\boldsymbol{1} - \boldsymbol{A_p}^{\#}_{\boldsymbol{W_p}} \boldsymbol{A_p} \right) \right)^{\#}_{\boldsymbol{W}_s} \\ & \left(\dot{\boldsymbol{y}}^{\circ}_{d,s} + \boldsymbol{B}_s \widehat{\dot{\boldsymbol{\chi}}}_{\boldsymbol{u}} - \boldsymbol{A}_s \boldsymbol{A_p}^{\#}_{\boldsymbol{W}_p} \left(\dot{\boldsymbol{y}}^{\circ}_{d,p} + \boldsymbol{B}_p \widehat{\dot{\boldsymbol{\chi}}}_{\boldsymbol{u}} \right) \right), \end{aligned} \tag{22}$$

where the subscripts p and s denote primary and secondary, respectively. $\boldsymbol{A}_p$, $\boldsymbol{B}_p$ and $\dot{\boldsymbol{y}}^{\circ}_{d,p}$ are constructed using the primary constraints (7), (13) and (18), while $\boldsymbol{A}_s$, $\boldsymbol{B}_s$ and $\dot{\boldsymbol{y}}^{\circ}_{d,s}$ are constructed using the secondary constraints (19).

VI. MODEL UPDATE AND ESTIMATION BLOCK

The goal of model update and estimation is threefold: *(1)* to provide an estimate for the system outputs $\boldsymbol{y}$ to be used in the feedback terms of constraint equations (21), *(2)* to provide an estimate for the uncertainty coordinates $\boldsymbol{\chi_u}$ and their derivatives, to be used in the control input (20), and *(3)* to maintain the consistency between the joint and feature coordinates $\boldsymbol{q}$ and $\boldsymbol{\chi_f}$ based on the loop constraints.

Model update and estimation makes use of a prediction/correction procedure [1] and is based on an extended system model and on the measurement equations. The extended system model for this example is constructed according to [1], while the measurement equations are given by (8), (14) and the measurement equation for z_6.

The extended system model contains a motion model for the uncertainty coordinates. For example, if a constant velocity model is used for the uncertainty coordinates, the motion model is expressed as: $\dot{\boldsymbol{\chi}}_{\boldsymbol{u}} = C^{te}$ or $\frac{d}{dt}\dot{\boldsymbol{\chi}}_{\boldsymbol{u}} = \boldsymbol{0}$.

The uncertainty coordinates $\boldsymbol{\chi_u}$, $\dot{\boldsymbol{\chi}}_{\boldsymbol{u}}$, ...can also be estimated by a dedicated, external estimator. In the example task, both the position of the closest person and the vision-based estimation of the contour tangent and curvature are provided by external estimators. The next two subsections

[3] There is no feedback term for $y_2 = \dot{s}^a$, only a feedforward term.

outline the operation of the people tracker and the vision-based estimator.

A. People tracker

Estimating the position of multiple moving persons is significantly harder than estimating the position of a single person. First, one has to determine the number of persons that are currently in the field of view. Furthermore, the update process is harder, since observations may result in ambiguities, features may not be distinguishable, objects may be occluded, or there may be more features than persons. Hence, a system for tracking multiple moving persons must be able to estimate the number of persons and must be able to assign the observed features to the persons being tracked.

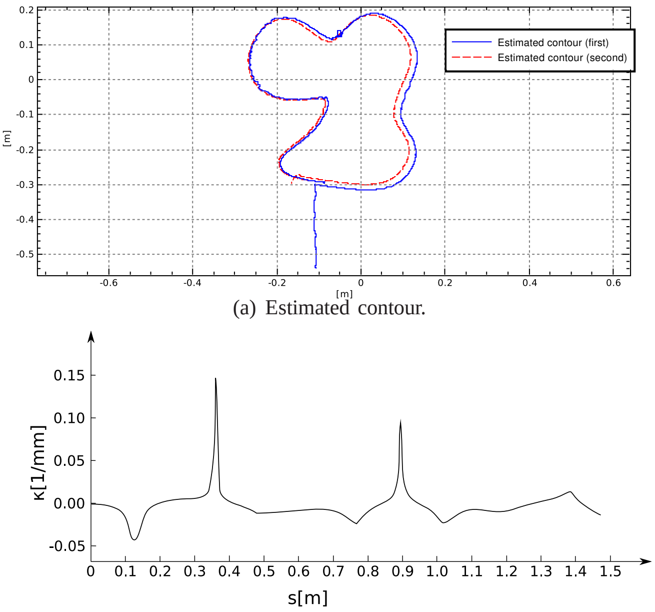
Fig. 5 presents the process flow of the people tracker. A laser scanner measures the range of the objects over $180°$. From the measured distances, the measurement resulting from the environment is selected if the probability that the measurement does result from the environment is higher than a threshold: $P\,(z \notin \text{environment}\,|\geq)\,p_{ud}$, with p_{ud} a user-defined threshold. From the selected measurements, low level features are extracted using a Variational Bayesian cluster finding algorithm (VBC) [17]. The VBC provides automatic relevance detection, that is, it automatically selects the most probable number of clusters.

To keep track of multiple moving persons requires estimation of the joint probability distribution of the state of all persons. In practice, however, this is already intractable for a small number of persons, since the size of the state space grows exponentially with the number of persons. To overcome this problem, a common approach is to track the different persons independently, using factorial representations for the individual states. A general problem in this context is to determine which measurement is caused by which person. In this paper we apply a Joint Probabilistic Data Association Filter (JPDAF) [18] and/or a Sequential Joint Probabilistic Data Association Filter (SJPDAF) for this purpose [19]. The data association algorithm computes a Bayesian estimate of the correspondence between the low level features and the different persons to be tracked. Using these correspondences the individual filter for each person is updated. The JPDAF uses a Kalman filter to track the individual persons while SJPDAF uses a particle filter. To improve the tracking of individual persons, even in case of occlusions, a motion model is used. In this paper a constant velocity model was incorporated in the estimation.

The data association filters assume that the number of persons to be tracked is known. In our application, however, the number of objects often varies over time. As suggested by [19] this is handled by an extra discrete estimator estimating the number of persons from the low level features.

B. Vision-based estimator

The visual contour estimator estimates the tangent and curvature of the unknown contour at the current contact point using camera 2 mounted on robot 2 (Fig. 2(a)). At the beginning of the contour following task, once contact is established between contour and probe, an initialization is carried out in which the user indicates the foreground, that is, a point on the object. Next, information on properties of foreground and background are gathered using cue integration through voting for defocus, hue and saturation cues. Then, based on the integrated cues a floodfill segmentation is carried out using an automatic optimal threshold through a quality number. This quality number is obtained through a voting procedure in which the cues are: the total number of segmented pixels (the more the better), the increase in number of segmented pixels between two thresholds (if the floodfill segmentation overflows in the next valley, the previous threshold has reached an intensity ridge), the average pixel value of the image obtained through cue integration, with the segmented pixels as a mask and an indicator to check whether the segmentation reaches the edges of the camera image. The image segmented with this optimal threshold is smoothed to find a two dimensional edge and subsequentially, a natural cubic spline is fitted to the obtained edge.



(a) Estimated contour.

(b) Curvature profile computed from a CAD-model of the contour.

Fig. 6. Contour and curvature profile of contour of workpiece

For efficiency reasons, the initialization is not repeated on line: during the motion, the spline knots are corrected using line segments perpendicular to the spline [20]. Within each of these 1D search spaces, another voting procedure (cues: a minimal average correct hue/saturation, a maximal average difference in correct hue/saturation, an ISEF edge detector [21], and a weight on the distance to the previous knot position.) determines the optimal new knot position. A base for this procedure is the cue integration of hue and saturation along these line segments.

VII. EXPERIMENTAL RESULTS

In our experimental setup robot 1, holding the workpiece, is a Kuka K160 robot while a Kuka K361 robot is holding

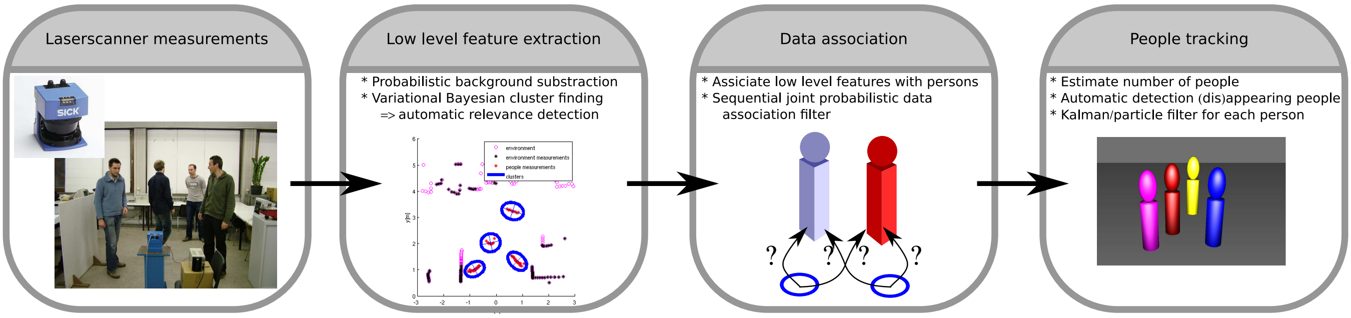
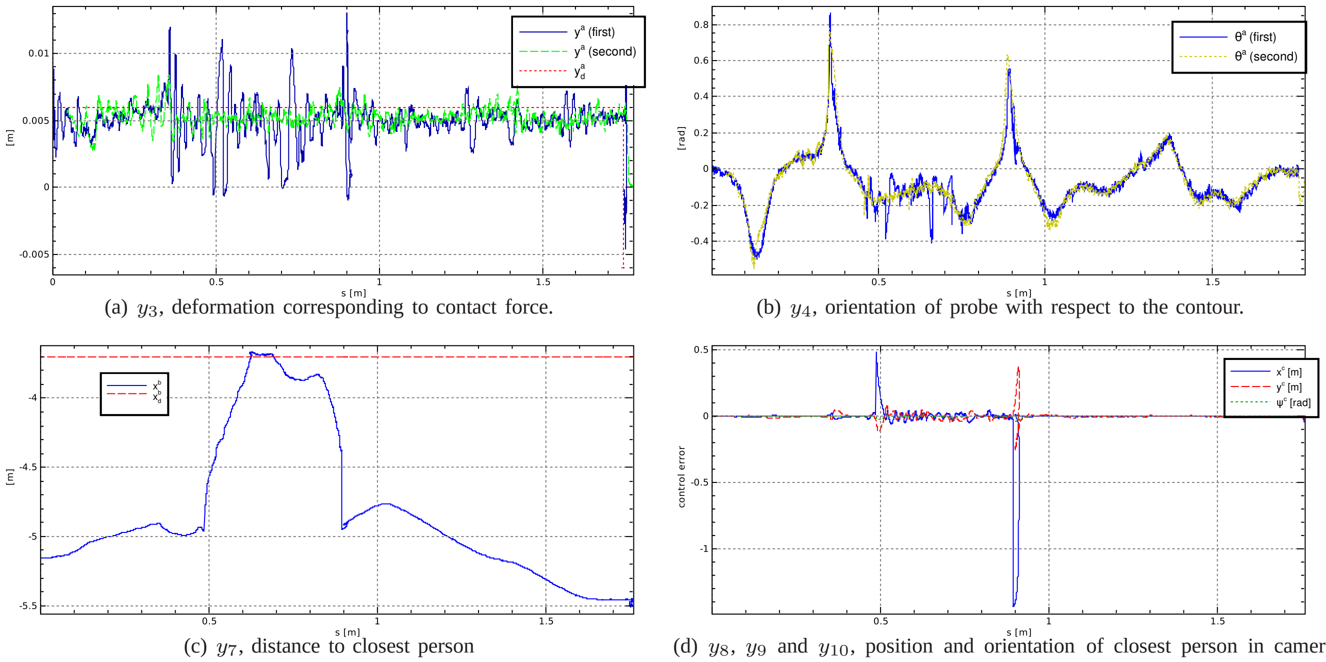

Laserscanner measurements
SICK
Low level feature extraction
* Probabilistic background substraction
* Variational Bayesian cluster finding
=> automatic relevance detection
environment
environment measurements
people measurements
clusters
Data association
* Assiciate low level features with persons
* Sequential joint probabilistic data association filter
?
?
?
?
People tracking
* Estimate number of people
* Automatic detection (dis)appearing people
* Kalman/particle filter for each person

Fig. 5. Scheme for people tracker.



(a) y_3, deformation corresponding to contact force.

(b) y_4, orientation of probe with respect to the contour.

(c) y_7, distance to closest person

(d) y_8, y_9 and y_{10}, position and orientation of closest person in camera image.

Fig. 8. Experimental results for outputs y_3-y_{10} for both first and second experiment together with desired values.

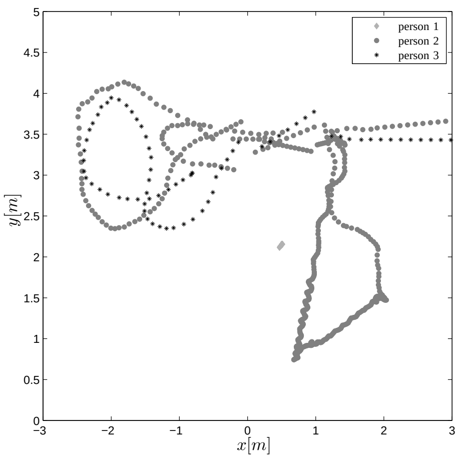


Fig. 7. Estimated motion of persons.

the probe. The force sensor is a JR3 100M40A3-I63-DH. A Sick laserscanner (range $8m$) is used to track the persons in the neighborhood of the robot.

We carried out two different experiments. During the *first experiment* different persons, one standing still and two walking, are present in the robot environment while all constraints are switched on. In the *second experiment*, only the contour following constraints (7) are activated, by putting the weights of the constraints for the minimum distance to the closest person (13) and the camera tracking of the closest person (18) to zero.

A constant velocity along the contour of $0.01\frac{m}{s}$ ($\dot{y}_2$) was applied in both experiments, while a contact force of $30N$ was desired, resulting in a desired y_3 of $0.006m$. The desired angles between contour and follower were set to $0°$ ($y_4 = y_5 = y_6 = 0°$). A minimum distance of $3.7m$ to the closest person was commanded (y_7), while the closest person was kept straight in the middle of the image ($y_8 = y_9 = 0m$, $y_{10} = 0°$).

The contour of the $2.5D$ workpiece estimated during the two experiments is shown in Fig. 6(a). The difference between the estimated contour in the two experiments is due to the imperfect kinematics and relative position of the two robots. Fig. 7 shows the estimated position of the persons resulting from the people tracker (Section VI-A). Fig. 8 shows some of the realized outputs in the two experiments. The difference between the desired and realized outputs is smaller for the second experiment, as expected. Fig. 8(c) shows that the inequality constraint on the distance of the closest person is only active when $y_7 < 3.7m$. As expected, the orientation error of the probe with respect to the contour in the plane of the contour, y_4, (Fig. 8(b)), is proportional to the curvature of the contour (Fig. 6(b)). This suggests that the tracking would benefit from feedforward of the curvature information, which can be obtained from the vision-based estimator (Section VI-B).

The closest person switches from one to the other when $s = 0.5m$ and $s = 0.9m$. This causes a sudden difference in the distance of the closest person (y_7) (Fig. 8(c)) and of position of the closest person in the image (y_8 and y_9) for the first experiment (Fig. 8(d)). This sudden error is also reflected in y_3 and y_4 (Fig. 8(a), Fig. 8(b)).

Remark that the realized outputs are influenced by the imperfectly known kinematics, dynamics and relative position of the two robots, as well as the relatively stiff contact (we did not add any flexibility between contour and probe!).

Movies of different experiments are available at `http://people.mech.kuleuven.be/~orocos/`.

VIII. CONCLUSIONS AND FUTURE WORK

Using an example task, this paper showed that iTASC is a powerful tool for multi-sensor integration in robot manipulation. The example task includes multiple sensors: encoders, a force sensor, cameras, a laser distance sensor and a laser scanner. The paper detailed the systematic modeling procedure, derived control and estimation equations for the task involving 10 primary constraints, 7 uncertainty coordinates, 6 scalar measurements and 12 secondary constraints, presented a laser scanner based people tracker and presented experimental results for the example task.

Future work includes the development of a user friendly interface to support the task specification to enable high-level task programming. We will also perform experiments with an acceleration-based control scheme including dynamic models of the robots, and compare the results with the velocity-based scheme.

REFERENCES

[1] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *Int. J. Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007.

[2] J. De Schutter and H. Van Brussel, "Compliant Motion I, II," *Int. J. Robotics Research*, vol. 7, no. 4, pp. 3–33, Aug 1988.

[3] N. Hogan, "Impedance control: An approach to manipulation. Parts I-III," *Trans. ASME J. Dyn. Systems Meas. Control*, vol. 107, pp. 1–24, 1985.

[4] ——, "Stable execution of contact tasks using impedance control," in *Int. Conf. Robotics and Automation*, Raleigh, NC, 1987, pp. 1047–1054.

[5] H. Kazerooni, "On the robot compliant motion control," *Trans. ASME J. Dyn. Systems Meas. Control*, vol. 111, pp. 416–425, 1989.

[6] J. Baeten, H. Bruyninckx, and J. De Schutter, "Integrated vision/force robotics servoing in the task frame formalism," *Int. J. Robotics Research*, vol. 22, no. 10, pp. 941–954, 2003.

[7] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-11, no. 6, pp. 418–432, 1981.

[8] H. Bruyninckx and J. De Schutter, "Specification of force-controlled actions in the "Task Frame Formalism": A survey," *IEEE Trans. Rob. Automation*, vol. 12, no. 5, pp. 581–589, 1996.

[9] A. P. Ambler and R. J. Popplestone, "Inferring the positions of bodies from specified spatial relationships," *Artificial Intelligence*, vol. 6, pp. 157–174, 1975.

[10] C. Samson, M. Le Borgne, and B. Espiau, *Robot Control, the Task Function Approach*. Oxford, England: Clarendon Press, 1991.

[11] J. De Schutter, J. Rutgeerts, E. Aertbelien, F. De Groote, T. De Laet, T. Lefebvre, W. Verdonck, and H. Bruyninckx, "Unified constraint-based task specification for complex sensor-based robot systems," in *Int. Conf. Robotics and Automation*, Barcelona, Spain, 2005, pp. 3618–3623.

[12] T. De Laet, W. Decré, J. Rutgeerts, H. Bruyninckx, and J. De Schutter, "An application of constraint-based task specification and estimation for sensor-based robot systems," in *Proc. IEEE/RSJ Int. Conf. Int. Robots and Systems*, San Diego, California, 2007, pp. 1658–1664.

[13] W. Decré, T. De Laet, J. Rutgeerts, H. Bruyninckx, and J. De Schutter, "Application of a generic constraint-based programming approach to an industrial relevant robot task with uncertain geometry," in *IEEE Int. Conf. Comp. as a Tool*, Warsaw, Poland, September 2007, pp. 2620–2626.

[14] T. De Laet and J. De Schutter, "Control schemes for constraint-based task specification in the presence of geometric uncertainty using auxiliary coordinates," Dept. Mech. Eng., Katholieke Univ. Leuven, Belgium, Internal report 07RP001, 2007.

[15] K. L. Doty, C. Melchiorri, and C. Bonivento, "A theory of generalized inverses applied to robotics," *Int. J. Robotics Research*, vol. 12, no. 1, pp. 1–19, 1993.

[16] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, reprinted ed. Huntington, NY: Robert E. Krieger Publishing Company, 1980.

[17] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[18] I. J. Cox, "A review of statistical data association techniques for motion correspondence," *Int. J. of Computer Vision*, vol. 10, no. 1, pp. 53–667, 1993.

[19] D. Schulz, W. Burgard, and D. Fox, "People tracking with mobile robots using sample-based joint probabilistic data association filters," *Int. J. Robotics Research*, vol. 22, no. 2, pp. 99–116, 2003.

[20] M. Isard and A. Blake, "CONDENSATION—conditional density propagation for visual tracking," *Int. J. Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[21] J. Shen and S. Castan, "An optimal linear operator for step edge detection," *Computer vision, graphics, and image processing: graphical models and understanding*, vol. 54, no.2, pp. 112–133, 1992.