

Application of a Generic Constraint-Based Programming Approach to an Industrially Relevant Robot Task with Geometric Uncertainties

Abstract—This paper shows the application of a generic constraint-based task specification approach for sensor-based robot systems to a laser tracing example. Key properties of the used approach are (i) its ability to specify complex robot tasks by introducing auxiliary task-oriented feature coordinates, defined with respect to user-defined object and feature frames, (ii) its support for both underconstrained and overconstrained robot tasks, and (iii) its ability to integrate sensor measurements in a unified way, using auxiliary uncertainty coordinates, to estimate geometric uncertainties in the robot system or its environment. Simulation and real world experimental results are presented.

Index Terms—constraint-based programming, task specification, estimation, geometric uncertainty, laser tracing

I. INTRODUCTION

INDUSTRIAL robot control software contains adequate motion primitives to specify robot tasks of limited complexity, such as pick and place operations, in well known structured environments. However, specifying (geometrically) more complex tasks using off the shelf robot control software can quickly become an inefficient, cumbersome, time consuming and therefore costly operation. Additionally, no generic solutions exist for the integration of sensor information in the robot control software to estimate uncertain geometric parameters. Incorporating the estimation of geometric uncertainties in the robot software can however substantially increase the flexibility of the robot system and improve the execution quality of its task. Especially applications where structuring the environment is impossible or demands prohibitively high costs can benefit from the estimation of geometric uncertainties. This type of applications arises in for instance domestic environments or in small series industrial production facilities.

The goal of our research is to develop programming support so that robot systems can more fully realize their potential. The backbone of this programming support is a novel generic and systematic approach to specify and control a task while dealing properly with geometric uncertainty.

Our preliminary work on a task specification framework was presented in [1], while the mature framework is thoroughly discussed in [2]. The contribution of this paper lies at (i) the application of this approach to a laser tracing task, and (ii) the implementation of this application both in a simulation environment and on a real robot, hence substantiating the framework's power and practical advantages.

Previous work on specification of sensor-based robot tasks, such as force controlled manipulation [3]–[6] or force controlled compliant motion combined with visual servoing [7], was based on the concept of the *compliance frame* [8] or *task frame* [9]. The drawback of the task frame approach is that it

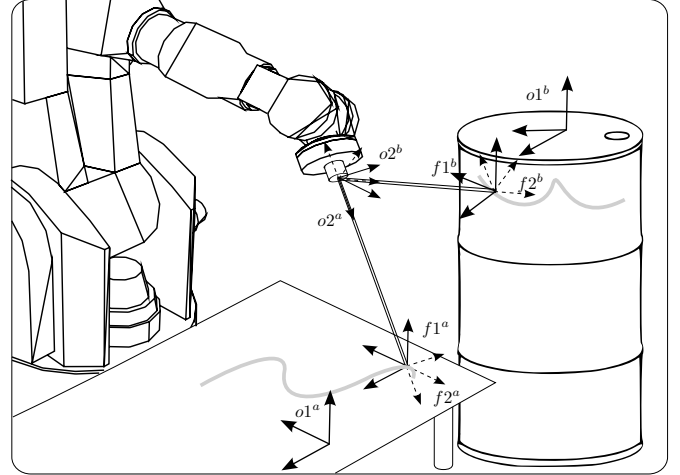


Fig. 1. The object and feature frames for simultaneous laser tracing on a plane and a barrel.

only works well for task geometries with limited complexity, that is, task geometries for which separate control modes can be assigned independently to three pure translational and three pure rotational directions along the axes of one *single* frame.

A more general 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*, lies at the foundation of the developed framework.

Seminal theoretical work on constraint-based programming of robot tasks was done by Ambler and Popplestone [10] and by Samson and coworkers [11]. Also motion planning research on configuration space methods (see [12] for an overview) specifies the desired relative poses as the result of applying several (possibly conflicting) constraints between object features.

This paper is structured as follows. Section II introduces the example application of laser tracing. Section III defines the auxiliary feature and uncertainty coordinates that are used to model task constraints and geometric uncertainty. Subsequently, Section IV details a velocity based control scheme which uses these auxiliary coordinates. Section V explains the procedure used for updating the models and estimating the geometric uncertainties in the system. Simulation as well as experimental results are presented in Section VI. Finally, Section VII discusses the proposed approach and summarizes the main conclusions.

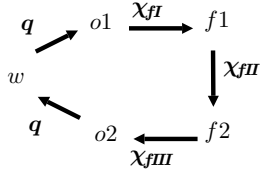


Fig. 2. Object and feature frames and feature coordinates.

II. APPLICATION

The robot task consists of simultaneously tracing a path on a plane and on a cylindrical barrel with radius R using two lasers which are rigidly attached to the end effector of a robot with six degrees of freedom. This is illustrated in Figure 1. As shown later on, the task specification easily allows imposing extra task constraints such as maintaining a fixed distance or orientation of the lasers with respect to their associated surfaces. Additionally, the lasers measure the distances to their associated surfaces. This measurement information is used to estimate geometric uncertainties. In the presented simulations and experimental setup, the position and orientation of the plane are initially unknown, while the barrel's axis is known to be vertical but its exact position is unknown. Furthermore, this paper deals with the *external* calibration of the laser sensors, i.e. ascertaining the lasers' positions and orientations with respect to the robot end effector. Key properties of this task that frequently occur in industrial robot tasks, are (i) the underconstrained specification, which means that the robot is a redundant system for the specified task: in this case the robot has six degrees of freedom, while the task only imposes four motion constraints, two for the plane and two for the barrel, and (ii) the estimation of uncertain geometric parameters. Note that this example application can serve as a model for industrial applications with similar task topology such as robot spray painting, robot laser welding and robot scanning.

III. MODELING

The general framework presents a novel modeling approach that introduces two types of auxiliary task related coordinates additional to the standard robot joint coordinates q :

- *feature coordinates* χ_f , to facilitate the modeling of constraints and measurements by the user,
- *uncertainty coordinates* χ_u , to represent modeling errors, uncontrolled degrees of freedom in the robot system or geometric disturbances in the robot environment.

These two types of coordinates are defined in *object* frames and *feature* frames that are chosen by the task programmer in a way that simplifies the specification of the task at hand.

This section applies this novel modeling approach to the laser tracing example. The modeling of other applications such as visual servoing, cooperating robots, localization and path tracking of a nonholonomic mobile robot, contour tracking and human-robot co-manipulation is presented in [2].

A. Object and feature frames

A typical robot task accomplishes a relative motion between objects¹. The first modeling step is to introduce a set of reference frames in which these relative motions are easily expressed. The first frame is the “world” reference frame, denoted by w . In this application the world frame is placed at the base frame of the robot. The other frames are attached to *objects* and *features* that are relevant for the task at hand:

- an *object* can be any rigid object in the robot system (for example a robot end effector or a robot link) or in the robot environment,
- a *feature* is linked to an *object*, and indicates any *entity* of that object that is relevant for the specification of the robot's task: a *physical entity* (such as a vertex, edge, face, surface), or an *abstract geometric property* of a physical entity (such as the symmetry axis of a cylinder, or the reference frame of a sensor connected to the object, such as a camera).

In the application four objects are relevant: the two lasers, the plane and the barrel. Relevant features are the intersection points between the first laser and the plane and between the second laser and the barrel since these points have to trace a specified path.

For an application in 3D space, there are in general six degrees of freedom between two objects. Given two objects with associated object frames $o1$ and $o2$ and two feature frames $f1$ and $f2$ linked to $o1$ respectively $o2$. 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 (I, II, III) as shown in Figure 2. In the application, two relevant kinematic chains are recognized, one for the laser-plane combination (denoted *a*) and another one for the laser-barrel combination (denoted *b*). The relative motion between the two objects is then *specified by imposing constraints* on one or on a combination of these submotions.

Figure 1 shows the chosen object and feature frames for the laser tracing example. For the laser-plane feature:

- frame $o1^a$ fixed to the plane and with the z -axis perpendicular to the plane,
- frame $o2^a$ fixed to the first laser on the robot end effector and with its z -axis along the laser beam,
- frame $f1^a$ has the same orientation as $o1^a$, but is located at the intersection of the laser with the plane,
- frame $f2^a$ has the same position as $f1^a$, but the same orientation as $o2^a$,

and for the laser-barrel feature:

- frame $o1^b$ fixed to the barrel and with its origin on and the x -axis along the axis of the barrel,
- frame $o2^b$ fixed to the second laser on the robot end effector and with its z -axis along the laser beam,
- frame $f1^b$ located at the intersection of the laser with the barrel, z -axis perpendicular to the barrel surface and x -axis parallel to the barrel axis,

¹In general, the framework also supports controlled dynamic interactions between objects, but the laser tracing example does not involve such interaction.

- frame $f2^b$ with the same position as $f1^b$ and the same orientation as $o2^b$.

B. Feature coordinates

Task related *feature coordinates* χ_f are introduced to facilitate the task specification by the user. These coordinates represent the submotions between $o1$ and $o2$. The choice of feature coordinates is not unique, yet by *wisely* (i.e. task-oriented) choosing the object and feature frames, the mathematical representation of the submotions is simplified, as shown in the next paragraphs.

For the laser-plane combination the feature coordinates expressing the submotions are:

$$\chi_{fI}^a = (x^a \ y^a)^T, \quad (1)$$

$$\chi_{fII}^a = (\phi^a \ \theta^a \ \psi^a)^T, \quad (2)$$

$$\chi_{fIII}^a = (z^a), \quad (3)$$

where x^a and y^a are expressed in $o1^a$ and represent the position of the laser dot on the plane, while z^a is expressed in $o2^a$ and represents the distance of the robot to the plane along the laser beam. ϕ^a, θ^a, ψ^a represent Euler angles between $f1^a$ and $f2^a$ and are expressed in $f2^a$.

For the laser-barrel combination feature coordinates expressing the submotions are:

$$\chi_{fI}^b = (x^b \ \alpha^b)^T, \quad (4)$$

$$\chi_{fII}^b = (\phi^b \ \theta^b \ \psi^b)^T, \quad (5)$$

$$\chi_{fIII}^b = (z^b), \quad (6)$$

where x^b and α^b are cylindrical coordinates expressed in $o1^b$ representing the position of the laser dot on the barrel, while z^b is expressed in $o2^b$ and represents the distance of the robot to the plane along the laser beam. ϕ^b, θ^b, ψ^b represent Euler angles between $f1^b$ and $f2^b$ and are expressed in $f2^b$.

All feature coordinates are grouped into a single vector χ_f :

$$\chi_f = (\chi_{fI}^a \ \chi_{fII}^a \ \chi_{fIII}^a \ \chi_{fI}^b \ \chi_{fII}^b \ \chi_{fIII}^b)^T. \quad (7)$$

C. Uncertainty coordinates

To represent modeling errors, uncontrolled degrees of freedom in the robot system or geometric disturbances in the robot environment, *uncertainty coordinates* χ_u are introduced. In the laser tracing example uncertainty coordinates are needed to represent the unknown position and orientation of the plane, the position of the barrel and, in the case of calibration, the position and orientation of the lasers with respect to the robot end effector.

The unknown position and orientation of the plane is modeled by²:

$$\chi_{uI}^a = (h^a \ \alpha^a \ \beta^a)^T, \quad (8)$$

with h^a the z -coordinate of the intersection point of the plane and the z -axis of w with respect to the world, and α^a and β^a the two Euler angles which determine the orientation of the

plane with respect to the world³. The unknown position of the barrel is modeled by:

$$\chi_{uI}^b = (x_u^b \ y_u^b)^T, \quad (9)$$

with x_u^b and y_u^b the x - and y -position of the barrel with respect to the world. If the laser position and orientation with respect to the end effector is also unknown, for example during the calibration phase, additional uncertainty coordinates are introduced:

$$\chi_{uIV} = (x_l \ y_l \ z_l \ \phi_l \ \theta_l)^T. \quad (10)$$

x_l, y_l and z_l represent the positions of the laser with respect to the robot end effector $o2$. ϕ_l and θ_l denote the two Euler angles which determine the orientation of the laser with respect to the robot end effector $o2$ ⁴.

All uncertainty coordinates are grouped into a single vector χ_u . When calibration has been carried out the uncertainty coordinates reduce to⁵:

$$\chi_u = (\chi_{uI}^a \ \chi_{uI}^b)^T. \quad (11)$$

D. Task specification

Task specification consists of imposing constraints on a user-defined *system output* $y(t)$. For the laser tracing example, the task goal is to generate desired paths on the plane and cylinder. Hence, the system output and the *constraint equation* are given by⁶:

$$y(t) = (x^a \ y^a \ x^b \ \alpha^b)^T \quad \text{and} \quad y(t) = y_d(t), \quad (12)$$

where $y_i(t)$ represents a system output, and $y_{di}(t)$ the imposed constraint.

Finally, the *measurement equations* for the lasers measuring the distance to the plane and the barrel are given by:

$$z(t) = (z^a \ z^b)^T. \quad (13)$$

IV. CONTROL

For the example, we assume a velocity controlled robot and neglect its system dynamics. Hence, the *system equation* is given by:

$$\dot{q} = u = \dot{q}_d, \quad (14)$$

where the control input u corresponds to the desired joint velocities \dot{q}_d .

The *output equation* relates the system state to the outputs y :

$$f(q, \chi_f) = y. \quad (15)$$

The system state, consisting of q and χ_f is nonminimal. The dependency relation between q and χ_f corresponds to the

³Since the plane is considered infinite, the rotation around the normal of the plane is irrelevant.

⁴Since the laser beam is rotation symmetric, the rotation around the beam axis is irrelevant and two Euler angles suffice.

⁵For calibration, (11) is extended with χ_{uIV} .

⁶Imposing additional constraints on the relative distance or orientation between lasers and associated surfaces is done by extending (12) with the corresponding constraint equations.

²The numbering of the uncertainty coordinates follows the notations of the framework [2].

loop closure equations which are perturbed by the uncertainty coordinates χ_u , and is expressed as:

$$l(q, \chi_f, \chi_u) = 0. \quad (16)$$

For the derivation of velocity based control the output and the loop closure equations are differentiated with respect to time to obtain equations at velocity level. The output equation at velocity level is written as:

$$C_q \dot{q} + C_f \dot{\chi}_f = \dot{y}, \quad (17)$$

with $C_q = \frac{\partial f}{\partial q}$ and $C_f = \frac{\partial f}{\partial \chi_f}$. C_q and C_f can be easily found by inspecting (7) and (12), which yields:

$$C_q = 0 \quad \text{and} \quad C_f = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0_{4 \times 4} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (18)$$

On the other hand, the velocity loop constraint becomes:

$$J_q \dot{q} + J_f \dot{\chi}_f + J_u \dot{\chi}_u = 0, \quad (19)$$

with $J_q = \frac{\partial l}{\partial q}$, $J_f = \frac{\partial l}{\partial \chi_f}$ and $J_u = \frac{\partial l}{\partial \chi_u}$. J_q represents the robot jacobian. J_f and J_u can be found by inspection, based on the definitions of the feature and uncertainty coordinates. Solving $\dot{\chi}_f$ from (19) yields:

$$\dot{\chi}_f = -J_f^{-1} (J_q \dot{q} + J_u \dot{\chi}_u). \quad (20)$$

Note that J_f is invertible⁷. Substituting (20) into (17) yields the modified output equation:

$$A \dot{q} = \dot{y} + B \dot{\chi}_u, \quad (21)$$

where $A = C_q - C_f J_f^{-1} J_q$ and $B = C_f J_f^{-1} J_u$ are introduced for simplicity of notation.

In (21), constraint equation (12) is expressed at velocity level. To compensate for drift, modeling errors and disturbances, a position feedback term is added:

$$\dot{y} = \dot{y}_d + K_p (y_d - y) = \dot{y}_d^\circ, \quad (22)$$

with K_p a matrix of feedback constants and \dot{y}_d° the *modified constraint at velocity level*. Since the outputs y cannot be measured directly, they are replaced in (22) by their estimates \hat{y} provided by the estimator (as explained in Section V).

Applying constraint (22) to (21), while also substituting system equation (14) and replacing $\dot{\chi}_u$ by its estimate $\hat{\dot{\chi}}_u$ (since its real value may be unknown during the task execution), results in:

$$A \dot{q}_d = \hat{y}_d^\circ + B \hat{\dot{\chi}}_u. \quad (23)$$

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

$$\dot{q}_d = A_W^\# (\hat{y}_d^\circ + B \hat{\dot{\chi}}_u), \quad (24)$$

where $A_W^\#$ denotes the weighted pseudoinverse [13], [14] with weighting matrix W . Furthermore, since the plane and the barrel are not moving or, more accurately, are modeled to be non-moving, $\hat{\chi}_u = 0$ in this example. Therefore the control input reduces to:

$$\dot{q}_d = A_W^\# \hat{y}_d^\circ. \quad (25)$$

⁷The fact that J_f is invertible implies that the global number of feature coordinates must equal the number of independent loop equations.

Since the task specification is underconstrained, additional constraints can be added without compromising the task execution. In the overconstrained case constraints are weighted with W . Alternatively, redundancy can be used to accomplish *secondary* task objectives (e.g. maintaining a relative orientation or distance between laser and surface). These secondary constraints are solved in the subspace remaining after applying the primary (i.e. the original) constraints, so that primary constraints have absolute priority over secondary constraints if primary and secondary constraints are conflicting.

V. MODEL UPDATE AND ESTIMATION

The goal of the model update and estimation step is three-fold: (i) to provide an estimate for the system outputs y to be used in the feedback terms of constraint equations (22), (ii) to provide an estimate for the uncertainty coordinates χ_u to be used in the loops (16), and (iii) to maintain the consistency between the dependent system states q and χ_f based on the loop constraints. The model update and estimation step is based on an extended system model consisting of (i) the robot system model, (ii) the velocity loop constraints, and (iii) the dynamic model for the uncertainty coordinates. The states of this extended model are q , χ_f , χ_u .

A predictor-corrector procedure is proposed here. The predictor consists of two steps. The first step generates predicted estimates based on the extended system model. The extended system model for the example application is given by⁸:

$$\frac{d}{dt} \begin{pmatrix} q \\ \chi_f \\ \chi_u \end{pmatrix} = 0_{n \times n} \begin{pmatrix} q \\ \chi_f \\ \chi_u \end{pmatrix} + \begin{pmatrix} I_k \\ -\frac{1}{\alpha_m \times k} I_l \end{pmatrix} \dot{q}_d, \quad (26)$$

with $k = 6$ the number of joints, $m = 5$ the number of uncertainty coordinates, $n = k + l + m$ the total number of state variables in the extended model and $l = 12$ the number of feature coordinates.

This model consists of three parts: the first line corresponds to the system model (14), the second line corresponds to the velocity loop constraint (20), and the last line corresponds to the model used for the uncertainty coordinates. Since in this case the uncertainty coordinates which are estimated are constant (plane position and orientation and barrel position) the model for the uncertainty coordinates is very simple:

$$\chi_u = C^{te} \quad \text{or} \quad \frac{d}{dt} \chi_u = 0_{m \times 1}. \quad (27)$$

A second step eliminates any inconsistencies between these predicted state estimates: the dependent variables χ_f are made consistent with the other estimates (q , χ_u) by iteratively solving the position loop constraints (16)⁹. Similarly, the corrector consists of two steps. The first step generates updated estimates based on the predicted estimates and on the information contained in the sensor measurements. Since these measurement equations are expressed in χ_f (13), the position

⁸In general the extended system model is more involved [2].

⁹Since in this step no extra information on the geometric uncertainties is available, and since there is no physical motion of the robot involved, both χ_u and q are kept constant. Adapting χ_f is sufficient to close any opening in the position loops, since χ_f can be solved unambiguously from these position loops.

loop constraints (16) have to be included in this step to provide the relationship with the other coordinates. In the case of the laser tracing example each laser distance measurement fits to one feature coordinate. Therefore the measurement equations contain a simple selection matrix:

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{2 \times 5} & \begin{vmatrix} 1 \\ 0 \end{vmatrix} & \mathbf{0}_{2 \times 5} & \begin{vmatrix} 0 \\ 1 \end{vmatrix} \end{pmatrix} \mathbf{x}_f. \quad (28)$$

Different estimation techniques can be used to obtain optimal estimates of the geometric uncertainties. Extended Kalman filtering is an obvious choice because of its low computational cost and since all equations for extended Kalman filtering are straightforwardly derived using (26) and (28). For other estimators, detailed numerical procedures are described in literature [15]–[19].

Finally, since the corrector procedure may re-introduce inconsistencies between the updated state estimates, the second step of the predictor is repeated, but now applied to the updated state estimates.

After prediction and correction of the states, estimates for the system outputs $\hat{\mathbf{y}}$ follow from (15).

Note that the basic Kalman filter, which is an exact Bayesian filter, is limited to linear system and measurement equations. The extended Kalman filter however is only an *approximate* Bayesian filter since it relaxes this linearity assumption. Hence, for non-linear models, it only exhibits acceptable behavior for *sufficiently small* state errors and uncertainties.

VI. RESULTS

A simulation experiment (Matlab®) is carried out in which desired paths are traced on a plane and on a barrel with radius $R = 0.285m$. The desired paths on the plane and barrel are two different Lissajous curves, both with a period of 8s. Results of this simulation are shown in Figure 3. The initial estimate of the plane differs from the real one by 0.3m for the z -position of the intersection point of the plane and the z -axis of w , 5° for Euler angle α^a and 5° for Euler angle α^b . The initial estimate of the barrel's position differs from the real one by 0.2m in the x -direction and 0.1m in the y -direction. An extended Kalman filter is used for the estimation in which extra process uncertainty is applied to enhance the convergence (by multiplying the covariance with a fading factor 1.1 for the plane and 1.12 for the barrel, as explained in [16]).

Furthermore a real world experiment is carried out in which the position of a barrel with radius $R = 0.285m$ is estimated. To estimate this barrel the laser is moved back and forth along the y -axis of the world at a fixed height as shown in Figure 4a. Figure 4 also shows experimental results for the estimation of the barrel position using a Baumer laser distance sensor (OADM 2016480/S14F) mounted on a Kuka361 industrial robot. Additionally, to compensate for robot system dynamics and numerical inaccuracies, joint angles \mathbf{q} are replaced by encoder values, which avoids integrating $\dot{\mathbf{q}}$. For the same reason, the joint velocities are replaced by a finite difference of encoder values in the extended system equation (26). The experiment is implemented using Orocos, a C++ framework for advanced machine and robot control¹⁰. The initial estimation

errors for the barrel are approximately 0.4m in the x -direction and 0.15m in the y -direction. The same filtering procedure as explained above is used for the experiment. The figure shows the distance to the surface measured by the laser and the estimation results. The position of the barrel is estimated with reasonable accuracy after approximately 5s. Higher estimation performance can be obtained by (i) reducing measurement noise, (ii) increasing the information content (entropy) of the measurements by altering the prescribed robot motion, (iii) reducing modeling errors (such as calibration errors) and (iv) using more advanced estimation techniques, as discussed in Section V.

VII. CONCLUSIONS AND FUTURE WORK

This paper shows the application of a novel generic approach for constraint-based task specification in the presence of geometric uncertainty. One of the innovative concepts in this approach is the introduction of auxiliary feature and uncertainty coordinates. Using these coordinates greatly simplifies task specification and dealing with geometric uncertainties. Simulation and experimental results for the laser tracing application show the validity and the potential of the approach. Current research focuses on: (i) applying more advanced, mainly Bayesian estimation techniques to handle larger and higher-dimensional uncertainties, (ii) linking the presented framework with high-level task planners, and (iii) developing an Integrated Development Environment (IDE) for sensor-based robot tasks, including a graphical user interface. Given the generality of the approach, large parts of the described task specification procedure can be automated. This way, the required time and user skills for setting up complex sensor-based robot tasks will be substantially reduced.

REFERENCES

- [1] 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 *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005, pp. 3618–3623.
- [2] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbelien, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," Department of Mechanical Engineering, Katholieke Universiteit Leuven, Belgium, Internal report 06PP122, 2006.
- [3] J. De Schutter and H. Van Brussel, "Compliant Motion I, II," *The International Journal of Robotics Research*, vol. 7, no. 4, pp. 3–33, Aug 1988.
- [4] N. Hogan, "Impedance control: An approach to manipulation. Parts I–III," *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 1–24, 1985.
- [5] —, "Stable execution of contact tasks using impedance control," in *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, Raleigh, NC, 1987, pp. 1047–1054.
- [6] H. Kazerooni, "On the robot compliant motion control," *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, vol. 111, pp. 416–425, 1989.
- [7] J. Baeten, H. Bruyninckx, and J. De Schutter, "Integrated vision/force robotics serving in the task frame formalism," *The International Journal of Robotics Research*, vol. 22, no. 10, pp. 941–954, 2003.
- [8] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-11, no. 6, pp. 418–432, 1981.
- [9] H. Bruyninckx and J. De Schutter, "Specification of force-controlled actions in the "Task Frame Formalism": A survey," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 581–589, 1996.

¹⁰Open ROBOT CControl Software, <http://www.orocos.org/>.

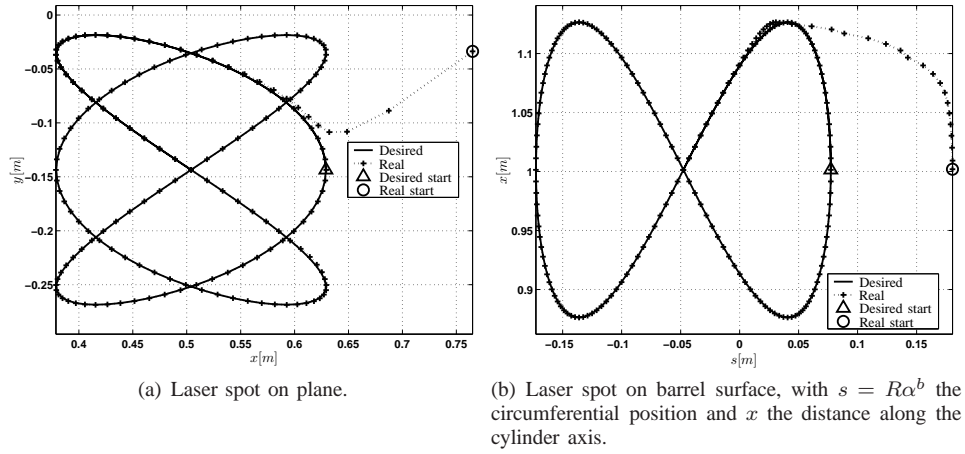


Fig. 3. Laser tracing on plane and barrel while estimating the plane height (h^a), plane orientation (α^a and β^a) and x - and y -position of the barrel (simulation). Data points are displayed for time steps of 0.1s.

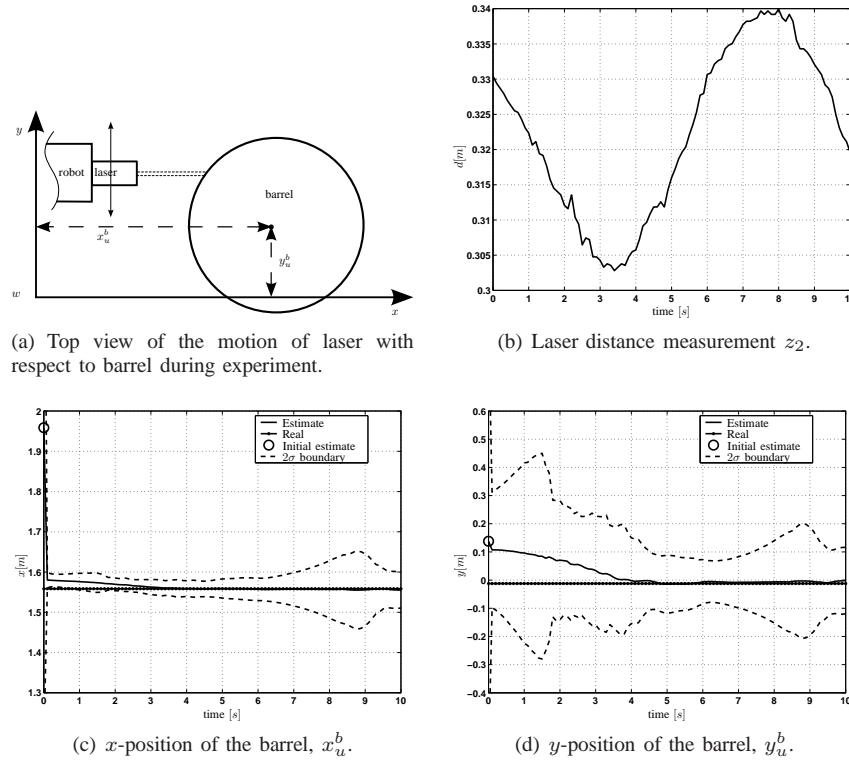


Fig. 4. Laser tracing on barrel: estimation of x - and y - position of the barrel (x_u^b and y_u^b) (experiment). The increase of the covariance as observed from 7s to 9s in (c,d) is due to the limited movement of the laser spot on the barrel resulting in little extra measurement information (b). Note that the initial 2σ boundaries do not fit on the displayed area.

- [10] A. P. Ambler and R. J. Popplestone, "Inferring the positions of bodies from specified spatial relationships," *Artificial Intelligence*, vol. 6, pp. 157–174, 1975.
- [11] C. Samson, M. Le Borgne, and B. Espiau, *Robot Control, the Task Function Approach*. Oxford, England: Clarendon Press, 1991.
- [12] J. C. Latombe, *Robot motion planning*, ser. Int. Series in Engineering and Computer Science. Boston, MA: Kluwer Academic Publishers, 1991, vol. 124.
- [13] K. L. Doty, C. Melchiorri, and C. Bonivento, "A theory of generalized inverses applied to robotics," *The International Journal of Robotics Research*, vol. 12, no. 1, pp. 1–19, 1993.
- [14] Y. Nakamura, *Advanced robotics: redundancy and optimization*. Reading, MA: Addison-Wesley, 1991.
- [15] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME, Journal of Basic Engineering*, vol. 82, pp. 34–45, 1960.
- [16] Y. Bar-Shalom and X. Li, *Estimation and Tracking, Principles, Techniques, and Software*. Artech House, 1993.
- [17] A. E. Gelb, *Optimal Estimation*, 3rd ed. Cambridge, MA: MIT Press, 1978.
- [18] H. Tanizaki, *Nonlinear Filters. Estimation and Applications*. Springer-Verlag, 1996.
- [19] A. Doucet and V. B. Tadic, "Parameter Estimation in General State-Space Models using Particle Methods," *Annals of the Institute of Statistical Mathematics*, vol. 55, no. 2, pp. 409–422, 2003.