



# LUND UNIVERSITY

## Optimal Tracking and Identification of Paths for Industrial Robots

Olofsson, Björn; Nilsson, Henrik; Robertsson, Anders; Åkesson, Johan

*Published in:*

Proceedings of the 18th IFAC World Congress, 2011

2011

*Document Version:*

Publisher's PDF, also known as Version of record

[Link to publication](#)

*Citation for published version (APA):*

Olofsson, B., Nilsson, H., Robertsson, A., & Åkesson, J. (2011). Optimal Tracking and Identification of Paths for Industrial Robots. In *Proceedings of the 18th IFAC World Congress, 2011* (pp. 1126-1132). IFAC.

*Total number of authors:*

4

*Creative Commons License:*

Unspecified

### General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117  
221 00 Lund  
+46 46-222 00 00

## Optimal Tracking and Identification of Paths for Industrial Robots

Björn Olofsson \* Henrik Nilsson \* Anders Robertsson \*  
Johan Åkesson \*\*,\*\*

\* Department of Automatic Control, LTH, Lund University, SE-221 00 Lund,  
Sweden (E-mail: Bjorn.Olofsson@control.lth.se).

\*\* Modelon AB, Ideon Science Park, SE-223 70 Lund, Sweden.

---

**Abstract:** This paper presents results from time-optimal path tracking for industrial robots. More specifically, three subproblems are studied and experimentally evaluated. The first is a contact-force control approach for determining the geometric robot motion, such that the tool centre point of the robot is moved according to the specification. The second problem is off-line solution of the optimisation problem describing the time-optimal path tracking problem, by using software which allows high-level implementation and solution of optimisation problems. The third problem is robust control of the robot during real-time path tracking based on the optimisation results determined off-line. An earlier developed control structure for robust control is implemented and tested in a robot system. This paper discusses the theory behind time-optimal path tracking and presents experimental results. Both contact-force controlled path identification and real-time path tracking of the identified path are evaluated on a 6-DOF industrial robot of type IRB140 from ABB.

*Keywords:* Industrial robots, robot control, optimal control, path tracking, force control.

---

### 1. INTRODUCTION

In many application areas in production, industrial robots are utilised for performing various tasks. A few examples of those, like moving a part from point *A* to point *B*, painting in the automotive industry and assembly in the manufacturing industry can be mentioned. Commonly, a predefined path exists or can be determined, such that the robot is to track this path with its tool centre point (TCP). To determine a path to be tracked in advance, the so called *decoupled approach* (LaValle (2006)), is often to be preferred in order to reduce the complexity of the whole path- and trajectory planning problem. It is common that the tracking is to be performed within minimal time conditions. The path traverse time is limited by the constraints on the actuators of the robot. Consequently, the time-optimal path tracking problem conveniently can be formulated and solved off-line as an optimisation problem. The solution determines control signals to be sent to the robot along the path.

In order to obtain real-time path tracking, a robust control strategy has to be used. Due to modelling errors and disturbances, control of the robot in open-loop, with the control signals obtained in the optimisation, will not work satisfactory. Both feedback and an optimal strategy for reducing the speed of the path traverse in case of actuator saturation are methods that can be introduced to handle the model uncertainty.

The path to be tracked can be the result of a mathematical path planning process. In other cases, where the path to be tracked is instead specified by the motion of a tool mounted on the robot along the contour of an object, experimental methods are required in order to determine the geometric robot motion, such that the tool is moved according to the specification. We consider a contact-force control approach (Siciliano and Villani (1999)) for this task.

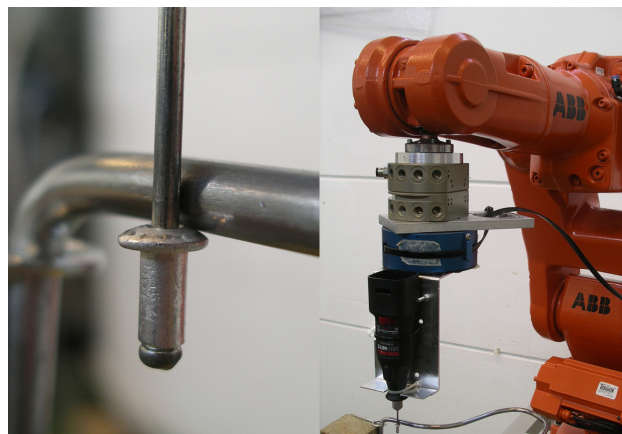


Fig. 1. Experimental setup for time-optimal path tracking. The metal bar is first to be identified using contact-force control and then tracked as close as possible under time-optimal conditions.

In this paper, experimental results from contact-force controlled path identification, off-line optimisation for determining a time-optimal control strategy using the optimisation platform JModelica.org as well as experimental results from real-time path tracking are presented and discussed. The experimental evaluations are made on an industrial robot of type IRB140 from ABB, with the setup displayed in Fig. 1.

### 2. BACKGROUND

The problem of time-optimal path tracking for industrial robots has been investigated in several papers over the past decades. An early method which determined the time-optimal solution to the path tracking problem was presented in the 1980's by

(Bobrow et al. (1985)) and (Shin and McKay (1985)). The method utilises integration in a two-dimensional phase plane, *regardless of* the number of joints in the robot. The phase plane method has then been developed in among others (Pfeiffer and Johanni (1987); Shiller and Lu (1992)).

### 2.1 Optimisation formulation and robust control strategy

In (Dahl (1993)) a strategy for formulating a general optimisation problem describing the path tracking problem for a robot is presented. The optimisation problem can be solved—*e.g.*, with collocation-based optimisation methods—in order to determine a control strategy of the robot. Further, in (Dahl and Nielsen (1989)) a robust strategy for controlling the robot during time-optimal path tracking is presented. The strategy uses the optimisation results obtained off-line as a basis, but modifies those online based on feedback. The optimisation formulation and control strategy presented in (Dahl (1993); Dahl and Nielsen (1989)) has previously been tested using optimisation software and in simulations with satisfactory results (Hast et al. (2009)).

A recent contribution to the area of time-optimal path tracking is (Verscheure et al. (2009a)), which shows how to formulate a convex optimisation problem describing the path tracking problem. A convex formulation is advantageous, since all *locally optimal* solutions are also *globally optimal*, see, *e.g.*, (Boyd and Vandenberghe (2004)). Further, the convex optimisation formulation allows online optimisation for obtaining time-optimal path tracking in cases where the whole path to be tracked is not already determined when the traverse starts. An algorithm for obtaining this is presented in (Verscheure et al. (2009b)).

### 2.2 The optimisation platform JModelica.org

In practice, path optimisation problems cannot be solved analytically for realistic paths. Hence, in this paper the optimisation problem is solved using *simultaneous collocation* (Biegler et al. (2002)). This collocation method transforms the original continuous problem—*i.e.*, a problem of infinite dimension—to a problem with a large, however finite, number of optimisation variables, which is called a non-linear program (NLP).

In order to perform the collocation and solve the resulting NLP, the open-source optimisation software JModelica.org (Åkesson et al. (2010)), initiated at the Department of Automatic Control, Lund University, has been utilised. Using the software, the user utilises the modelling language Modelica (Modelica Association (2010)) for expressing the dynamics in the shape of a differential algebraic equation (DAE) system. However, in Modelica it is not possible to express optimisation problems with arbitrary cost functions and constraints. The optimisation problem as such is therefore formulated with an extension of Modelica called Optimica (Åkesson (2008)). The specific version of collocation implemented in JModelica.org is called orthogonal collocation, where Lagrange polynomials are used to describe the state and variable profiles and the positions of the collocation points are chosen as the corresponding Radau points.

JModelica.org is interfaced with the DAE-simulation software SUNDIALS (SUNDIALS (2010)) and the NLP-solver IPOPT (Wächter and Biegler (2006)). Hence, initial values for the optimisation can be simulated and provided to the solver IPOPT in order to make the convergence to a solution robust. The interface between the user and JModelica.org is the scripting

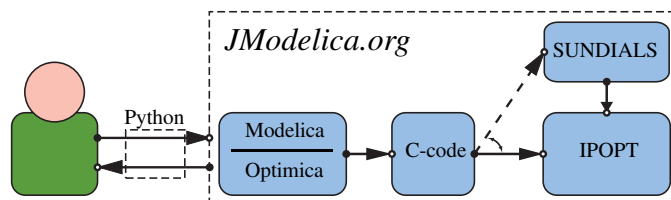


Fig. 2. Flow-chart of the optimisation software JModelica.org.

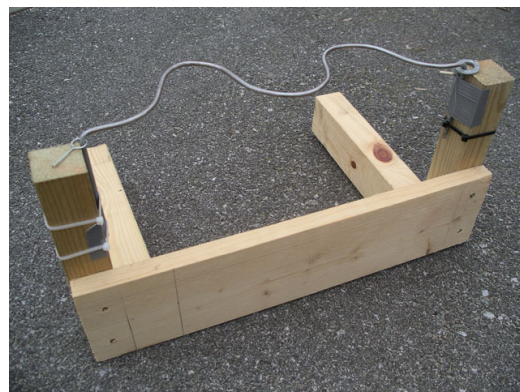


Fig. 3. The metal bar with circular cross section, which is to be tracked.

language Python. The user provides a file with the optimisation problem described in Modelica and Optimica syntax and issues certain commands in the Python environment, whereby the software translates and compiles the optimisation problem into C-code, which can be used in SUNDIALS and IPOPT. A schematic description is displayed in Fig. 2.

## 3. CONTACT-FORCE CONTROL FOR PATH IDENTIFICATION

Contact-force control (Siciliano and Villani (1999)) is an interesting approach for obtaining interaction between the robot and its environment, especially in environments where the geometric conditions are uncertain. In the context of optimal path tracking, contact-force control can be used as an experimental method, which determines the geometric robot motion, when the path to be tracked is only defined as a result of a desire to move a tool along the contour of an object.

### 3.1 Preliminaries

As a case-study of time-optimal path tracking, the curved metal bar in Fig. 3 is to be tracked. This can for instance be thought of as preparation for a future grinding process of the metal bar, which is a common application of industrial robots. A force sensor of model 100M40A-I63 from JR3 (JR3, Inc. (2011)), measuring the forces and torques exerted on it, is attached to the tool changer mounted on the robot flange. The forces and torques are measured in two Cartesian coordinate systems with a common origin. A small metal stick with a thick bottom is attached orthogonal to the force sensor, see Fig. 1.

The control problem can be described as follows: Initially, contact has to be established between the metal stick attached to the force sensor and the metal bar, then the robot should be controlled in such a way that the contact-force is held constant while moving the metal stick along the metal bar, which has an unknown shape. Furthermore, the tool attached to the robot has

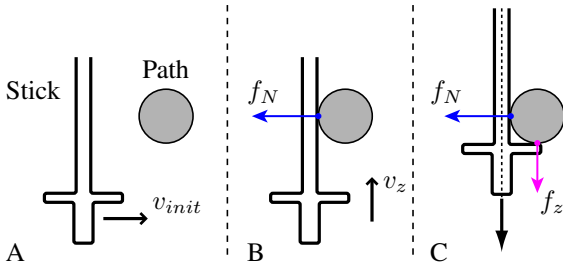


Fig. 4. Illustration of the first steps in the path identification. The stick and the cross section of the path are displayed.

to be reoriented along the metal bar such that the orientation of the tool is kept the same with respect to the metal bar. During the traverse along the metal bar, the positions of the joints in the robot are stored. These positions then define the motion of the robot along the path.

### 3.2 Force control strategy

**Force control** The force control strategy aims to keep the contact-forces constant in two different directions, see Fig. 4, both on the side of and underneath the metal bar. The force measurements from the force sensor are utilised in order to close a feedback loop for position control. The main part of the force controller is two PI controllers keeping the contact-forces constant in each of the controlled directions. The control error in the PI controller in the normal direction is formed as the difference between the norm of the measured normal force vector and a reference value. The reference value is chosen as a compromise between the risk of losing contact with the surface with a low reference value and the increase of the friction forces with higher reference values. Further, the controller parameters are tuned experimentally.

The direction of the control is defined as the opposite direction of the measured normal force vector. The control signal and corresponding direction are then interpreted as the velocity vector of the TCP, *i.e.*, the TCP is moved in the direction and with the velocity given by the control signal. The structure of the PI controller in the vertical direction is the same as that of the controller in the normal direction.

**Torque control** In order to reorient the tool correctly along the metal bar, the torque measurements from the force sensor are utilised. When the metal stick is in contact with the metal bar, the normal force gives rise to a torque, see Fig. 5, about the  $z$ -axis of the force sensor coordinate system. By controlling this torque to be held constant, the correct orientation of the tool is obtained along the metal bar. The chosen control structure is a PI controller, which acts on the difference between the torque in the  $z$ -direction and a constant reference value, corresponding to the desired orientation relative the path.

The control signal  $u_{rot}$  from the PI controller is interpreted as an angular velocity about the  $z$ -axis of the TCP coordinate system, which is attached to the metal stick in contact with the metal bar. The angular velocity  $\omega$ , expressed in the TCP coordinate system, can be written according to

$$\omega = [0 \ 0 \ -u_{rot}]^T. \quad (1)$$

The rotation axis is important since the reorientation has to be done about the point of contact between the stick and the metal bar, otherwise contact will be lost.

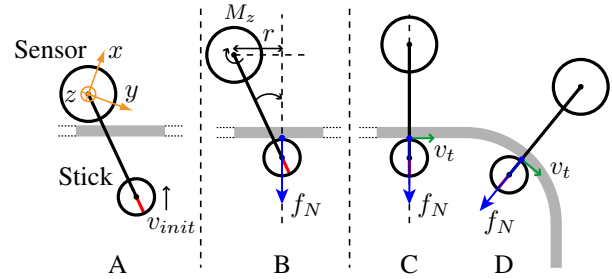


Fig. 5. Illustration, seen from above, of the reorientation of the tool during the path identification, such that its orientation is the same with respect to the metal bar.

### 3.3 Motion along the path

In order to move the tool along the metal bar with the unknown shape, a new tangential direction of the metal bar is calculated in every sample based on the information in the force sensor measurements. The tangential direction  $v_t$  is calculated such that it is perpendicular both to the measured normal force vector  $f_N$  and the normal vector  $n$  of the plane in which the path is located, *i.e.*, as

$$v_t = n \times f_N. \quad (2)$$

The calculated vector is normalised such that its length corresponds to a motion of the TCP along the metal bar with a constant velocity.

## 4. OFF-LINE OPTIMISATION

In order to determine an optimal control strategy of the robot off-line using optimisation, an accurate model of the robot has to be determined, *e.g.*, using system identification methods. With the model at hand, an optimisation problem can be formulated.

### 4.1 Robot model

It is assumed that the robot can be described by a simplified version of the rigid body model for a serial kinematic robot with  $n$  joints, found in, *e.g.*, (Spong et al. (2006)), according to

$$\tau = M\ddot{q} + D\dot{q}, \quad (3)$$

where  $\tau \in \mathbb{R}^n$  is the applied torques on the joints;  $q \in \mathbb{R}^n$  the joint positions;  $\dot{q} \in \mathbb{R}^n$  the joint velocities;  $\ddot{q} \in \mathbb{R}^n$  the joint accelerations;  $M \in \mathbb{R}^{n \times n}$  the inertia matrix and  $D \in \mathbb{R}^{n \times n}$  the viscous friction matrix.

It is to be noted that the model (3) can be generalised if position or velocity control of the individual joints in the robot is utilised instead of torque control. Actually, in the implementation presented in this paper, the joint velocity reference is considered as the control signal. However, the structure of the subsequent optimisation problem is exactly the same.

### 4.2 Optimisation problem

The path which is to be tracked,  $f \in \mathbb{R}^n$ , is expressed in the joint positions of the robot. The path is parametrised in a path coordinate denoted  $s(t)$ , with  $s(t) \in [s_0, s_f]$ ,  $[s_0, s_f] \subset \mathbb{R}$ , and  $t$  denoting time. For path tracking it is required that  $q = f(s)$ , and consequently

$$\dot{q} = f'(s)\dot{s}, \quad \ddot{q} = f'(s)\ddot{s} + f''(s)\dot{s}^2, \quad (4)$$



where  $(\prime)$  denotes  $\frac{\partial}{\partial s}$  and  $(\dot{\phantom{x}})$  denotes  $\frac{d}{dt}$ . With these relations, the robot dynamics (3) can be rewritten, see, e.g., (Bobrow et al. (1985)),

$$\begin{aligned}\tau(s) &= M[f'(s)\ddot{s} + f''(s)\dot{s}^2] + Df'(s)\dot{s} = \\ &= \Gamma_1(s)\ddot{s} + \Gamma_2(s, \dot{s}),\end{aligned}\quad (5)$$

where  $\dot{s}$  is referred to as the path velocity and  $\ddot{s}$  the path acceleration. When the robot dynamics has been expressed according to above, an optimisation problem can be formulated with the path coordinate  $s$  and its derivatives as variables following (Dahl (1993)) and similar to (Verschueren et al. (2009a)). Firstly, the state variable  $x(s) = \dot{s}(s)^2$  is introduced. The optimisation criterion is the final time  $t_f$  of the path traverse, which can be expressed in the state  $x(s)$  according to

$$t_f = \int_0^{t_f} 1 dt = \int_{s_0}^{s_f} \frac{dt}{ds} ds = \quad (7)$$

$$= \int_{s_0}^{s_f} \frac{1}{\dot{s}} ds = \int_{s_0}^{s_f} \frac{1}{\sqrt{x(s)}} ds, \quad (8)$$

if it is assumed that  $\dot{s} \geq 0$ . Further, assuming that the robot is to start and stop at rest and that the joint torques are limited to be in the range  $\tau_{min} \leq \tau \leq \tau_{max}$ , the optimisation problem can be formulated similar to (Dahl (1993)) according to

$$\text{minimise } \int_{s_0}^{s_f} \frac{1}{\sqrt{x(s)}} ds \quad (9)$$

$$\text{such that } \tau(s) = \Gamma_1(s)\ddot{s} + \Gamma_2(s, \dot{s}) \quad (10)$$

$$\dot{s}(s_0) = \dot{s}(s_f) = 0, \quad x'(s) = 2\dot{s}\ddot{s} \quad (11)$$

$$\dot{s}(s) \geq 0, \quad \tau_{min} \leq \tau(s) \leq \tau_{max} \quad (12)$$

$$x(s) = \dot{s}(s)^2, \quad (13)$$

where the path acceleration  $\ddot{s}(s)$  is considered as the input which is to be determined and  $s$  is the independent variable. This optimisation problem is solved using the software JMod-elica.org. Note that the problem only has one differential state  $x(s)$ , regardless of the number of joints in the robot. This can be compared to an ordinary optimal control problem with a rigid-body model of a  $n$ -DOF robot which has  $2n$  states.

## 5. REAL-TIME PATH TRACKING

Under ideal circumstances it would be satisfying to directly apply the optimised values of the control signals, i.e., the joint torques, on the robot. However, in a time-optimal solution to the path tracking problem it can be shown that, under mild assumptions, one and only one of the joints is saturated in each time instance in terms of applied torque (Chen and Desrochers (1989)). Consequently, the robustness is low since there is no margin in the control signals left for handling modelling errors and disturbances.

### 5.1 Path velocity control

In order to make the control more robust, a previously developed control strategy, first presented in (Dahl and Nielsen (1989)), and later referred to as *path velocity control* (PVC) is implemented. The idea behind this strategy is to perform the path tracking with the optimised values of the path velocity  $\dot{s}(s)$  and the path acceleration  $\ddot{s}(s)$  as basis, and modify those values online based on feedback from the robot. More specifically, the speed of the path traverse is reduced if the algorithm realises

that the speed is too high due to the errors mentioned above. Consequently, real-time versions of the path coordinate  $s$  and its time-derivatives, denoted  $\sigma$ ,  $\dot{\sigma}$  and  $\ddot{\sigma}$ , are introduced. The integrator  $\dot{\sigma} = u$  is then driven with the modified path acceleration as input signal in order to obtain  $\dot{\sigma}$  and  $\sigma$ . The nominal values of  $\ddot{\sigma}$  are obtained from the optimisation.

**Bounds on path acceleration** In the PVC an internal tracking controller is used for feedback from the robot system. Any controller can be used as long as the control law can be parametrised in  $\sigma$  and its time-derivatives on the following general form

$$\tau = \beta_1 \ddot{\sigma} + \beta_2, \quad (14)$$

where  $\beta_1$  and  $\beta_2$  are functions of  $\sigma, \dot{\sigma}, q$  and  $\dot{q}$ , but do not depend on  $\ddot{\sigma}$ . With this parametrisation of  $\tau$  and the limits on the torques,  $\tau_{min} \leq \tau \leq \tau_{max}$ , the upper and lower bounds on the path acceleration can be calculated online according to (Dahl and Nielsen (1989))

$$\ddot{\sigma}_{max} = \min_i \ddot{\sigma}_{max}^i, \quad \ddot{\sigma}_{min} = \max_i \ddot{\sigma}_{min}^i, \quad (15)$$

where

$$\ddot{\sigma}_{max}^i = \begin{cases} \frac{\tau_{min} - \beta_2^i}{\beta_1^i}, & \beta_1^i < 0 \\ \frac{\tau_{max} - \beta_2^i}{\beta_1^i}, & \beta_1^i > 0 \\ \infty, & \beta_1^i = 0 \end{cases} \quad (16)$$

$$\ddot{\sigma}_{min}^i = \begin{cases} \frac{\tau_{max} - \beta_2^i}{\beta_1^i}, & \beta_1^i < 0 \\ \frac{\tau_{min} - \beta_2^i}{\beta_1^i}, & \beta_1^i > 0 \\ -\infty, & \beta_1^i = 0 \end{cases} \quad (17)$$

In (15)–(17), superscript  $i$  denotes joint  $i$ ,  $i = 1, \dots, n$ , and the assumption was made that all joints have the same torque limitations.

**Feedback from path velocity** The bounds on the path acceleration  $\ddot{\sigma}$  stated in the last paragraph are used to saturate the nominal values of the path acceleration. However, this saturation might result in that the robot deviates from the path since it can be driven to a point where there are no allowed torques which can be applied in order to retain the path tracking. Therefore, in the PVC, internal feedback  $v_f(\sigma)$  is introduced from the optimised path velocity according to (Dahl and Nielsen (1989))

$$v_f(\sigma) = \frac{\alpha}{2}(\dot{s}(\sigma)^2 - \dot{\sigma}(\sigma)^2), \quad (18)$$

where  $\alpha$  is a parameter used to choose the gain of the feedback and  $\dot{s}(\sigma)$  is the optimised path velocity. This choice of feedback achieves asymptotic tracking of the optimised path velocity when the path acceleration is not saturated. The calculated signal  $v_f(\sigma)$  in (18) is added to the nominal value of the path acceleration  $\ddot{\sigma}$ .

**Path velocity scaling** The PVC scheme also contains a scaling of the optimised path velocity based on the saturation of the path acceleration. The scaling factor is updated adaptively. This scaling was also used in the implementation. For further details, the reader is referred to the reference stated above.

## 6. EXPERIMENTAL RESULTS

Contact-force controlled path identification and time-optimal path tracking have been experimentally evaluated on a 6-DOF

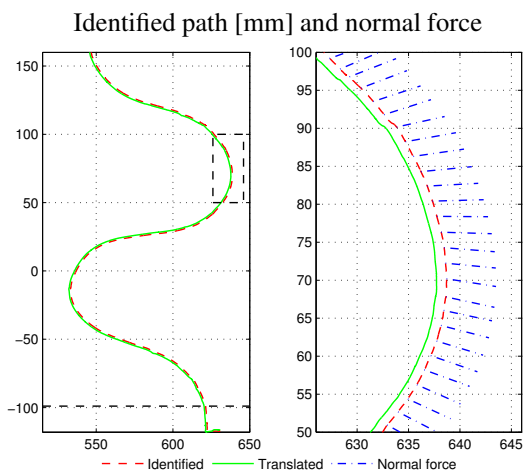


Fig. 6. The identified path in its whole (left) and detail with the direction of measured contact force (right). The TCP  $y$ -position is plotted as function of the TCP  $x$ -position.

industrial robot from ABB of type IRB140 (ABB Robotics (2009)), together with a control cabinet of model IRC5. The software architecture is an open robot control architecture called ORCA, developed at Lund University, which allows implementation of controllers, designed in Simulink in MATLAB, in the robot system (Blomdell et al. (2010)). C-code is automatically generated by the toolbox *Real-Time Workshop*. The C-code is compiled and dynamically linked into the robot system.

### 6.1 Path identification

Contact-force controlled path identification was experimentally evaluated on the metal bar displayed in Fig. 3. The result is displayed in Fig. 6. Initially, the robot is given a search direction such that contact between the metal stick and the metal bar can be established. When a contact-force is measured, the force- and torque controllers are activated and motion along the metal bar with the unknown shape is started.

### 6.2 Time-optimal path tracking

*Optimisation in JModelica.org* The joint positions of the robot along the metal bar, which were stored during the path identification, are used in the optimisation software JModelica.org when solving the optimisation problem (9)–(13). To represent the joint positions in the Modelica model, spline approximations are utilised. Spline approximations are suitable since also the derivatives of the joint position profiles are required during the solution.

In order to reduce the wear of the robot joints, weighting of the derivatives of the control signals is introduced in the cost function (9) in the optimisation. This turns out to considerably reduce the rates of change for the control signals. However, it also means that the time-optimality of the solution is compromised with a few percent, given that the weighting coefficient is chosen reasonably.

The optimised values of the path velocity  $\dot{s}(s)$  and the path acceleration  $\ddot{s}(s)$  are displayed in Fig. 7. When solving the optimisation problem, initial values for the optimisation variables are determined by simulations. For this problem and the studied robot models and paths, the initial values are not that critical for obtaining convergence. For further details on the solution of

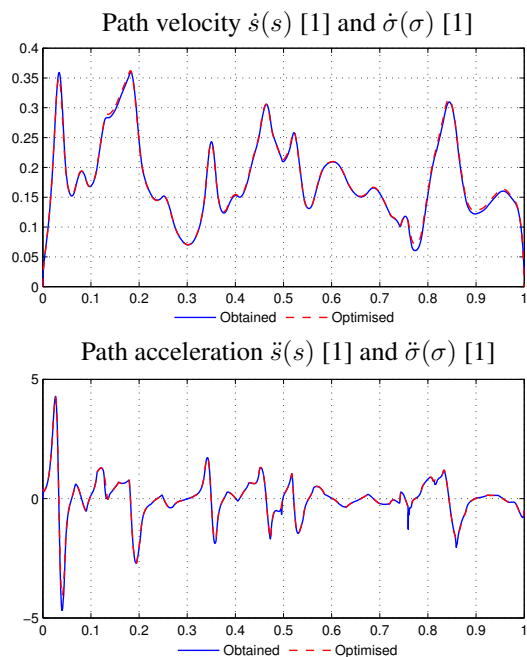


Fig. 7. Optimised values of the path acceleration and the path velocity as well as corresponding quantities during real-time path tracking, with respect to normalised path coordinate.

the optimisation problem, the reader is referred to (Nilsson and Olofsson (2010)).

*Real-time path tracking* Real-time path tracking is experimentally evaluated on the robot system. For visual evaluation of the tracking performance, a small ring, with a diameter in the range of the diameter of the metal bar, is placed around the bar and mounted on the robot. The ring must not come into contact with the metal bar during the time-optimal path traverse.

In the experiments on the robot system, the PVC structure is used together with the result of the optimisation in JModelica.org. In the PVC, an internal controller for the robot system is used to obtain the tracking of the path as described earlier. In the current experiment, a feed-forward controller combined with a standard PD controller, see, e.g., (Spong et al. (2006)), is chosen. This controller is tuned by changing the gains in the position control loop and in the velocity control loop.

The result from the experiments is displayed in Figs. 7–9. From Fig. 8 it is clear that the path tracking is working satisfactory. However, it can be noted that certain parts of the path are harder to track accurately. A detailed analysis of these parts showed that the path deviations in those cases are mainly because of the spline approximations in the optimisation and not due to the tracking performance of the robot. Further, it can be concluded from Fig. 9 that the PVC structure keeps the control signals within their limits during the whole path traverse.

## 7. DISCUSSION AND CONCLUSIONS

### 7.1 Path identification

It is apparent that contact-force controlled path identification is working satisfactory and the metal bar is identified with

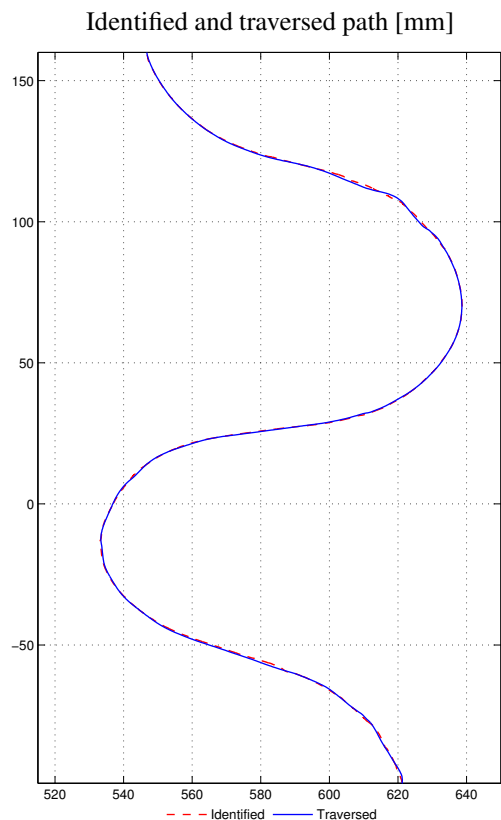


Fig. 8. Force control identified path and time-optimal path traverse obtained with the PVC. The  $y$ -coordinate of the TCP-position is plotted as function of the  $x$ -coordinate.

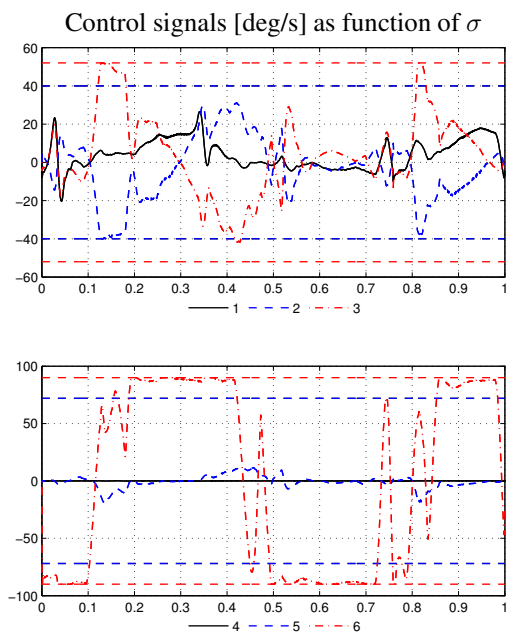


Fig. 9. Control signal and saturation limits for each joint as function of  $\sigma$ . Notice that the sixth joint is the joint which is saturated most part of the path traverse.

a high accuracy even though the shape of the metal bar is unknown *a priori*. However, it is to be noted that the friction forces arising between the metal stick and the metal bar result in a measured contact-force in the normal direction which is not completely perpendicular to the path tangent, see Fig. 6. However, as the discrepancy is low this issue was neglected in the implementation. This issue is of course more prominent if other materials with higher friction than metal are used.

### 7.2 Optimisation with JModelica.org

Using the optimisation software JModelica.org for solving the time-optimal path tracking problem proved to be successful. The software allows straightforward implementation of the robot dynamics and explicit high-level formulation of the optimisation problem as such. Further, the software allows an iterative procedure for obtaining a satisfactory result when derivative weighting of the control signals in the cost function is utilised. Also, it is straightforward to change the geometric joint position profiles to be tracked when a new path identification has been performed.

### 7.3 Time-optimality

From Fig. 9 the time-optimality of the path traverse can be examined, since time-optimality means that one of the joints has to be saturated, in terms of applied control signal, in every time instance. One of the joints is indeed saturated during almost the whole path traverse, which indicates near time-optimality. It can also be noted that it is the sixth joint which is saturated most of the path traverse, which is a result of the reorientation of the tool and the experimental setup.

Another significant measure of the time-optimality is the path traverse time. The experimentally measured traverse time is 6.62 s. This could be compared to the theoretical minimum traverse time of 6.49 s, which is calculated in the optimisation when derivative weighting of the control signals is utilised. The pure time-optimal traverse time—without weighting of the derivatives—is 6.30 s. This means that the experimentally measured traverse time is approximately 5 % longer than the theoretically time-optimal. Since the increase compared to the optimal case is low, the experimental result has to be considered as reasonable.

### 7.4 Tracking performance

The tracking error for each of the joint controllers during the path traverse is displayed in the plots in Fig. 10. The error is measured as the difference between the reference position calculated in the PVC and the actual joint position. As expected, the joints with the highest velocities and fastest changes have the least accurate tracking performances. This is especially the case for the sixth joint.

It is to be noted that the plots in Fig. 10 do not give any information about the direction of the tracking error. If the error is directed tangential to the path, the path tracking is not that severely affected as compared to when the error is directed perpendicular to the path. In the current experiment, it is likely that the tracking error is directed along the path based on the satisfactory tracking result displayed in Fig. 8.

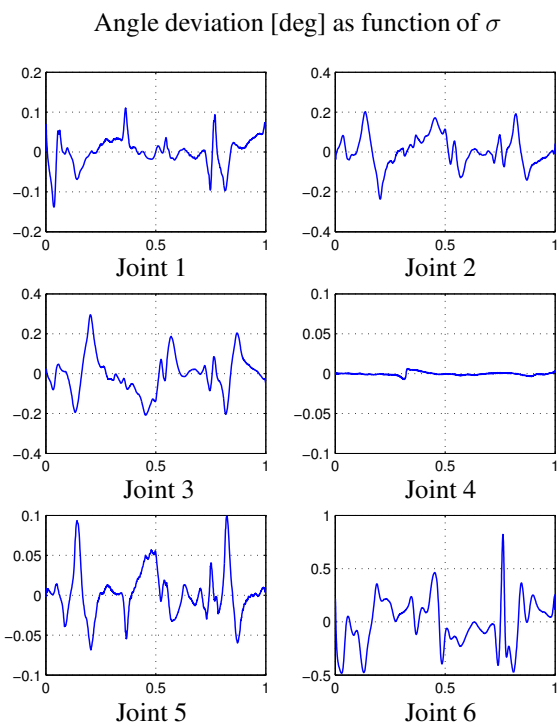


Fig. 10. The error in the joint positions during the path traverse.

### 7.5 Conclusions

To conclude, the experimental results presented in the previous section indicate that the procedure for time-optimal path tracking outlined in this paper—*i.e.*, path identification using contact-force control, optimisation using the software JModelica.org and real-time tracking control of the robot utilising the PVC structure—is a competitive method, which can be used to improve both the accuracy and the efficiency of industrial robots in production and manufacturing.

## 8. ACKNOWLEDGMENTS

This research was partially made in the framework of the project ProFlexA, supported by the Swedish Foundation for Strategic Research under the programme ProViking. This financial support is gratefully acknowledged.

## REFERENCES

- ABB Robotics (2009). ABB IRB140 Industrial Robot Data sheet. Data sheet nr. PR10031 EN\_R8.
- Åkesson, J. (2008). Optimica—an extension of Modelica supporting dynamic optimization. In *6th Int. Modelica Conf. 2008*. Modelica Association.
- Åkesson, J., Årzén, K.E., Gäfvert, M., Bergdahl, T., and Tummescheit, H. (2010). Modeling and optimization with Optimica and JModelica.org—Languages and tools for solving large-scale dynamic optimization problems. *Computers and Chemical Engineering*.
- Biegler, L.T., Cervantes, A.M., and Wächter, A. (2002). Advances in simultaneous strategies for dynamic process optimization. *Chemical Engineering Science*, 57, 575–593.
- Blomdell, A., Dressler, I., Nilsson, K., and Robertsson, A. (2010). Flexible application development and high-performance motion control based on external sensing and reconfiguration of ABB industrial robot controllers. In *Proc. of the workshop of "Innovative Robot Control Architectures for Demanding (Research) Applications—How to Modify and Enhance Commercial Controllers"*, the 2010 IEEE Int. Conf. on Robotics and Automation (ICRA2010), 62–66. Anchorage, Alaska, USA.
- Bobrow, J.E., Dubowsky, S., and Gibson, J.S. (1985). Time-optimal control of robotic manipulators along specified paths. *The Int. Journal of Robotics Research*, 4(3), 3–17.
- Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge, UK.
- Chen, Y. and Desrochers, A.A. (1989). Structure of minimum-time control law for robotic manipulators with constrained paths. In *Proc. of 1989 IEEE Int. Conf. on Robotics and Automation*, volume 2, 971–976. IEEE.
- Dahl, O. (1993). Path constrained motion optimization for rigid and flexible joint robots. In *Proc. of the 1993 IEEE Int. Conf. on Robotics and Automation*, 223–229. Atlanta, Georgia.
- Dahl, O. and Nielsen, L. (1989). Torque limited path following by on-line trajectory time scaling. In *IEEE Int. Conf. on Robotics and Automation*, 1122–1128. Scottsdale, Arizona.
- Hast, M., Åkesson, J., and Robertsson, A. (2009). Optimal robot control using Modelica and Optimica. In *Proc. of the 7th Int. Modelica Conf. 2009*. Modelica Association.
- JR3, Inc. (2011). JR3 100M40A Force and Torque sensor, Data sheet.
- LaValle, S.M. (2006). *Planning Algorithms*. Cambridge University Press. URL: <http://planning.cs.uiuc.edu/>.
- Modelica Association (2010). URL: <http://www.modelica.org>.
- Nilsson, H. and Olofsson, B. (2010). Optimal tracking and identification of paths for industrial robots. Master's thesis, Dept. of Automatic Control, Lund University, Sweden.
- Pfeiffer, F. and Johanni, R. (1987). A concept for manipulator trajectory planning. *IEEE Journal of Robotics and Automation*, RA-3(2), 115–123.
- Shiller, Z. and Lu, H.H. (1992). Computation of path constrained time optimal motions with dynamic singularities. *Journal of Dynamic Systems, Measurement, and Control*, 114, 34–40.
- Shin, K.G. and McKay, N.D. (1985). Minimum-time control of robotic manipulators with geometric path constraints. *IEEE Transactions on Automatic Control*, AC-30(6), 531–541.
- Siciliano, B. and Villani, L. (1999). *Robot Force Control*. Kluwer Academic Publishers.
- Spong, M.W., Hutchinson, S., and Vidyasagar, M. (2006). *Robot Modeling and Control*. John Wiley & Sons.
- SUNDIALS (2010). (SUite of Nonlinear and Differential/ALgebraic equation Solvers), URL: <https://computation.llnl.gov/casc/sundials/main.html>.
- Verschuere, D., Demeulenaere, B., Swevers, J., De Schutter, J., and Diehl, M. (2009a). Time-optimal path tracking for robots: A convex optimization approach. *IEEE Transactions on Automatic Control*, 54(10), 2318–2327.
- Verschuere, D., Diehl, M., De Schutter, J., and Swevers, J. (2009b). On-line time-optimal path tracking for robots. In *2009 IEEE Int. Conf. on Robotics and Automation*, 599–605. Kobe, Japan.
- Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), 25–57.