8th Int. IFAC Symposium on Robot Control: SYROCO 2006
Bologna, Italy, Sept. 2006

IS IMPEDANCE-BASED CONTROL SUITABLE FOR TRAJECTORY SMOOTHING?

Friedrich Lange [*,1] **Mirko Frommberger** [*]
Gerd Hirzinger [*]

[*] *Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR)*
Oberpfaffenhofen, D-82234 Weßling, Germany
http://www.robotic.de/Friedrich.Lange/

Abstract: The paper discusses the difference between explicit sensor-based control and impedance control, using an example of fast vision-based contour tracking. While explicit sensor-based control may cause a rough desired robot path with sharp vertices, impedance control is known to filter the trajectory at the expense of a bigger control error. We present a new implementation and a new approach to impedance-based control where the latter turns out to produce smooth trajectories with small control errors. Both methods are independent of actual control errors but only depend on the desired speed and the sensed shape of the contour.

Keywords: impedance control, tracking, vision, predictive control

1. INTRODUCTION

Impedance control has been introduced for force control tasks. We concentrate on those approaches which instead of joint torque control use a positional interface which is existing for industrial robots, see e.g. Heinrichs and Sepehri (1999) or Ferretti et al. (2004). Such an interface allows in each sampling step to send positional commands and to receive measurements of the current position (see e.g. Pertin and Bonnet-des-Tuves (2004)). Typically, either variables in joint coordinates or in Cartesian space are processed with sampling rates of about 100 Hz or more. For force control the positional commands are computed from forces and torques of a wrist mounted sensor. They typically affect the system in an outer loop that is created in addition to the usual (inner loop) PID-controllers. Specification of impedance control is implemented in the outer loop as well.

Our approach is not limited to forces and torques as input but we especially consider non-tactile sensors as distance sensors or even cameras. This generalizes the usual problem of a robot in contact with the environment to non contact scenarios. In this case, at least with cameras as sensors, we have to tolerate very low sampling rates that are not acceptable for direct control.

In this paper we distinguish between the underlying control of the robot and the motion which is desired. So our approach to impedance control is independent of a particular controller or model of the robot. We assume an *ideal* robot and compute a desired path which depends on the sensor values. The *ideal* robot may be implemented by predictive control according to Clarke et al. (1987) or an equivalent method by Lange and Hirzinger (1996).

The motivation for this approach is that internal positional control is sampled much faster than the camera frame rate. Therefore the inner positional loop is better suited for control of the robot

[1] Partially supported by the Bayerische Forschungsstiftung

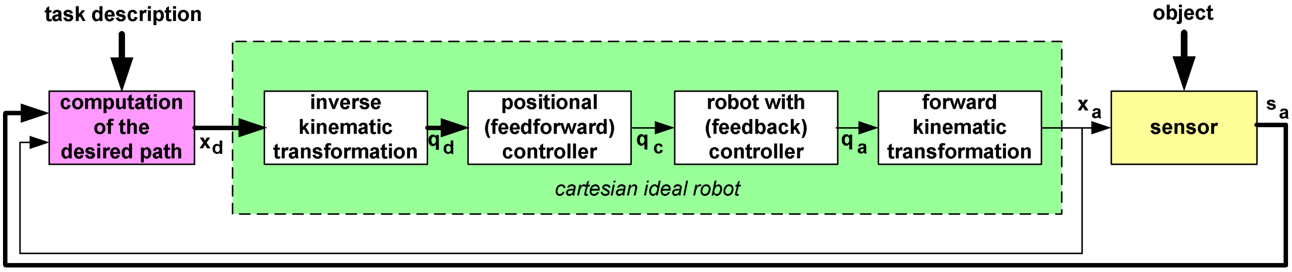
task description
object
computation of the desired path
x_d
inverse kinematic transformation
q_d
positional (feedforward) controller
q_c
robot with (feedback) controller
q_a
forward kinematic transformation
x_a
sensor
s_a
cartesian ideal robot

Fig. 1. Control architecture with positional control as inner loop (green) and the determination of the desired path as outer loop

dynamics including nonlinear effects as friction or joint elasticity. So in contrast to other papers as Ferretti et al. (2004), in the outer loop we can restrict on a linear consideration and a relatively low sensor sampling rate.

The task we consider in this paper is high speed tracking of a visible contour by an industrial robot using a camera. Figure 2 shows that the contour consists of straight edges with vertices at the junctures. The vertical distance between the camera and the edge is known.

The challenge of this task comes from saturation effects in the inner loop, which means that the robot can only reach a good accuracy for smooth or low-speed trajectories. So the desired path has to be determined executable. In contrast, explicit sensor-based control in the strict sense reproduces vertices of the sensed contour or a rough surface. Instead, we can define a desired impedance to specify the desired reaction due to sensor signals, i.e. a law how to deviate from the path of explicit sensor-based control.

The separation of control and task specification effects that this desired motion is independent of the actually executed motion but depends only on the specification of the reference task and the impedance parameters, and on the real environment.

So impedance-based control is understood in this paper as the specification and execution of a trajectory that realizes the given impedance law in presence of a sensible fixed contour.

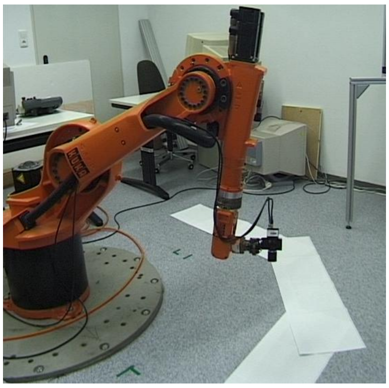

Fig. 2. Robot and edges for visual servoing

Our basic approach to sensor-based control has been presented by Lange and Hirzinger (2005) who address the problem of explicit sensor-based control using an ideal position controlled robot. This will be recapitulated in the next section.

Section 3 then gives a motivation for impedance control by showing the differences with respect to explicit sensor-based control. Though the latter theoretically is optimal for control of sensor data, we prefer an impedance approach because motion is then smoother and more robust to disturbances.

Section 4 presents a second approach to impedance-based control which in contrast to the other method has no offset.

Section 5 then shows the difference between the individual methods in experiments with an industrial robot.

2. BASIC STRUCTURE OF SENSOR-BASED CONTROL

The method is based on the separation between the positional control and the integration of sensor data, as shown in Figure 1. The first is realized by a combination of the existing industrial feedback controller and a feedforward filter that realizes predictive control. Sensor data are then used to compute a desired path of the robot from the specification of the reference task and sensor data of the real environment.

2.1 Inner loop positional control

In Figure 1 positional control is executed by joint controllers since on joint level there are less couplings than in Cartesian space. The kinematic transformations between the Cartesian pose vectors $\mathbf{x}$ and the vectors of joint positions $\mathbf{q}$ are known since we do not consider redundant robots or singular configurations. $\mathbf{q}_d$ and $\mathbf{q}_a$ are the desired and the actual joint values respectively. $\mathbf{q}_c$ is the vector of commanded positions which is the usual input to industrial control systems similar to that of Pertin and Bonnet-des-Tuves (2004). $\mathbf{q}_c$ is computed using an adaptive feedforward controller presented in Lange and Hirzinger (1996) or Grotjahn and Heimann (2002), which processes in each sampling step the desired values of the current and of some future sampling steps (bold face lines in Figure 1 denote values of several sampling steps). A sufficient number of future values theoretically allows the realization of an *ideal robot*, i.e. a robot which precisely executes the desired trajectory without time delays.

As feedforward controller we tune the usually diagonal matrices $\mathbf{K}_{qi}$ of

$$\mathbf{q}_c(k) = \mathbf{q}_d(k) + \sum_{i=0}^{n_d} \mathbf{K}_{qi} \cdot (\mathbf{q}_d(k+i) - \mathbf{q}_d(k)) \quad (1)$$

by a parameter adaptation method using a fixed path which is not defined by sensor data.

A limitation of the performance of the *ideal robot* may result from the requirement of future values $\mathbf{q}_d(k+i)$ for the computation of $\mathbf{q}_c(k)$. So if sensor data are processed, this means that they have to be extrapolated or predicted.

Further restrictions are given by limits for the robot joint accelerations. In case of a non smooth desired trajectory the acceleration commands are cut frequently, yielding deviations in joint space. This is one of the motivations for a Cartesian trajectory smoothing by impedance control.

2.2 Explicit control of sensor data

In this paper a sensed contour is used to define the desired robot motion. In the general case we sense position and orientation of an object and define the distance the robot should keep from it while moving in a given direction.

This gives

$${}^r\mathbf{x}_d = {}^r\mathbf{x}_a + {}^a\mathbf{x}_d \quad (2)$$

where the indices $_r$, $_d$, and $_a$ stand for reference, desired tcp, and actual tcp, respectively. $\mathbf{x}$ represents the position of the tool center point (tcp).

The positions are expressed with respect to a reference pose that is the desired pose in the case of nominal sensor values. A reference system is introduced for three reasons: First, sensor data provide only geometrical information. The velocity profile has to be given, e.g. according to a reference trajectory. Second, for non sensible components the desired pose is defined to coincide with the reference pose. And third, the differences of the orientations of the actual and the desired system are usually small. This allows to formulate (2) instead of a matrix equation in homogeneous coordinates.

The control error ${}^a\mathbf{x}_d$ can be computed from the sensor signal and the desired value by

$${}^a\mathbf{x}_d = {}^a\mathbf{x}_o + {}^o\mathbf{x}_d = {}^a\mathbf{x}_o - {}^d\mathbf{x}_o. \quad (3)$$

The sensor signal ${}^a\mathbf{x}_o$ measures the object with respect to the tcp. This requires that the sensor is calibrated with respect to it.

Depending on the task description, the origin of the object system will be the center of gravity of the object or a point on the object surface. In all cases the desired tcp position ${}^o\mathbf{x}_d$ is expressed with respect to this object system.

Future sensor values ${}^a\mathbf{x}_o(k+i)$, that are required for feedforward control in the inner loop, will be computed from the actual position $\mathbf{x}_a(k)$ and from the sensed object shape.

Using (2) and (3), the desired path $\mathbf{x}_d$ can be computed from the sampled values of the reference pose $\mathbf{x}_r$, the actual pose $\mathbf{x}_a$ of the tcp and the control errors ${}^a\mathbf{x}_d$ between desired and actual tcp positions, which are all given or sensed.

Then the *ideal robot* will control the tcp along the desired trajectory. Theoretically only one measurement is required for this as long as the object is time-invariant. In practice however, visibility restricts the predictions. Nevertheless the method tolerates low sensor sampling rates as the frame rate of vision systems.

3. IMPEDANCE-BASED CONTROL

This section begins with a motivation for impedance control instead of explicit sensor-based control. Then we formulate the equations of a new method that specifies the desired motion using an impedance law instead of directly using the impedance parameters for control. Thereafter the characteristics of this method are presented. Finally details for the integration into the structure of Section 2 are given.

3.1 Motivation

What are the differences between explicit sensor-based control (as hybrid position and force control) and impedance control? Of course, the control error of the sensor values will be less without an impedance law. But there are also advantages. Impedance control is favorable

- with big initial control errors or big changes of the sensed values, since in this case explicit sensor control as in Section 2.2 tries to reduce the sensed control errors within the next sampling step.

- to slow down accelerations that otherwise would be caused by vertices or rough objects. The specification of an impedance law may prevent to exceed the maximum joint torques of the robot or damp oscillations that are caused by the high gains of explicit sensor-based control.
- to reduce the deviations from the reference path.

Especially the first two items have been the motivation for this paper since always existing restrictions with respect to accelerations may be a limitation for the performance of tracking tasks. Otherwise tracking is only possible with filtered sensor values, smooth references and continuous object contours, or with low sampling rates.

3.2 Specification of the desired motion using an impedance law

Figure 3 shows a simple model. The reference pose $\mathbf{x}_r$ is the pose of the tcp when the object pose is the nominal object pose $\mathbf{x}_n$. With explicit control of sensor data (as in the last section), the desired sensor data ${}^d\mathbf{s} = {}^d\mathbf{x}_o$ are identical to the reference values $\mathbf{s}_r$.

In this section ${}^d\mathbf{s}$ may differ from $\mathbf{s}_r$, since in contrast to explicit sensor-based control, the goal of impedance control is not to reach and keep a specified reference value of sensor data. Instead a weighted compromise between the errors of such reference values and the deviations from the reference values of position, speed, and acceleration is desired. With respect to the reference frame we define

$$ {}^d\mathbf{s} - \mathbf{s}_r = \mathbf{E}_d \cdot {}^r\mathbf{x}_d + \mathbf{D}_d \cdot {}^r\dot{\mathbf{x}}_d + \mathbf{M}_d \cdot {}^r\ddot{\mathbf{x}}_d \qquad (4) $$

where we call $\mathbf{E}_d$, $\mathbf{D}_d$, and $\mathbf{M}_d$ the desired impedance parameters. While $\mathbf{x}_r$ and $\mathbf{s}_r$ are given, (4) and the real object pose $\mathbf{x}_o$ give a trajectory $\mathbf{x}_d$. When this trajectory is executed, desired sensor values ${}^d\mathbf{s}$ will be sensed.

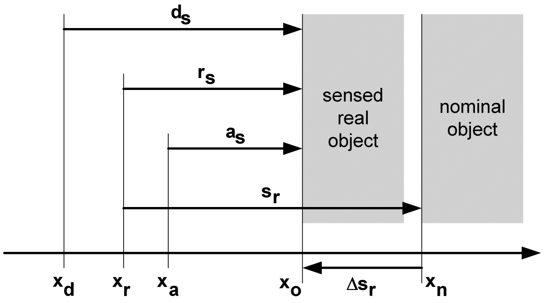

Fig. 3. Robot tool positions and sensor values that measure the distance to a fixed object

The object pose $\mathbf{x}_o$ can be computed from any pair of tcp pose and sensor values (see Figure 3). So with

$$ {}^r\mathbf{x}_o = {}^r\mathbf{x}_d + {}^d\mathbf{s} = {}^r\mathbf{x}_a + {}^a\mathbf{s} \qquad (5) $$

we can replace (4) by

$$ \begin{aligned} (\mathbf{I} + \mathbf{E}_d) \cdot {}^r\mathbf{x}_d + \mathbf{D}_d \cdot {}^r\dot{\mathbf{x}}_d + \mathbf{M}_d \cdot {}^r\ddot{\mathbf{x}}_d \\ = {}^a\mathbf{s} - \mathbf{s}_r + {}^r\mathbf{x}_a = {}^r\mathbf{s} - \mathbf{s}_r = \Delta\mathbf{s}_r \qquad (6) \end{aligned} $$

which is a differential equation to compute ${}^r\mathbf{x}_d$. [2]

3.3 Features of this kind of impedance control

Equation (4) shows substantial differences with respect to the literature of (mechanical) impedance laws since it is characterized by the absence of the actual values $\mathbf{x}_a$ or ${}^a\mathbf{s}$. Instead only references and desired values are used. The actual sensor value is then used only to detect the object position. Thus the impedance law depends only on the configuration. So in contrast to other impedance-based approaches here the desired motion is independent of the actual robot motion. This allows to compute the desired motion in advance, as long as the object shape can be sensed.

Thus the desired motion and finally the actual robot motion is independent of delays or low sampling rates in sensor processing. Therefore, as in Lange and Hirzinger (2005) for explicit sensor-based control, stability is controlled only by the inner positional loop which already has been tuned to be stable by the robot manufacturer.

The right hand side of (6) is the fixed difference between the sensor value that would be sensed at the reference pose and the reference sensor value. The former is constant if the shape of the reference trajectory is parallel to the shape of the object surface (${}^r\mathbf{x}_o = const.$). On the other hand the system reacts compliant if either the reference sensor value changes or the reference trajectory has a component that proceeds normal to the object position or vice versa. A typical example is the initial step of the object position from the nominal to the real position (see Figure 3).

Equation (4) states that with $\mathbf{E}_d = \mathbf{D}_d = \mathbf{M}_d = \mathbf{0}$ deviations from the reference trajectory are not weighted, i.e. explicit sensor-based control is desired.

In contrast, with positive high values of the impedance parameters the sensor values are almost ignored, i.e. control is not sensor-based. Then ${}^r\mathbf{x}_d \approx \mathbf{0}$.

[2] It might be more usual to define $\mathbf{E}_d$ as the factor weighting ${}^r\mathbf{x}_d$, instead of $(\mathbf{I} + \mathbf{E}_d)$.

In a typical application we shall use $\mathbf{E}_d = \mathbf{0}$ to guarantee statically the reference sensor values. $\mathbf{M}_d > \mathbf{0}$ will be used to smooth the desired trajectory in case of noisy sensor data or abrupt changes with respect to the reference values. This may prevent that the limitations of the actuators are reached. Finally $\mathbf{D} > \mathbf{0}$ will be specified to damp oscillations, where in the scalar case $D = 2\sqrt{E \cdot M}$ is a good choice.

3.4 Integration of impedance-based control

By replacing the differentials in (6) by differences of the values at the sampling steps, with the sampling time T_0, we get

$$\begin{aligned}
&(\mathbf{I}+\mathbf{E}_d)\cdot {}^r\mathbf{x}_d(k) \\
&+\mathbf{D}_d \cdot \frac{{}^r\mathbf{x}_d(k+1) - {}^r\mathbf{x}_d(k-1)}{2\cdot T_0} \\
&+\mathbf{M}_d \cdot \frac{{}^r\mathbf{x}_d(k+1) - 2\cdot {}^r\mathbf{x}_d(k) + {}^r\mathbf{x}_d(k-1)}{T_0^2} \\
&\qquad = \Delta\mathbf{s}_r \qquad (7)
\end{aligned}$$

for each sampling step k [3].

With $n_i \approx 2 \cdot n_d$ as prediction horizon for the impedance law, in each time step k we get a tridiagonal linear set of equations for the time steps $k+i = k, \cdots, k+n_i-1$ that might be solved to get ${}^r\mathbf{x}_d(k+i)$. Thus the required data of n_d sampling steps for the computation of the desired path according to Section 2 are available. So control may be ideal and yield the desired motion.

Instead of solving (7), which might be sensitive to noise, we estimate the ${}^r\mathbf{x}_d(k+i)$ thus accounting for noise. To save computing power we use an algorithm for band matrices which is only linear in n_i. ${}^r\mathbf{x}_d(k-1)$ and ${}^r\mathbf{x}_d(k-2)$ are taken from previous estimation steps while ${}^r\mathbf{x}_d(k+n_i)$ and ${}^r\mathbf{x}_d(k+n_i+1)$ are assumed to coincide with explicit control. Strictly speaking, the latter is not correct and may yield non optimal smoothing. However, the effect will be minimized by setting $n_i \gg n_d$ if sensor data allow such a prediction.

4. TRAJECTORY SMOOTHING USING IMPEDANCE-BASED CONTROL

The impedance-based control method described so far works well in many cases, but there is a drawback which sometimes affects the performance: Equation (7) minimizes the weighted sum of acceleration error, velocity error and position error. This means that for example a negative acceleration error may compensate a positive position error. Instead of the weighted sum, we want to minimize all three types of error independently, namely

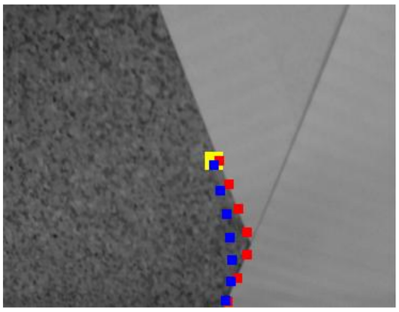

Fig. 4. View of the robot-mounted camera with sensed edge points (red) and smoothed path (blue) computed from the present camera pose (yellow).

$$(\mathbf{I}+\mathbf{E}_d)\cdot {}^r\mathbf{x}_d(k) = \Delta\mathbf{s}_r \qquad (8)$$

$$\mathbf{D}_d \cdot \frac{{}^r\mathbf{x}_d(k+1) - {}^r\mathbf{x}_d(k-1)}{2\cdot T_0} = 0 \qquad (9)$$

$$\mathbf{M}_d \cdot \frac{{}^r\mathbf{x}_d(k+1) - 2\cdot {}^r\mathbf{x}_d(k) + {}^r\mathbf{x}_d(k-1)}{T_0^2} = 0\,. \qquad (10)$$

This means that instead of computing ${}^r\mathbf{x}_d$ by minimizing the error of (7) for all time instants k we now minimize the errors of the three equations (8) to (10) for all time-steps k. This is possible using the same algorithm and the same weights and yields non biased results.

5. EXPERIMENTS

While the experiments of Lange and Hirzinger (2005) are quite smooth and therefore can reach control errors of less than a millimeter with explicit camera-based control, Figure 2 represents a task with edges that can only be tracked after smoothing, since with the given speed of 0.7 m/s,

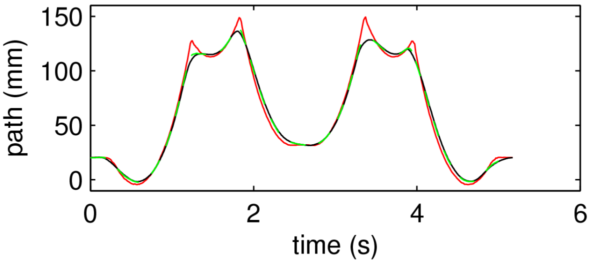

Fig. 5. Plot of deviations from the reference path when tracking the sensed edge (red), back and forth, using impedance-based control according to (7) with $E_d = 0$, $D_d = 0.2$ s, $M_d = 0.01$ s^2. Desired (black) and actual (dashed green) path are almost identical.

[3] We use a KUKA robot with $T_0 = 12$ ms and $n_d = 16$.

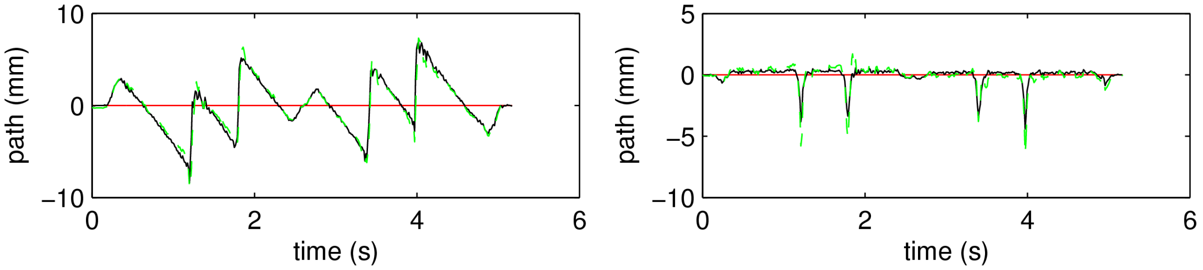

Fig. 6. Plots of distances with respect to the sensed edge, using impedance-based control according to (7) (left) or according to (8) to (10) (right). In both experiments we use $E_d = 0$, $D_d = 0.02$ s, $M_d = 0.0001$ s^2. Desired (black) and actual (dashed green) path are almost identical.

explicit sensor-based control would cause too high accelerations at the vertices.

A typical view of the robot-mounted camera is shown in Figure 4. Seven edge points are sensed and represented by a polynomial. The latter is used to compute the edge still beyond the image border.

Equation (7) yields Figure 5[4]. The really executed path differs from the so computed desired path by only 0.3 mm in the mean. This is reached because excessive accelerations are avoided. In contrast, without the specified impedance law the robot would exceed the maximum joint torques at the first vertex. Figure 5 clearly demonstrates that the path is smoothed in regions where the sensed edge is not smooth.

However, also in regions of small acceleration there are deviations. They can only be avoided using the approach of Section 4. This is demonstrated in Figure 6 which shows a similar experiment. This time the curves are zoomed and displayed with respect to sensed edge. In contrast to the first approach, our second method smoothes only at the vertices. The resulting executed trajectory shows a mean distance to the sensed edge of about a millimeter.

In both cases smoothing is limited by the existing field of view. With a speed of 0.7 m/s, the camera is not more able to sense all n_i future time-steps. Thus the computed path changes during motion and is therefore not optimally tracked. In contrast, execution with 0.4 m/s will be even more accurate.

6. CONCLUSION

The paper discusses that explicit sensor-based tracking of a contour may not be executable at high speed if the contour is not smooth. Therefore two new approaches of impedance-based control have been presented, that consider sensor noise or vertices in the contour. Both methods are implemented using a control architecture that distinguishes between the (predictive) computation of the desired motion and predictive control of the robot position, where the latter is correct if the desired motion is smooth enough. This kind of vision-based impedance-based control is able to determine a trajectory that is smooth enough to be tracked, whereas, at least with the second method, the control error with respect to the edge is small. The amount of smoothing can be specified by the operator.

REFERENCES

D. W. Clarke, C. Mohtadi, and P. S. Tuff. Generalized predictive control - part I. the basic algorithm. *Automatica*, 23(2):137–148, 1987.

G. Ferretti, G. Magnani, and P. Rocco. Impedance control for elastic joints industrial manipulators. *IEEE Trans. on Robotics and Automation*, 20(3):488–498, 2004.

M. Grotjahn and B. Heimann. Model-based feedforward control in industrial robotics. *The International Journal on Robotics Research*, 21 (1):45–60, January 2002.

B. Heinrichs and N. Sepehri. A limitation of position based impedance control in static force regulation: Theory and experiments. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 2165–2170, Detroit, Michigan, May 1999.

F. Lange and G. Hirzinger. Learning of a controller for non-recurring fast movements. *Advanced Robotics*, 10(2):229–244, April 1996.

F. Lange and G. Hirzinger. Stability preserving sensor-based control for robots with positional interface. In *Proc. 2005 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1712–1717, Barcelona, Spain, April 2005.

F. Pertin and J.-M. Bonnet-des-Tuves. Real time robot controller abstraction layer. In *Proc. Int. Symposium on Robots (ISR)*, Paris, France, March 2004.

[4] The reference trajectory is a quarter of a circle back and forth around the robot base. In this case the straight lines of the edges result in a curve since the reference frame is rotated. Only the radial component is controlled by sensor data and impedance-based control.