Joint 41th Int. Symp. on Robotics and 6th German Conf. on Robotics ISR/ROBOTIK 2010, Munich, Germany, June 2010

Control of Large Forces and Torques Using an Asymmetrically Arranged Compliant Sensor

Friedrich Lange, Bertram Willberg, and Gerd Hirzinger
Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Weßling, Germany

Abstract

Control of forces and torques seemed to be solved. But the usual method for position controlled robots is not optimal when using a slow interface. In addition, there are differences between control with a compliant sensor and with a stiff sensor, even more if the sensor is distant from the contact point and with respect to the vector of the desired force. The method proposed in this paper further considers a point-surface contact, from which the sensor cannot infer on rotation. We also consider the automatic detection of this case. A further obstacle to force control with compliant sensors is the fact that only small scale sensors are available. Therefore a method for the scaling of existing sensors and an example of its use are presented.

1 Introduction

Force control has been used for almost three decades, but there are still some aspects that have to be discussed, especially with position-based robot controllers and with compliant sensors.

There are two main approaches to force control. Most papers, as [1], opt for controlling the joint torques of the robot. However, this is not possible with standard industrial robots. For these systems, inner loop / outer loop control as in [2] is applied, with position control as inner loop and force control as outer loop. Most robot controllers for the outer loop use an interface [3], which is sampled every 10 ms, or even less often. This limits performance unless a special approach, presented in Section 2, is used.

Another classification of force control schemes distinguishes between impedance control [4] and explicit force control [5], where the latter minimizes force and torque errors independently from position errors which may be controlled in a different subspace. See [6] for a more comprehensive discussion on force control since this goes beyond the scope of this paper. Here we concentrate on explicit force control and adapt the desired forces if compliance is desired (see Section 2.2).

Force control can be also classified by the type of sensor. While the majority of publications assumes a stiff force-torque sensor with negligible positional displacement, certain assembly tasks may require such a displacement, especially in dynamic environments [7]. A compliant sensor [8, 9] will yield passively when forces or torques are applied. In contrast to active force control by a controller, passive compliance possesses almost infinite bandwidth, which is only limited by the mass of the displaced body. Thus control is only required to reduce low bandwidth or steady-state force control errors.

This paper concerns compliant sensors. In contrast to forces acting on stiff sensors, the deflection of a compliant sensor affects the pose of the tool center point (tcp). This effect has to be compensated by the control algorithm (Section 4).

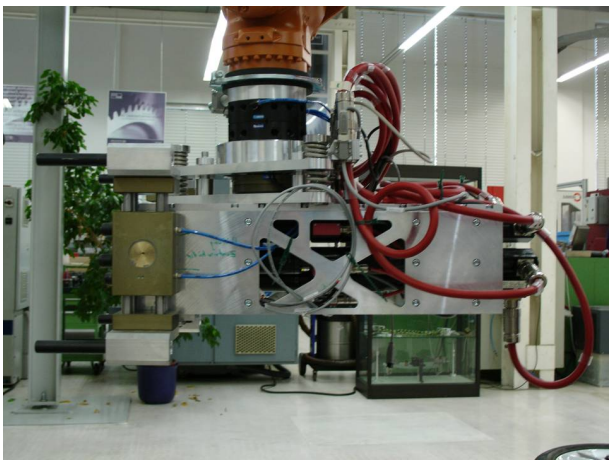

Figure 1: Setup at *iwb* with heavy end-effector for the assembly of a wheel to a car body. Between the springs a sensor is integrated that perceives forces and torques that are exerted by the robot.

Since a force-torque sensor has to measure all contact forces that act on the end-effector, the sensor is typically located between the robot tool and the robot flange. **Figure 1** gives an example in which the end-effector is mounted under the robot flange, which is advantageous for heavy tools. In this way the sensor has only to bear the weight of the end-effector. In other configurations, the weight also exerts a big moment. Since compliant force-torque sensors are only available for small scale applications, this paper also presents a method to design a com-

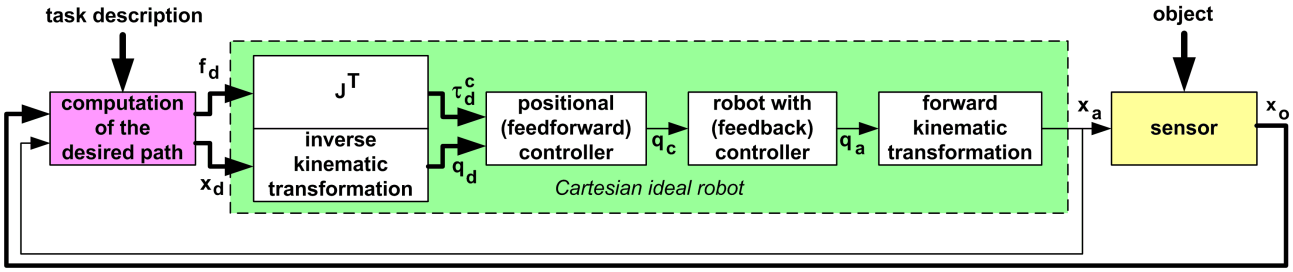

Figure 2: Control architecture for position-based control of sensor data.

pliant sensor for large forces and torques using standard components (Section 5).

Force control is studied using the task of [10] in which wheels are assembled to a car body. This is done using an end-effector (Figure 1) with a compliant sensor at the robot flange. The contact is modeled as a point-surface contact every time the screws reach the wheel hub, and as a peg-in-hole problem when the wheel rim is in contact with the wheel hub and the screws are fixed. Then all 6 degrees of freedom (dof) of the contact are present and measured by the sensor. In the first phase, a force at the tcp will additionally generate a moment in the sensor. Thus the control of this moment will cause a rotation, which is not wanted. Section 4 explains how this can be inhibited. It includes the distinction between moments that are measured in the first phase, and those that have to be controlled because they are caused by the assembly process itself.

The paper is organized as follows: First, the fundamental control architecture for the consideration of sensor signals is outlined (Section 2). Then it is applied to force control when using a compliant sensor (Section 3). The handicap of an asymmetrically arranged sensor is considered in Section 4. Section 5 finally shows how large-scale sensors can be built. The method is verified in experiments with the wheel assembly robot of [10].

2 Position-based control of sensor data

Sensor-based control can be realized independently of a specific task or sensor set-up if sensors are used to determine the desired robot motion, while the motion control is done by a lower control level. This is different to direct sensor control, where the signals directly generate motion increments. However, the task specific part and the robot dependent position control are only independent of each other if the interface is chosen in such a way that the controller characteristics give no feedback to the task level. This requires that the desired motion is executed in an ideal way. **Figure 2** shows the proposed control architecture. An ideal robot is defined by $\mathbf{x}_a(k) = \mathbf{x}_d(k)$, meaning that the actual robot pose is identical to the desired one.

An ideal robot cannot be implemented if only the desired pose is given. In addition, the desired speed and acceleration has to be provided or, as used in this paper, a part of the robot trajectory that covers the period beginning at the current time step and lasting for some robot time constants. In this way the interface becomes predictive. See [10] for a possible implementation of the position control.

Since the stability of position control is not affected by a feed-forward controller, the ideal robot is stable as long as the feedback controller provided by the robot manufacturer remains stable. In this way the whole system is stable as long as the computation of the desired pose really computes a *desired* pose, and does not feed back the *actual* robot motion. This will discussed in Section 3.

In the sample project, as in all complex assembly tasks, several types of sensors are used [11]. Besides a force-torque sensor there are a distance sensor and vision. In order to use a common interface for all sensors, the selection of a predictive *position* interface is generic. Furthermore, since at least the outcome of the vision system is received time-delayed, a position *trajectory* is well suited for integrating modifications that are result of sensor data. In contrast, asynchronously appearing sensor data can hardly be processed in direct control approaches.

The idea of computing a desired pose is advantageous with respect to direct feedback of sensor values to motion increments, because the sensor interface is usually sampled less often than the internal position controllers. In the proposed approach this does not affect performance since the computed desired poses are invariant to that changes of the sensed values, which are caused by the robot motion.

Since sensors typically represent the current *state*, the required *trajectory* has to be generated in a model based way. For stationary objects this will be done by maintaining the desired pose. With moving objects or contour following we propose extrapolation, preferably of the deviation of the computed desired pose from the reference pose.

2.1 Force control

A force-torque sensor is a special sensor: Firstly, position is not its natural output. Secondly, forces and torques can affect the robot motion. The first aspect is discussed in Section 3 using a compliant sensor. The effect of the interacting forces on the position control can be neglected since typical desired contact forces are at least an order of magnitude lower than the robot weight or its motor torques.

If significant forces or torques are present, the additional input $\mathbf{f}_d$ to the ideal robot in Figure 2 has to be used. $\mathbf{f}_d$ is the desired Cartesian force. τ_d^c is that part of the desired joint torque, that is caused by this contact force.[1] These inputs serve for disturbance rejection and by no means as a reference for control. $\mathbf{J}$ is the Jacobian of the robot.

2.2 Impedance-based control

If, instead of explicit force control, a compromise between a force error and a deviation from the programmed path is desired, the desired values $\mathbf{x}_d$ and $\mathbf{f}_d$ have to be adapted. This can be done using an impedance-based approach in which the trajectory is modified by minimizing

$$\mathbf{K} \cdot {}^r\mathbf{x}_d(k+i) = {}^r\mathbf{x}_e(k+i) \tag{1}$$

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

$$\frac{\mathbf{M}}{T_0^2} \cdot ({}^r\mathbf{x}_d(k+i+1) - 2{}^r\mathbf{x}_d(k+i) + {}^r\mathbf{x}_d(k+i-1)) = 0. \tag{3}$$

Here all poses are expressed with respect to the reference system ^{r}, which is the system of the current programmed pose of the tool center point (tcp). ${}^r\mathbf{x}_d(k)$ is the resulting current modification with respect to the reference path, computed from the modification ${}^r\mathbf{x}_e(k)$ which would be active for explicit force control, and of the impedance parameters $\mathbf{K}$, $\mathbf{D}$, and $\mathbf{M}$. T_0 is the sampling time. With $\mathbf{K} = \mathbf{I}$, in the static case, there is no bias of the desired force. But at times, the desired forces may differ from the original value, in order to smooth the trajectory. ${}^r\mathbf{x}_d(k)$ is computed by minimizing the equation errors for $i = 0, \cdots, n_{imp}$ where n_{imp} is sufficiently large. The desired force $\mathbf{f}_d(k)$ will be computed from ${}^r\mathbf{x}_d(k)$, the sensor stiffness, and the force present in the nominal case. See [12] for details.

3 Control using a compliant force-torque sensor

A compliant force-torque sensor is better suited for assembly tasks than a stiff one since, firstly, it instantaneously yields to disturbances, without generating extreme forces that may result with a stiff sensor. Secondly, it creates position differences that can be processed by a position controller. In contrast to stiff sensors, compliant ones measure translational and rotational deflections of the sensor and transform them to forces and torques, using the calibrated sensor stiffnesses. At least using a SCHUNK® sensor [13], it is possible to read the sensor displacements instead of the forces. This directly allows position control. In this way a desired deflection $\mathbf{s}_d$ is given, instead of a desired force $\mathbf{f}_d$.

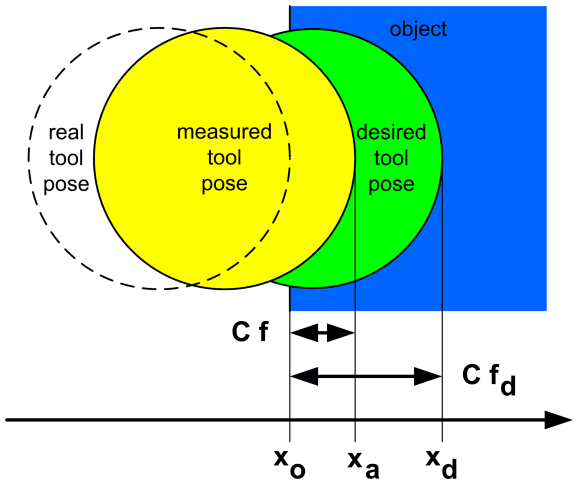

Figure 3: Notation when pressing with a pin against an object.

Figure 3 visualizes the effect of sensor compliance when the tcp is in point-surface contact. The real position is clearly identical with the object position $\mathbf{x}_o$. This means that the size of the tool tip is already considered by the measured position. Alternatively the object position could be shifted by the radius of the tool tip.
The forward kinematics do not compute the real but rather a virtual position $\mathbf{x}_a$, which would result in free space. Both positions differ by the sensor deflection, transformed from the sensor to the tcp. The goal of control is another virtual position $\mathbf{x}_d$ that, assuming compliance $\mathbf{C}$, yields the desired force $\mathbf{f}_d$.
The assumed compliance $\mathbf{C}$ results from the environmental compliance, the robot compliance, and the sensor compliance, and is computed as the sum of them. In most cases, using a compliant sensor, its known compliance will be dominant so that it can be taken as the resulting compliance. In this way the generated forces could be slightly too small, but the stability is not affected by this assumption. In contrast, when processing computed forces, errors in the assumed compliance may feed back the robot motion to the desired poses, so that the separation of position control is not valid anymore.
With full contact, all 6 dof of the real tcp pose as well as of the object pose are identical. Then virtual positions and orientations can be defined in a similar way from forces $\mathbf{f}_t$ and moments $\mathbf{m}_t$, at the tcp.[2]

3.1 Computation of the desired pose

For the computation of the desired pose we refer to the notation of homogeneous transformation matrices $\mathbf{T}$, which

[1] Unless otherwise stated, within this paper the actual and desired poses $\mathbf{x}$ and forces $\mathbf{f}$ are vectors with 6 elements, thus comprising translation and rotation or forces and torques, respectively.

[2] Forces and moments, related to the tcp or to the sensor, represent only 3 dof (in contrast to the definition in Section 2).

consist of a 3×1 position vector $\mathbf{p}$ and a 3×3 rotation matrix $\mathbf{R}$.

$$\mathbf{T} = \begin{bmatrix} & \mathbf{R} & & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The desired pose is computed from the pose of the object $\mathbf{T}_o$ and the desired offset ${}^o\mathbf{T}_d$ from it. Here, $\mathbf{T}_o$ is the contact pose, the pose of the tcp when both are in contact. ${}^o\mathbf{T}_d$ corresponds to the desired contact force. The displacements may be computed from the desired forces $\mathbf{f}_{td}$ and torques $\mathbf{m}_{td}$ by

$$\begin{aligned} {}^o\mathbf{p}_d &= \mathbf{C}_p \cdot \mathbf{f}_{td} && (5) \\ {}^o\boldsymbol{\varphi}_d &= \mathbf{C}_\varphi \cdot \mathbf{m}_{td}, && (6) \end{aligned}$$

where $\mathbf{C}_p$ and $\mathbf{C}_\varphi$ are the translational and rotational compliances of the sensor, and ${}^o\boldsymbol{\varphi}_d$ is the rotation vector that represents ${}^o\mathbf{R}_d$. This yields

$$\mathbf{T}_d = \begin{cases} \mathbf{T}_o \cdot {}^o\mathbf{T}_d & \text{sensed components} \\ \mathbf{T}_r & \text{other components} \end{cases} \tag{7}$$

The upper row is only valid if all components can be sensed. Non-measurable dof have to be replaced by a programmed reference pose $\mathbf{T}_r$. For point-surface contact with (non measurable) orientation ${}^o\mathbf{R}_d = \mathbf{I}$, equation (7) will be replaced by

$$\begin{aligned} \mathbf{p}_d &= \mathbf{p}_o + \mathbf{R}_d \cdot {}^o\mathbf{p}_d && (8) \\ \mathbf{R}_d &= \mathbf{R}_r. && (9) \end{aligned}$$

The object pose $\mathbf{T}_o$ is computed from the actual pose of the tcp $\mathbf{T}_a$, which is computed from the robot forward kinematics and from the sensed distance ${}^a\mathbf{T}_o$, which is represented by the displacement at the tcp:

$$\mathbf{T}_o = \mathbf{T}_a \cdot {}^a\mathbf{T}_o \tag{10}$$

${}^a\mathbf{T}_o$ is denoted as ${}^a\mathbf{T}_{a'}$ where the index $_{a'}$ stand for the displaced tcp. This displacement is caused by the measured deflection of the sensor ${}^s\mathbf{T}_{s'}$. Thus

$$\begin{aligned} {}^a\mathbf{T}_o &= {}^a\mathbf{T}_s \cdot {}^s\mathbf{T}_{s'} \cdot {}^{s'}\mathbf{T}_{a'} \\ &= {}^s\mathbf{T}_a^{-1} \cdot {}^s\mathbf{T}_{s'} \cdot {}^s\mathbf{T}_a && (11) \end{aligned}$$

where ${}^s\mathbf{T}_a$ is the transformation from the sensor to the tcp. For point-surface contact with ${}^s\mathbf{R}_a = \mathbf{I}$ this yields

$$\begin{aligned} {}^a\mathbf{p}_o &= -{}^s\mathbf{p}_a + {}^s\mathbf{p}_{s'} + {}^s\mathbf{R}_{s'} \cdot {}^s\mathbf{p}_a \\ &= {}^s\mathbf{p}_{s'} + ({}^s\mathbf{R}_{s'} - \mathbf{I}) \cdot {}^s\mathbf{p}_a && (12) \\ {}^a\mathbf{R}_o &= {}^s\mathbf{R}_{s'}. && (13) \end{aligned}$$

We know that for a point-surface contact ${}^a\mathbf{R}_o$ is not defined by the measurements and therefore we set it to

$${}^a\mathbf{R}_o = \mathbf{R}_a^{-1} \cdot \mathbf{R}_r. \tag{14}$$

Then, with (7), (10), (12), (5), and $\mathbf{m}_{td} = 0$,

$$\begin{aligned} \mathbf{p}_d &= \mathbf{p}_a + \mathbf{R}_a \cdot {}^s\mathbf{p}_{s'} + \mathbf{R}_a \cdot ({}^s\mathbf{R}_{s'} - \mathbf{I}) \cdot {}^s\mathbf{p}_a \\ &\quad + \mathbf{R}_r \cdot \mathbf{C}_p \cdot \mathbf{f}_{td} && (15) \\ \mathbf{R}_d &= \mathbf{R}_r. && (16) \end{aligned}$$

4 Compensation of the motion generated by an asymmetrical arrangement

If all components are sensed, the upper row of (7), (10), and (11) completely define the desired motion. Otherwise these equation can yield a drift if noise in ${}^s\mathbf{R}_{s'}$, or in a component of ${}^s\mathbf{p}_{s'}$ that is parallel to the surface, have a bias. Therefore the point-surface contact has to be detected (Section 4.1).

In addition, if there is a point-surface contact, the non-measurable dof yield arbitrarily wrong sensor values that depend on the sensor characteristics. For example, with the configuration of **Figure 4**, from the experimental setup (see Figure 1), when pressing in z direction the deflection at the sensor is not restricted to this component, but a rotational deflection around y (β) is measured as well (see **Figure 5**). This is true because $\mathbf{d} = {}^s\mathbf{p}_a$ has a component in x.

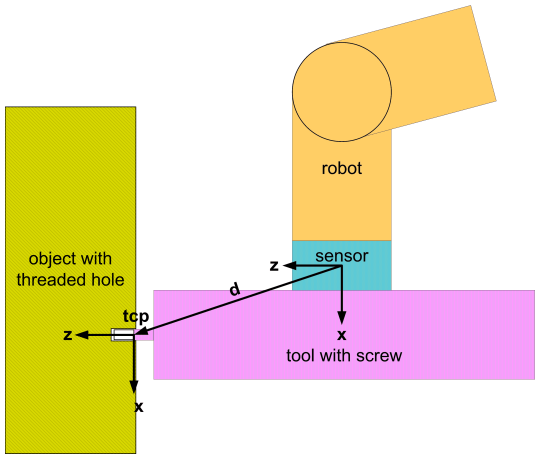

Figure 4: Setup with heavy end-effector that has to insert a screw or a pin.

In this way, the displacement at the tcp consists of the component in z direction, which will be controlled, and of a rotational deflection in β, which causes further translational displacement in x. These are disturbances that are caused by the sensor characteristics (e.g. the end-effector dimensions). If these disturbances would be used for control in the case of point-surface contact, the end-effector would permanently rotate in β and, with small friction, also drift in x.

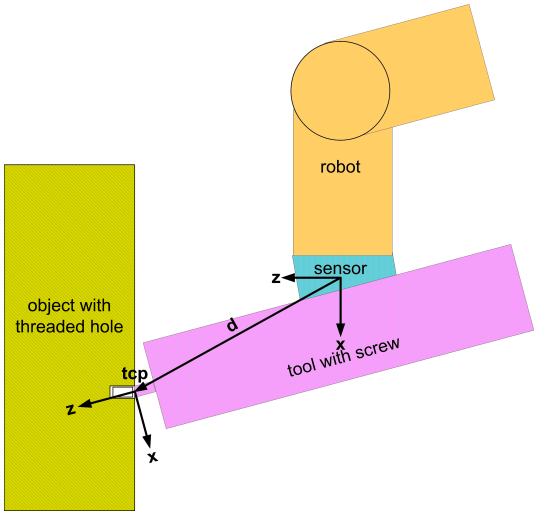

Figure 5: Real posture when pressing in point-surface contact.

4.1 Detection of point-surface contact

The detection of a point-surface contact depends on two conditions. A necessary condition is $\mathbf{m}_t = 0$ but it may be also true in the goal pose. Therefore a second condition is required, which has to consider that $\mathbf{m}_t$ will not vary during rotations with point-surface contact.
With the wrench equations[3]

$$\mathbf{f}_{sd} = \mathbf{f}_{td} \quad (17)$$
$$\mathbf{m}_{sd} = \mathbf{m}_{td} + \mathbf{d} \times \mathbf{f}_{td} \quad (18)$$

and uncoupled compliances at the sensor

$${}^s\mathbf{p}_{s'} = \mathbf{C}_p \cdot \mathbf{f}_s \quad (19)$$
$${}^s\boldsymbol{\varphi}_{s'} = \mathbf{C}_\varphi \cdot \mathbf{m}_s, \quad (20)$$

the first condition gives

$${}^s\mathbf{p}_{s'} = \mathbf{C}_p \cdot \mathbf{f}_{td} \quad (21)$$
$${}^s\boldsymbol{\varphi}_{s'} = \mathbf{C}_\varphi \cdot (\mathbf{d} \times \mathbf{f}_{td}). \quad (22)$$

This can be transformed to tcp displacements where, for convenience, the twist equations

$${}^a\mathbf{p}_{a'} = {}^s\mathbf{p}_{s'} - \mathbf{d} \times {}^s\mathbf{R}_{s'} \quad (23)$$
$${}^a\boldsymbol{\varphi}_{a'} = {}^s\boldsymbol{\varphi}_{s'} \quad (24)$$

are used.[4]
With $d_y = {}^s y_a = 0$ and $\mathbf{C}_s = diag(c_x, c_y, c_z, c_\alpha, c_\beta, c_\gamma)$, the MATLAB® symbolic toolbox yields

$${}^a\alpha_{a'} = \frac{d_x \cdot c_\gamma \cdot {}^a y_{a'}}{c_y + d_x \cdot d_x \cdot c_\gamma + d_z \cdot d_z \cdot c_\alpha} \quad (25)$$
$${}^a\beta_{a'} = \frac{c_\beta \cdot c_z \cdot d_z \cdot {}^a x_{a'} - c_\beta \cdot c_x \cdot d_x \cdot {}^a z_{a'}}{c_x \cdot c_z + d_z \cdot d_z \cdot c_\beta \cdot c_z + d_x \cdot d_x \cdot c_\beta \cdot c_x} \quad (26)$$
$${}^a\gamma_{a'} = \frac{-d_z \cdot c_\alpha \cdot {}^a y_{a'}}{c_y + d_x \cdot d_x \cdot c_\gamma + d_z \cdot d_z \cdot c_\alpha}. \quad (27)$$

Point-surface contact may be active if these conditions apply.
The second condition is implemented by computing the rotational differences with respect to the previously determined object (surface) orientation, which is estimated using a Kalman filter. The covariance matrix gives information about the uncertainty of the estimations. In the absence of a torque at the tcp, a point-surface contact is assumed if either the estimation of the object orientation is uncertain or its difference with respect to the tcp orientation is big. The first case is here implemented since usually contact begins at a single point. The second criterion distinguishes from the case with a small torque that may be present in the goal position (with 6 dof contact). Without this criterion, an oscillation would occur since the reference orientation would be assumed as desired until a substantial torque would be measured.
Finally, a factor $0 <= \delta <= 1$ is computed; $\delta = 1$ represents point-surface contact, i.e., uncertainty or orientational deviation.

4.2 Inhibition of unwanted rotational evasive motion

The decision whether a point-surface contact is present or not is not a binary question. Because of noise, a region φ_1 around the values of (25) to (27) will be partially treated as point-surface contact. This is done by

$$\boldsymbol{\varphi}_d := \begin{cases} \boldsymbol{\varphi}_r + (\boldsymbol{\varphi}_d - \boldsymbol{\varphi}_r) \cdot (1 - \delta \cdot (1 - \frac{|{}^a\boldsymbol{\varphi}_{a'} - \boldsymbol{\varphi}_0|}{\varphi_1})) \\ \qquad \text{for } |{}^a\boldsymbol{\varphi}_{a'} - \boldsymbol{\varphi}_0| < \varphi_1 \\ \boldsymbol{\varphi}_d \qquad \text{else} \end{cases} \quad (28)$$

where $\boldsymbol{\varphi}_0$ is the orientational displacement of (25) to (27) and ${}^a\boldsymbol{\varphi}_{a'}$ is the measured orientational displacement at the tcp. $\boldsymbol{\varphi}_d$ is the desired orientation, which is revised in the case of point-surface contact.

4.3 Inhibition of unwanted translational evasive motion

A translational drift may be inhibited in a similar way. This appears with point-surface contact and small friction. Then

[3] The following equations implicitly use ${}^s\mathbf{R}_a = \mathbf{I}$, i.e. parallel coordinate systems for the sensor and the tcp.
[4] The twist equations are an approximation of the equation of the homogeneous transformations (11).

the lateral forces f_x and f_y at the tcp are zero. This is detected by ${}^s x_{s'} = 0$ and ${}^s y_{s'} = 0$. A translational drift, e.g. in x, will be inhibited by

$$x_d := \begin{cases} x_r + (x_d - x_r) \cdot (1 - \delta \cdot (1 - \frac{\sqrt{{}^s x_{s'}{}^2 + {}^s y_{s'}{}^2}}{x_1})) \\ \qquad \text{for} \sqrt{{}^s x_{s'}{}^2 + {}^s y_{s'}{}^2} < x_1 \\ x_d \qquad \text{else} \end{cases} \tag{29}$$

4.4 Ensuring the desired orientation for assembly

Sections 4.2 and 4.3 inhibit drifts with respect to components that are not provided by the sensor in the case of point-surface contact. However, the rotational deflection, as in Figure 5, remains. Instead, a pose like in **Figure 6** is desired.

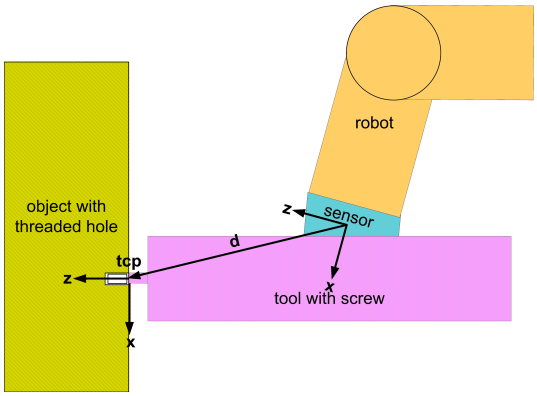

Figure 6: Desired posture with nominal orientation of the screw.

This can be reached by using

$$\varphi_d := \begin{cases} \varphi_r + (\varphi_d - \varphi_r) \cdot (1 - \delta \cdot (1 - \frac{|{}^a\varphi_{a'} - \varphi_0|}{\varphi_1})) \\ \quad -{}^a\varphi_{a'} \cdot \delta \cdot (1 - \frac{|{}^a\varphi_{a'} - \varphi_0|}{\varphi_1}) \\ \qquad \text{for} |{}^a\varphi_{a'} - \varphi_0| < \varphi_1 \\ \varphi_d \qquad \text{else} \end{cases} \tag{30}$$

instead of (28), since in point-surface contact this gives an additional rotation by the rotational deflection ${}^a\varphi_{a'}$. (29) is not affected by the orientation and is therefore not modified.

5 Design of a large scale compliant force-torque sensor

A compliant sensor [9] is available from SCHUNK®. It is implemented by springs between the two flanges, and the deflection of the springs is measured. Unfortunately the biggest sensor [13] is only designed for forces and torques up to 300 N and 7 Nm (15 Nm around the main axis x).[5] Especially the torque range is not sufficient for set-ups with a long end-effector. Therefore, for large scale applications, three external springs are suggested (see Figure 1). Then the measurement units can be used unchanged. The drawback of this design is an oversize construction, which is tolerable in the case of heavy loads. **Figure 7** shows a cross-section.

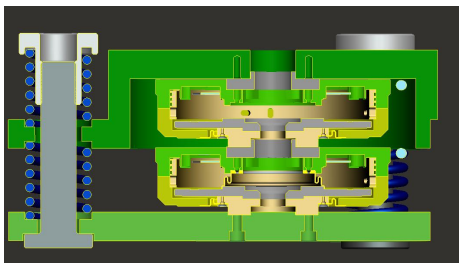

Figure 7: Design of a large scale sensor from two small scale sensors and three external springs.

In this design not only forces and torques are scaled, but also the range of measurable displacements is twice as big as the original ± 1 mm and $\pm 1^o$ of the measurement units. The 2×3 springs are chosen to have a stiffness s_x, so that 1/6 of the maximum force is reached with the maximum deflection of 2 mm, since

$$f_x = 6 \cdot s_x \cdot {}^s x_{s'}. \tag{31}$$

Furthermore, a sufficient length for the springs has to be provided, such that rotational deflections are also possible, and such that the end-effector weight can be compensated by pre-stressing the springs. The pre-stressing is possible since the axes, as shown at the left-hand side of Figure 7, are implemented as threaded studs with screw nuts. This is a crucial difference with respect to the original sensor.
Besides, the lateral stiffness of the springs s_{yz} has to be considered:

$$f_y = 6 \cdot s_{yz} \cdot {}^s y_{s'} \tag{32}$$
$$f_z = 6 \cdot s_{yz} \cdot {}^s z_{s'} \tag{33}$$

The distances r of the springs from the center of the sensor give the tolerable moments:

$$m_x = 6 \cdot r^2 \cdot s_{yz} \cdot {}^s\alpha_{s'} \tag{34}$$
$$m_y = (2 + 4 \cdot \cos^2(60^o)) \cdot r^2 \cdot s_x \cdot {}^s\beta_{s'} \tag{35}$$
$$m_z = 4 \cdot \sin^2(60^o) \cdot r^2 \cdot s_x \cdot {}^s\gamma_{s'} \tag{36}$$

Since the factors in (35) and (36) are equal, the stiffness in the yz-plane is independent of orientation.
Strictly speaking, the stiffnesses of the springs that are integrated in the original sensors also have to be considered - usually they are much smaller. For the experiments, a sensor with integrated elastomer springs is used. This provides some positive damping characteristics but also shows substantial hysteresis.

[5] This section adopts the notation of the preceding figures, though originally the sensor's main axis is denoted by z.

6 Experiments

The experiments are executed with the set-up of **Figure 8**, where wheels are assembled to a car. In contrast to the assumption of Section 3 the car may yield to contact forces. Therefore the equations of motion differ from the equations that exclusively assume linear compliance of the sensor. In addition, instead of the single screw as in the figures, the real end-effector shows five power screw drivers, such that excessive rotation may cause a torque at the tcp. In all test runs the screws were present in the end-effector, but not actuated (screwing). A wheel is not used because else there is not a point-surface contact when the wheel rim touches the wheel hub. The robot control includes an oscillation damping algorithm that is not shown in Figure 2. It may cause a suboptimal transient behavior.

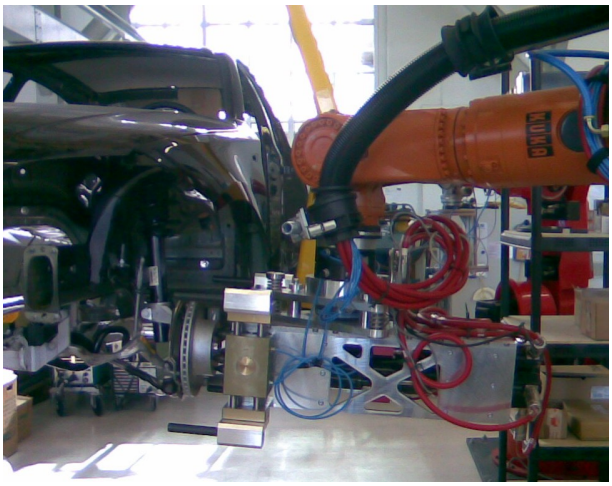

Figure 8: Experimental set-up at *iwb* for assembly using a KUKA robot.

There are two types of experiments. In the first type the robot is pressing against the wheel hub, missing the threaded holes. Thus there is a real point-surface contact. In the other case the holes are met. Then, even before screwing, lateral forces can be exerted. Nevertheless the steering angle and the camber of the wheel hub are not measurable.

Figures 9 and 10 show that the translational and rotational drifts that were predicted in Section 4 are only compensated when using (28) and (29). There is a small displacement with respect to the reference since the sensed orientational deflection slightly differs from the theoretical one. Nevertheless (30) aims at the correct real orientation for assembly, which is the sum of the desired orientation and the orientational deflection of about 1 mrad.

In **Figures 11 and 12** the threaded holes are met. Therefore the drift in x is limited by the backlash within the screw drivers and the mount of the screws. m_y is partially compensated by f_x, especially without the compensation according to Section 4.3, so that the drift in β is limited as well.

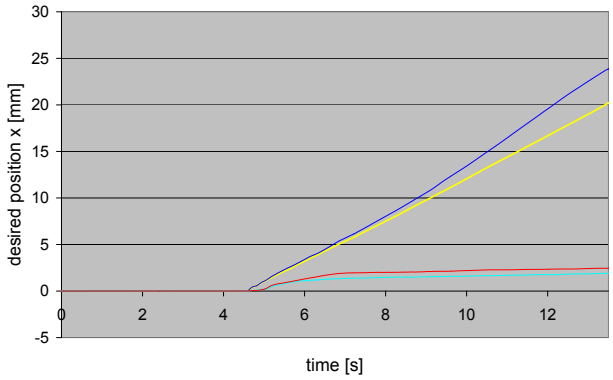

Figure 9: Vertical component (x) of the resulting desired position in point-surface contact without lateral forces (without compensation, with compensation according to Section 4.2, with compensation according to Section 4.3, with compensation according to Section 4.4).

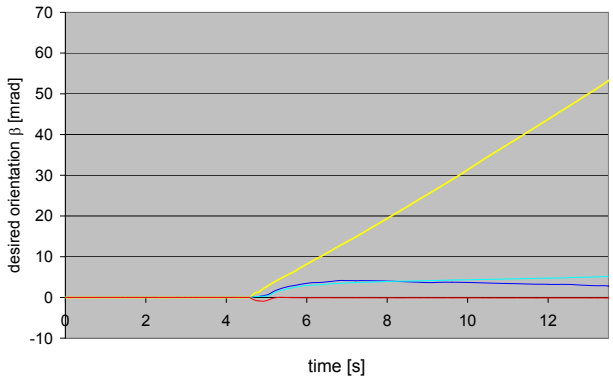

Figure 10: Tilting angle (β) of the resulting desired pose in point-surface contact without lateral forces (curves as with Figure 9).

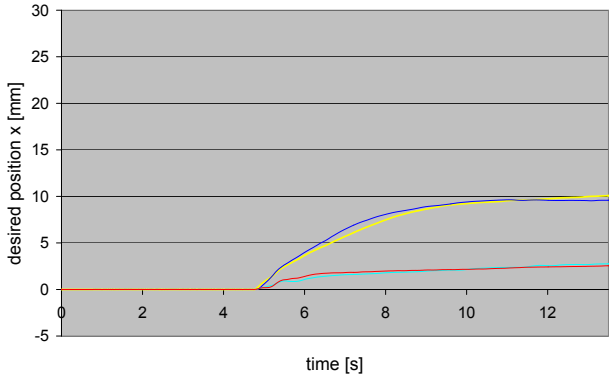

Figure 11: Vertical component (x) of the resulting desired position in point-surface contact with the screws fixed by the form fit (curves as with Figure 9).

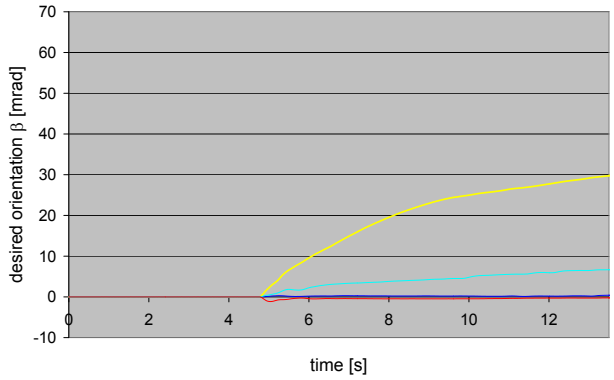

Figure 12: Tilting angle (β) of the resulting desired pose in point-surface contact with the screws fixed by the form fit (curves as with Figure 9).

Other experiments, not displayed here, show that when assembling the wheel the displacements are controlled in all dof, including the camber of the wheel.

7 Conclusion

The paper presents an approach to force control of position controlled robots, thus typical industrial robots. The approach is straightforward when using a compliant sensor.
With an asymmetrically arranged sensor, similar to other control approaches, control of all dof yields unwanted evasive movements in the case of a point-surface contact. The compensation of these movements includes the detection of the point-surface contact, since the control of torques is desired when the tcp is in full contact. By using the proposed method this is reached and inhibits any drift of the pose when pressing against a surface.
Future work will improve the force control and the position control in such a way that the desired forces and torques are accurately reached and are held constant, irrespective of the arrangement and the type of contact.

Acknowledgment

This work is funded by the KUKA Roboter GmbH, Augsburg, Germany. The experiments have been executed at the Application Center Augsburg of the Institute for Machine Tools and Industrial Management (*iwb*) of the Technical University of Munich. The authors would like to thank the head of this institute for sponsorship and Johannes Scharrer for his support.

References

[1] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, RA-3, 1987.

[2] J. De Schutter and H. Van Brussel. Compliant robot motion II. A control approach based on external control loops. *The Int. Journal of Robotics Research*, 7(4):18–33, 1988.

[3] 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] N. Hogan. Impedance control: An approach to manipulation: Part I-theory; part II-implementation; part III applications. *ASME J. Dynam. Sys. Measurement Control*, 107:1–324, 1985.

[5] R. Volpe and P. Khosla. A theoretical and experimental investigation of explicit force control, strategies for manipulators. *IEEE Trans. on Automatic Control*, 38(11), Nov. 1993.

[6] B. Siciliano and O. Khatib, editors. *Handbook of Robotics*. Springer, 2008.

[7] G. Reinhart, J. Werner, and F. Lange. Robot based system for the automation of flow assembly lines. *Production Enginering Research and Development*, 3(1):121–126, March 2009.

[8] D. E. Whitney and J. L. Nevins. What is the remote centre compliance (rcc) and what can it do? In *Proc. of the 9th Int. Symposium on Industrial Robots*, pages 135–152, 1979.

[9] J. Grewe, G. Hirzinger, R. Koeppe, C. Strobl, and B. Willberg. Compliant-force-torque-sensor. In *Proc. 10th Int. Workshop on Robotics in Alpe-Adria-Danube Region*, Vienna, Austria, May 2001.

[10] F. Lange, J. Scharrer, and G. Hirzinger. Is a linear axis really required when assembling parts to a moving conveyor. In *Proc. Joint 41th Int. Symp. on Robotics and 6th German Conf. on Robotics ISR/ROBOTIK 2010*, Munich, Germany, June 2010.

[11] G. Reinhart and J. Werner. Flexible automation for the assembly in motion. *Annals of the CIRP*, 56(1):25–28, 2007.

[12] F. Lange, M. Frommberger, and G. Hirzinger. Impedance-based smoothing for visual servoing along edges. In *Proc. Joint Conference on Robotics ISR 2006 / ROBOTIK 2006*, Munich, Germany, May 2006.

[13] Schunk GmbH & Co KG. *SCHUNK - Workholding Solutions, Automation Components, Toolholding Components, Gripping Systems, Linear Systems*. Lauffen, Germany, 2010. http://www.schunk.com/schunk/schunk_websites/products/products_level_3/product_level3.html?country= INT&lngCode=EN&lngCode2=EN&product_level_1=244&product_level_2=252&product_level_3=296.