Proc. Int. Conf. on Robotics and Automation (ICRA) Anchorage, AK, May 2010

Assembling Wheels to Continuously Conveyed Car Bodies Using a Standard Industrial Robot

Friedrich Lange, Jochen Werner, Johannes Scharrer, and Gerd Hirzinger

Abstract— **Within assembly lines, wheel assembly to continuously conveyed car bodies is still executed by human workers using a device that compensates the weight of the wheel. This paper presents a solution in which a robot autonomously assembles and fixes the wheels. The approach uses a sensor-driven control strategy that compensates a possible temporal or spatial offset. Three types of sensors are proposed for adequate perception of the wheel hub. Their signals are fused by a Kalman filter that allows predictions in the time domain. Finally, a feed-forward controller is used, that is designed to consider the predictions in order to minimize dynamical delays. The control is driven by a special task description that extents usual robot programming methods.**

I. Introduction

Automated robot-based final assembly has not become fully accepted in car manufacturing, in contrast to, e.g., car body construction. Wheel assembly, e.g., is still performed predominantly manually, while the car body is continuously[1] moved by a conveyor. Only the assembly of bulky parts which can be hardly handled by humans is executed by robots. In that case, the car body has to be stopped for assembly, and accelerated thereafter. This realization consumes time and space, thus it is more expensive. This automated assembly is used for the adhesive bonding of the windshield, the integration of the cockpit, or the insertion of a retractable roof.

The reasons for the exclusion of robots in final assembly originate from arbitrary disturbances of the conveyor system. Therefore the standard approach to explicitly program the nominal robot path, and then to control the robot to follow this path with high accuracy, cannot be used.

Instead, at a conveyor, both the motion of the conveyed object and the actual robot trajectory have to be sensed. The perception of the object trajectory is fundamental since the object is not fixed within immobile clamps. It requires different types of sensors which are addressed in relation to the assembly state. A single tracking of the conveyor is not sufficient because of disturbances that may affect all 6 degrees of freedom (dof) [1]. For instance a bucking conveyor will not only impair the translation in the conveying direction, but it may also excite rotational oscillations with respect to the suspension of the object within the conveyor.

It is not possible to compensate deviations from the desired trajectory by off-line shifting positions in a programmed path. Instead, the robot path has to be modified in real time in accordance to the sensors' readings. These deviations are always present with typical robot controllers that are designed for low positioning errors instead of low path errors, i. e. small deviations from the desired trajectory. Thus the actual robot path has to be perceived on-line as well, in order to be able to comply with very small assembly tolerances and to keep contact forces small.

The existing approaches to automated assembly in motion can be differentiated into mechanical, guided or controlled synchronization principles [1]. So far, all synchronized robot systems applied lack of accuracy compared to stationary assembly. Furthermore, economic application in industry is additionally hindered by the complexity and the costs of the system.

The mounting of wheels, is typically done by humans at either side of the car. They use a mechanical device to compensate for the weight of the wheels. After manual insertion of the screws, they are fixed by power screw drivers which are mechanically attached to the wheel and automatically released after completion.

The paper presents a system for wheel assembly to a car body that is being moved by a power-and-free conveyor (see Fig. 1). The fit between wheel rim and wheel hub limits

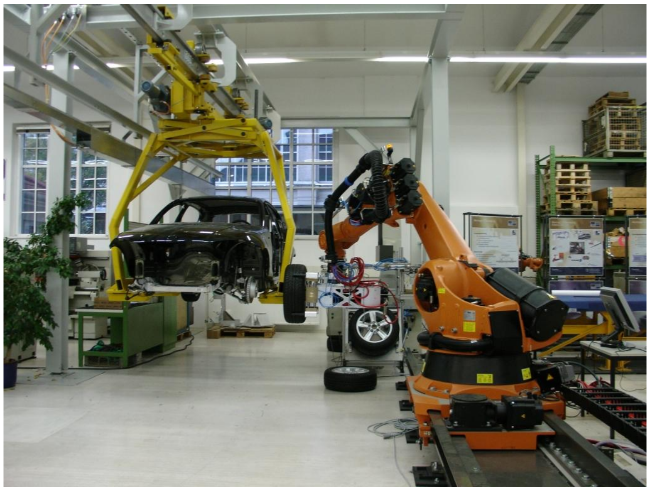

Fig. 1. Set-up at *iwb* for mounting wheels to a continuously moving car body

F. Lange and G. Hirzinger are with the German Aerospace Center (DLR), Institute of Robotics and Mechatronics, D-82234 Wessling, Germany `friedrich.lange@dlr.de`

J. Werner and J. Scharrer are with the Technical University of Munich, Institute for Machine Tools and Industrial Management (*iwb*), Application Center Augsburg, D-86153 Augsburg, Germany

[1] Continuously conveyed objects are objects that are conveyed without being stopped for assembly as with assembly at clocked cycles. The term continuously does not necessarily mean a smooth and thus predictable trajectory or even constant speed.

possible position errors to about 0.2 mm, while the system is moving with about 100 mm/s. Such a high accuracy during motion can only be reached by, firstly, robust perception using multiple redundant sensors that track the conveyed object and secondly, a new control approach that minimizes stochastic control errors as well as offsets.

At the 2009 Hannover Fair, robot-based wheel assembly has been demonstrated by [2], [3]. There, the pose of the wheel hub is first surveyed by images of a CCD-camera. The position and the orientation with respect to the wheel axis is extracted from an image of the wheel hub. Since tilting angles are hard to be determined in this way, laser stripes are subsequently projected from a distant source onto the hub and measured with the same camera, this time without flashing the scene. However, the sensor cannot sense while the assembly operation is performed. Therefore encoders at the conveyor are used to measure the motion in the conveying direction.

Two problems exist within this setup. Firstly, oscillations of the car cannot be perceived. Only the conveyor position is propagated. This is insufficient if the conveyor stops, or accelerates after a stop. Secondly, in the contact state it is not possible to avoid substantial contact forces or torques since neither a force sensor nor passive compliance are used. The latter would not be possible without an additional sensor, since with an elastic part between the robot flange and the wheel, the actual tool center point (tcp) may vary in presence of accelerations of the end-effector. Thus the position of the ideal contact point is uncertain.

The first problem is solved in [4] by continuous evaluation of camera data, but the control is restricted to 1 dof nevertheless, thus neglecting couplings of the conveyor motion to the other dofs. Recently another realization of wheel assembly has been published by [5], [6]. In this setup, however, the camera is arranged lateral to the wheel, such that occlusion may limit the close-range observations. None of these implementations mentions an instrumented mechanical compliance.

Our paper is organized as follows. Next, the operational sequence is listed including the control set-up (Sect. II). Then the different sensors and sensing algorithms are outlined (Sect. III). This allows to survey the computation and the control of the desired trajectory (Sect. IV). Finally, we demonstrate the performance in real experiments with the system of Fig. 1.

II. Set-Up and Work Flow

The generic approach for robots working at a conveyor belt is to include a linear axis in parallel to the conveyor. The robot is mounted on this linear axis. For controlling purposes, a 7 dof system is considered. Such a system is proposed for tasks as the cockpit assembly.

For assembly tasks on the outer body, like wheel or door assembly, a linear axis is not required anymore as long as the working range of the robot is sufficient. For assembly tasks at both sides of the car such a system is required on either side of the conveyor.

A. Operational Sequence

The procedure for wheel assembly is as follows:

1) Five screws are fed to the robot tool. In our test scenario they are first unscrewed from tapped holes using multiple power screw drivers. This is possible without the use of a sensor, just by programmed pose and compliance in the shaft of the screw driver.
2) A wheel is feeded to the gripper. This includes different steps:
    a) The robot moves towards the wheel.
    b) From a predefined distance the wheel pose is measured, in particular the orientation.
    c) The robot approaches the wheel and simultaneously aligns the end-effector.
    d) Finally the gripper is closed and the robot takes the wheel away.
3) Then the wheel is assembled to the conveyed wheel hub. This includes:
    a) The robot moves to the conveyor and waits until the car approaches.
    b) The robot moves parallel in synchronized motion to the conveyor. It measures simultaneously the pose of the wheel hub, including its orientation.
    c) The end-effector is aligned while the wheel is being approached to the wheel hub.
    d) After mechanical contact, forces and torques are controlled while the screws are fastened.
    e) The gripper is opened to release the wheel, but the end-effector is still moving in a synchronized way with the car.
    f) For the final fastening, the screws are partly removed and screwed synchronously thereafter to ensure minimum warping.
    g) The end-effector retreats from the conveyor, still synchronized to prevent any collisions even if the conveyor is stopped.

In the case of intolerable forces or another operational fault the robot immediately retreats from the conveyor since otherwise the conveyor had to be stopped. Simultaneously, the gripper is opened if the power screw drivers are active. These faults are not present in the current implementation. Nevertheless, this procedure emphasizes the applicability of the method since the normal course of operation of the production line as a total is not affected even in the case of a breakdown of its components.

B. Signal Flow

For control of sensor data we use an online interface to the robot position controller. Such interfaces are available for most robots [7], [8]. They provide the robot joint positions $\mathbf{q}_a(k)$ or the Cartesian poses $\mathbf{x}_a(k)$ in each sampling step k of, e.g., 12 ms. At the same time they accept pose commands or commanded joint angles $\mathbf{q}_c(k)$ that are processed by the internal feedback position controller.

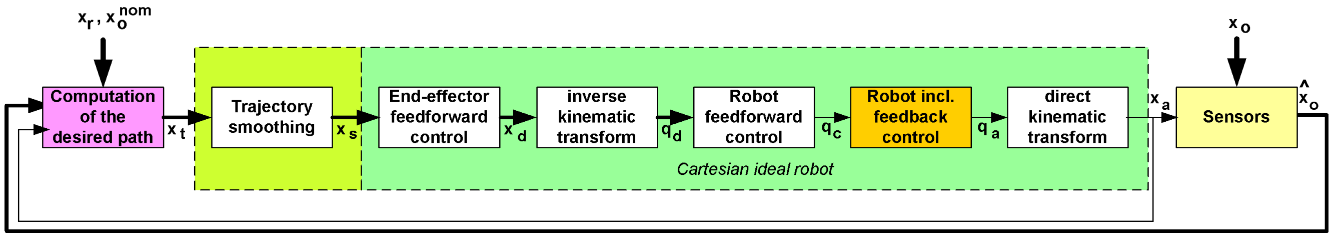
x_r, x_o^{nom}
Computation of the desired path
x_t
Trajectory smoothing
x_s
End-effector feedforward control
x_d
inverse kinematic transform
q_d
Robot feedforward control
q_c
Robot incl. feedback control
q_a
direct kinematic transform
x_a
Sensors
x_o
$\hat{x}_o$
Cartesian ideal robot

Fig. 2. Signal flow for predictive trajectory control of the robot

In contrast to well-known position-based force control architectures as [9] (also referred as external hybrid control scheme [10]) there is no direct feedback from sensor data to motion increments. Instead, similar to [11], we use sensor data to compute the so called desired poses which are then regulated by the original position controller (see Fig.2). This approach tolerates low sampling rates of sensors or of the sensor-robot interfaces since, at least with a constant reference, the transient behavior is only characterized by the internal control loop.

Thus the desired pose $\mathbf{x}_t$ of the tcp is computed by [2]

$$\mathbf{x}_t(k) = \mathbf{x}_a(k) + \mathbf{s}(k) - \mathbf{s}_d(k) \tag{1}$$

where $\mathbf{s}$ is the vector of sensor data which - for a robot mounted sensor - typically represents

$$\mathbf{s}(k) = \mathbf{x}_o - \mathbf{x}_a(k) \tag{2}$$

with the object pose $\mathbf{x}_o$. $\mathbf{s}_d$ represents the desired sensor data, e.g. a time varying distance between the wheel rim and the wheel hub or a desired sensor deflection that is caused by the contact force.

The separation of the determination of the desired pose / desired trajectory and the control unit design gives way to independent extensions. The next section explains the selection of sensors and their use for the computation of the pose of the designated assembly point - the wheel hub. Then Sect. IV closes the loop by presenting the trajectory generation and the controllers.

III. Sensing of the Pose of the Wheel Hub

It is not sufficient to measure the conveyor position because oscillations of the conveyed object are possible, [1]. Instead, a set-up of three kinds of sensors has proved to be generally useful [12].

Firstly, a coarse sensor that informs the system whether the target object is present. This sensor might be a simple sensor barrier. Alternatively it may be implemented as an encoder within the conveyor, that gives the approximate location of the conveyed object in the conveying direction (Sect. III-C).

Secondly, a non-contact sensor that surveys the target pose (Sect. III-A). A preferred realization of this sensor is a CCD camera at the robot end-effector, since cameras are flexible sensors and this location leads to more accuracy and less occlusion. The output of the vision system will be used to let the robot end-effector approach the designated contact point.

After the impact, the third sensor, a compliant force-torque sensor [13], takes control (Sect. III-B). This is a sensor that elastically gives way to acting forces and torques. A compliance of at least one millimeter or degree is crucial, since conveyors are known to generate small scale oscillations, and also because the robot path accuracy is typically worse than the assembly tolerance. With a compliant sensor it is sufficient to keep the mean control error small instead of tracking high frequency disturbances of the conveyor.

All sensors measure the pose of the conveyed object absolutely or with respect to the end-effector pose. The required trajectory is then obtained by prediction (Sect. IV-B).

In contrast to the pose of the conveyed object, the pose $\mathbf{x}_a$ of the robot end-effector is coarsely accessible when evaluating the measured robot joint encoders $\mathbf{q}_a$. This is sufficient if the pose of the conveyed object is sensed with respect to the end-effector, since then disturbances of the robot pose, e.g. caused by joint compliance of the robot, cancel out when computing the desired robot trajectory. For the consideration of the robot pose for dynamic control purposes the joint encoders suffice.[3]

A. Non-Contact Sensing

Since in contrast to [2] in the presented implementation the wheel is gripped at the circumference (tire), it is possible to arrange the camera in a central position (Fig. 3), as in [4]. All wheel rims exhibit a bore to house the kingpin of the wheel hub. In this location the camera is close to the scene and it is free of occlusion.

When measuring the wheel pose (steps 2.b and c of the procedure in Sect. II-A), the bores for the screws and the kingpin are detected by a model-based search within each image of the wheel rims. This is supported by a film that reflects the flashlight which surrounds the camera (see upper part of Fig. 6). This way during the whole approach the desired end-effector pose is tracked in world coordinates.

[2] The notation of this paper is for small orientational sensor corrections. Otherwise the sum of vectors has to be replaced by a product of homogeneous transformation matrices. In addition, in Sect. IV-B the definition of $\mathbf{x}_t$ is extended.

[3] When computing the pose $\mathbf{x}_a$ from the robot joint encoders $\mathbf{q}_a$, the measured deflections of the tactile sensor, caused by gravity and accelerations, are considered in the forward kinematics.

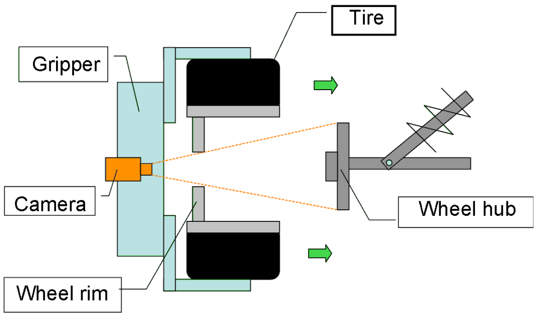

Fig. 3. View of the camera for pickup and assembly of a wheel

During the non-contact tracking of the wheel hub (steps 3.b and c of the procedure in Sect. II-A), the kingpin of the wheel hub is traced model-based, so that its 3d position can be tracked, despite of disturbances in the car motion. The tapped bores for the screws are evaluated as well in order to determine the hub orientation. This information is held constant with lower distance when the features disappear of the images. In addition, tilting angles might be sensed using lines that are projected from aside.

B. Tactile Sensing

The tactile sensor is evaluated to determine the pose of the wheel hub with respect to the virtual pose of the wheel rim, where the latter is computed from the robot pose without considering the sensor deflection.[4]This pose of the hub is measured by the internally sensed displacements within the force-torque sensor, which usually are transformed to forces and torques [13], [14]. In contrast to the virtual pose of the wheel rim, its actual pose coincides with the pose of the hub, when both are in contact.

When opening the gripper (step 3.e of the procedure in Sect. II-A), sensing has to be disabled in the Cartesian directions that are affected by gravity. Otherwise, the end-effector would both rise and tilt when releasing the wheel.

Since compliant sensors of the required dimensions (1000 N, 300 Nm, 2 mm) were not available, a small scale sensor was strengthened by external springs and a second measuring unit in order to fulfill requirements (Fig. 4). Fig. 5 shows a prototype of the resulting multi-sensory end-effector, which includes five power screw drivers.

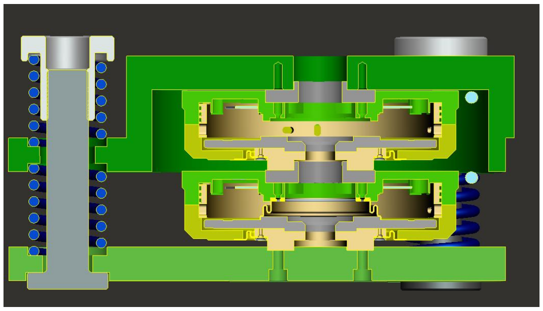

Fig. 4. Compliant force-torque sensor for high forces and torques and twice as much deflection as usual (Exo-Compliance)

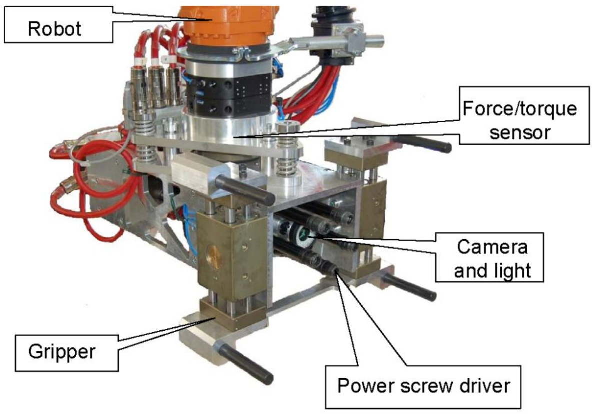

Fig. 5. Multi-sensory end-effector

C. Redundant Sensing

With respect to the procedure in Sect. II-A, the non-contact sensor is required for the pick-up of the wheel (steps 2.b and c) and for the approach to the wheel hub (steps 3.b and c), while the tactile sensor is evaluated when the wheel rim and the wheel hub are in contact (steps 3.d-f). At the beginning and at the end of the robot motion at the conveyor only the coarse position of the conveyor is available. The latter might be insufficient to prevent a collision during the retraction (step 3.g) if the position sensor is implemented as a sensory barrier. Therefore, we recommend a coarse position sensor that is attached to the conveyor.

In addition, such a sensor has advantages while sensing a conveyor stop. Especially when using a camera, processing time and potential filtering delay might be significant, so that an extra sensor is useful. Additional sensing at the suspension of the conveyed object is also meaningful if a conveyor stop excites oscillations of the body, which initially approximately compensate the translational stop. Therefore a conveyor stop will be sensed at the contact point only with delay.

The application of a conveyor-fixed sensor requires that the bias between it and the more accurate sensors that measure at the contact point is identified and considered. In addition, filtering might be advantageous to account for low resolution. A Kalman filter might be useful to do so and to recognize any disturbance with respect to continuous motion.

The three types of sensors predominantly sense one after the other. The encoder is used in the beginning, when camera data are nor yet available. The next phase until contact is made is driven by the vision system. From that point onwards the image of the hub remains constant. Therefore in the contact phase only the tactile sensor will be evaluated. In this way sensor fusion might be implemented by switching between the sensors. Instead, we use a Kalman filter for

[4] Strictly speaking, only that part of the deflection is omitted, that is caused by the contact. The other part which is caused by gravity or by accelerations, does contribute to the virtual pose, to be consistent with the non-contact phase.

fusion, since then there is no jerk at the switching point and since a Kalman filter enables prediction. See [15] for details on the algorithm for sensor fusion.

IV. Computation and Control of the Desired Trajectory

A. Task Description and Programming

Applications for sensor-based control in assembly lines require programming techniques different from the typical industrial robot languages. In addition to the definition of the nominal robot path, for sensor-based tasks the application program has to incorporate the desired sensor values as well. These values usually vary, e.g. during vision-controlled motion towards the tracked object. Therefore, within this project both, the programmed robot trajectory $\mathbf{x}_r$ (reference) and the nominal object trajectory $\mathbf{x}_o^{nom}$ are used for the online computation of the desired sensor values.

$$\mathbf{s}_d(k) = \mathbf{x}_o^{nom}(k) - \mathbf{x}_r(k) \tag{3}$$

In particular, the nominal object motion (the nominal pose and trajectory of the hub) is firstly extracted from the programmed robot path, assuming a conveyor-like motion with ideal contact in the contact phase. In this way, the existing robot languages can still be used with small extensions, e.g. to specify the active sensors or phases.

B. Transition from the Programmed to the Sensed Trajectory

When the first sensor is selected, the car body is still far away, out of the working range of the robot. Therefore the non conveyed directions are regulated first and the robot waits for the conveyor (step 3.a). When it arrives, the robot is started to move in parallel.

Then, as with hybrid force control [16] only those components are sensor controlled, for which there are sensor data that contribute to the estimated pose of the hub. The other directions are desired to track the programmed path, or a trajectory that has been modified because of previous sensor data. This yields $\mathbf{x}_f$, the fused sensed desired trajectory,

$$\mathbf{x}_f(k) = \begin{cases} \mathbf{x}_a(k) + \mathbf{s}(k) - \mathbf{s}_d(k) & \text{sensed components} \\ \mathbf{x}_r(k) & \text{else} \end{cases} \tag{4}$$

When the first data are received just after selecting the vision sensor, there may be a discontinuity in the desired pose, e.g. the orientation of the hub. Then a transient trajectory is computed that defines the desired path $\mathbf{x}_t$ which is then processed according to Sect. IV-C. This transient motion between $\mathbf{x}_f$ and $\mathbf{x}_t$ has a quadratical shape, in order to merge smoothly with the sensed trajectory, shortly before the contact is expected. In contrast, the switching to the tactile sensor is smooth, if the camera is properly calibrated.

Like this, a desired pose is computed in each sampling step, which is continuous to the previous steps. In addition, since the control of Sect. IV-C uses predictions, the desired poses have to be extrapolated to create a current desired trajectory. This extrapolation uses a priori data, e.g. the feed rate of the power screw driver when fixing the screws or the possible accelerations of the conveyor.

C. Predictive Trajectory Control

The computation of the desired path is realized according to Sects. III and IV-B. The control of this path consists of three modules according to Fig. 2.

Since Sect. IV-B generates a continuous but not necessarily a smooth trajectory, $\mathbf{x}_t(k)$ has to be smoothed. This has to be done without delay. Therefore according to [17] an impedance law is formulated, that generates a new trajectory $\mathbf{x}_s(k)$ which smoothes the current trajectory $\mathbf{x}_t(k)$ symmetrically with respect to the time.

Next, the real control is done. The dynamical system can be divided into the robot and the end-effector, where the latter is dominated by the elastic suspension, the force sensor. This suspension is deflected in Cartesian space. Therefore we use a Cartesian oscillation damping module to minimize the effect of the end-effector. This is realized by an input shaping filter [18] which filters the possible oscillation modes from $\mathbf{x}_s(k)$. Feedback is not required since besides the control input there is no significant excitation of oscillations.

The robot controller is realized in joint space, since then there are less couplings than in Cartesian space. Again, a feed-forward controller is used [19], [20]. In contrast to the internal joint controllers which at time step k process the joint positions $\mathbf{q}_c$, the use of desired joint trajectories allows to restrict to feed-forward. In the case of a predictable trajectory, adapted controller matrices $\mathbf{R}_i$ in

$$\mathbf{q}_c(k) = \ \mathbf{q}_d(k) + \sum_{i=1}^{n_d} \mathbf{R}_i \cdot (\ \mathbf{q}_d(k+i) - \mathbf{q}_d(k)) \tag{5}$$

minimize the control errors, if the number n_d of predicted desired joint angles $\mathbf{q}_d(k+i)$ is sufficiently large.

Since the added controllers have no feedback component, the total control system is inherently stable as long as the internal feedback controllers are stable [21]. And this is always true.

The prediction horizon consists of different parts that are required to realize the joint feed-forward, the oscillation damping, and the smoothing, since otherwise a delay would occur. These parts are summed to the total prediction n_{tot} which is the length of the sensed trajectory $\mathbf{x}_t(k+i)$ with $i = 0, .., n_{tot}$ at time step k. For the wheel assembly n_{tot} covers about 0.5 s. Small changes of the trajectory may take place during this prediction horizon.

See [15] for more details on trajectory control.

V. Experiments

According to Sect. II, for wheel assembly there is no linear axis required. Nevertheless, we present experiments with the generic set-up. Also in this case, for simplicity, control is restricted to the 6 axes of the robot. The speed of the linear axis is only modified, if the sensed desired poses reach the limits of the robot workspace, e.g. when completing the assembly at a stopped conveyor.

With the latest implementation, all assembly experiments have been successful. Figs. 6 and 7 show the performance of a sample run, documented by the robot mounted camera

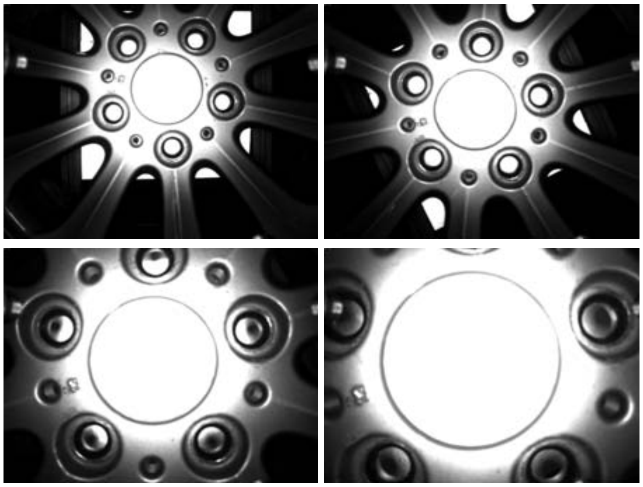

Fig. 6. Images of the robot mounted camera while approaching the wheel

during the approach to the wheel and to the wheel hub. The images show the decreasing distance and the adjustment of the orientation. They prove that the contact point is reached without deviation. The attached video-clip illustrates the experiments. Similar clips are available at our web site www.robotic.de/212/.

The performance is demonstrated in Fig. 8 by plots of the sensor corrections with respect to the reference path. After 6 s the computed trajectory reaches the sensed one since then a contact is expected. The left hand side diagram is dominated by the drift, due to a conveyor speed that is different from the nominal speed of 100 mm/s. The right hand side shows the real orientation of the wheel hub which requires a rotation of 0.3 rad (18 deg). Fig. 9 displays that the control errors are small. Note the different scale with respect to the sensed deviations. The displayed control errors represent the distance between the wheel hub and the wheel rim only before the contact which is established at 7.3 s. Afterwards the assembly tolerance is kept by the positive locking. Then Fig. 9 demonstrates that the robot accurately tracks the motion of the hub and exerts minimum forces and torques on the car body.

VI. Conclusion

This project proves that fail-safe assembly to moving objects is possible, and furthermore that the accuracy is sufficient for very challenging tasks. In contrast to existing industrial assembly applications that are executed by robots, it is not required to use clocked cycles, i.e. to stop the conveyed object for assembly. Instead, robots can be seamlessly integrated into flow assembly lines. The method contributes to reduce production costs, thus enhance productivity, and to release workers from manipulating heavy objects.

The assembly is executed by firstly computing the desired robot motion using sensors to perceive the target to which the part has to be fixed. Secondly, the desired motion is executed with high accuracy, which is possible if reliable predictions of the trajectory are available. This is reached by a proper calibration of the camera [22] that prevents a hard impact. The remaining prediction errors are within the range of permissible deflections of the compliant sensor.

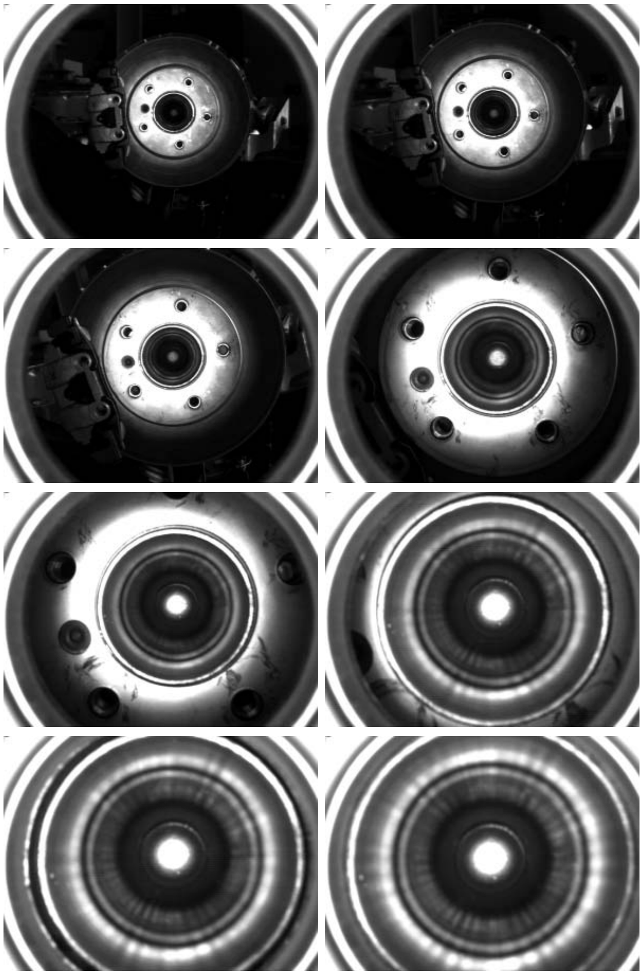

Fig. 7. Images of the robot mounted camera while approaching the wheel hub

The shown approach is not limited to wheel assembly but can be applied to other assembly tasks as well. Thus there is no more need to stop the conveyor for the assembly of heavy parts.

Acknowledgment

This work is funded by the KUKA Roboter GmbH. It is based on a previous project that had been supported by the Bayerische Forschungsstiftung. The authors would like to thank the whole project team for their cooperation, in particular Mirko Frommberger, Stefan Jörg, Amine Kamel, Jörg Langwald, Peter Meusel, Klaus H. Strobl, Bertram Willberg (all DLR), Bernhard Gruber, Jens Klein (both QUISS), and Bruno Fellhauer (SCHUNK).

References

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

[2] W. Meyer. Mounting wheels automatically on moving car bodies: Increasing cost-effectiveness with on-the-fly assembly. ISRA VISION SYSTEMS, 2006. http://www.isra.de/media/public/pdf2006/97_Pressenotiz_Radmontage_e_final.pdf.

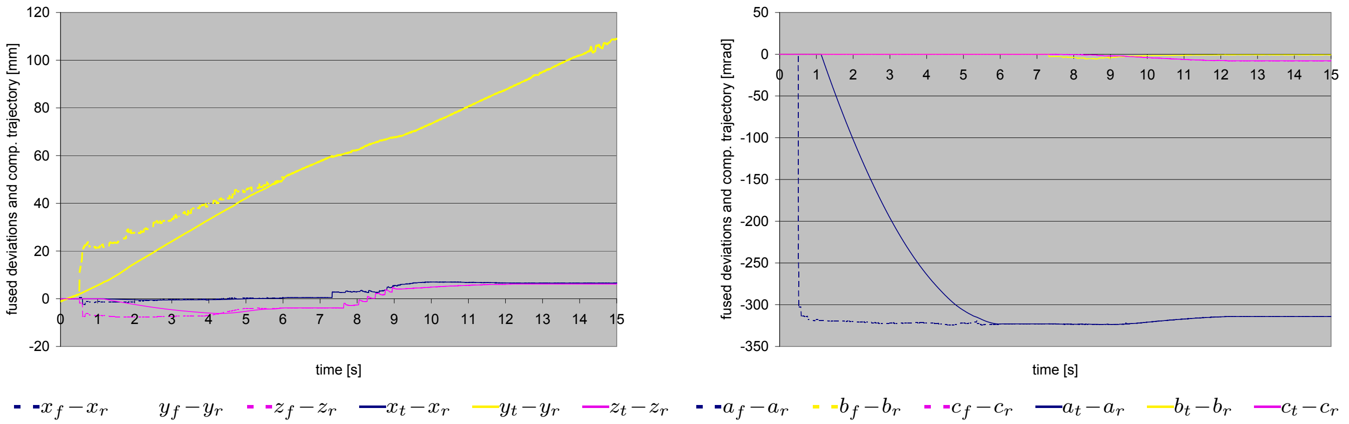

Fig. 8. Fused sensed deviations from the reference path $\mathbf{x}_f - \mathbf{x}_r$ (dashed) and computed desired robot trajectory $\mathbf{x}_t - \mathbf{x}_r$ (solid)

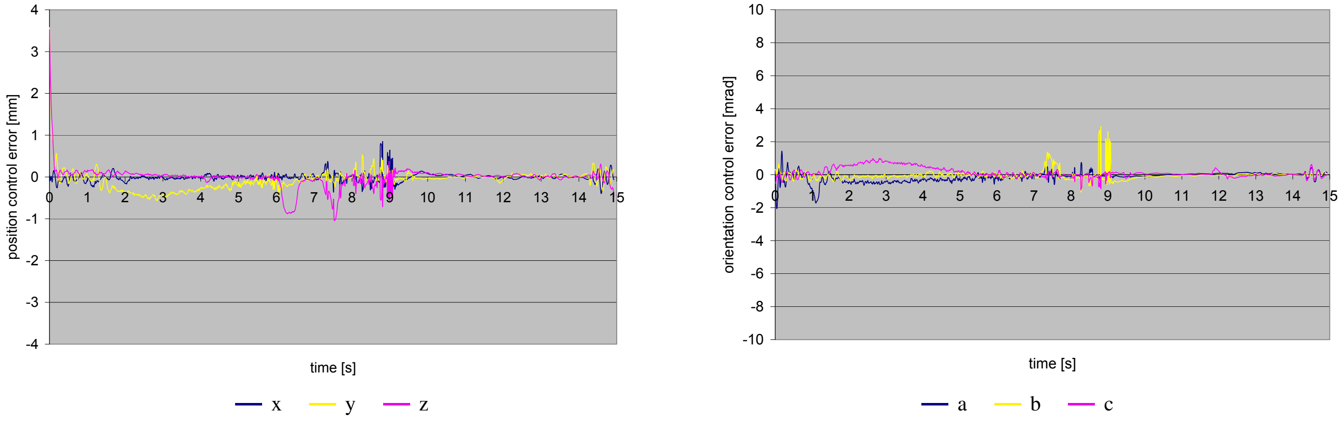

Fig. 9. Control error during the wheel assembly

[3] GOEKE Technology Group. Hannover fair 2009. http://p58084.ts01.wwwcenter.de/GTG.html?jsid=228'&jsl=1.

[4] C. Cho, S. Kang, M. Kim, and J.-B. Song. Macro-micro manipulation with visual tracking and ist application to wheel assembly. *The Int. Journal of Control, Automation, and Systems*, 3(3):461–468, Sept. 2005.

[5] H. Chen, W. Eakins, J. Wang, G. Zhang, and T. Fuhlbrigge. Robotic wheel loading process in automotive manufacturing automation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3814–3819, St. Louis, USA, Oct. 2009.

[6] H. Chen, G. Zhang, W. Eakins, and T. Fuhlbrigge. Assembly on moving production line based on sensor fusion. *Assembly Automation*, 29(3):257–262, 2009.

[7] 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.

[8] KUKA Roboter GmbH. *Robot Sensor Interface (RSI), Release 2.0*, 2006.

[9] KUKA Robot Group. *KUKA.ForceTorqueControl(FTCrl) 2.1*, 2007.

[10] W. Khalil and E. Dombre. *Modeling, Identification and Control of Robots*. Hermes Penton Sience, London, 2002.

[11] V. Lipiello, B. Siciliano, and L. Villani. A framework for force and visual control of robot manipulators. In *Proceedings of the International Conference on Robotics Research (ISRR)*, Hiroshima, Japan, Nov. 2007.

[12] J. Werner, M. F. Zäh, and F. Lange. System to realize a conveying belt synchronous assembly. In *Proc. 17th Int. DAAAM Symposium*, Vienna, Austria, Nov. 2006.

[13] J. Grewe, G. Hirzinger, R. 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.

[14] F. Lange and G. Hirzinger. Learning force control with position controlled robots. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 2282–2288, Minneapolis, Minnesota, April 1996.

[15] F. Lange, J. Scharrer, and G. Hirzinger. Classification and prediction for accurate sensor-based assembly to continuously conveyed objects. In *Proc. 2010 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Anchorage, AK, USA, May 2010.

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

[17] F. Lange, M. Frommberger, and G. Hirzinger. Is impedance-based control suitable for trajectory smoothing? In *Preprints 8th IFAC Symposium on Robot Control (SYROCO 2006)*, Bologna, Italy, Sept. 2006.

[18] A. Kamel, F. Lange, and G. Hirzinger. New aspects of input shaping control to damp oscillations of a compliant force sensor. In *Proc. 2008 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2629–2635, Pasadena, CA, May 2008.

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

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

[21] 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.

[22] K. H. Strobl, W. Sepp, S. Fuchs, C. Paredes, and K. Arbter. DLR CalDe and DLR CalLab. Institute of Robotics and Mechatronics, German Aerospace Center (DLR). http://www.robotic.dlr.de/callab/.