A Manipulator Plays Jenga

Applying Multisensor Integration in Industrial Manipulation Control

BY TORSTEN KRÖGER, BERND FINKEMEYER, SIMON WINKELBACH, LARS-OLIVER EBLE, SVEN MOLKENSTRUCK, AND FRIEDRICH M. WAHL

One reason why industrial manipulators are mainly used for trajectory-following operations, i.e., pure position control, is that today commercial control units are rarely open for sensor integration. Research institutions often replace the existing control units with their own to perform experiments in control engineering. On the other hand, when studying literature, we get the impression that problems of force control, visual servoing, distance control, manipulator dynamics, and control software architectures are solved sufficiently. In concrete cases, this is correct. Problems appear when merging all these fields together. Developers should be encouraged to have a better view of the overall manipulation control system. When considering any kind and any number of sensors (force/torque, distance, vision, pressure, light barriers, etc.), how can even nonexperienced programmers implement guarded and guided motion commands with respect to any reference coordinate system? What could such a hybrid control architecture look like? How can sensor signals be consistently mapped to stable, unambiguous, and deterministic manipulator motions? Questions such as these are supposed to be answered by the research community to bring existing control approaches into industrial practice.

The application described in this article, which plays Jenga [1], is not an industrial one. Nevertheless, it is well suited to show the potential of multisensor integration and to demonstrate the possibilities when concepts of the aforementioned fields are combined. Jenga is a parlor game that consists of 54 wooden rectangular blocks. All blocks are set up together as a tower with three blocks on each level. The aim of the game is to find a loose block in the tower, take it out, and put it back onto the top of the tower. With each move, the tower gets more unstable. Even for human players, this is a task requiring a very high tactile sensitivity. The manipulator plays against itself with the aim of building a tower as high as possible. The setup of the manipulator work cell is shown in Figure 1. Sensor feedback is given by a six-dimensional (6-D) force/torque sensor, a 6-D acceleration sensor, and a laser triangulation distance sensor. To get visual feedback, two charge-coupled device (CCD) cameras observe the tower. To reduce the demands on the image-

Digital Object Identifier 10.1109/MRA.2008.921547

How can even nonexperienced programmers implement guarded and guided motion commands with respect to any reference coordinate system?

processing system, each block was colored black and has two white lines and two dots (see Figure 1).

Related Work

There has been a multitude of research in each specific aspect of this manipulation problem (force/torque control, sensor fusion, task specification, vision, and robot programming). Each area has been developed independently without considering how they can be integrated together.

Whitney [2], Mason [3], and Raibert and Craig [4] published initial works on force control concepts and compliant motion control. Today, literature provides basically three different approaches for force control: 1) impedance control [5], which uses relationships between acting forces and manipulator position to adjust the mechanical impedance of the end effector with external forces; 2) parallel control [6], which enables control of both force and position along the same task space direction; and 3) force/position control, which controls force and position in two orthogonal subspaces [8]. (The problem of orthogonality has to be taken into account here as stated by Duffy [7], who extended the approach such that it is consistent, independent of units, and independent of any origin coordinate system.) The third approach, force/position control, is used within this work, and selection matrices are used here to address the concerns mentioned previously [9].

When applying force/torque control, in practice, it is very helpful to separate forces and torques caused by inertia and those caused by environmental contacts. A recently published approach for force/torque and acceleration sensor data fusion is based on observer techniques to estimate noncontact forces [10]. A more traditional approach that is based on inertia tensor identification, measured forces, and accelerations to calculate the desired contact forces and torques was presented by Kozłowski [11].

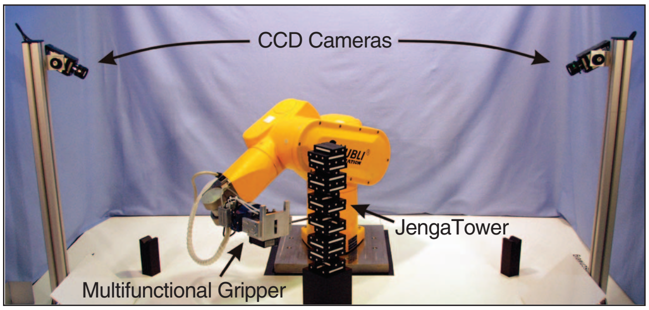

Figure 1. *Work cell setup for the Jenga-playing manipulator.*

Khatib published the operational space approach in 1987 [12], which has been one of the most significant frameworks to realize sensor-guided manipulator motions. De Schutter, Van Brussels, and Bruyninckx published significant articles on compliant motion specification [13], [14] and compliant motion control [15]. Schimmels and Huang wrote about force-guided assembly in a very theoretical manner [16]. The Ph.D. thesis of Natale [17] constitutes the objective in clear theoretical expressions as well as in good practical experiments.

For the realization of visual servoing, Baeten presents a good overview in theory and practice [18]. To achieve advanced vision integration, robust and fast image-processing algorithms have to be implemented [19] for applications in this field.

Considering any kind and any number of sensors in a manipulation work cell, the question of programming concepts or even automated programming arises. Approaches on task specifications can be found in [14], [20], and [21]. Since the complexity and the demands on manipulation control systems have been continuously growing during the last decades, the development of suitable software control architectures gains importance. Two widely known approaches are the Open Robot Control Software (OROCOS) project [22] and the Open System Architecture for Controls Within Automation Systems (OSACA) [23].

Our recent works address task-level programming [24], force/torque control [25], [26], sensor data fusion [27], multisensor integration [28], [29], online trajectory generation [30], and control software architectures [9]. The following sections briefly present the respective approaches and describe the Jenga-playing manipulator. The experiments were also published as a video sequence [31]. In comparison with all the works mentioned previously, this article gives an overview of a concrete implementation and contains parts of all the mentioned fields.

Technical Realization

The hardware and software setup is explained in the subsequent sections, which is then followed by a brief introduction on manipulation primitives (MP) that constitute the interface between the underlaying control levels and the user application. Finally, the underlying control architecture and the user application for playing Jenga are outlined.

Hardware

The robot used for all experiments in this context is a Stäubli RX60 industrial manipulator. Its original controller has been replaced as described in [9], and only the original power electronics has been retained. Depending on the experiment or application, the control system consists of several PC nodes, and, here, we use four PCs. With the high-level hybrid controller, we achieve a control rate of 2 kHz, whereas the low-level joint controller runs at rate of up to 20 kHz. Figure 2 illustrates the developed gripper, which is mounted to the end effector of the manipulator. The gripper is equipped with a 6-D force/torque sensor, a 6-D acceleration sensor, and a laser triangulation distance sensor. The acceleration sensor can be used for the application of a 6-D sensor data fusion approach to compute forces and torques established by environmental contacts [10], [27].

Software Architecture

One of the overall aims has been to provide a manipulation control system that is open for any kind and any number of sensor systems while offering a unique and intuitive programming interface (the so-called MPs) to the user. One important requirement was to develop a scalable real-time hardware and software system. Middleware for robotics and process control applications (MiRPA) constitutes the communication base [9]. It is a distributed real-time middleware, which runs on several PC nodes, with QNX running as the real-time operating system. Every software process in the system has only one communication partner: MiRPA. As a result, a very high modularity is achieved, as can be seen in Figure 3, which depicts the global software architecture for all PC nodes.

The software modules MP Interface and MP Execution form the core of the adaptive hybrid control system and feed the joint control module. The user application robot task is programmed by means of MP nets and sends single MPs to the MP interface module. All software modules on the right of Figure 3 are drivers for actuators, drivers for sensors, or controllers (open or closed loop), which are triggered by the MP execution module. For example, the modules `Force_Ctrl`, `Distance_Ctrl`, and `Vision_Ctrl` are closed-loop controllers, whereas all trajectory-generating modules like `Position_Ctrl` and `Velocity_Ctrl` are feed-forward controllers, which are addressed in the next two sections. The Gripper module is the only actuator in this concrete setup. All the modules mentioned on the right of Figure 3 use the same communication profile so that they can be added and exchanged very easily.

MPs

MPs constitute the interface from the control level to the user application. They are used to specify sensor-guided and sensor-guarded motion commands. A hybrid motion command enables the user to assign set points of any physical magnitude to each degree of freedom (sensor-guided motion). Here, the manipulator motion is generated directly on the base of the respective sensor signal(s). To obtain the universality as demanded in the introduction, any sensor signal can principally be addressed here.

Additionally, a boolean expression called the stop condition, which can contain any sensor signal in any coordinate frame, can be set up to determine the end of a single MP (sensor-guarded motion). As soon as this expression becomes true, the execution of a single MP is finished. For instance, one can specify a hybrid motion command such that the manipulator performs force control in x-direction, distance control in y-direction, vision-based control in z-direction, and trajectory-following control in all rotational degrees of freedom (DoF). The execution ends, if, for example, the force in x-direction exceeds a certain value or a position in y-direction gets under a certain minimum value.

How can sensor signals be consistently mapped to stable, unambiguous, and deterministic manipulator motions?

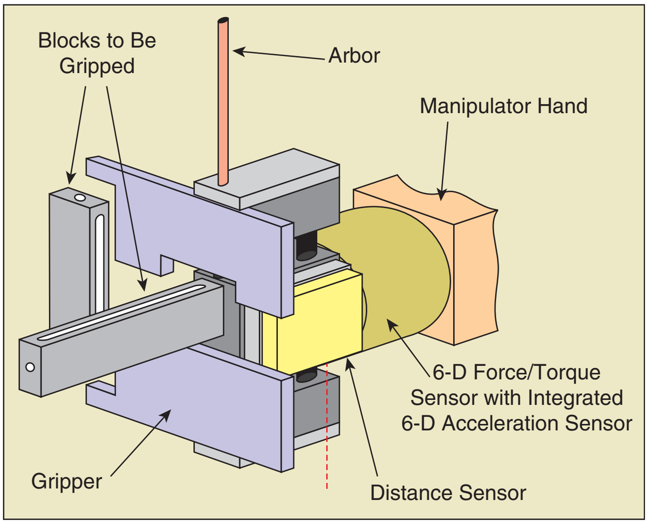

Figure 2. *Multifunctional gripper with sensor devices. Blocks of the game can be gripped in two different configurations. The arbor is used to push single blocks out of the Jenga tower.*

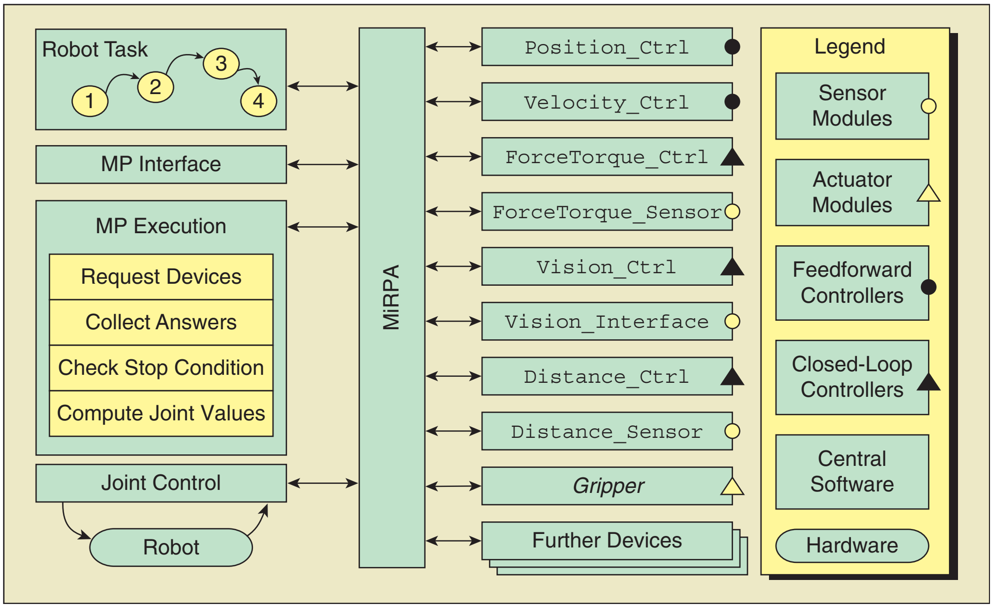

Figure 3. *Modular control software architecture based on the distributed real-time MiRPA.*

This exhibit shows the potential of multisensor integration and opens new possibilities for industrial manipulation.

Besides the hybrid motion command and the stop condition, a MP consists of a third part, the tool command. It addresses the control of further actuators in work cells. In this concrete case, it only opens or closes the gripper.

The consistency of this approach is ensured by using the adaptive selection matrix [9], which assigns exactly one appropriate controller to each DoF. Depending on the current system state, each controller checks whether it is able to control the manipulator in the current state or not. As a result, the responsibility for stability has been taken away from the user. A force control module, for example, would not be able to generate any reasonable output signal as long as the manipulator's end effector remains in free space, i.e., the fused data of the acceleration and force/torque sensors remains under a certain threshold value. In such a case, an alternative controller can be chosen depending on the parameters of the currently executed MP, e.g., the module `Velocity_Ctrl` in Figure 3 could lead the end effector into contact such that the `Force_Ctrl` module could, subsequently, take over in the moment of contact transition.

Control Architecture

The concept of MPs requires a hybrid control system that is able to react on (sensor) events within one control cycle of the MP Execution module, since the point in time where the stop condition becomes true is unpredictable. (In this context, a hybrid controller is a switching control system, which discretely switches between a number of continuous subsystems [32].) As a result, set points for the hybrid controller might change arbitrarily from one control cycle to another. Also, coordinate frames, e.g., the task frame (see [13], [14], and [15]), might change ad hoc. The stability of the resulting overall system is another issue that cannot be neglected [32]. The key part for the realization of these requirements is the online trajectory generator [30], which is represented by the blocks `Position_Ctrl` and `Velocity_Ctrl` in Figure 3. These modules are able to handle and proceed with any state of motion (position, velocity, and acceleration) and in any space (task space, world coordinate, joint space, etc.).

Both modules, `Position_Ctrl` and `Velocity_Ctrl`, compute set points for a synchronized trajectory as desired for the currently executed MP. The `Position_Ctrl` generates a manipulator trajectory, which lets the manipulator reach a desired pose exactly. The boundary conditions for this trajectory (maximum velocity, maximum acceleration, and maximum jerk) depend on the currently executed MP as well as the space in which the trajectory is generated. In comparison to the `Position_Ctrl` module, the `Velocity_Ctrl` module independently generates a trajectory that accelerates all the desired DoF to a certain MP-dependent target velocity.

The `Vision_Interface` provides the position and orientation data about each block of the Jenga tower. This data is subsequently used by the `Vision_Ctrl` module for visual servoing. It computes the poses in space for all blocks and monitors the tower's behavior (to detect a vibrating or collapsed tower).

To achieve high robustness while operating in contact with the environment, model-following-control structures are used for force/torque control in the block `ForceTorque_Ctrl`, which receives feedback signals via the `ForceTorque_Sensor` module. Similar to all controller modules on the right of Figure 3, the force/torque controller computes only a pose difference, which is subsequently interpreted by the MP Execution module to calculate a new absolute pose.

For rapid control prototyping purposes, the developer of a system has the possibility to create a process on one of the QNX PC nodes directly out of MATLAB/Simulink [33]. Here, Real-Time Workshop, Opal RT-Lab [34], and an in-house software interface module were used to establish the connection to MiRPA. Processes can be added and exchanged even during runtime such that users of the system can experiment with new sensor systems and new control approaches (e.g., trajectory computation, force/torque control, distance control, visual servoing, parameter estimation/identification, sensor data fusion etc.) very easily as desired for research purposes. During the development state, the processes on the right of Figure 3 are generated directly with MATLAB/Simulink, and these modules can subsequently be used for experimental verifications on the real target system.

Playing Jenga

Finally, we briefly describe the application program for playing Jenga. During the entire game, no strategy is applied. The blocks to be pushed out are selected randomly. The first step is to always try to push a block few centimeters out of the tower such that it can be subsequently gripped and pulled out from the opposite side. If the counter force during pushing gets too high or if the cameras detect a dithering tower, the manipulator stops immediately, moves back, and tries to push out the next randomly chosen block. In order not to damage the tower when gripping a block, we have to ensure that the block to be gripped does not move when closing the gripper, i.e., each block has to be gripped exactly centered. The spatial resolution of the three dimensional (3-D) model of the tower, which is estimated based on the CCD camera images, is approximately 1 mm in our setup, but this is not enough to calculate an accurate grip pose. This is the reason why a triangulation distance sensor signal is additionally needed to measure the pose of a block in the range of micrometers. The respective functionality is provided by the processes `Distance_Sensor` and `Distance_Ctrl.` To perform a high-precision measurement of the pose of a single block, the manipulator moves its end effector along the block such that the distance sensor records a distance profile, which is subsequently used to determine the block's exact pose. When the block has been gripped,

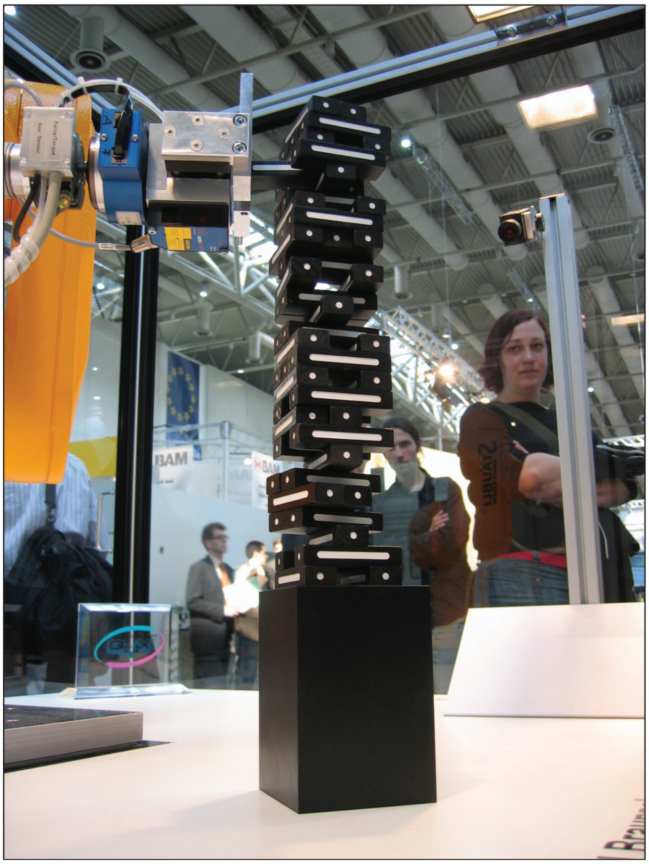

Figure 4. *Jenga-playing manipulator with a tower of 27 levels. Videos of this project can be found in [31] and [35].*

a force-guided MP is set up to pull the block out of the tower very carefully and to eliminate all transversal forces and respective torques. The last part of a single move is to put the block on to the top of the tower. This is done by a simple force-guarded MP, which moves the manipulator carefully toward the tower. After a certain force threshold is exceeded and contact has been established, the motion is stopped, and the gripper can be opened.

Figure 4 illustrates the manipulator with a tower of 27 levels. The record height was a tower of 28 levels, i.e., ten additional levels consisting of 29 ($9 \times 3 + 2$) blocks were put onto the top of the tower.

Conclusions and Future Work

This article describes an overview of a prototypical manipulation control system, which is able to play Jenga. The implementation of the Jenga game has been chosen to verify the integration of existing concepts such as force/torque control, distance control, real-time behavior of distributed control systems, sensor data fusion, online trajectory computation, and visual servoing in one exhibit. This exhibit has no direct industrial use, but it clearly shows the potential of multisensor integration and opens new possibilities for industrial manipulation. The scope of these kinds of systems is certainly not limited to industrial manipulation applications, e.g., the potential in the field of medical robotics is also very high and has to be investigated. A key part of this work is the online trajectory generator. Within this work, a second-order generator (rectangular acceleration signals) has been applied [30], but, for the aim of bringing the mentioned concepts into industrial practice, a jerk-limited generator will be necessary. This and further research on sensor fusion methods will be a major focus of our future work. Besides, the Jenga game could be used as an international benchmark for manipulation control concepts (including force/torque control, distance control, and visual servoing). It is well-known and, since the manufacturer [1] has only one place of production, the game is exactly the same all over the world.

Acknowledgment

The authors would like to thank QNX Software Systems for providing free software licenses.

Keywords

Robotic manipulation, multisensor integration, hybrid control, manipulation primitives, real-time middleware.

References

[1] Hasbro, Inc. (2007, Aug. 13). [Online]. Available: http://www.jenga.com

[2] D. E. Whitney, "Force feedback control of manipulator fine motion," *ASME J. Dynam. Syst., Measurement and Control*, vol. 98, pp. 91–97, 1977.

[3] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. 11, pp. 418–432, June 1981.

[4] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *ASME J. Dyn. Syst., Meas. Control*, vol. 102, pp. 126–133, June 1981.

[5] C. Canudas, B. Siciliano, and G. Bastin, *Theory of Robot Control*. New York: Springer, 1996.

[6] B. Siciliano and L. Villani, *Robot Force Control*. Norwell, MA: Kluwer, 1999.

[7] J. Duffy, "The fallacy of modern hybrid control theory that is based on "orthogonal complements" of twist and wrench spaces," *J. Robot. Syst.*, vol. 7, no. 2, pp. 139–144, 1990.

[8] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed. Engelwood Cliffs, NJ: Prentice Hall, 2003.

[9] B. Finkemeyer, T. Kröger, and F. M. Wahl, "Executing assembly tasks specified by manipulation primitive nets," *Adv. Robot.*, vol. 19, no. 5, pp. 591–611, June 2005.

[10] J. Gámez García, A. Robertsson, J. Gómez Ortega, and R. Johansson, "Generalized contract force estimator for a robot manipulator," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2006, pp. 4019–4024.

[11] K. Kozłowski, *Modelling and Identification in Robotics*. New York: Springer, 1998.

[12] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Autom.*, vol. 3, no. 1, pp. 43–53, Feb. 1987.

[13] J. De Schutter and J. Van Brussel, "Compliant robot motion. I: A formalism for specifying compliant motion tasks," *Int. J. Robot. Res.*, vol. 7, no. 5, pp. 3–17, Aug. 1988.

[14] H. Bruyninckx and J. De Schutter, "Specification of force-controlled actions in the task frame formalism—A synthesis," *IEEE Trans. Robot. Autom.*, vol. 12, pp. 581–589, Aug. 1996.

[15] J. De Schutter and J. Van Brussel, "Compliant robot motion. II: A control approach based on external control loops," *Int. J. Robot. Res.*, vol. 7, no. 4, pp. 18–33, Aug. 1988.

[16] S. Huang and J. M. Schimmels, "Sufficient conditions used in admittance selection for force-guided assembly of polygonal parts," *IEEE Trans. Robot. Autom.*, vol. 19, no. 4, pp. 737–742, Aug. 2003.

[17] C. Natale, *Interaction of Robot Manipulators—Six-Degress-of-Freedom Tasks*, vol. 3 (Springer Tracts in Advanced Robotics). New York: Springer, 2003.

[18] J. Baeten and J. De Schutter, *Integrated Visual Servoing and Force Control*, vol. 8 (Springer Tracts in Advanced Robotics). New York: Springer, 2004.
[19] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall, 2003.
[20] O. Khatib, "A framework for task-level robotic manipulation," in *Proc. 6th Symp. Robotics Research*, 1994, vol. 6. pp. 287–304.
[21] J. De Schutter, J. Rutgeerts, E. Aertbeliën, F. De Groote, T. De Laet, T. Lefebvre, W. Verdonck, and H. Bruynickx, "Unified constraint-based task specification for complex sensor-based robot systems," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2005, pp. 3618–3623.
[22] OROCOS. (2007, Aug. 13). *Open robot control software*. [Online]. Available: http://www.orocos.org
[23] OSACA. (2007, Aug. 13). *Open system architecture for controls within automation systems*. [Online]. Available: http://www.osaca.org
[24] T. Kröger, B. Finkemeyer, and F. M. Wahl, "A task frame formalism for practical implementations," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2004, pp. 5218–5223.
[25] T. Kröger, B. Finkemeyer, M. Heuck, and F. M. Wahl, "Adaptive implicit hybrid force/pose control of industrial manipulators: Compliant motion experiments," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robotic Systems*, 2004, pp. 816–821.
[26] R. Osypiuk, T. Kröger, B. Finkemeyer, and F. M. Wahl, "A two-loop implicit force/position control structure, based on a simple linear model: Theory and experiment," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2006, pp. 2232–2237.
[27] T. Kröger, D. Kubus, and F. M. Wahl, "Force and acceleration sensor fusion for compliant manipulation control in six degrees of freedom," *Adv. Robot.*, vol. 21, no. 14, pp. 1603–1616, 2007.
[28] T. Kröger, B. Finkemeyer, S. Winkelbach, S. Molkenstruck, L.-O. Eble, and F. M. Wahl, "Demonstration of multi-sensor integration in industrial manipulation (poster)," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2006, pp. 4282–4284.
[29] U. Thomas, S. Molkenstruck, R. Iser, and F. M. Wahl, "Multi sensor fusion in robot assembly using particle filters," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2007, pp. 3837–3843.
[30] T. Kröger, A. Tomiczek, and F. M. Wahl, "Towards on-line trajectory computation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robotic Systems*, 2006, pp. 736–741.
[31] T. Kröger, B. Finkemeyer, S. Winkelbach, S. Molkenstruck, L.-O. Eble, and F. M Wahl, "Demonstration of multi-sensor integration in industrial manipulation (video)," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2006.
[32] M. S. Branicky, "Multiple lyapunov functions and other analysis tools for switched and hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 475–482, Apr. 1998.
[33] The MathWorks. (2007, Aug. 20). [Online]. Available: http://www.mathworks.com
[34] Opal-RT Technologies, Inc. (2007, Aug. 20). [Online]. Available: http://www.opal-rt.com
[35] T. Kröger. (2007, Aug. 20). *Experimental videos of robot tasks*. [Online]. Available: http://www.rob.cs.tubs.de/research/projects/manipulation

Torsten Kröger studied electrical engineering at the Technical University of Braunschweig in 2002. In 2001, he did an industrial internship at Lenze Corp. in Atlanta. He is pursuing a Ph.D. degree at the Institute for Robotics and Process Control of the Technical University of Braunschweig. His major research interests are online trajectory generation, hybrid manipulation control, multisensor integration in robot work cells, and new robot programming paradigms. Since 2006, he has worked as a technical consultant for several companies dealing with robot technologies.

Bernd Finkemeyer studied electrical engineering at the Technical University of Braunschweig. Since 1998, he has been a research collaborator at the Institute for Robotics and Process Control of the Technical University of Braunschweig, from where he received his Ph.D. degree in 2004. His main fields of interest are modular distributed control architectures, multisensor integration in robot work cells, hybrid manipulation control, and new robot programming paradigms. Since 2005, he has been working as a project manager in the Research and Predevelopment Department of Kuka Robotics in Augsburg, Germany.

Simon Winkelbach studied computer science at the Technical University of Braunschweig. Since 2001, he has been a research assistant at the Institute for Robotics and Process Control, and he received his Ph.D. degree in September 2006. He was awarded by the Siegfried Werth Foundation and the Foundation for the Advancement of Science of the Technical University of Braunschweig for his research. His research interests focus on 3-D computer vision, surface acquisition and registration, object recognition, and medical imaging.

Lars-Oliver Eble studied electrical engineering with a specialization in the field of control, automation, and measurement engineering at the Technical University of Braunschweig. Since 2005, he has been working as a software developer for technical information systems at S&P Infosysteme GmbH in Braunschweig.

Sven Molkenstruck studied computer science at the Technical University of Braunschweig. He became a research assistant at the Institute for Robotics and Process Control at the Technical University of Braunschweig in 2005. His main research interests include vision systems in robot assembly, automated assembly and sensor planning, and 3-D data acquisition.

Friedrich M. Wahl received his diploma, Ph.D., and the "venia legendi" in digital signal and image processing at the Technical University of Munich in 1974, 1980, and 1984, respectively. From 1974 to 1981, he conducted research at the Institute of Communication Engineering at the Technical University in Munich in the fields of pattern recognition and signal and image processing. From 1981 to 1986, he worked at the IBM Research Labs in San Jose and Zurich in the areas of document analysis, industrial image analysis, and machine vision. Since 1986, he has been a professor of computer science at the Technical University of Braunschweig, Germany, where he set up the Institute for Robotics and Process Control. He became advisory professor of Shanghai University in 1992 and chair of the collaborative research center on robotic systems for handling and assembly in Braunschweig in 2000. His main interests are in robotics, computer vision, and algorithmic aspects of computer science.

Address for Correspondence: Torsten Kröger, Institute for Robotics and Process Control, Technical University of Braunschweig, Muehlenpfordtstrasse 23, 38106 Braunschweig, Germany. E-mail: t.kroeger@tu-bs.de.