Extending an industrial robot controller: Implementation and applications of a fast open sensor interface

Article *in* IEEE Robotics & Automation Magazine · October 2005

DOI: 10.1109/MRA.2005.1511872 · Source: IEEE Xplore

Gunnar Bolmsjö
Linnaeus University

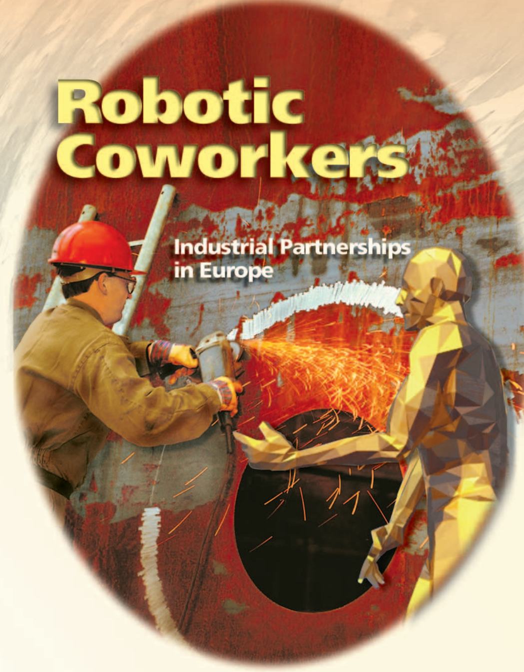
Mats Isaksson
Swinburne University of Technology

Rolf Johansson
Lund University

Mathias Haage
Lund University

Extending an Industrial Robot Controller

Implementation and Applications of a Fast Open Sensor Interface



BY ANDERS BLOMDELL, GUNNAR BOLMSJÖ, TORGNY BROGÅRDH, PER CEDERBERG, MATS ISAKSSON, ROLF JOHANSSON, MATHIAS HAAGE, KLAS NILSSON, MAGNUS OLSSON, TOMAS OLSSON, ANDERS ROBERTSSON, AND JIANJUN WANG

Many promising robotics research results were obtained during the late 1970s and early 1980s. Some examples include Cartesian force control and advanced motion planning. Now, 20 years and many research projects later, many technologies still have not reached industrial usage. An important question to consider is how this situation can be improved for future deployment of necessary technologies.

Today, modern robot control systems used in industry provide highly optimized motion control that works well in a variety of standard applications. To this end, computationally intensive, model-based robot motion control techniques have become standard during the last decade. While the principles employed have been known for many years, deployment in products required affordable computing power, efficient engineering tools, customer needs for productivity/performance, and improved end-user competence in the utilization of performance features.

However, applications that are considered nonstandard today motivate a variety of research efforts and system development to package results in a usable form. Actually, robots are not useful for many manufacturing tasks today, in particular those found in small and medium enterprises (SMEs). Reasons include complex configuration, nonintuitive (for the shop floor) programming, and difficulties instructing robots to deal with variations in their environment. The latter challenge includes both task definitions and definition of motion control utilizing external sensors. The key word here is flexibility, and flexible motion control is particularly difficult since the user or system integrator needs to influence the core real-time software functions that are critical for the performance and safe operation of the system. We must find techniques that permit real-time motion controllers to be extended for new, demanding application areas.

 1070-9932/05/$20.00©2005 IEEE

We must find techniques that permit real-time motion controllers to be extended for new, demanding application areas.

Open Control

Most robot control systems today support some type of user inputs/outputs (I/Os) connected on local networks or buses. A crucial issue is the achievable bandwidth for the control loops with the external feedback. For many applications, the effect of the bandwidth limitations only shows up at longer duty cycles, whereas for some applications (like contact force control between the robot and the environment/workpiece; see Figure 1), stability problems and severe performance degradation may result [1]–[3].

Viewing robotics research from a control perspective, direct access to the driving torques of the manipulator and fast feedback are very valuable, or even crucial, for algorithm evaluation and implementation of high-performance control schemes. This made early robot systems like PUMA 560 popular. Unfortunately, this kind of low-level access is not present in the commercial robot control systems of today. The difference is that today we should not only be able to close fast feedback loops at a low level, we also need to do so in a consistent manner, supporting supervision and coordination with the application-oriented programming on higher hierarchical levels. Therefore, alternative ways to obtain high-bandwidth control based on external sensors, which maintain the existing supervision and coordination functionality, are necessary.

An examination of five major European robot brands (ABB, Comau, Kuka, Reis, and Stäubli) shows that they all, to some extent, provide support for application-specific motion control. Some controllers are fully open but only if all original safety and programming features are disabled. In the project considered in this article, we have used the ABB S4CPlus controller as an example. Whereas S4CPlus is not an open system, its internal design provides some features for development of open control. Similar results have been reported for other systems (see, for example, [4]).

Open Issues

Developments up to the current state of the art raise fundamental questions that form the motivation of this article.

- Today, industrial robot controllers provide highly optimized, model-based motion control that claims to be fully programmable and configurable. Still, when new autonomous or service robot systems are developed, systems developed for industrial manipulation are hardly ever used. Instead, manipulator control is redeveloped but without the full performance and system robustness that would be possible if results/systems from industrial control were used. Can current industrial controllers be useful as components in future advanced robot systems?
- Twenty years ago, proceeding from a textbook algorithm to a functional implementation required extensive engineering efforts. Today, we have engineering tools and code generation from specifications, descriptions, and simulations of control principles. Comparing experimental work within the academic community with industrial robot development, engineering tools such as MATLAB, Maple, and the like are quite similar, whereas the code generation and deployment of controllers/components appear to be quite different. Deployment in a product requires substantially more verification, optimization, and tailoring to the system at hand. Then, the question is this: Could commercial/optimized systems be structured to permit flexible extensions, even on a hard, real-time level?

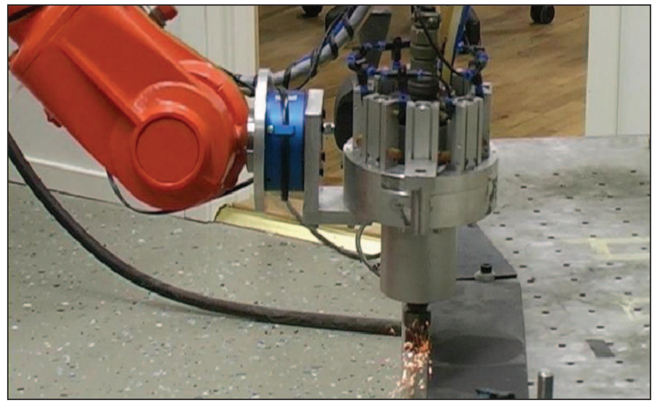


Figure 1. *High-bandwidth, force-controlled grinding using an ABB IRB2400 industrial robot with the extended ABB S4CPlus control system.*

Objectives

We try to answer these questions by confronting theoretical and experimental laboratory results with actual industrial reality. A most challenging case, also representing the 20-year lag between experiment and product, is high-performance, 6-degrees-of-freedom (DOF) control of the contact forces between the robot and its environment. As a part of the off-line automated fettling and finishing (AUTOFETT) project, where the main objective was to develop flexible support and handling devices for castings, force-controlled grinding was accomplished and brought to industrial tests; this will serve as our primary example.

We will consider different aspects of incorporating a fast "sensor interface" into an industrial robot controller system, where the ABB S4CPlus system will be taken as the primary example. The term *sensor interface* may be a bit restrictive because, in practice, the interface not only allows for feedback from external sensor data but also allows for code and algorithms to be downloaded and dynamically linked to the robot control system (Figure 2).

Considerations and Design of System Extensions

The architecture of the ABB S4CPlus control system and its extensions are shown in Figure 2. Task descriptions, as given by the robot programming language RAPID, are passed through the trajectory generation and turned into references for the low-level servo controllers. Extensions to the system, based on present and future applications requiring the use of external sensor-based control, could be made by modifying references on any level (task, Cartesian, joint, or motor currents level). We will discuss the underlying design considerations and our implementation of the platform.

On a high level, the present ABB S4CPlus (and earlier) systems already feature the ability to read sensors via customer I/O to influence the robot task as expressed in programs written in the ABB RAPID language. The RAPID program reading sensor information via the I/O system can be referred to as a *pull protocol,* which requires no external computing. The sensor reading/handling, however, must be expressed in the user program. Today, it is also possible to change programmed motion targets via remote procedure calls (RPC) during robot motion, which can be referred to as a *push protocol*. This requires external computing but less RAPID programming since the logic of how sensor data should influence motions is expressed in external software. Both these alternatives are of great value and should be maintained, but there are also two major problems that must be resolved in future systems.

- *Performance:* Restricting the use of external sensors to the RAPID level only implies that new types of high-performance motions cannot be introduced with a reasonable engineering effort. Some simple cases have been solved, such as the control of external welding equipment, but the fundamental support for motion sensing is missing. Whereas force control is much needed within several application areas, such as foundry and assembly, it is currently quite difficult to accomplish in the robot work cell.
- *Flexibility:* The use of port-based I/O data without self-description leads to less flexible application programs that require manual configuration, limiting development of high-level application program packages.

Therefore, today we have high-level (user-level) usage of low-level (primitive) sensors. To overcome the two aforementioned problems, we also need low-level (motion control) usage of high-level (force, vision, etc.) sensors. As a first step, interfacing with force sensors should be supported. This is both a technically demanding case and a desired one from a customer point of view.

With this overall goal, some specific topics will now be covered.

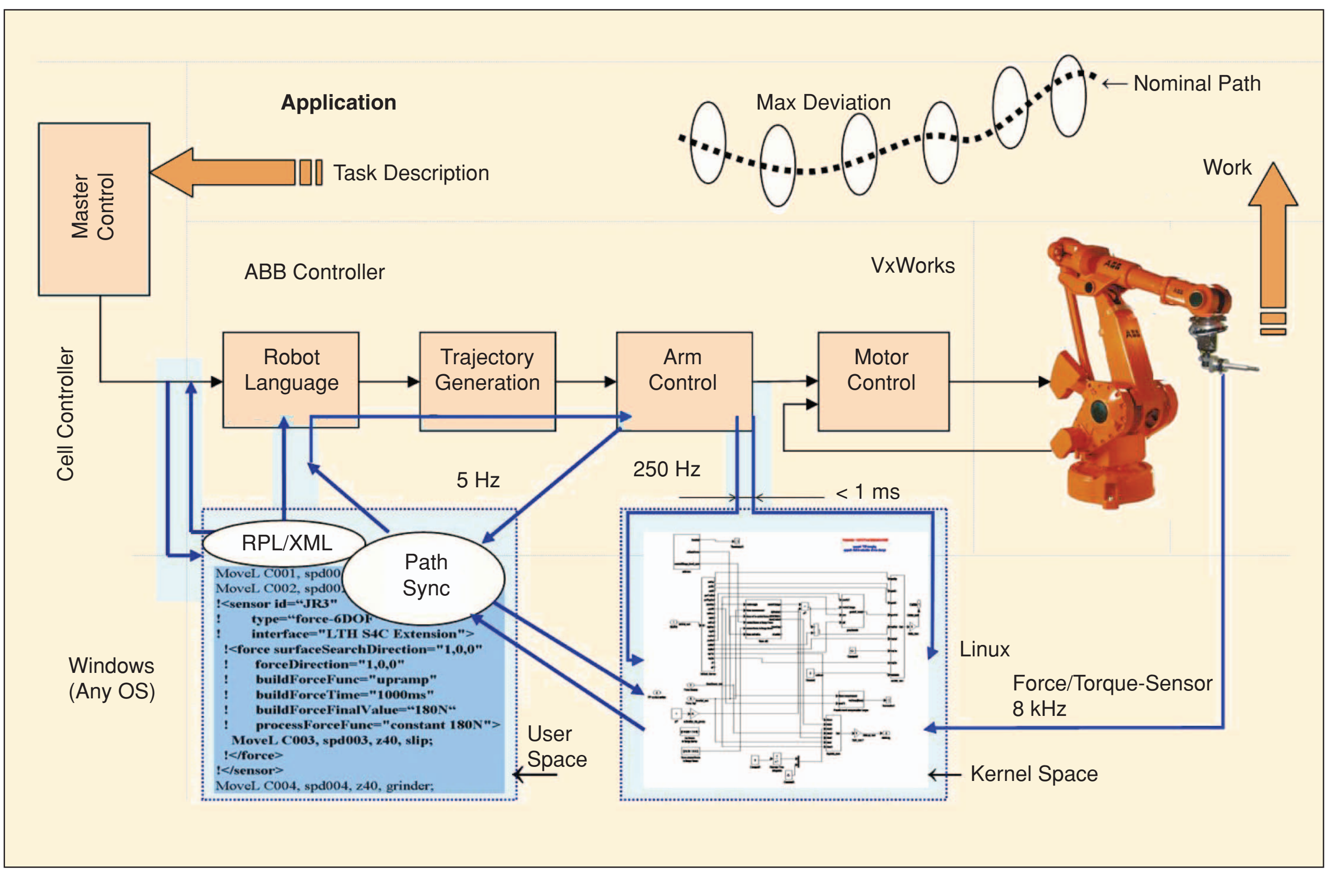


Figure 2. *The extension of an industrial robot controller with sensor interface and support for external computations and synchronization. The configuration uses a Motorola PPC-G4 PrPMC-800 processor board mounted on a Alpha-Data PMC-to-PCI carrier board with a local PCI bus.*

Hardware and I/O Protocol

The most promising hardware interfacing possibilities (from a cost and performance point of view) are shared memory access via the peripheral connection interface (PCI) system bus and standard high-speed Ethernet communication. To some extent, these techniques are already used in S4CPlus.

Shared Memory

Obtaining sensor data directly in shared memory simplifies system development since the system programming model is unchanged. Shared memory is assumed to be provided via the PCI bus. S4CPlus already supports installation of PCI plug-in boards, although this feature is not made available to customers.

Presently, most sensors do not come with a PCI interface, even if some simple sensors can be connected to PCI-based I/O boards. However, some advanced sensors, such as the JR3 force-torque sensor, provide a PCI-based interface. The trend is that more and more PCI-based sensors are becoming available. This method of interfacing external sensors also allows for adding "intelligent sensors" (sensor fusion or sensors with additional computational power).

Networked

Sensor interfaces can also be networked based on field buses, which are available on the user level for all modern controllers and on the servo level for some controllers. However, it appears that field-bus interfaces and communication introduce delays and limited performance compared to the shared memory interface.

As an alternative, our experiences with Ethernet communication using raw scheduled Ethernet or UDP/IP show promising results [5]. The bandwidth is comparable to that of the PCI bus, and the standard network-order of data bytes simplifies interfacing. Also, with proper network/interrupt handling, the latency can be very short, showing great potential for future applications utilizing distributed sensors.

Safety and Quality Issues

Open systems require careful engineering to avoid exhibiting unpredictable or even unsafe behavior when confronted with inexperienced users and extended with novel features at the customer site. One significant challenge in the development of open systems is the complexity in the systems engineering, where several difficulties, which are discussed below, must be addressed.

Hardware Reliability

Installing third-party hardware means there is an additional risk for system failures, despite the high and ensured quality of the basic robot system.

The added hardware may fail without affecting the robot hardware, but it can still lead to system failure from an application point of view if the application was made dependent on the added hardware. Also, third-party modules may severely interfere with the communication on the data buses used by the control computers. Such a failure can be due to faulty added hardware, to bad configuration, or to incorrect access of the bus interface of the added hardware.

To avoid these problems, customers or in-house application developers should write the application software in such a way that functionality can be tested based on some dummy sensor data without using the actual hardware. This can also be accomplished by running the application with a virtual

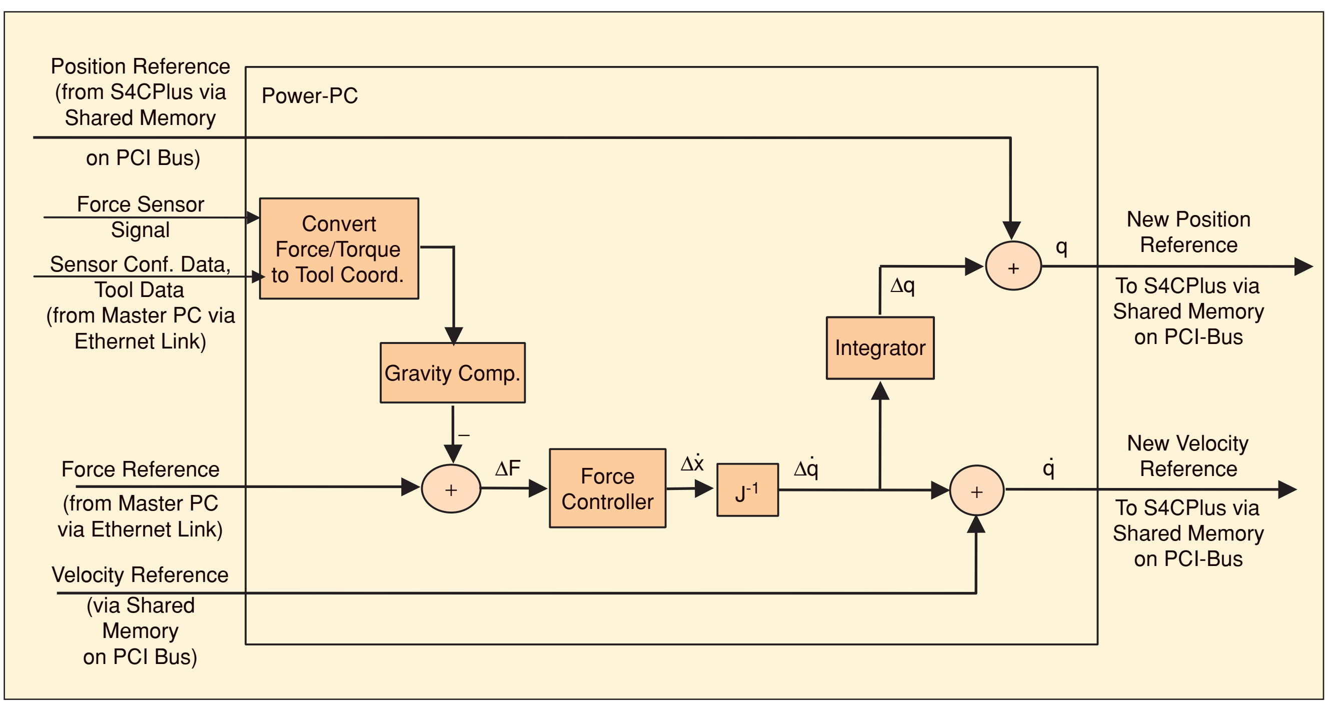


Figure 3. *The force controller block structure. The force control algorithm is implemented inside the block labeled Force Controller.*

controller. Hence, guides for developing sensor-based applications could and should be supported within graphical robot simulation and programming tools.

Sensor failures are inevitable and have long since been an important obstacle in real applications. In cost-efficient production, it is not as simple as saying that there should be redundant sensors; this configuration is costly and increases the risk of system overload/failure. A combination of system structure, proper interface design, testing methodology (including simulation support), and well-defined fall-back control is needed.

Data Integrity

A serious problem, from a safety point of view, is the risk of external software damaging important robot system control data (for instance, due to bad pointers or bad array indices in the external software). Therefore, common data areas should normally be located on the added board and then accessed by the robot controller.

Classified System Properties

The definition of shared data (control signals and other internal states) could be achieved by providing available header files. However, using the ordinary header files would expose potentially classified motion control techniques in too much detail. Therefore, there should be a neutral definition of (possibly) exposed variables, preferably based on information from textbooks or articles and possibly suggested as an open standard.

Robot Safety

Even with hardware and software functioning as intended (as described in previous sections), in a strict technical sense, there is a potential risk that the external logic interacts with the control logic in an unforeseen manner. That is, even if the externally added software does what the program states, it can potentially still compromise robot safety functions.

To overcome this difficulty, the states exposed to external software should be copies of the internal true state, and external states need to be cross-checked before influencing the modes of the standard robot control. Updating can be periodic in some cases, whereas other states (such as run-mode and brake states) should be updated in an event-driven fashion in order to improve consistency between internal and external states, including generation of interrupts to the external software.

Perhaps the most important part of safety is the ability to keep the internal safety functions activated (possibly with adjusted tolerances), even during sensor-based motions. While this problem has been solved, the remaining challenge is to combine safety with performance.

Performance

For industrial robots, control performance means productivity. Specific force control algorithms (inside the Force Controller block in Figure 3) are outside the scope of this article, but the imposed requirements on the open system deserve some attention.

Sampling and Bandwidth Considerations

As an example, force control in a noncompliant environment typically requires fast sampling since excessive contact forces may build up very quickly, for instance, during the impact phase. It is also well known from control theory that feedback from a sampled signal decreases the stability margin, thereby decreasing the robustness to varying operating conditions.

In the architecture of the ABB S4CPlus system, there are a number of levels in which external control actions can enter the system. First, high-level feedback using the high-level ABB RAPID language to modify the generated trajectories gives a sampling time of $h = 0.1$ s. The interface to the built-in arm servo control has a higher sampling frequency, $h = 4$ ms. Finally, $h = 0.125$ ms gives the maximum internal sampling frequency of the JR3 force/torque sensor.

A simple simulation will illustrate the effects of different sampling intervals for a highly simplified model of a typical force control task. A linear model with 1 DOF of a controlled robot is given by the transfer function model $F(s) = 35000\,k/(20s^2 + 1500s + 35000)X_r(s)$, where F is the contact force, k is the stiffness, and X_r is the commanded position reference.

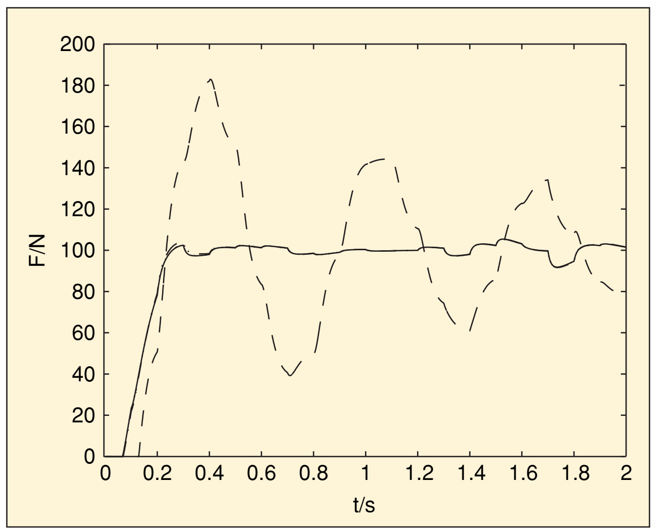


Figure 4. *The contact force for continuous time design (solid), h = 0.1s (dashed), h = 4 ms (dash-dotted), and h = 0.125 ms (dotted).*

In Figure 4, we can see the results of the simulation using a rough surface of stiffness $k = 25$ N/mm, with a continuous time proportional controller as well as discrete time control with the sampling times described above. The desired force F_r was 100 N. It can be seen that the response for $h = 4$ ms is almost identical to the continuous time design, thanks to the fast position control in the inner control loop. The 4-ms level has been determined to be a good trade-off in many force control applications, considering also the limited available computational power. However, in some applications, such as force control in extremely stiff environments or applications where high approach velocities are required, a sampling rate higher than 4 ms may be desired.

Alternative ways to obtain high-bandwidth control based on external sensors, which maintain the existing supervision and coordination functionality, are necessary.

User and System Aspects

Open controllers need ways of expressing the usage of sensors. The RAPID language of the S4CPlus controller supports sensor feedback through a RAPID concept called correction generator. This language mechanism allows the robot program to correct the robot path during operation with information typically derived from a sensor.

Unfortunately, the built-in RAPID mechanism is not applicable for use by force sensor feedback, primarily for two reasons. The correction generators only support position-based path correction, while force feedback may require torque-based path correction. Second, the update bandwidth supported is much too low to apply to most force control applications. Whereas some servo-level extension is needed to accommodate the bandwidth requirements, the user programming level requires extensions for force control.

Language Extensions

The specification of desired force control should be available on the user level, where the rest of the robot application is specified. The solution was to introduce two new language scopes into the RAPID language, integrating handling of sensor-influenced trajectories into the language itself. In order to be backwards compatible with standard RAPID, the new code was encoded as XML scopes and tags within RAPID comments (Figure 5). The processing of the XML comments is conducted in a new master PC module acting as a robot proxy (see Master PC in Figure 2), which then communicates both with the original program server of the S4CPlus controller and with the added low-level control on the added PCI board.

System Connections

The communication between the master PC and the S4CPlus controller is over the Ethernet using TCP/IP and UDP/IP. This is not within the force control loop as such; its purpose is to synchronize the robot program execution with the low-level force control along the programmed path. This was accomplished by the following design elements:

- The force scopes in the ExtRAPID program are replaced by calls to a generic motion-server, which is written in RAPID and downloaded with the rest of the application. The force-controlled MoveL instructions are kept in the master PC (see Figure 2) and fed to the program server of the S4C controller via the ABB Robot Action Protocol (RAP).
- The embedded motion server carries out the motions by executing TriggL instructions (instead of the original MoveL) with extra arguments that form a subscription of an I/O byte output. Later, when the S4C servo actually performs the motion, that output (the sync signal from servo to master PC in Figure 2) forms the lower byte of the integer value of a system-wide path coordinate.
- The master PC uses the received path coordinate as the basis for the 4-ms advancement along the path, maintaining the overall path coordinate and computing force control set points and parameters accordingly.

Due to the limitations on buffering according to the first design element, and since an external set point in real time can influence the set points to the force control loop (Figure 2), any external sensors connected to or communicating with the master PC in real time can be used for instant feedback to the motion control. Note that the ABB controller is then kept aware on the top level of the robot's commanded target.

External Motion Control

With language extensions and system connections in place, the implementation of the actual external controller (to the S4C) can be accomplished. This is the force control in Figure 2. To accomplish hard, interrupt-driven, real-time execution with shared memory communication, the force controller is run as a Linux kernel module. Such a module can be replaced without rebooting the system, but programming for kernel mode is a complication. However, all parts of the force controller (including the shared memory interfaces) were implemented in C as Simulink blocks, which (apart from being used for simulating the system) were cross-compiled to the target computer and incrementally linked to form a Linux kernel module. The porting of the Linux kernel to the specific computing and I/O hardware was carried out in our laboratory as was the tailoring of the build procedure for making Linux kernel modules for sensor feedback.

The host computer version of the Simulink blocks are first translated for embedding by using MathWorks Real-Time Workshop, then compiled and linked with external libraries. In the resulting system, the control engineer can graphically edit the force control block diagram and then build and deploy it in the robot controller.

Simulation

Apart from simulating the force control as such, it is also highly desirable to be able to simulate sensor-based robot control from an application point of view.

Simulation of Low-Level Control

Designing the controllers using MATLAB/Simulink (which as described above also gives the implementation) means that the Simulink models can also be connected directly to existing models of the robot; furthermore, models of the environment and sensors can be used for simulation. Tools such as Modelica and Dymola were used and proved to be very effective for the modeling and simulation of many types of dynamical systems, including industrial robots [6]–[9].

External Sensing in the Digital Factory

Traditional off-line programming does not use the full potential of virtual models and simulation systems in industrial robot applications. The interface between the off-line programming system and the robot controller is today restricted to program transfer. Considerable improvements have been made in the accuracy of programs created off-line, especially since the introduction of technologies such as RRS (Realistic Robot Simulation; http://www.realistic-robot-simulation.org). However, extensive problems remain. For instance, when high-level sensors such as vision, force, and laser scanning are used, no mechanism is available to relate the sensor information to prior knowledge actually existing in the model created in the off-line system.

If the virtual model could be accessed during the execution of the robot task, intelligent decisions could be made despite changes in the state of the robot work cell that were not anticipated when the robot task was planned. Instead of using a simple feedback loop to the robot movement, the virtual model is continuously updated, allowing new information and previous knowledge to be accumulated in a common format. High-level replanning of the robot task can then be automatically performed. Typical limitations of robot systems that are hard to handle online include collisions due to obstacles unknown to the robot program and deviations of the setup and kinematic singularities during linear movements. Successful implementation and experiments have been made in the present case project.

Case Study—Force-Controlled Deburring

The use of industrial robots for automated deburring, grinding, and polishing is an interesting example of a process where external sensing capabilities are crucial. Accurate control of the contact forces can help increase the quality of the final product, as well as flexibility in the deburring process. To handle the deviations from the nominal workpiece geometry that are inevitable consequences of the foundry process, some compliant behavior needs to be included in the system used for deburring. As an alternative to physically adding mechanical compliance to the system setup, for instance by using a compliant tool, force control can be used to program a desired compliant behavior and to maintain a desired contact force during the deburring process.

Hybrid Force/Position Controller

During the deburring task, only the direction perpendicular to the surface of the workpiece is constrained, and a hybrid force/position control strategy is employed [10]. In this type of structure, 1 or several DOF become force controlled while ordinary position control is used in the remaining directions. Typically, the force-controlled direction is perpendicular to the surface, while the motion tangent to the surface and the orientation are controlled using position control. The directions that should be force controlled are selected using a diagonal selection matrix, which is set as part of the high-level task specification. There have also been extensions to the hybrid position/control approach that take the robot dynamics into account [3].

```
MoveL C001, spd001, z40, grinder;
MoveL C002, spd002, z40, grinder;
!<sensor id=“optidrive”
!            type=“force”
!            interface=“LTH+ABB S4C Extension”>
   !<force surfaceSearchDirection=“1,0,0”
!              forceDirection=“1,0,0”
!              buildForceFunc=“upramp”
!              buildForceTime=“1000ms”
!              buildForceFinalValue=“150N”
!              processForceFunc=“constant 150N”>
    MoveL C003, spd003, z40, grinder;
   !</force>
!</sensor>
MoveL C004, spd004, z40, grinder;
```

Figure 5. *A sample ExtRAPID program. The extended language constructs are located in RAPID comments and are modeled as XML tags in order to be easily modifiable.*

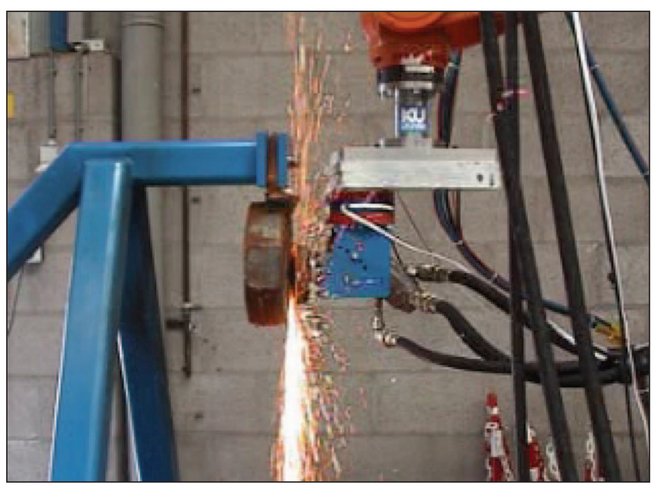


Figure 6. *Grinding with IRB6400 at Kranendonk, The Netherlands, using a compliant grinding tool developed at KU Leuven, Belgium, together with force feedback control. The video can be found at http://www.robot.lth.se.*

Today, modern robot control systems used in industry provide highly optimized motion control that works well in a variety of standard applications.

Experimental Results

Also included in the controller is a finite state machine responsible for switching between position and force control, and for handling control during the transition and impact phases.

The grinding experiments were carried out on an ABB IRB6400 robot (see Figure 6) at the facilities of the Kranendonk company in The Netherlands. A special grinding tool developed at KU Leuven, Belgium, was used in the experiments. The results from an experiment with reference $F_r =$ 150 N are shown in Figure 7. The disturbance that can be seen at time $t \approx 13$ s is caused by a resonance in the workpiece/fixture, which occurs when the grinding tool moves across a hole in the center of the workpiece.

Discussion

Robots are distinguished from other types of machines in terms of flexibility, i.e., the ability to change their behavior through reprogramming in order to be able to cope with new situations. Stemming from the initial research on programmable manipulation systems, recent research approaches typically fall into one of two categories: 1) autonomous robotics, with a focus on handling unstructured environments but largely neglecting performance for industrial productivity, and 2) industrial robots, with a focus on motion performance in structured environments but neglecting most of the perception and navigation issues.

The fact that robots today effectively handle fully structured and specified tasks in industry, combined with the lack of experience/knowledge from small-scale manufacturing within the research community, has created the misconception that industrial robotics is solved. In future manufacturing, however, it is apparent there will be an increased need for industrial robots that (typically in small enterprises) understand human instructions and are able to handle larger task/workpiece variations. Then, we will need to combine both theory and system technologies from various fields of robotics research. An important feat is to package experimental results as useful components (for verification and reuse). Another is to find techniques that permit real-time motion controllers to be extended for new demanding applications, typically using external sensing to substantially improve flexibility.

Many robotics labs have reported activities in open control systems that fully satisfy the need for the abovementioned aspects of evaluation and implementation [11], [12].

The close cooperation and technology transfer between industry and academia have been instrumental during the development of the platform, since control and software need to be tightly integrated for performance and applicability. Robotics is multidisciplinary and researchers from many fields and different university departments have been active in the development of the field.

Conclusions

This article describes the design and implementation of a platform for fast external sensor integration in an industrial robot control system (ABB S4CPlus). As an application and motivating example, we report on the implementation of force-controlled grinding and deburring within the AUTOFETT-project (EU Growth Programme).

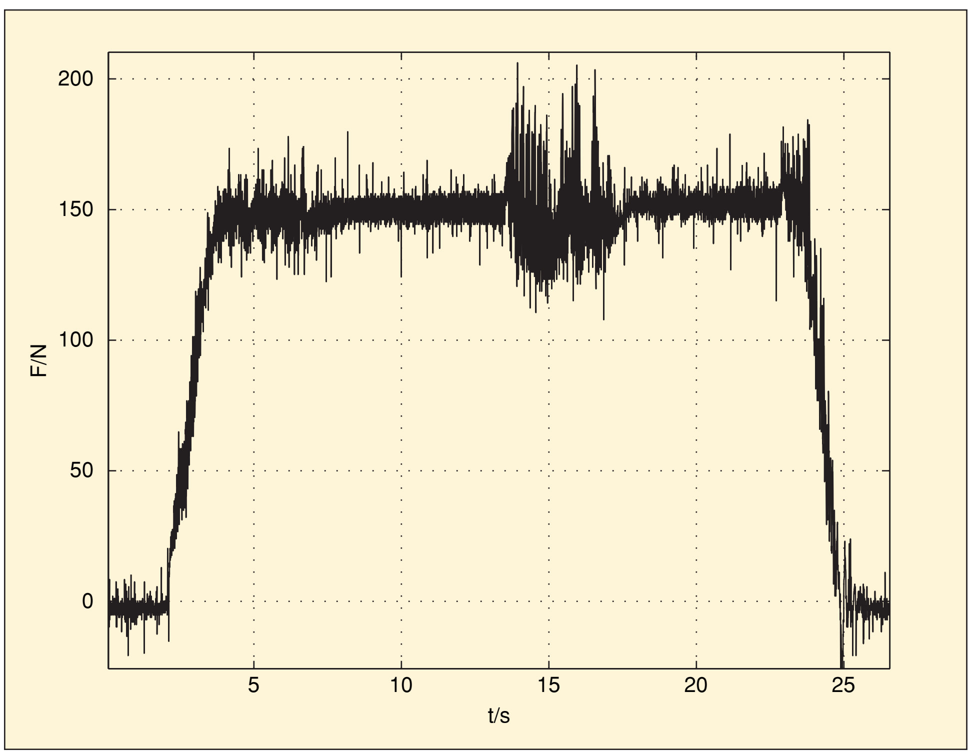


Figure 7. *The contact force during grinding experiment with force reference 150 N. The disturbance at time $t \approx 13$ s is caused by a resonance in the workpiece.*

The accomplished sensor interface, we believe, is unique because of the following elements:

- A shared memory interface to the built-in motion control enables fast interaction with external sensors.
- High-level and low-level controls are integrated in such a way that low-level instant compensation (within the tolerances of system supervision limits) propagates to higher levels of execution and control, providing state and path coordinate consistency.
- The external sensing and control is built on top of a standard industrial controller with (due to the previous item) the built-in system and safety supervision enabled, making it possible for the end user to utilize all the features (language, I/O, etc.) of the original system.
- The add-ons to the original controller can be engineered (designed and deployed) by using standard and state-of-the-art engineering tools, thereby bridging the gap between research and industrial deployment of new algorithms.

Experiences from industrial usage of the fully developed prototype confirms the appropriateness of the design choices, thereby also confirming the fact that control and software need to be tightly integrated.

The new sensor platform may be used for the prototyping and development of a wide variety of new applications. It also offers an open experimental platform for robotics research explored on many hierarchical levels (from control algorithms with high bandwidth to robot programming and task modeling with online sensor information). The preserved high-level support and the integration of the supervision and safety system with the standard industrial robot system constitute major differences from most open robot systems that have been reported for academic research.

The mutual benefits of collaboration between academia and industry tend to be, in our opinion, crucial for the future development of flexible, productive robots. With open systems, external partners will be able to extend the system also on the motion control level. The richness of applications and their dynamics will radically increase the need for more research. In this process, going from theory to practice is not only a matter of technology transfer but a bidirectional flow of ideas and knowledge.

Acknowledgments

The authors are grateful for the friendly system-integration atmosphere and support at Kranendonk Production Systems in The Netherlands. We also appreciate the support within our organizations, including Håkan Brantmark and Peter Eriksson at ABB and many others. The work reported in this article was partly funded by the EU 5th Framework Growth Project GRDI-2000-25135: Affordable, flexible system for off-line automated fettling and finishing (AUTOFETT). The initial development of the external control platform and techniques was supported by VINNOVA, the Swedish Agency for Innovation Systems.

The authors of this article received the EURON Technology Transfer Award 2004 for the "Open platform for skilled motions in productive robotics."

Keywords

Open robot control, force control, industrial robots, sensor-based control, sensor interfaces, flexible motion control.

References

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

[2] D. Gorinevsky, A. Formalsky, and A. Schneider, *Force Control of Robotic Systems*. Boca Raton, FL: CRC, 1997.

[3] T. Yoshikawa, "Force control of robot manipulators," in *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, Apr. 2000, pp. 220–226.

[4] H. Born and J. Bunsendal, "Programmable multi sensor interface for industrial applications," in *Proc. Int. Conf. on Multisensor Fusion and Integration for Intell. Syst.*, Baden-Baden, Germany, 2001, pp. 189–194.

[5] A. Martinsson, "Scheduling of real-time traffic in a switched Ethernet network," M.S. thesis, Dept. Automat. Control, Lund Institute of Technology, Lund, Sweden, Mar. 2002.

[6] F. Calugi, "Observer-based adaptive control," M.S. thesis, Dept. Automat. Control, Lund Institute of Technology, Lund, Sweden, Apr. 2002.

[7] Modelica [Online]. Available: http://www.modelica.org

[8] S.E. Mattsson and H. Elmqvist, "An overview of the modeling language Modelica," in *Proc. Eurosim'98 Simulation Congress*, Helsinki, Finland, Apr. 1998, pp. 182–186.

[9] Dymola [Online]. Available: http://www.dynasim.se

[10] M.H. Raibert and J.J. Craig, "Hybrid position/force control of manipulators," *ASME J. Dyn. Syst., Meas. Control*, vol. 103, no. 2, pp. 126–133, 1981.

[11] K. Nilsson and R. Johansson, "Integrated architecture for industrial robot programming and control," *J. Robot. Auton. Syst.*, vol. 29, no. 4, pp. 205–226, 1999.

[12] H. Bruyninckx, "Open robot control software: The OROCOS project," in *Proc. Int. Conf. Robotics and Automation*, Seoul, Korea, 2001, vol. 3, pp. 2523–2528.

Anders Blomdell has been a research engineer at the Department of Automatic Control, Lund University, Sweden, since 1988. His research interests include man-machine interaction, real-time programming, hardware design, network communication protocols, and computer languages for control engineering.

Gunnar Bolmsjö received M.Sc. and Ph.D. degrees in engineering from Lund University, Sweden, in 1981 and 1986, respectively. He continued his research at Lund University within the area of applied robotics and industrial automation. The primary application areas have been arc welding and deburring operations, including issues related to sensors and to controlling these applications in an industrial context. He is also involved in developing concepts for service robotics, specifically within the area of healthcare and physically disabled persons. He currently holds a professorship in robotics at Lund University.

Torgny Brogårdh received the M.Sc. degree in electrical engineering and the Ph.D. degree in instrumentation technology from Lund University in 1971 and 1975, respectively. He became an assistant professor at Lund University in

1976. Between 1982 and 1986 he was an associate professor in instrumentation technology at the Royal Institute of Technology in Stockholm. He is currently a robot motion control specialist at the robotic unit of ABB Automation Technology. He has been granted more than 60 patents, and in 1998, he received the Swedish Royal Academy of Engineering gold medal for his contributions to the robotics development at ABB.

Per Cederberg received an M.Sc. degree in mechanical engineering and a Ph.D. degree in robot technology from Lund University, Sweden, in 1988 and 2004, respectively. His main research interests are in the areas of sensor-guided robots and simulation of robotic off- and online systems, focusing on one-off manufacturing models. He is currently working on interactive satellite TV distribution systems in Los Angeles, California.

Mats Isaksson received the M.Sc. degree from Lund University in 1997. He is currently a robot motion control specialist at the robotic unit of ABB Automation Technology.

Rolf Johansson, M.D., Ph.D., is a professor of control science at Lund University. In his scientific work, he has been involved in research in adaptive system theory, mathematical modeling, system identification, robotics, and signal processing. Since 1987, he has also had side appointments as a graduate advisor at the Faculty of Medicine, Lund University Hospital. Johansson was awarded the 1995 biomedical engineering prize (the Ebeling Prize) of the Swedish Society of Medicine for distinguished contribution to the study of human balance through application and development of system analysis and robotics. In 1993, he published the book *System Modeling and Identification* (Prentice-Hall).

Mathias Haage received the M.Sc. and Techn. Lic. degrees in computer science from Lund University, Sweden, in 2000 and 2005, respectively. His main research interests are in the areas of sensor-guided robots and manufacturing, focusing on process-oriented robot programming methods. He is currently pursuing the Ph.D. degree in computer science at Lund University.

Klas Nilsson received the M.Sc. degree in mechanical engineering from the Lund Institute of Technology, Sweden, and the Ph.D. degree in automatic control from Lund University, Sweden, in 1982 and 1996, respectively. Between 1982 and 1988 he worked at ABB Robotics Products in Västerås, Sweden, where he developed motion control software and algorithms. He is currently an assistant professor at the Department of Computer Science. Within both teaching and research, his main interests are software techniques for real-time systems and factory automation.

Magnus Olsson received the M.Sc. degree in mechanical engineering and the Techn. Lic. and Ph.D. degrees in robot technology from Lund University, Sweden, in 1992, 1996, and 2002, respectively. His main research interests are in the areas of computer-aided robotics and manufacturing, focusing on off-line programming and simulation of robots and their application processes. He is currently a research associate at the mechanical engineering department of Lund University.

Tomas Olsson received the M.Sc. degree in engineering physics and the Techn. Lic degree in automatic control from Lund University, Sweden, in 2001 and 2004, respectively. His main research interests are multisensor-based robot control, especially vision-based control and robotic force control. He is currently pursuing a Ph.D. degree in automatic control at Lund University.

Anders Robertsson received the M.Sc. degree in electrical engineering and the Ph.D. degree in automatic control from the Lund Institute of Technology, Sweden, in 1992 and 1999, respectively. He is currently a research associate at the Department of Automatic Control, Lund University. His research interests are in nonlinear control, real-time systems, and robotics, particularly the output feedback control problem and force feedback control for robot manipulators.

Jianjun Wang received his M.S. and Ph.D. degrees in mechanical engineering from Pennsylvania State University in 1999 and 2001, respectively. He is currently working for ABB Corporate Research Center in the United States as a robotics research scientist. His main research areas include sensor-based robot control, especially force control and visual servoing, robotized manufacturing process modeling, and advanced programming methods for industrial robots.

Address for Correspondence: Anders Robertsson, Department of Automatic Control, Lund Institute of Technology, Lund University, P.O. Box 118, SE-221 00 Lund, Sweden. E-mail: Anders.Robertsson@control.lth.se.