Proceedings of the 2003 IEEE
International Conference on Robotics & Automation
Taipei, Taiwan, September 14-19, 2003

DLR Hand II: Hard- and Software Architecture for Information Processing

S. Haidacher, J. Butterfass, M. Fischer, M. Grebenstein, K. Joehl, K. Kunze, M. Nickl, N. Seitz and G. Hirzinger

German Aerospace Center - DLR
Institute for Robotics and Mechatronics
E-mail: Steffen.Haidacher@dlr.de

Abstract — *In the robotic community more and more hands have been developed. These newly designed manipulators greatly outperform their ancestors in terms of available sensor signals, applicable grasping force, mechanical stability, reliability, kinematic design and more. This development extends the possible range and complexity of applications of robotic grippers also to areas outside of well structured laboratories and simple tasks. It also calls for more flexible control structures to provide a framework for implementing and executing these newly arising tasks without having to start from scratch for each new task. During the last few years we developed a control system architecture for DLR Hand II that proved to be useful for a great variety of different applications. This paper presents the basic ideas behind DLR Hand II's hard- and software architecture adapted to new needs in data processing.*

1 Introduction

In the last few years, lots of robotic hands have been developed [3, 10, 7]. These hands are now generously equipped with sensors, their reliability increased significantly and through studies of human grasping advances have been made in kinematics and anthropomorphic behavior [9]. On the other hand, algorithms and demands on applications for robotic grippers have increased [6, 1, 8]. In order to handle this complexity of hardware, sensors and applications, an appropriate hard- and software structure has to be provided [5]. The recently developed DLR Hand II [4]

Figure 1: DLR Hand II.

Hardware Level
Communication Protocol Level
Data Processing Level
Lower Controller Level
Higher Controller Level
External Comand Level

Figure 2: Communication Levels

provides an excellent platform for the development of future grasping strategies. The hand system's information processing architecture consists of hardware, software and communication structures. In the scope of the hardware, to minimize cabling and weight as well as to preserve extendibility, the hand uses a serial communication system, allowing the hand to be integrated into any robot system, in particular in different generations of DLR's lightweight robot with a customized tool adapter. The result is a system that is both compact (cf. figure 1) and easy to use. The communication system of the DLR Hand II has been structured as modular as possible in order to provide for easy access to measured data, simple maintenance and quick replacement or enhancement of parts of the system to adapt to new needs in experiments and applications. In this context data processing has been parted in multiple levels of abstraction, which in most cases are implemented as a single module in either hardware, software or both (cf. fig. 2). In the lowest *Hardware Level*, data is collected and conditioned in analog, time-continuous hardware. Section 2 describes the hardware basis needed for this communication level. The second *Communication Protocol Level*, concerned about control of the data collection process and the logical transmission of digitized sensor values to higher levels is described in section 3. Both levels are shaded dark in fig. 2. Finally on levels three and higher (shaded bright), software implementation of algorithms is applied to process measurements and produce control signals at multiple levels of complexity. This architecture is introduced in section 4.

2 Hardware Level

This lowest level of data processing consists of analog time-continuous or quasi time continuous hardware. Figure 3 gives an overview of the steps of hardware data processing and its interface to higher levels, mostly implemented in software. Gathered experience

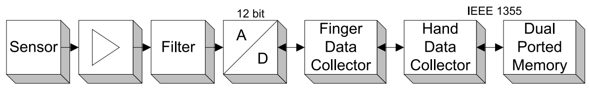

Figure 3: Sensor signal routing

with DLR Hand I [3] suggested a redesign of the then used concept of data flow. In the use of DLR Hand I, power dissipation and space requirements of the electronic parts became a major issue. Additionally, an increase in performance could be reached by reducing the electronic components moved by the robot. Thus, former digital signal processors were exchanged with off-the-shelf control hardware components outside of the DLR Hand II (cf. sec. 2.3). Reduced power dissipation inside the hand itself hence rendered active cooling obsolete. As surplus and in compliance with modular design considerations, the capacity of the external processor can be customized without an expensive redesign. In order to realize the shift of computational power, the data in DLR Hand II is now collected from the sensors and conditioned for further processing (cf. sec 2.1), especially as it has to be transported further than in DLR Hand I. Wiring effort in and towards the hand had to be kept low to use the hand flexibly at different robot systems. Thus, the sensor data in DLR Hand II is collected by a network of FPGAs and sent serially to a higher level digital signal processing controller. The serial data transport is governed by a protocol implemented on the *Communication Protocol Level* (cf. sec. 3). The use of FPGAs allows a flexible structure of data collectors. Any new data source can be integrated in the system. This was used to include also the reconfigurable palm. The data is made available to the *Data Processing Level* and higher levels on the control computers through a dual ported memory board that allows mutual access to data from the FPGA network as well as from external processor hardware. Finally, a pluggable connection in the data line between hand and robot had to be realized in order to guarantee the exchangeability of hands on different robot arms (cf. sec. 2.4).

2.1 Data Collection and Signal Conditioning Hardware

In order to allow sensitive interaction with the environment, DLR Hand II has been equipped with hall sensors and potentiometers for position and velocity measurement, with strain gauges for joint torque readings, a in-house developed six-dimensional force/torque sensor in the fingertip and temperature sensors for compensation purposes and system monitoring. The effort to collect and condition a given sensor signal is dictated by the intended use of this data, but generally the path depicted in figure 3 is followed. The physical data of the sensor, e.g. the voltage output of a specially designed potentiometer board based on conductive plastic, is led to the sensor signal preprocessing. This is located directly beside the sensor itself and consists of an impedance converter and an instrumentation amplifier with a Butterworth low pass filter of first or 3rd order. This circuit is followed by a 12 bit A/D converter, which is located as close as possible to the sensor signal conditioning circuits in each finger link in order to keep wiring effort and noise induction low. The converters are sampled asynchronously at a rate higher than 10 KHz, which is more than ten times faster than the actual control system clock rate, guaranteeing a quasi continuous data stream and leaving space for further extensions. The data is sent through a serial communication link to a *finger data collector* implemented in an FPGA. This chip also provides appropriate interface logic for every sensor and A/D converter. Each *finger data collector* routes its data serially to the *hand data collector* which reroutes all collected hand data to the dual ported RAM board. The serial link between the two data collectors is galvanically decoupled by opto couplers. The route is depicted in figure 5. For simplicity, all sensors irrespective of their actual need of resolution use the same 12 bit A/D converters. Therefore, each moving link of the finger is equipped with one 8-channel 12 bit A/D converter. The non moving base board of each finger houses two of these A/D converters. In total, this results in five converters per finger providing 40 channels. Currently 25 channels are used, leaving 16 channels available for further applications like additional tactile sensors. Further signal conditioning, e.g. temperature compensation for torque sensors, is done in external control computers on the *Data Processing Level*.

2.2 Command Signal Distribution

Control values for the motors coming from the control software are distributed in the reverse way of the sensor signal collection. To separate the power electronics from signal processing devices an additional galvanical decoupling is introduced between the finger controller and the power electronics.

2.3 Controller Hardware

In order to be easily scalable and to guarantee simple access to both code and data of the controller, all higher level data processing was implemented on two industrial PowerPC VME-bus boards. For communication with external computers and data sources, both boards are provided with LAN connection and serial interface. Both VME boards run the commercial real-time operating system VxWorks. In this setup, user-software can be designed nearly independent of the underlying control/computer hardware, easily be started and monitored in a Unix-shell like manner on a desktop computer.

2.4 Customized Tool Adapter and Hand Supply

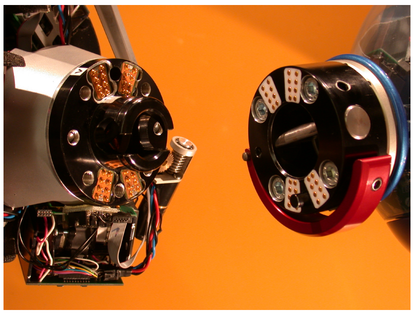

Figure 4: Customized tool adapter

The DLR Hand II is equipped with a specially designed tool adapter for quick mounting as shown in figure 4. This adapter can be used to exchange the hand for a different tool with different applications. Also, to provide higher availability, the hand in use can be replaced by a freely exchangeable spare hand, an advantage of the modular information processing architecture of the DLR Hand II. The adapter itself consists of a robot-side fixture and a tool-side plug. The adapter is designed to not only attach DLR Hand II but also other tools used in combination with two generations of our light weight robots. The customized tool adapter is based on a commercially available mechanical coupling which has been modified also implementing 32 electrical connections. Twelve of them are used by the hand at the moment leaving capacity for different tools. Eight of the twelve contacts dedicated to the hand, are used for communication between the hand and external control computers, four contacts provide electrical power. The communication lines consist of parallel signal lines two directed up and two down. Both are transmitted differentially. The four power supply lines include two lines of DC power for the actuators and two lines of AC power for the electronics. For the latter, we use a 50 V 20 kHz power supply and tiny transformers with AC/DC converters to provide the hand with different needed voltages. Galvanic decoupling was achieved using different secondary windings on the transformer core. Completing this concept, DLR light weight robots II and III provide internal cables for hand communication and power supply. Therefore no external lines are necessary on the arm. Using this customized tool adapter, the hand can be replaced without effort. To replace the hand, it is enough to turn the motor power supply off, open the latch of the tool adapter, replace the hand, close the latch, and turn on the motor power supply.

2.5 EEPROM for calibration data

Calibration parameters for sensors vary in-between sensors as well as between fingers. To maintain modularity and exchangeability of hardware, these parameters are stored in the individual fingers themselves. Therefore each finger is equipped with an EEPROM to store the parameters for the corresponding finger. Upon powering up the system, the parameters are read.

3 Communication Protocol Level

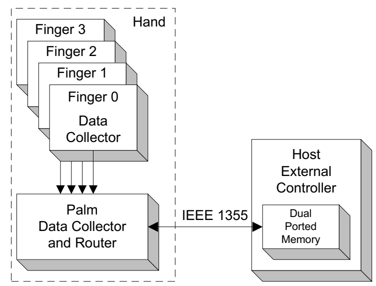

Figure 5: Serial Communication Hardware.

In order to collect and transmit data, a hierarchical serial communication system has been implemented in the palm and the fingers of DLR Hand II (s. fig. 5). The serial communication inside the hand and to the external control computers is written in VHDL and implemented in FPGAs according to the IEEE 1355 standard, which specifies a slim protocol layer, prevents collisions, and supports different physical transmission mediums. According to the IEEE 1355 DS, the character links of the network are realized with data and strobe lines in both directions resulting in four lines as described in sec. 2.4. A special heartbeat event activates the data collector to take the shared memory data by a package and send it to the host. There is no priority mechanism for data packages defined in IEEE 1355. Therefore, a special character with high priority is used to transmit the heartbeat event to all data collectors for real-time system synchronization. A host interface controller with a dual ported memory interface to the host CPU handles the incoming packages with the sensor values, prepares the outgoing packages with the actuator values to send them to the data collectors, and generates the heartbeat event to synchronize the system. In the actual implementation of DLR Hand II, a physical data rate of 10 Mbits allows a full cycle sampling sensor data, computing and setting actuator values every millisecond.

4 Control Software Architecture

The multilevel, modular structure of the whole hand system is also introduced in the hand's software architecture. As shown in figure 6, data processing in soft-

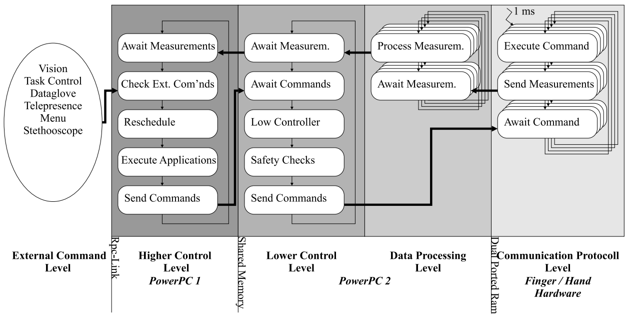

Figure 6: Logical levels of signal flow in DLR Hand II

ware is realized in four more major logical levels. In figure 6 levels are increasing from right to left with the *Communication Protocol Level* being rightmost. The following *Data Processing Level* performs all computations needed to convert digitized sensor values to applicable measurements in SI units, e.g. position and torque, and hereof derived values like velocities. Following is the *Lower Controller Level*, which performs basic control, system monitoring and safety tasks concerned with individual fingers, joints and sensors. The fourth software level, the *Higher Controller Level* implements coordinated controllers for all fingers and performs basic operations that can be used by external command generators from the highest *External Command Level*. All the levels shown in this picture represent digital, time-discrete levels with the signals being transmitted at a sampling rate of 1 kHz.

4.1 Data Processing Level

In this level for each finger, one independently scheduled task waits for newly arrived sensor data signalized by interrupt. In most cases, the data has to be processed to be interpreted by other tasks. Most signals can be converted using a linear measurement function $processedValue = gain * (rawValue - zeroOffset)$. However some signals can better be fit by higher order polynomials or special computations. For example, position measurements from two hall sensors $\tilde{A}$ and $\tilde{B}$ measuring the magnetic motor field 120^o apart can be computed according to:

$$\theta = atan2(\tilde{A} - \bar{A}, \beta * (\tilde{B} - \bar{B}) + \alpha * (\tilde{A} - \bar{A}))$$

with $\bar{A}$, $\bar{B}$ being the mean value of the respective signal and α, β are correcting factors for $atan2$ because the sensors are 120^o apart instead of 90^o. For computation of velocities, position measurements of high resolution, the hall sensor signals, are fed to a Kalman filter to clear out disturbances and usual ripples in motor hall readings.

4.2 Lower Controller Level

In this level, the controller loop awaits a heartbeat interrupt and processed measurements of all fingers to synchronize with the hardware clock. In a second step, joint or Cartesian commands are received from the *Higher Controller Level*. Next to these command values, control values are also received, which govern the choice and parameterization of controllers and the on-off status of motors etc. Consecutively, one step in the desired type of controller is performed. Implemented types of controllers contain joint level impedance control with friction and gravity compensation, zero torque control and two different implementations of Cartesian control. In contrast to higher levels, in this *Lower Controller Level*, no smooth transition between the different types of controllers is provided, since almost no experimental setup requires their online change. Finally, extensive safety checks are performed including joint and temperature limits, sensor signal validity and communication system integrity. Resulting errors are reported at a lower priority in an extra task to ensure proper time behavior also in faulty cases. In case of a major system fault, a post mortem dump of sensor values and commands is possible to help reconstructing and removing errors. The special cardanic base joint of DLR Hand II (cf. [4], section 3.3) requires at least joint limitation violation checks at this high logic level, because workspace limitations have to be tested in true joint angles rather than the directly measurable motor or transmission angles. Joint and motor angles are related through a coordinate rotation. Through this rotation, the easily testable rectangular workspace in joint coordinates is represented by a polygon in motor coordinates (cf. figure 7).

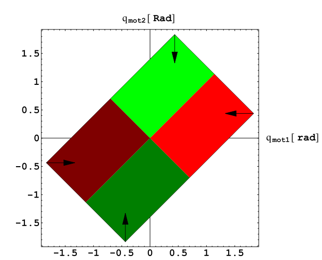

Figure 7: Limits of joints transformed in motor angles

4.3 Higher Controller Level

In this level basic skills are provided like coordinated controllers and higher level data processing necessary to perform complex tasks. These basic skills comprise for example detection of contact, estimation of the velocity of a grasped object or an interface to a data glove. Examples of complex tasks as performed in experiments at DLR are presented in a companion paper [2]. In contrast to the *Lower Controller Level*, here it is urgently necessary to be able to switch from one type of control to another in a smooth transition without having to terminate one control type first. This for example makes approaching, grasping and manipulating objects feasible at all. For this reason, operations and controllers spawned in this level are controlled by a specially designed scheduler, governing a state machine for each operation, in this context called an *application*. Context dependent security checks have to be performed on this level. With a sampling rate of 1 KHz this can be done even without additional interrupts.

4.3.1 Application Scheduler

In this context, two types of applications are distinguished: *commanding applications* and *assisting applications*. The first kind comprises for example closed loop controllers and different kinds of reference position generators, e.g. data glove interfaces. There can be only one active *commanding application* at a time. The latter type includes all applications providing and processing data for one *commanding application*. There can be multiple *assisting applications* active at once. A schematic of the state machine of applications is illustrated in figure 8. Each application

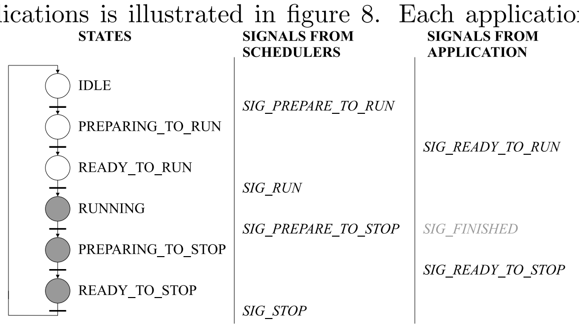

Figure 8: Petri Net of Application

has to implement six states: *IDLE*, *PREPARING_TO_RUN*, *READY_TO_RUN*, *RUNNING*, *PREPARING_TO_STOP* and *READY_TO_STOP*. In the last three states (marked grey in the figure) the application is required to produce valid commands or output data. These states are active states in contrast to the remaining three passive states. On the other hand, the states *PREPARING_TO_RUN* and *PREPARING_TO_STOP* are preparation states whereas the states *READY_TO_RUN* and *READY_TO_STOP* are ready states. When a change of applications is requested from an external source, the scheduler requests all running applications that have to be terminated and all idle applications that have to be started into the respective preparation state. Now all the applications perform their respective preparations, i.e. get ready and adjust internal values for later taking over or shut down and initiate safe hand over to other applications. Each application can indicate its readiness by issuing the appropriate signal and change to the respective ready state. When all applications have reached their ready state the active applications can be scheduled from *READY_TO_STOP* to the passive state *IDLE* and passive applications can be switched from *READY_TO_RUN* to *RUN* by issuing the respective signal. As a special case, a *commanding application* can signalize a desire to stop when all its jobs are finished. In this case all applications go to state *IDLE* and there exists no active *commanding application*. Table 1 lists all states and their meaning.

4.3.2 Program Flow of the Higher Controller Level

The described scheduling mechanism is included in the program flow of level five. The control task awaits measurements from the *Lower Controller Level* and hereby indirectly synchronizes to the hardware clock. Consecutively it checks for commands from external sources, reschedules and executes applications and sends the resulting commands. Interfaces to external command generators are constructed as separate RPC or serial interface drivers that communicate through an *assisting application* with the control task. This way an easy expandability to new external command generators as well as to new *commanding applications* is achieved.

4.4 External Command Level

The developed modular level concept was designed mainly to perform multiple different tasks on a high abstraction level. These tasks can be implemented in this level. The realization can be done on any computer platform. The *External Command Level* is provided with interfaces to the lower levels through a RPC connection. Realized interfaces so far include a connection to any scripting language, e.g. shell scripts to change and parameterize applications, command positions and receive data. Second a graphical task oriented programming system has been connected to DLR Hand II as well as several control and monitor menus to evaluate the state of the system. Using these basic interfaces, many applications have been developed, amongst others catching a soft ball, following an object, autonomously grasping glasses and bottles and teleoperation experiments. (cf. [2])

5 Conclusion

Based on our long experience with robotic hands and the great demand for a flexible working and operation environment for DLR Hand II we developed a modular hard- and software architecture for information processing. This structure is based on six levels of increasing complexity and abstraction and decreasing direct contact to hardware. Through this modular

State	Requirement for Application
PREPARING_TO_RUN	get ready and adjust internal values for taking over control
READY_TO_RUN	stay ready and await to be set active
RUNNING	produce valid active output
PREPARING_TO_STOP	shut down and initiate safe hand over to possible other application
READY_TO_STOP	stay ready and await to be set passive
IDLE	passive, no special behavior is required

Table 1: Meaning of Application States

design we are able to realize a wide variety of applications. These applications have to be analyzed and the respective elements on each level must be implemented. It is possible for example to complete tasks that consist of multiple different operations, like locating a room, opening its door with the hand, locating an object in this room and grasping the object. Through exchangeability and inherent switch mechanisms other tasks can be implemented on the same system like observing and catching a thrown ball without developing and maintaining two independent lines of data processing.

A New Finger-Tip Sensor

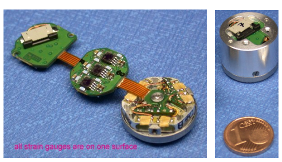

Figure 9: Tip sensor

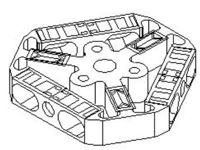

Figure 10: Structure

A miniaturized six-component force/torque sensor (20mm in diameter and 16mm in height, cf. figure 9) with full digital output has been developed for the fingertip. It needs only 6 wires including power supply for the high speed data transmission(15.6 kHz sampling frequency). Based on a former design, the elastic body is made from only one part and all strain gauges are on one surface (figure 10), rendering the sensor extremely flat and very appropriate for thin film technology of strain gauges for easier assembly. A signal processing circuit and high speed serial A/D converter(12 bit) are also integrated in the sensor. The force and torque measure ranges are 30N and 200Nmm respectively. Also a 200% mechanical overload protection is provided in the structure.

References

[1] A. Bicchi and V. Kumar. Robotic grasping and manipulation. In S. Nicosia, B. Siciliano, A. Bicchi, and P. Valigi (eds.), editors, *Ramsete: Articulated and mobile robots for services and Technology*, volume 270, chapter 4, pages 55–74. Springer-Verlag, Berlin Heidelberg, Germany, 2001. .

[2] Ch. Borst, M. Fischer, S. Haidacher, H. Liu, and G. Hirzinger. Dlr hand ii: Experiments and experiences with an anthropomorphic hand. In *submitted to ICRA 2003*, 2003.

[3] J. Butterfass, G.Hirzinger, S. Knoch, and H. Liu. DLR's Multisensory Articulated Hand. part I: Hard- and software architecture. In *Proc. IEEE Conf. on Robotics and Automation*, pages 2081 – 2086, Leuven, 1998. .

[4] J. Butterfass, M. Grebenstein, H. Liu, and G. Hirzinger. DLR-Hand II: Next Generation of Dextrous Robot Hand. In *Proc. IEEE Conf. on Robotics and Automation*, pages 109 – 114, Seoul, Korea, May 2001. .

[5] E. Coste-Manire and R. Simmons. Architekture, the backbone of robotic systems. In *Proceedings of the IEEE International Conference on Robotics and Automation April 2000, San Francisco, California*, 2000.

[6] S. Haidacher and G. Hirzinger. Contact point identification in multi-fingered grasps using kinematic constraints. In *Proceedings of the IEEE International Conference on Robotics and Automation May 2002, Washington,D.C. ,USA*, 2002.

[7] C. S. Lovchik and M. A. Diftler. The robonaut hand: A dextrous robot hand for space. In *Proc. IEEE Conf. on Robotics and Automation*, pages 907 – 912, Detroit, Michigan, USA, May 1999.

[8] L. Petersson, P. Jensfelt, D. Tell, M. Strandberg, D. Kragic, and H. I. Christensen. System Integration for Real-World Manipulation Tasks. In *Proc. of IEEE Intl. Conference on Robotics and Automation*, pages 2500 – 2505, Washington DC, USA, May 2002.

[9] Nancy S. Pollard and Richards C. Gilbert. Tendon Arrangement and Muscle Force Requirements for Humanlike Force Capabilities in a Robotic finger. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 3755 – 3762, Washington, DC, USA, May 2002. .

[10] S. Schulz, Ch. Pylatiuk, and G. Bretthauer. A new ultralight anthropomorphic hand. In *Proc. IEEE Conf. on Robotics and Automation*, pages 2437–2441, Seoul, Corea, May 2001.