A Humanoid Two-Arm System for Dexterous Manipulation

Ch. Ott, O. Eiberger, W. Friedl, B. Bäuml, U. Hillenbrand, Ch. Borst,
A. Albu-Schäffer, B. Brunner, H. Hirschmüller, S. Kielhöfer, R. Konietschke,
M. Suppa, T. Wimböck, F. Zacharias, and G. Hirzinger

German Aerospace Center (DLR e.V.)
Institute of Robotics and Mechatronics
82234 Wessling, Germany
Email: christian.ott@dlr.de

Abstract— **This paper presents a humanoid two-arm system developed as a research platform for studying dexterous two-handed manipulation. The system is based on the modular DLR-Lightweight-Robot-III and the DLR-Hand-II. Two arms and hands are combined with a three degrees-of-freedom movable torso and a visual system to form a *complete* humanoid upper body. In this paper we present the design considerations and give an overview of the different sub-systems. Then, we describe the requirements on the software architecture. Moreover, the applied control methods for two-armed manipulation and the vision algorithms used for scene analysis are discussed.**

I. Introduction

The development of humanoid robotic platforms has always gained great attention. In recent years the advances in mechatronics and robotics have enabled the development of a large number of humanoid robots [1], [2], [3], [4], [5]. This also has lead to a rapid growth of the humanoid research community. The popularity of humanoids may mainly be due to their attraction to the public and the speculations about the moment they reach human performance. But from a robotics (engineering) point of view they also have great attraction as a challenging and intuitively coherent testbed for robotics and mechatronics research. Humanoids do not only drive the development of special fields like the design of highly integrated miniaturized mechatronic components or the development of robust control methods. They also require an integrated design and development process to handle the system complexity. Therefore, human performance serves as an intuitive and principled reference for specialists on different fields.

So far, attention has mainly been directed towards the locomotion of humanoid platforms. Therefore, systems that can dynamically walk, lie down and stand up again are now available [2], [4], [5]. In contrast, the manipulation skills of these systems are usually relatively poor. Most have only simple and light gripping devices. Often the allowed payload for the arms is quite small. To tackle human performance in tool usage and manipulation, strong but small arms and flexible gripping devices like dexterous hands seem necessary [6], [7] on the hardware side. On the software side robust control methods for two arms and hands as well as perceptual capabilities have to be developed and verified.

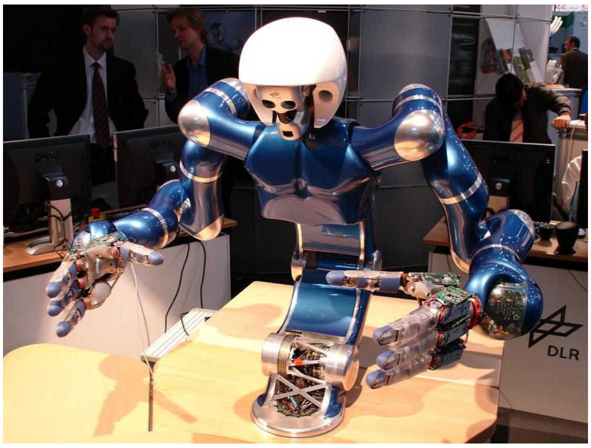

Fig. 1. System overview.

Based on our experience in developing lightweight robotic arms [8] and dexterous hands [9] we have designed a two arm/hand system with a movable torso as a research platform to contribute to the manipulation skills of humanoid robots. As we are well aware of the system complexity we have concentrated on the upper body and mounted the system on a table at first. To allow for the verification of different manipulation and control strategies the system architecture is kept as flexible as possible. The total of 43 degrees-of-freedom (DoF) of the system can be controlled either within one special manipulation controller in a 1 kHz control loop or with separated controllers for the arms, the hands, and the torso.

The paper is organized as follows. In Sec. II the design considerations for our humanoid platform are summarized. The mechanical design is discussed in Sec. III. The software structure is described in Sec. IV. An overview of the control methods for object manipulation with two arms is offered in Sec. V. Section VI briefly touches upon the vision algorithms used for scene analysis. A summary is given in Sec. VII.

II. Design Considerations

The concept of our humanoid manipulator ”Justin”[1] is based on the modular 7-DoF DLR-Lightweight-Robot-III (DLR-LWR-III) [8] and the four-fingered DLR-Hand-II [9]. It is designed as a versatile platform for research on two-handed manipulation and service robotics in standard human environments.

For the mechanical design the following requirements have been taken into account: The system should be able to reach objects on the floor as well as objects on a shelf up to a height of about 2 m. It should have an anthropomorphic kinematic configuration for research on bi-manual grasping. Since in future versions the system may also operate on a mobile platform, it has to be slim enough to pass standard doorways of about 90 cm width. The integration of link-side torque sensors in the joints has already proved very useful for the arm and the hand, and therefore should be maintained throughout the system. Finally, a sensor head mounted on a 2-DoF pan-tilt-unit should be included in order to allow for scene analysis based on stereo vision and perhaps other range sensors [10].

In particular, the modular concepts of the DLR-LWR-III and the DLR-Hand-II are exploited by building the system symmetrically with a right-handed and a left-handed subsystem. Figure 2 shows a visualization of the system in four different configurations. Besides two-handed manipulation on the table, the movable torso also enables the system to reach objects on the floor and to grasp highly elevated objects. As the system is prepared to be integrated with a mobile platform, it is worth noting that the backward *seated* position allows for a lower center of gravity. This is useful to prevent tipping over in curves despite the overall weight of approximately 45 kg. Mounted on a 60 cm table or platform the torso reaches a *human-like* shoulder height of up to 150 cm. Therefore, the hands can reach a vertical range similar to a human despite lacking the leveling function of human legs.

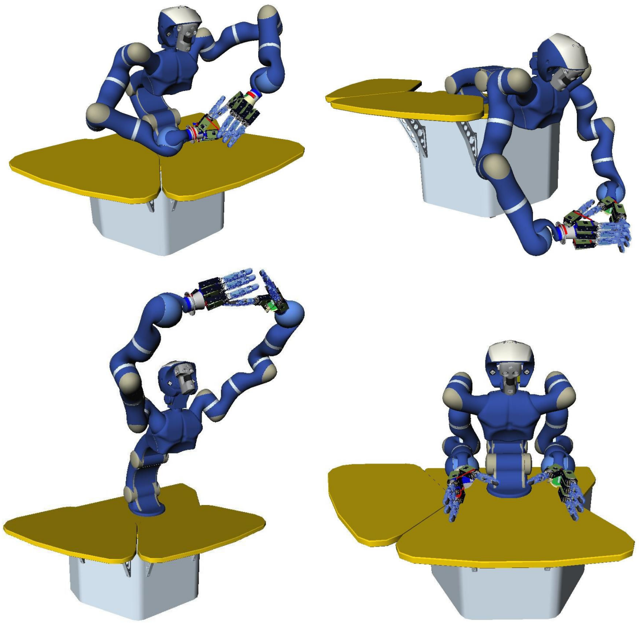

Fig. 2. Different configurations: manipulation on the table, grasping of objects on the floor, grasping of elevated objects, and a *seated* configuration for transport.

Table I gives an overview of the 43 actuated DoF. In the following, the mechanical design of the different sub-systems is discussed in more detail.

TABLE I
System Overview

Sub-system	DoF
Torso	3
Arms	2 x 7
Hands	2 x 12
Head & Neck	2
$\sum$	43

III. Mechanical Design

A. Arms and Hands

The DLR-LWR-III is a 7-DoF service manipulator arm which has integrated torque sensors in each of its joints [8]. These sensors allow for the implementation of highly sensitive torque and impedance controllers which take account of the joint flexibility. When moving with maximum velocity the arm can carry a load of up to 7 kg, while for slow motion a payload of up to 15 kg is admissible. The overall weight of the arm itself is about 14 kg. The first four joints are in roll-pitch-roll-pitch configuration with kinematics and ranges similar to the human arm. The wrist is in roll-pitch-pitch configuration. The arms are mounted on the torso in a 60 degrees tilted configuration, see Fig. 1, allowing for an operation of the arms far from joint limits and singular configurations. Notice that this also allows to move the elbows under the shoulders (see the *seated* configuration in Fig. 2) and the arms may work in front as well as in the backside of the torso.

As dexterous hands the four-fingered DLR-Hand-II is used in a right-handed and a left-handed configuration. The DLR-Hand-II is equipped with joint torque sensors and a 6-DoF force/torque sensor in each of the finger tips. Each hand has 3 DoF per finger and a reconfigurable palm offering configurations for power grasp and precision grasp. Further details on the hand can be found in [9].

B. Torso

The torso consists of four moving axes, three of which are actuated. A first actuated roll axis is followed by two actuated pitch axes. These three joints define the posture of the torso. Similar to the technology of the DLR-LWR-III these joints are equipped with Harmonic Drive gears and strain-gauge based joint torque sensors. Each of the actuated joints is controlled by a local signal processor allowing for the implementation

[1] The system was named ”Justin” because it was finished *just in time* for the *Automatica* fair 2006 in Munich.

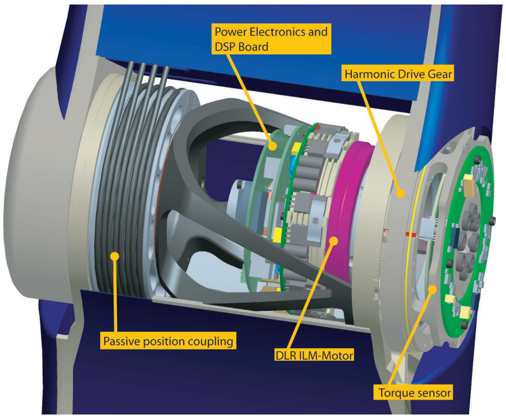

Fig. 3. Assembly of the third torso joint. The *passive position coupling* on the left-hand side indicates the cable actuation of the passive fourth joint.

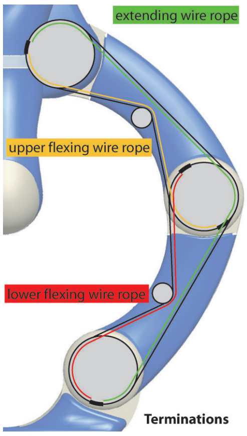

Fig. 4. Sketch of the cable actuation of the fourth torso joint.

of fast inner joint position and joint torque control loops with 3 kHz sampling rate. A serial digital bus connects these signal processors to a central computing unit which receives all measurements of the system at a sampling rate of 1 kHz (see also Section IV). As an example the assembly of the third joint is shown in Fig. 3.

The tilt of the chest is fixed via a fourth pitch joint. This joint is coupled to the base via tendons, such that the tilt resulting from the preceding two joints is compensated. Figure 4 shows a sketch of the tendon actuation. Any torque around the rotation axis of the fourth joint is transmitted to the base due to this construction, and not to the preceding joints. This allows to use smaller gears as well as torque sensors with higher resolution for the actuated joints.

The first joint ranges from -140 to +200 degrees enabling the robot to reach also objects at the back side. The second joint ranges from zero (straight upright) to ±90 degrees. The joint limits of the third joint stem from the tendon actuation and depend also on the second joint. This joint may move in the range $[\max(0, -q_2), 135°]$, where q_2 is the second joint angle. The fourth joint, as mentioned, is passively coupled to the first. Together both active pitch joints allow an operating area of 30 cm behind to 60 cm ahead of the first rotation axis and of 21 cm below to 60 cm above the first pitch joint. A sectional view of the resulting reachable workspace of the torso is shown in Fig. 5.

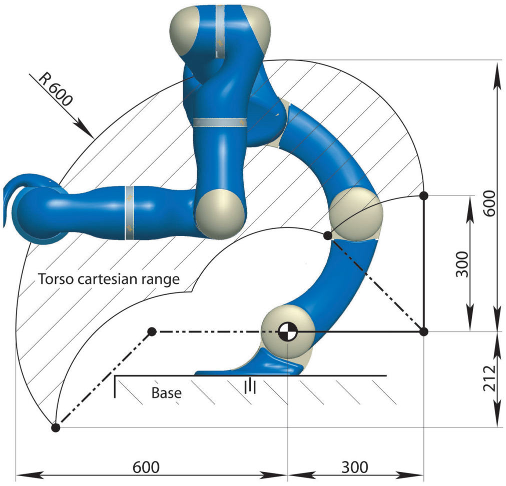

Fig. 5. Workspace of the torso.

C. Head

The head consists of an articulated visual system, which allows for scene analysis, tracking of objects, and world model generation. Due to the versatility of environments Justin might encounter, a multisensory approach for the vision system was chosen. Laser-sensing provides high accuracy in the short to mid range, whereas stereo triangulation is more suitable for large distance measurements. Texture acquisition enhances realistic world modeling, useful for all kinds of service robotics applications. Generally speaking, data fusion of range and object information with this variety contributes complex environment perception for path planning, navigation, model generation, and object recognition.

These considerations supported the use of a multisensory head, called *3D-Modeller*, as vision system. In early work [10], the 3D-Modeller was used for hand-guided applications. A redesigned 3d-Modeller, now connectable to a robot system, suited the system for use with Justin. It integrates a *laser-range scanner*, a *laser-stripe profiler*, and a *stereo camera sensor*. Internal electronics allow for sensor synchronization. Local computational power (embeddedPC running Linux) enables pre-processing as well as parameter adjustment for the sensors. Data is transfered via FireWire, which also supplies power to the 3D-Modeller. The design considerations led to the sensor particularities described in Table II.

IV. Software Architecture

The software architecture of Justin is based on the 'agile Robot Development' (aRD) concept [11] we have developed at our institute. The aRD concept is a flexible, pragmatic and distributed software concept designed to support the development of complex mechatronic and robotic systems. It

TABLE II

TECHNICAL DATA OF THE 3D-MODELLER'S SENSOR COMPONENTS

Sensor	Laser-range scanner	Dual Laser Stripe Profiler	Stereo Camera Sensor
Principle	triangulation	image processing and triangulation	stereo vision
Range $[mm]$	$50 - 300$	$150 - 500$	$250 - 2000$
Resolution $[mm]$	$0.1 - 2.5$	$0.3 - 2.5$	$2 - 50$
Base distance $[mm]$	20	103	50
Field of view	$270°$	58°(f=6mm) 30°(f=12mm)	58°x 44° 30°x 22°

gives easy access to scalable computing performance and is based on the abstract view of a robotic system as a decentral 'net of calculation blocks and communication links'.

A. Software Design

Justin consists of a number of complex robot components. The complexity of a single robotic component stems from the high number of DoF and sensors and the necessity of sophisticated control algorithms requiring considerable computational power. In the overall system the complexity naturally arises through the delicate and tight interaction of its sub-systems. This makes scalable computational resources an essential requirement, both for the non-realtime and hard realtime parts.

Fortunately, the currently available computing power of commodity systems together with their fast communication links and high-speed buses connecting the robot components allows for a completely new solution for the system architecture of a robot system. It offers the possibility to realize the above mentioned *decentral net* of calculation blocks and communication links, resembling the view of component-based software engineering.

This component-based approach is particularly well suited, since complex robot systems are usually developed by teams of researchers. Additionally the team is heterogenous with experts from various fields each demanding their specific standard tools. Therefore, an appropriate software concept should support a simple and easy way of integrating such tools.

Moreover, for building a complex system it is highly advantageous that the software concept supports rapid prototyping to allow an iterative development process. Availability of modern graphical development tools provides additional means of simplifying software handling and structuring.

B. Software Overview

The robot components of Justin (arms, hands, torso, ...) each have a separate fast serial digital bus with a bandwidth $>$ 4Mbit/s, which allows to read sensor values and write actuator commands with a rate of 1kHz. The robot components are connected to a realtime target running the controller blocks at a rate of 1kHz for all 43 DoF beside other realtime blocks, like the inverse kinematics or the collision detection. The realtime target is currently a PC with Dual-Pentium 4,3 GHz, but can be easily scaled up to a cluster of PCs as more computation performance is needed.

The realtime part communicates over switched gigabit ethernet and point-to-point gigabit connection with non-realtime blocks. Those blocks implement user interaction (e.g. 3D-viewer, GUI) and higher level intelligence (e.g. vision system, path planning) and run distributed on a net of PCs.

Additional hosts run tools for development (edit-compile-debug) and tools for monitoring and profiling of the different parts of the system during runtime.

The system can also be used as a simulator when simply replacing the real robot components by blocks simulating the robot dynamics. As the other parts of the system are unchanged and the simulation of the robot dynamics can also be run on the realtime target, because of easy scalability of computation performance, the simulator can resemble the behavior and timing of the real system very accurately (only depending on the quality of the model of the robot dynamics).

C. Implementation

The current implementation of the aRD concept of a decentral net of calculation blocks and communication links consists of aRDnet, a simple software suite developed at our institute and a toolchain based on Matlab/Simulink/RTW[2] [12] and RTLab [13]. As operating systems (OS) we use QNX Neutrino [14], a POSIX-compliant microkernel realtime OS, for the realtime target, Linux for the non realtime computers, and Windows XP for the development hosts.

Matlab/Simulink is the quasi-standard tool for simulation of robot dynamics and controller design. A Simulink model resembles the functional view on a system as being a net of communicating blocks. With RTW it is possible to automatically generate from such a model executables running on a realtime target. Via RTLab even the semi-automatic parallelization of the model to run on multi CPU or even distributed computing resources is feasible.

Besides blocks that are part of the Simulink model and implement mainly controller-related functionality important parts of the net are made of standalone blocks. A standalone block is an individual process running an arbitrary application which sends and receives data packets. The aRDnet suite supports and standardizes the communication between blocks and consists of three parts. Firstly, a library for easy implementation of block input and output ports. Secondly, the `ardnet` executable for allowing communication between blocks running on different computers (currently using UDP sockets). Finally, a template for a Simulink stub block that allows to easily represent any standalone block in the Simulink model and so seamlessly integrates the two parts of the net of blocks.

While the system architecture is decentral it is still necessary to have a central description of the overall configuration and a mechanism for central startup. We achieve both points by using a hierarchy of shell scripts, similar to the startup process in Unix systems. By adding remote login tools like `rsh` and `ssh` the whole distributed system can be started from a single console.

[2] *RTW ... Real-Time Workshop*

D. Preliminary tests

The aRD concept already proved successful in the preliminary experiments reported in Section V-D. It could flexibly fit all computational needs, especially in the hard realtime part, brought up by the researchers. In addition, it allowed for an efficient and 'agile' development flow of the heterogenous team of experts, in the spirit of the 'Agile Software Development' methodology [15], [16].

V. Control

Since Justin is based on DLR's lightweight arms and hands, the control methods developed for these systems can serve as a basis. The following description focuses on the control of the arms and the torso, while the hands are treated as passive end-effectors. In the experiments reported in Sec. V-D the hands were controlled by joint impedance controllers. Concerning manipulation tasks the following two questions arise:

- How to cope with the joint elasticity due to the *Harmonic Drive* gears of the arms and of the torso joints?
- What is an useful/appropriate impedance behavior for a two-arm manipulator system?

The first question, how to cope with the joint elasticity, was treated in detail in our previous works on position-, torque-, and impedance control of flexible joint robots, see [17] and the references therein. These concepts form the basis for the impedance controllers in Sec. V-B which are designed especially for a two-arm manipulator system.

A. Treatment of the joint elasticity

The *Harmonic Drive* gears of the n joints of the arms and the torso present the main amount of elasticity. Accordingly, the reduced flexible joint robot model from [18] is considered for the controller design

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) &= \boldsymbol{K}(\boldsymbol{\theta} - \boldsymbol{q}) + \boldsymbol{\tau}_{ext}, \quad (1) \\ \boldsymbol{B}\ddot{\boldsymbol{\theta}} + \boldsymbol{K}(\boldsymbol{\theta} - \boldsymbol{q}) &= \boldsymbol{\tau}_m \ . \quad (2) \end{aligned}$$

Herein, $\boldsymbol{\theta} \in \mathbb{R}^n$ and $\boldsymbol{q} \in \mathbb{R}^n$ are the motor angles and link angles, respectively. The difference between the motor and the link angles correspond via the diagonal joint stiffness matrix $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ to the joint torques $\boldsymbol{\tau} = \boldsymbol{K}(\boldsymbol{\theta} - \boldsymbol{q})$. The matrix $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ is a diagonal matrix consisting of the motor inertia values. The components $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$, and $\boldsymbol{g}(\boldsymbol{q})$ of the link side rigid body dynamics (1) represent the inertia matrix, the Coriolis and centrifugal terms, and the gravity terms. Finally, $\boldsymbol{\tau}_m \in \mathbb{R}^n$ is the vector of motor torques which is the control input and $\boldsymbol{\tau}_{ext} \in \mathbb{R}^n$ is a vector of external torques which are exerted on the robot by the environment.

For the design of the controllers in [19], [20], [21] a passivity based approach was followed which endows the controller with advantageous robustness properties. The controller basically consists of two cascaded loops (see Fig. 6). In an inner loop a torque feedback controller of the form

$$\boldsymbol{\tau}_m = \boldsymbol{\tau}_d - \boldsymbol{K}_\tau(\boldsymbol{\tau} - \boldsymbol{\tau}_d) - \boldsymbol{K}_s\dot{\boldsymbol{\tau}} \quad (3)$$

with positive definite gain matrices $\boldsymbol{K}_\tau \in \mathbb{R}^{n \times n}$ and $\boldsymbol{K}_s \in \mathbb{R}^{n \times n}$ is used. Herein, the vector $\boldsymbol{\tau}_d \in \mathbb{R}^n$ is an intermediate control input corresponding to the desired torque from an outer loop impedance control law. In [19], [20] a detailed analysis[3] of this type of impedance controllers is given.

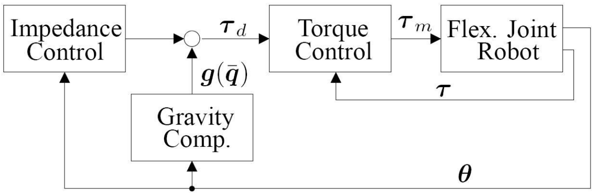

Fig. 6. Impedance controller structure for the arms and the torso.

The purpose of the inner torque feedback loop is twofold. On the one hand, the torque feedback causes a decrease of the effective motor inertia for forces acting on the link side [19]. Thereby it enhances the vibration damping effects of an additional outer control loop. On the other hand it also diminishes the effects of motor side friction since the joint torque sensors are placed on the link side.

In the next section an outer loop compliance behavior for the link side positions $\boldsymbol{q}$ will be designed by constructing a suitable potential function $V(\boldsymbol{q})$ which can be used together with an appropriate positive definite damping matrix $\boldsymbol{D}(\boldsymbol{q})$. In the following it is shown how this link side compliance can be combined with the underlying torque controller under consideration of the joint flexibility. Therefore, a *quasi-static* approximation $\bar{\boldsymbol{q}}(\boldsymbol{\theta})$ of $\boldsymbol{q}$ is computed which is a function of the motor side position only. According to [21] this function $\bar{\boldsymbol{q}}(\boldsymbol{\theta})$ can be chosen as the solution of

$$\boldsymbol{\tau} = \boldsymbol{K}(\boldsymbol{\theta} - \boldsymbol{q}) = \boldsymbol{g}(\boldsymbol{q}) - \frac{\partial V(\boldsymbol{q})}{\partial \boldsymbol{q}} \quad (4)$$

for $\boldsymbol{q}$. This equation ensures that statically the gravity compensation as well as the desired compliance relationship is fulfilled. Based on $\bar{\boldsymbol{q}}(\boldsymbol{\theta})$ the input $\boldsymbol{\tau}_d$ for the underlying torque controller is given by

$$\boldsymbol{\tau}_d = \boldsymbol{g}(\bar{\boldsymbol{q}}(\boldsymbol{\theta})) - \boldsymbol{D}(\bar{\boldsymbol{q}}(\boldsymbol{\theta}))\dot{\boldsymbol{\theta}} - \left(\frac{\partial V(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)_{\boldsymbol{q}=\bar{\boldsymbol{q}}(\boldsymbol{\theta})} . \quad (5)$$

More details on how to solve (4) and the stability and passivity properties of this controller design can be found in [21].

B. Cartesian compliance control

In impedance control of single manipulators the desired impedance is often chosen as a mass-spring-damper like system. From a practical point of view it is often sufficient to realize a desired stiffness and damping while the apparent inertia is not changed resulting in a *compliance control problem.* For Justin, a behavior is desired in which the two arms are controlled in a coordinated way rather than individually. A typical application for which such a coordinated controller is useful is the grasping of a large box.

[3] Therein a physical interpretation of the torque feedback was given in the sense that it scales the effective motor inertia from $\boldsymbol{B}$ to $(\boldsymbol{I} + \boldsymbol{K}_\tau)^{-1}\boldsymbol{B}$. This interpretation of torque feedback can be seized for the stability analysis.

A well known approach in the robotics literature for coordinated control of multiple robots, which also motivated our chosen control strategy, is the so-called IPC (Intrinsically Passive Controller) proposed by Stramigioli [22]. This controller was developed for the control of multi-fingered articulated hands and has an intuitive physical interpretation as it consists of several spatial springs which connect the end-effectors (i.e. finger tips) of a robotic hand with a virtual object. The controller structure for two-arm manipulation with Justin was designed similarly. Spatial stiffness components (and their respective potential functions) were used as building blocks in the controller design.

Firstly, we use two spatial springs $\mathcal{K}_r$ and $\mathcal{K}_l$ for the right and the left arm, which connect the end-effectors to the virtual equilibrium frames $\boldsymbol{H}_{r,d}$ and $\boldsymbol{H}_{l,d}$. If the springs are implemented as complete 6D-springs (i.e. with full rank (6×6)-stiffness matrices), then the complete Cartesian motion of the arms can be influenced already via only these two springs. Secondly, it is useful if some part of the compliance behavior is instead defined via an additional coupling spring $\mathcal{K}_c$ between the arms. As an example one might think of an impedance behavior in which only the rotational stiffness is implemented by the springs $\mathcal{K}_r$ and $\mathcal{K}_l$ while the translational stiffness is implemented via the coupling spring $\mathcal{K}_c$. Clearly, in such a configuration (as shown in Fig. 7) both the rest lengths as well as the stiffness values of the individual springs and the coupling spring should be chosen in a *compatible* way such that the springs do not interfere with each other. With this controller structure one can thus implement different behaviors, ranging from an independent control of the arms via $\mathcal{K}_r$ and $\mathcal{K}_l$ to a pure coupling based on $\mathcal{K}_c$ by choosing suitable stiffness matrixes of the individual spatial springs.

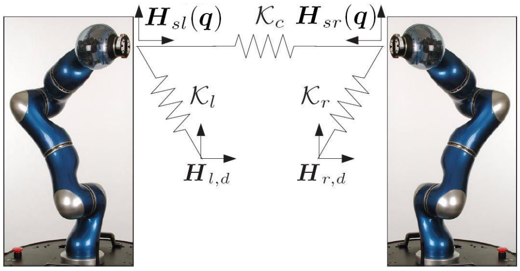

Fig. 7. Two-arm compliance behavior.

As described above, the controller structure basically consists of three spatial springs which connect the *desired* frames $\boldsymbol{H}_{r,d}$ and $\boldsymbol{H}_{l,d}$, the end-effector frame of the right arm $\boldsymbol{H}_{sr}(\boldsymbol{q})$, and the end-effector frame of the left arm $\boldsymbol{H}_{sl}(\boldsymbol{q})$ by pairs. In the following section an example implementation for these springs is discussed shortly.

C. Realization of the spatial springs

Consider a spatial spring between two general frames $\boldsymbol{H}_{sa} = [\boldsymbol{R}_{sa}, \boldsymbol{p}_{sa}] \in SE(3)$ and $\boldsymbol{H}_{sb} = [\boldsymbol{R}_{sb}, \boldsymbol{p}_{sb}] \in SE(3)$, where $\boldsymbol{R}_{sa} \in SO(3)$ and $\boldsymbol{R}_{sb} \in SO(3)$ denote the rotation matrices of the frames w.r.t. a static base frame $\boldsymbol{H}_s \in SE(3)$. The vectors $\boldsymbol{p}_{sa} \in \mathbb{R}^3$ and $\boldsymbol{p}_{sb} \in \mathbb{R}^3$ are expressed in $\boldsymbol{H}_s$ and point from the origin of $\boldsymbol{H}_s$ to the origins of $\boldsymbol{H}_{sa}$ and $\boldsymbol{H}_{sb}$, respectively. Concerning the orientation of the end-effectors, a quaternion representation is used (see [23]). The vector part of the unit quaternion representation from $\boldsymbol{R}_{ab} = \boldsymbol{R}_{sa}^T \boldsymbol{R}_{sb}$ shall be denoted by $\boldsymbol{\epsilon}_{ab}(\boldsymbol{R}_{ab})$.

The elastic part of the spring is implemented indirectly via a potential function $V_{ab}(\boldsymbol{H}_{sa}, \boldsymbol{H}_{sb})$ which is composed of a translational part and a rotational part in the form

$$\begin{aligned} V_{ab}(\boldsymbol{H}_{sa}, \boldsymbol{H}_{sb}) \quad = \quad & \frac{1}{2}(\boldsymbol{p}_{sa} - \boldsymbol{p}_{sb})^T \boldsymbol{K}_t (\boldsymbol{p}_{sa} - \boldsymbol{p}_{sb}) + \\ & 2\boldsymbol{\epsilon}_{ab}^T(\boldsymbol{R}_{ab}) \boldsymbol{K}_r \boldsymbol{\epsilon}_{ab}(\boldsymbol{R}_{ab}) \ . \end{aligned}$$

Herein $\boldsymbol{K}_t \in \mathbb{R}^{3 \times 3}$ and $\boldsymbol{K}_r \in \mathbb{R}^{3 \times 3}$ are the symmetric and positive definite translational and rotational stiffness matrices.

Based on the above definition of the individual potential functions, the potential for the elastic part of the compliance behavior can be summarized as

$$\begin{aligned} V(\boldsymbol{H}_{sd}, \boldsymbol{q}) \quad = \quad & V_{dr}(\boldsymbol{H}_{r,d}, \boldsymbol{H}_{sr}(\boldsymbol{q})) + V_{dl}(\boldsymbol{H}_{l,d}, \boldsymbol{H}_{sl}(\boldsymbol{q})) + \\ & V_{rl}(\boldsymbol{H}_{sr}(\boldsymbol{q}), \boldsymbol{H}_{sl}(\boldsymbol{q})) \ , \end{aligned}$$

and can subsequently be incorporated in (4) and (5). More details on spatial springs can be found, e.g., in [23].

D. First experiments

The compliance control strategy presented above was implemented using the control architecture described in Section IV and used for manipulation of large objects, see Fig. 8. As a test case the grasping and manipulating of a simple cylindrical trash bin was chosen. In the experiments the coupling spring $\mathcal{K}_c$ implemented a translational stiffness acting only in a horizontal plane. The springs $\mathcal{K}_l$ and $\mathcal{K}_r$ on the other hand implemented a translational stiffness according to the vertical movement of the arms, as well as an orientation stiffness.

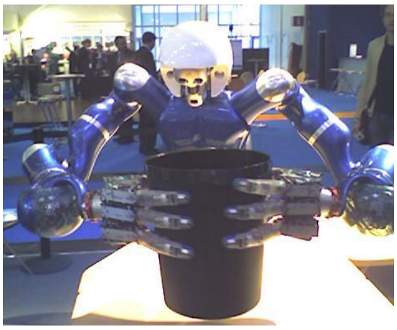

Fig. 8. Moving a trash bin by using the Cartesian compliance controller.

For the hands simple joint impedance controllers were used. That way the *grasping* was performed basically by the arms. In future experiments with Justin the impedance controllers for the arms will be augmented by force controllers for the fingers in order to keep a predefined grasping force. Also more sophisticated control strategies like an *object impedance* [24] of a combined arm-hand system will be evaluated.

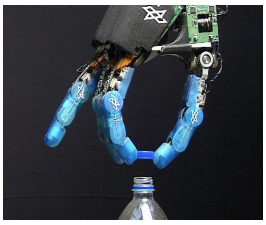

Fig. 9. The DLR four-finger hand, unscrewing the cap from a bottle.

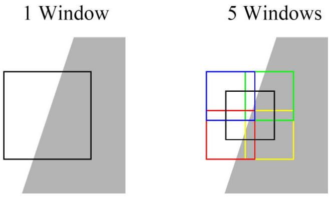

Fig. 10. Single and multiple correlation window configuration.

Finally, it should also be mentioned that in these preliminary two-armed manipulation experiments the initial position of the trash bin was considered to be known. Clearly, in a real application this knowledge about the scene must be established by some sensory information, notably by vision as described below.

VI. THE VISUAL SYSTEM

The robot system is equipped with a vision system. Scene data are acquired through a head-mounted pair of calibrated cameras. Their images are used for stereo processing and, based on the stereo data, subsequent scene analysis. The goal is to estimate a semantic world model that is accurate enough to enable fine manipulation of objects. We have previously demonstrated this type of sensory-motor capability with the Robutler system [25], [26] for a maximum of 3 DoFs (planar translation and single-axis rotation) of object configurations; see Fig. 9. Here we sketch a system that scales to the full 6 DoFs of arbitrary rigid objects.

A. *Stereo processing*

The Multiple Window, Multiple Filter (MWMF) stereo algorithm [27] is used to obtain range-data points from a scene. The MWMF algorithm has been developed to reduce the blurring of object borders, which is an inherent problem of all correlation-based stereo methods. The problem is tackled by a configuration of five partly overlapping correlation windows; see Fig. 10. The correlation score for each window is calculated by the Sum of Absolute Differences (SAD) of intensities. The correlation score for the five-window configuration is the sum of the scores for the central window and the two best outer windows. This configuration adapts well to sharp object borders and is also stable, because the small central window is always used.

Additionally, a combination of filters handle general matching errors. The method has been evaluated and recommended for real-time stereo processing [28].

Further improvements include a segment filter, which removes disparity patches below a certain size, because they usually represent matching errors. This is justified by considering that stereo correlation cannot detect objects smaller than the size of the correlation window. Thus, all small patches must be caused by errors. Finally, a segment-based interpolation method has been added [29]. It performs a bilinear interpolation of invalid disparity areas, if they are inside a segment. Otherwise, the background (i.e. the low disparity area) is propagated through the gap. This procedure ensures smooth interpolation of objects, while maintaining sharp object borders.

B. *Scene analysis*

From the stereo data, a set of objects taken from the everyday human environment, including bottles, glasses, and cups, is recognized by the vision system. The object poses are estimated to enable manipulation. Since we aim at analyzing scenes composed of an unknown number of objects, some of which may be unknown, robust estimation techniques are required.[4]

Pose estimation is based upon robust non-parametric location statistics in parameter space. To this end, a sample of pose parameters is obtained from a random data sample as follows. From each sampled data-point triple $[\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3] \in \mathbb{R}^{3\times3}$, a rotation $\hat{\boldsymbol{R}}$ and a translation $\hat{\boldsymbol{t}}$ are derived as a least-squares estimate,

$$[\hat{\boldsymbol{R}}, \hat{\boldsymbol{t}}] = \arg \min_{[\boldsymbol{R},\boldsymbol{t}]\in SE(3)} \sum_{i=1}^{3} ||\boldsymbol{R}\boldsymbol{m}_i + \boldsymbol{t} - \boldsymbol{d}_i||^2 \ , \qquad (6)$$

where $[\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3] \in \mathbb{R}^{3\times3}$ is a nearly congruent model-point triple and $||\cdot||$ denotes the Euclidean norm. An efficient solution to (6) is described in [30]. The obtained pose hypotheses $[\hat{\boldsymbol{R}}, \hat{\boldsymbol{t}}]$ are then mapped into a consistent parameter space. In consistent parameters, the invariant Haar measure of the Euclidean group $SE(3)$ is uniform, such that there is no bias of pose clustering incurred from the group topology [31].

Pose clusters are defined as the modes of the underlying parameter density. They are estimated from the parameter sample $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\} \subset \mathbb{R}^6$ through an adaptive mean-shift procedure [32]. More specifically, a sequence of pose parameters $\boldsymbol{p}^k \in \mathbb{R}^6$, $k = 1, 2, \ldots$, is obtained through iterative local averaging

$$\boldsymbol{p}^k = \frac{\sum_{i=1}^n w_i^k \boldsymbol{p}_i}{\sum_{i=1}^n w_i^k} \qquad (7)$$

with the weights

$$\begin{aligned} w_i^k &= u\big(||\boldsymbol{P}_{\text{rot}}\,(\boldsymbol{p}^{k-1} - \boldsymbol{p}_i)||/r_{\text{rot}}\big) \qquad (8) \\ &\times \ u\big(||\boldsymbol{P}_{\text{trans}}\,(\boldsymbol{p}^{k-1} - \boldsymbol{p}_i)||/r_{\text{trans}}\big) \ . \end{aligned}$$

Here $\boldsymbol{P}_{\text{rot}}$ and $\boldsymbol{P}_{\text{trans}}$ are the projectors on the rotational and translational parameter subspaces, respectively, and u is a unit

[4] Robust in the statistical sense means that each object model can be matched to a consistent part of the data while being largely unaffected by the rest of the data.

step function,

$$u(x) = \begin{cases} 1 & \text{if } x < 1, \\ 0 & \text{else.} \end{cases} \qquad (9)$$

The radii $r_{\mathrm{rot}} > 0$ and $r_{\mathrm{trans}} > 0$ of rotational and translational extension, respectively, of the averaging procedure are adapted to the local parameter density: a higher density affords smaller radii. The iteration converges to an estimate of the position of a local density maximum. By starting with $\boldsymbol{p}^0$ close to the dominant mode, as judged from a previous coarse density estimate, the sought pose estimate is thus obtained. Further modes may be explored in an analogous fashion to identify additional object instances.

VII. Summary

In this paper we presented a humanoid platform for research on two-handed manipulation. The system is based on the DLR-Lightweight-Robot-III and the DLR-Hand-II. A right-handed and a left-handed arm/hand sub-system are combined with a movable 3 DoF torso and a sensory head to form a complete upper body.

For the construction of the torso the technology of the DLR arms has been adopted. In particular, the system is equipped with *Harmonic Drive* gears and link-side joint torque sensors in the arms, the hands, and in the torso.

The complexity of the computation and communication required to control a humanoid manipulator is managed by a scalable and flexible software architecture. The identity and pose of objects to manipulate are provided by visual scene analysis base on range data.

The complete system was presented at the trade fair *Automatica* in May 2006 by using a two-arm Cartesian compliance controller for manipulation of large objects.

References

[1] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of honda humanoid robot," in *IEEE International Conference on Robotics and Automation*, 1998, pp. 1321–1326.

[2] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent ASIMO: System overview and integration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, pp. 2478–2483.

[3] T. Morita, H. Iwata, and S. Sugano, "Development of human symbiotic robot: WENDY," in *IEEE International Conference on Robotics and Automation*, 1999, pp. 3183–3188.

[4] K. Kaneko, F. Kanehiro, S. Kajita, K. Yokoyama, K. Akachi, T. Kawasaki, S. Ota, and T. Isozumi, "Design of prototype humanoid robotics platform for hrp," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, pp. 2431–2436.

[5] J.-Y. Kim, I.-W. Park, J. Lee, M.-S. Kim, B.-K. Cho, and J.-H. Oh, "System design and dynamic walking of humanoid robot KHR-2," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 1443–1448.

[6] M. A. Diftler, R. O. Ambrose, S. M. Goza, K. Tyree, and E. Huber, "Robonaut Mobile Autonomy: Initial Experiments," in *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005, pp. 1437 – 1442.

[7] R. Brooks, C. Breazeal, M. Marjanovic, B. Scassellati, and M. Williamson, "The Cog Project: Building a Humanoid Robot," in *Computation for Metaphors, Analogy and Agents*, ser. Springer Lecture Notes in Artificial Intelligence, C. Nehaniv, Ed. Springer Verlag, 1998, vol. 1562.

[8] G. Hirzinger, N. Sporer, A. Albu-Schäffer, M. Hähnle, R. Krenn, A. Pascucci, and M. Schedl, "DLR's torque-controlled light weight robot III - are we reaching the technological limits now?." in *IEEE International Conference on Robotics and Automation*, 2002, pp. 1710–1716.

[9] J. Butterfaß, M. Grebenstein, H. Liu, and G. Hirzinger, "DLR-Hand II: Next generation of a dextrous robot hand," in *IEEE International Conference on Robotics and Automation*, 2001.

[10] M. Suppa and G. Hirzinger, "A novel system approach to multisensory data acquisition," in *The 8th Conference on Intelligent Autonomous Systems IAS-8*, 2004.

[11] B. Bäuml and G. Hirzinger, "Agile Robot Development (aRD): A Pragmatic Approach to Robotic Software," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2006*, accepted for publication.

[12] The MathWorks. [Online]. Available: http://www.mathworks.com/

[13] OpalRT. [Online]. Available: http://www.opal-rt.com/

[14] QNX Software Systems. [Online]. Available: http://www.qnx.com/

[15] (2001) The Manifesto for Agile Software Development. [Online]. Available: http://agilemanifesto.org/

[16] A. Cockburn, *Agile Software Development*. Reading, MA: Addison-Wesley, 2001.

[17] A. Albu-Schäffer, Ch. Ott, and G. Hirzinger, "A unified passivity based control framework for position, torque and impedance control of flexible joint robots," in *International Symposium on Robotics Research*, 2005.

[18] M. Spong, "Modeling and control of elastic joint robots," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 291–300, 1987.

[19] Ch. Ott, A. Albu-Schäffer, A. Kugi, S. Stramigioli, and G. Hirzinger, "A passivity based cartesian impedance controller - part I: Torque feedback and gravity compensation," in *IEEE International Conference on Robotics and Automation*, 2004, pp. 2659–2665.

[20] A. Albu-Schäffer, Ch. Ott, and G. Hirzinger, "A passivity based cartesian impedance controller - part II: Full state feedback, impedance design and experiments," in *IEEE International Conference on Robotics and Automation*, 2004, pp. 2666–2672.

[21] ——, "Passivity based cartesian impedance control for flexible joint manipulators," in *IFAC Symposium on Nonlinear Control Systems*, 2004.

[22] S. Stramigioli, *Modeling and IPC Control of Interactive Mechanical Systems: A Coordinate-free Approach*, ser. Lecture Notes in Control and Information Sciences. Springer-Verlag, 2001, vol. 266.

[23] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, "Six-dof impedance control based on angle/axis representations," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 289–299, 1999.

[24] T. Wimböck, Ch. Ott, and G. Hirzinger, "Passivity-based object-level impedance control for a multifingered hand," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2006*, accepted for publication.

[25] U. Hillenbrand, Ch. Ott, B. Brunner, C. Borst, and G. Hirzinger, "Towards service robots for the human environment: the robutler," in *Mechatronics and Robotics (MECHROB)*, 2004, pp. 1497–1502.

[26] Ch. Ott, Ch. Borst, U. Hillenbrand, B. Brunner, B. Bäuml, and G. Hirzinger, "The robutler: Towards service robots for the human environment," Video, IEEE International Conference on Robotics and Automation Video Proceedings, 2005.

[27] H. Hirschmüller, P. R. Innocent, and J. Garibaldi, "Real-time correlation-based stereo vision with reduced border errors," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 229–246, 2002.

[28] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[29] H. Hirschmüller, "Stereo vision based mapping and immediate virtual walkthroughs," Ph.D. dissertation, De Montfort University, Leicester, UK, June 2003.

[30] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Am. A*, vol. 4, pp. 629–642, 1987.

[31] U. Hillenbrand, "Consistent parameter clustering: Definition and analysis," submitted to Pattern Recognition Letters.

[32] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 603–619, 2002.