

ARMAR-6: A Collaborative Humanoid Robot for Industrial Environments

Tamim Asfour, Lukas Kaul, Mirko Wächter, Simon Ottenhaus, Pascal Weiner, Samuel Rader, Raphael Grimm, You Zhou, Markus Grotz, Fabian Paus, Dmitriy Shingarey, Hans Haubert

Abstract—We present the collaborative humanoid robot ARMAR-6, which has been developed to perform a wide variety of complex maintenance tasks in industrial environments, collaborating with human workers. We present the hardware, software, and functional architecture of the robot as well as its current abilities. Those include the recognition of the need of help of a human worker, the execution of maintenance plans, compliant bimanual manipulation, vision-based grasping, fluent object handover, human activity recognition, natural dialog, navigation and more. We demonstrate the high level of technology readiness for real world applications in a complex demonstration scenario, shown more than 50 times at the CEBIT 2018 exhibition.

I. INTRODUCTION

Humanoid robotics has made progress in the last years, and several recent success stories can be reported: The achievements of the DRC humanoid robots in realistic unstructured environments, the application of Kawada’s collaborative robot NEXTAGE in manufacturing, the COMANOID humanoid robots in aircraft manufacturing, the NASA Robonaut for aerospace applications, just to mention a few.

In the SecondHands project, we aim at developing collaborative robots which are able to perform maintenance tasks in warehouse environments. Apart from executing grasping and manipulation tasks, the robot should be able to recognize that a human worker is in need of help, and provide such help in a proactive way. Based on our previous experience in building the ARMAR robots, we developed ARMAR-6 as a collaborative robot with highly integrated mechatronics and advanced sensorimotor and cognitive abilities. The design concept was driven by the application of providing assistance and support to technicians in maintenance tasks. Thus, the functional requirements were first identified based on a set of maintenance tasks, and then translated into technical requirements which were to be considered in the design process. These requirements include e. g. the maximum weight the robot has to carry, the maximum and minimum height that the robot would have to reach, the sensor systems for scene understanding and human perception, the required computational resources, etc. Key goals were to provide the robot with abilities such as grasping of maintenance tools, collaborative execution of bimanual mobile manipulation tasks, safe human robot collaboration as well as natural language interaction and dialog.

The authors are with the High Performance Humanoid Technologies Lab, Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany, asfour@kit.edu

The research leading to these results has received funding from the European Unions Horizon 2020 Research and Innovation programme under grant agreement No 643950 (SecondHands).



Fig. 1: The ARMAR-6 humanoid robot

In section II we give a brief overview over the hardware of ARMAR-6, the key components and overall specifications. section III provides an overview over the ArmarX-based software architecture underlying the robot’s capabilities. In section IV we highlight selected key abilities of ARMAR-6, and section V provides component-level evaluations as well as an account of the performance of the robot in a challenging demonstration scenario. Section VI concludes the paper with a summary and an outlook on future work.

II. THE HARDWARE OF ARMAR-6

The overarching considerations in the design of ARMAR-6 were robustness, reliability, modularity, a high level of component integration and high-level performance in providing support during maintenance tasks. The overall dimensions of ARMAR-6 are depicted in Figure 2.

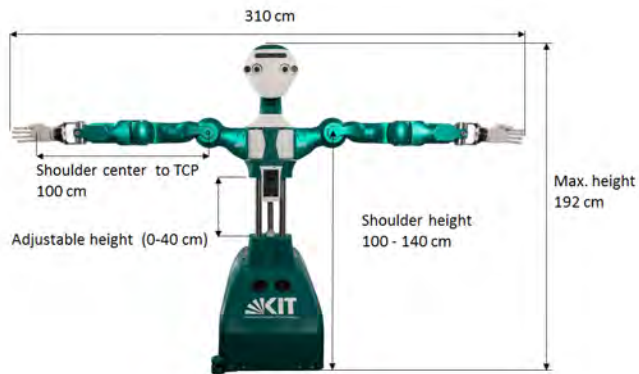


Fig. 2: Overall dimensions of ARMAR-6

A. Kinematics, actuation and sensing

ARMAR-6 features 27 actuated degrees of freedom (DOF): 8 actuated joints for the each arm, 2 motors for each underactuated hand, 2 neck joints, 1 prismatic joint that extends the torso and 4 individually driven Mecanum wheels in the mobile base. The robot further integrates a multimodal sensor setup for the measurement of torque, force, position, orientation, temperature, as well as laser scanners and stereo and depth vision systems. The following sections will detail the actuation and sensing systems contained in the different subsystems of the robot, as summarized in Table I.

1) *Dual Arm System:* Each of the two arms comprises 8 joints. The kinematic configuration of the dual arm system closely mimics the human anatomy. In particular, the arm includes an inner shoulder joint which replicates the clavicle joint of the human shoulder. 3 limitless rotating joints in the shoulder, arm and forearm significantly enlarge the bimanual workspace. Consequently, ARMAR-6 can operate equally well in its front and back[2].

The arm structure is based on an exoskeleton design approach. Highly integrated, custom designed sensor-actuator-controller (SAC) units are linked by a load bearing hollow structure that doubles as cover. Each SAC unit contains a brushless DC motor, a Harmonic Drive reduction gear, sensors for incremental position, absolute position, tempera-



Fig. 3: Large and medium sized SAC units that power the arm joints of ARMAR-6, entirely developed and assembled at KIT.

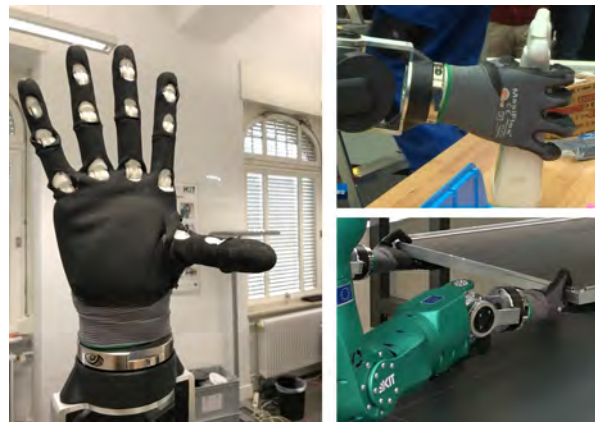


Fig. 4: The ARMAR-6's underactuated hands with protective work gloves designed for robust grasping and bimanual manipulation

ture, orientation and torque as well as all power and sensor signal conditioning electronics (see Figure 3 and [1]). To achieve a payload capacity of 10 kg we designed SAC units in three sizes with torque capacities of 176 Nm, 123 Nm and 64 Nm, respectively. All SAC units operate from the main 48 V power system and provide control interfaces and sensor readings over the EtherCAT bus.

2) *Hands:* Both arms feature anthropomorphic, underactuated five-finger hands (see Figure 4) based on the TUAT/Karlsruhe hand mechanism, which was first proposed in [3] and [4]. Each hand has 14 joints and is driven by 2 brushed DC motors (one motor for the thumb, one motor for the four fingers). Most mechanical parts of the hand are 3D-printed using a selective laser sintering out of PA2200, which displays remarkable strength and flexibility. Working gloves on both hands improve robustness and increase friction between the hands and objects. The hand integrates all drive electronics for the motors, incremental motor encoders and temperature monitoring. The included electronics expose an EtherCAT interface to directly control the hands and transmit sensor feedback through a standardized SCHUNK hand adapter, which provides well-defined mechanical and electronic interfaces. This allows to quickly exchange different end-effectors.

3) *Torso:* The torso can be extended vertically by up to 40 cm, significantly increasing the robot's workspace that reaches from the ground up to a height of 240 cm. The prismatic joint is driven by a brushless DC motor via a spindle drive and guided by twolinear guides. A draw wire sensor provides accurate absolute position feedback of the torso extension. An industrial-grade energy chain guides all cables running from the mobile base to the head and both arms.

4) *Head and neck:* The neck of ARMAR-6 has two DOF, yaw and pitch. Both are actuated by commercially available *Dynamixel Pro* actuators. The head features three different visual sensor systems: A *Roboception rc_visard 160* sensor, forming the eyes of the robot, delivers RGB stereo images, depth-maps and point clouds via Ethernet. A pair of *Point Gray Flea3* cameras forms a conventional stereo camera rig with a wide baseline of 270 mm. And a *Primesense Carmine*

TABLE I: Specifications and key hardware components of ARMAR-6

Weight				160 kg (without batteries)
Height				152-192 cm
Footprint				60 × 80 cm
Platform speed				1 m/s
Payload				10 kg per arm
Power consumption	Nominal			460 W
	Peak			1050 W
DOF	Neck	2	Dynamixel Pro	Pitch and Yaw
	Arms	2×8	Custom SAC-units [1] based on Robodrive brushless DC motors, Harmonic Drive reduction gears and ELMO motor controllers	4 per shoulder, 2 per elbow, 2 per wrist
	Hands	2×14	Maxon Brushed DC motors	Underactuated five-finger hands with one motor for 12 finger joints per hand and one motor for 2 thumb joints per hand
	Torso	1	Dunker Brushless DC motor with spindle drive and brake and ELMO motor controller	Linear actuator with 40 cm vertical travel
	Platform	4	Donkeymotion drive system with brushless DC motors and brake and ELMO motor controller	Omni-wheels for holonomic motion
Sensors	Head	1	Roboception rc_visard 160 stereo sensor	3D vision
		1	Primesense Carmine 1.09	RGB-D
	SAC-Units	2	Point Grey Flea3 USB-3 cameras	Stereo vision
		1	Renishaw AksIM MBA7/8	Absolute position
		1	AMS5306	Incremental position
		1	Texas Instruments ADS1220 ADC	Torque (strain gauge full bridge)
		1	Bosch BNO055 IMU	Acceleration and rotational rate
		5	Various temperature sensors	Temperature at different locations within the SAC units
	Torso	1	Waycon Draw Wire Sensor SX50	Absolute vertical position
	Platform	2	Hokuyo UST-10LX 2D lidars	Laser range finders for orientation and safety
Power Supply	Internal		NiMH battery with 38 Ah @ 48 V (1824 Wh)	Power-Autonomy for up to 4 h
	External		48 V power supply	Tethered power for development and testing
Computers	Real-time	1	Mini-ITX with Intel Core-i7 and 8GB RAM	Running Ubuntu with RT-PREEMPT patch, ArmarX and the open source SOEM EtherCAT master implementation
	Vision	1	Mini-ITX with Intel Core-i7 and 8GB RAM	Additionally equipped with a GeForce GTX-1080 GPU; running standard Ubuntu and ArmarX
	Speech	1	Mini-ITX with Intel Core-i7 and 8GB RAM	Additionally equipped an external Roland USB sound card for wireless microphone input and sound output; running standard Ubuntu and ArmarX
	Planning	1	Mini-ITX with Intel Core-i7 and 8GB RAM	Running standard Ubuntu and ArmarX
Communications	Internal		EtherCAT Bus	Real-time automation bus connecting all actuators (except for the neck DOFs) to the realtime computer; 1 kHz bus update rate
	External		GigaBit Ethernet	Either via LAN, 2.4 GHz or 5 GHz WLAN via the internal router and switch
Robot Operating System			ArmarX	Middleware, robot development environment
User Interface			ArmarX	Comprehensive high- and low-level APIs, GUI, natural language

1.09 RGB-D sensor outputs monocular color images and dense depth information.

5) *Mobile base*: The mobile base of the robot contains the battery packs, the power management system, the computers and GPU, the sound system, the prismatic joint, the remote e-stop receiver and the network peripherals along with their air cooling system. All components are enclosed in an easy-to-remove fiberglass cover. It also includes the robot's drive system consisting of four Mecanum wheels for unconstrained holonomic movement. The brushless DC motors, wheels, brakes and encoders are integrated in two commercial axes produced by *DonkeyMotion*. As in all brushless power sys-

tems of ARMAR-6, we use motor controllers from *Elmo Motion Control* to precisely command motor currents or velocities over the EtherCAT bus. Two *Hokuyo UTX-10* laser range finders in two diagonal corners of the platform provide planar 360° distance information for map-based navigation, safety and obstacle avoidance. A 1824 Wh battery pack allows for up to 4 h of power-autonomous operation.

B. Computing and communication

ARMAR-6 contains four identical computers in its mobile base. Each consists of a Mini-ITX motherboard with an Intel Core-i7 CPU, 32 GB of RAM, 500 GB of SSD storage and a dual GigaBit Ethernet link. We opted for the standard

Mini-ITX form factor to simplify maintenance and future upgrades. Each computer together with all its peripherals is mounted on a custom frame that slides in the back of the robot and guarantees quick access. One PC is dedicated to real-time control and runs ArmarX-RT based on the *Simple Open-Source EtherCAT Master (SOEM)* implementation on a Ubuntu Linux with RT-PREEMPT real-time patch. The second computer is dedicated to all computer vision applications and is therefore equipped with a GeForce GTX 1080 GPU. The remaining two computers are used for motion planning and natural language processing. All computers run the ArmarX software framework. A gigabit Ethernet switch connects all computers to the robot network in which a router provides access via WLAN and LAN for external communication and allows for seamless switching between wired and wireless operation. A custom-built EtherCAT hub connects the buses of the arms, the torso and the mobile base with the EtherCAT real-time computer.

III. THE SOFTWARE OF ARMAR-6

On the software side, ARMAR-6 is powered by the robot development environment ArmarX¹ [5], which offers the middleware tools and capabilities needed to develop new functionalities as well as a full functional three-level robot architecture from control up to symbolic reasoning. All the abilities of ARMAR-6 described in section IV and partially shown in the attached video are realized using ArmarX.

A. Functional architecture

The functional architecture of ARMAR-6 is a three-level architecture which links sensorimotor and symbolic representations through a memory layer and allows the realization of complex manipulation, collaboration and interaction tasks as well as natural language understanding and high-level reasoning (see Figure 5).

The *low-level* consists of the Hardware Abstraction Layer (HAL) and the real-time control framework ArmarX-RT (see subsection III-B). On this level, the sensors and actuators of ARMAR-6, i.e. the joints, the mobile platform, the hands and the visual sensors, are abstracted into hardware independent virtual devices to allow using the same controllers in simulation and on the real robot. These sensors and actors are used by the real-time single and multi-joint controllers, which in turn are parametrized and triggered by the statechart from the highest abstraction level.

The *middle level* consists of MemoryX and the perception modules. On this level, the memory system MemoryX and perception modules provide symbolic and subsymbolic information for the highest level such as grasp information for objects, 6D object poses of object instances, position of the robot in the environment, human posture, etc. A visualization of the working memory during the execution of a hand over action is shown in Figure 8.

The *highest level* consists of symbolic processing and understanding such as natural language understanding, symbolic planning, scene understanding and skill coordination

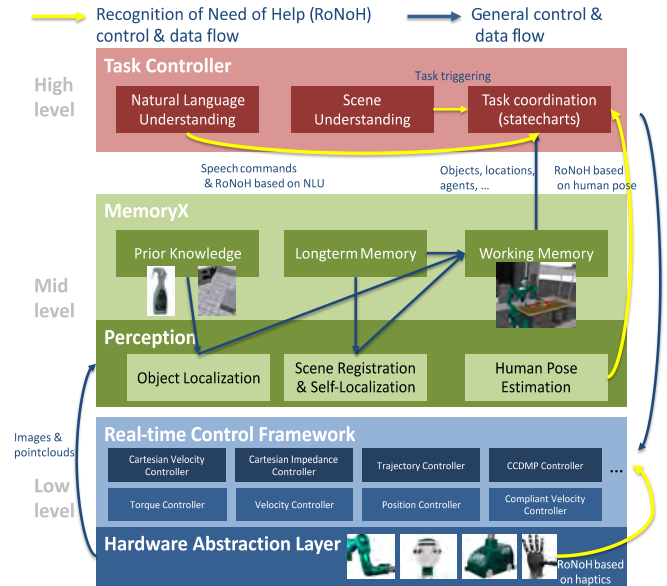


Fig. 5: Functional Architecture of ARMAR-6

via statecharts. On this level, the symbolic reasoning, scene understanding and natural language understanding (NLU) as well as task (i.e. symbolic planning) and skill coordination using ArmarX statecharts is located. The NLU provides new tasks for the task coordination directly from speech commands or inferred from utterances. The scene understanding triggers tasks based on the visual perception, which includes inferring when help is needed. The skill coordination is modeled for each task by statecharts ranging from simple skills such as moving single joints to complex tasks such as object handover or removing and carrying the guard of a conveyor belt. These statecharts retrieve the needed symbolic and subsymbolic information from the memory framework and parametrize and control the real-time controllers.

B. Real-time robot control framework ArmarX-RT

In order to be able to adequately control the high-performance actuators of ARMAR-6, we developed a robot-independent, real-time capable control framework *ArmarX-RT*. The developed concepts allow for combining and switching between different control approaches with varying cycle times and real-time requirements while maintaining full compatibility to our existing robot-framework ArmarX.

We abstract the robot in a set of control and sensor devices. These control devices represent actuators or the power management system, receive set-points from higher-level components, convert them into the hardware specific format and send them to the real hardware. Sensor devices receive real data (e.g. motor ticks) or virtual data (e.g. control frequency) and make it available to higher level components.

In order to develop flexible and reusable control mechanisms, the framework offers a two-level hierarchy for controllers:

¹<https://armarx.humanoids.kit.edu>

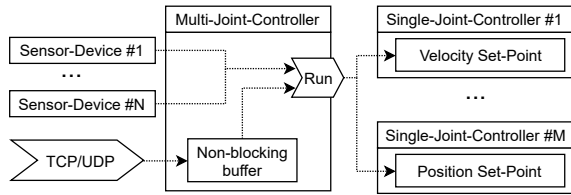


Fig. 6: Structure of a Multi-Joint-Controller. It receives input from sensor devices or remote function calls and calculates set-points for one or multiple control devices

- Hardware-specific *Joint-Controllers* consider low level control tasks such as encoding radian positions into encoder ticks or calculate motor current based on a torque target.
- Hardware-independent *Multi-Joint-Controllers* for mid- or high-level tasks such as joint trajectory execution or Cartesian velocity control (see Figure 6).

Using these mechanisms we are able to offer the same software interfaces on the real robot and in simulation which enables faster development since *Multi-Joint-Controllers* run in simulation and on the real robot. As shown in Figure 7, the control framework is divided in a real-time capable part (lower part) and a non real-time part (upper part). The non real-time part provides communication interfaces to non real-time software-modules and provides the possibility to use existing components, including the platform control, motion planning, our memory architecture and visualization of the internal state. It also handles all management functionalities such as controller creation and destruction. The real-time capable control thread handles bus communication, controller execution as well as the non-blocking interfacing with higher-level components via the TCP or UDP based communication protocols using ArmarX/Ice [6].

a) *EtherCAT Master*: ARMAR-6 uses EtherCAT as the main communication bus-technology. Thus, the sensor-actuator-controller units, the torso joint, the mobile base, the 6D-force-torque sensor as well as the hands of ARMAR-6

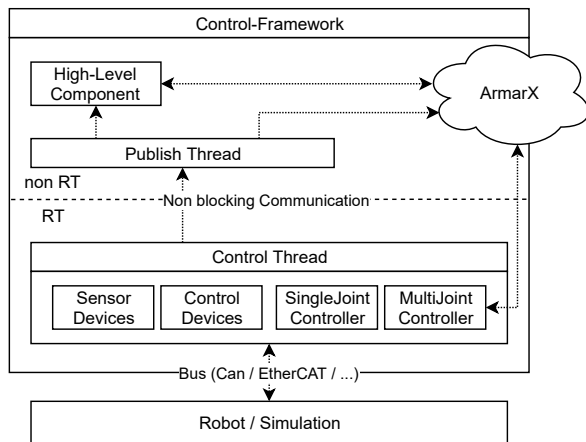


Fig. 7: Structure of the control framework split into a real-time section for controller execution and non real-time section for high-level tasks

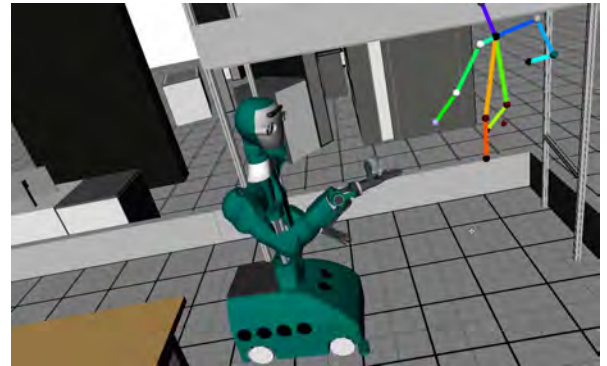


Fig. 8: 3D visualization of the working memory of ARMAR-6 during a handover action

are connected as EtherCAT slaves to the main control PC. The EtherCAT master implementation reads sensor devices and controls the actor devices. This EtherCAT master represents the *Hardware Abstraction Layer (HAL)* and utilizes the generic control-framework by abstracting all hardware devices to generic virtual devices. To increase reliability and achieve a communication frequency of 1 kHz between the EtherCAT slaves and the master implementation, the RT-PREEMPT extension of Linux is used.

With this extension the scheduling of threads is changed to reduce the blocking of high priority threads by the kernel and thus achieve more consistent cycle times.

b) *Control Modes of ARMAR-6*: Basic control modes of ARMAR-6 have been developed and integrated into the control framework. The available control modes for each arm joint of ARMAR-6 are position, velocity, torque and gravity-compensated torque control. Additionally, among others, the following robot-independent multi-joint controllers have been developed:

- *Trajectory controller* to execute pre-calculated multi-joint trajectories resulting e.g. from motion planning
- *Cartesian velocity controller* to execute given velocities of the end-effector in Cartesian space
- *Cartesian velocity controller for the mobile platform* which translate given Cartesian velocity targets into wheel velocities of the holonomic mobile platform
- *Task-space impedance controller* using the torque-sensor of each arm joint and allowing compliant task space motions

IV. ABILITIES

ARMAR-6 has a wide range of abilities, ranging from kinesthetic teaching and force-based interaction to recognizing when a human needs help. It is able to interact safely with the environment and humans in everyday scenarios. This is facilitated both by high-level reasoning and scene understanding as well as by the available low level controller architecture. In this section, we describe key abilities of ARMAR-6 that arise from its unique hardware/software architecture.

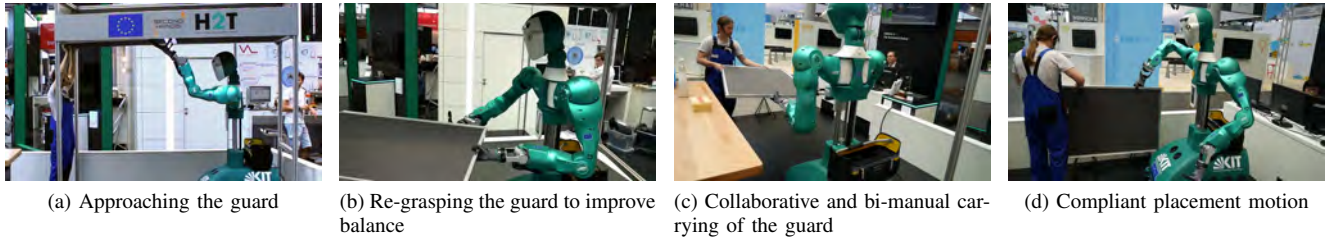


Fig. 9: The technician and ARMAR-6 remove the guard in a collaborative bimanual mobile manipulation task

A. Task space impedance control

Relying on the low level torque controllers, ARMAR-6 offers task space impedance control, which can be applied for several interactive tasks. The end-effector can be controlled via a virtual target pose in Cartesian space, while the whole arm remains compliant. This can be used for detecting contacts with the environment while respecting a given maximum interaction force. Additionally, the controller can be used to grasp and carry heavy objects while being able to react to additional perturbing forces. One example is the ability of ARMAR-6 to help a human co-worker to carry a heavy object. In this task the robot supports most of the object's weight, while following the desired path of the human. Physical interaction forces conducted through the object guide the robot to the desired target. The attached video shows examples, including the collaborative carrying of a large object.

B. Grasping of known and unknown objects

ARMAR-6 has the ability to grasp known, familiar and unknown objects. For known objects a multi-stage grasping pipeline is used to find the most suitable grasp for the task from a set of known grasps. Thereafter the optimal robot placement is determined by internal simulation. This includes planning of time-optimal collision free trajectories for fast grasping, as well as a natural approach direction and hand orientation. For familiar objects the object is first classified to match a set of reference classes. Then a feasible grasp is transferred based on the estimated object model.

In the case of unknown objects ARMAR-6 follows a reactive grasping scheme. Several grasp hypotheses are generated by fusing the available visual and tactile sensory information. Then the best grasp is chosen according to learned metrics. During grasp execution all available sensors in the hand and the wrist are used to adjust the grasp pose if necessary. An example of such a reactive grasp is shown in the attached video.

C. Physical human-robot interaction

Thanks to the torque-sensors in each arm joint and an accurate model of the dynamic properties, ARMAR-6 is able to detect external disturbances on the arms. This is used for active compliance and safe physical human-robot interaction. New skills can be taught by kinesthetic teaching or the robot can be guided while carrying a large object collaboratively (see Figure 10 and the attached video).

D. Bimanual mobile manipulation

With its two arms and its mobile platform ARMAR-6 is able to perform complex bimanual mobile manipulation tasks. ARMAR-6 uses learned movement primitives realized by two methods: Coordinate Change Dynamic Movement Primitive (CC-DMP) [7], which coordinates both arms by a leader-follower structure, and force-based admittance control, which emulates a coupling force between both arms by setting a virtual target pose for impedance control. Torque-based task execution enables compliant, collaborative manipulation of large objects (as shown in the top row of Figure 10).

Force-based bimanual control can be used for compliant collaborative manipulation, exemplified by the guard-support task of the demonstration scenario described in subsection V-B and shown in Figure 9 and in the attached video: First, a force based contact detector is used during the guard approaching motion to detect whether the robot's hands are in contact with the protective guard. Once the robot touches the guard, a dialog is triggered to inform the human collaborator that the robot safely supports the guard. During lowering the guard, both hands are controlled in a leader-follower framework (CC-DMP) to generate a synchronized motion. After lowering the guard, task space impedance control allows the human to pull or push on the robot's hands to guide it in the collaborative guard carrying. In this phase, the collaborator can decide where the guard should be transported. After reaching the desired spot for the guard placement, the robot utilizes the method developed in [8] to place the guard with detection of the user's intention.

E. Activity recognition and active hand over

ARMAR-6 is capable to estimate the pose of a human in front of the robot by only using RGB-D images, which is prerequisite for physical human-robot interaction skills that are able to adapt to the human. The computationally expensive human pose estimation algorithm is run on board on the GPU of the Vision PC and allows stable pose estimation at ~ 15 Hz. Based on the human pose estimation ARMAR-6 is capable of detecting activities the human is performing, or handing over tools directly into the hand of the human [9]. The working memory state during a hand over action, including the estimated human pose, is shown in Figure 8.



Fig. 10: During the CEBIT demonstrations, ARMAR-6 recognized the need of help as the technician carries out a maintenance task. ARMAR-6 and the technician lower and transport a guard cooperatively (top left) and place it at a place chosen by the human (top right). While the technician inspects the conveyor belt, ARMAR-6 brings necessary cleaning equipment from the workbench (bottom left). The handover based on human pose tracking allows the technician to comfortably receive the equipment (bottom right).

F. Recognition of the need of help

One of the key research objectives in the SecondHands project is to develop a robot that can recognize that its human co-worker needs help, and then pro-actively offer it. Its numerous sensor systems enable ARMAR-6 to recognize the need of help via different sensor modalities:

Using *vision-based* pose tracking and activity recognition, ARMAR-6 can infer from prior knowledge whether help is needed during a particular action.

The human can also explicitly request for help via the robot's *natural language* interface for speech and dialog, or indicate to the robot that he or she is about to start an activity that might require a helping pair of hands (e. g. carrying out a maintenance task).

The third way the robot can recognize that its help is needed is via *force and torque sensing*. An example for this is a collaborative carrying task: If the human changes the position of its hands on the carried object the robot can sense the change in the force distribution among its hands and position them in such a way that offers the best possible support for the human.

V. EXPERIMENTS AND EARLY EVALUATION

We conducted experiments for performance evaluation of component and system level. In particular, we demonstrate the ability of the robot as a complete system to robustly perform collaborative tasks with humans in real world scenarios.

A. Component-level

The accuracy of position repeatability of the arms was measured experimentally using a marker-based motion-capture system with sub-millimeter accuracy, and amounts

to 0.758 mm. Experiments confirm that our design target regarding the payload of the arm of 10 kg was reached. The holding force of the hands also amounts to 10 kg for a hook grasp. The sensitivity of the joint torque sensors in the arm joints is as low as 0.04 Nm (see also [1]).

ARMAR-6 operates from 48 V DC power, either provided by an external power supply or by the internal 1824 Wh battery. Based on experimentally measured data, the robot consumes 9.5 A in nominal operation mode (e. g. bimanual mobile manipulation), resulting in approximately 4 h of power-autonomous operation.

B. System-level

We developed an integrated demonstration to test and showcase the robot's ability to recognize the need of help in a challenging real-world setting via the modalities speech, vision and haptics/force, and its capabilities in terms of force-based bimanual manipulation, vision-based grasping, fluent object handover, human activity recognition, natural language based human-robot dialog, navigation and more. The demonstration scenario mimics a typical maintenance task at a highly automated customer fulfillment center (CFC) at Ocado Technology, normally carried out by two technicians: As an overhead conveyor system indicates malfunctioning, the technician starts to investigate the problem while the robot observes the process. The technician concludes that the protective guard needs to be removed and starts unscrewing it. ARMAR-6 visually recognizes that it can help and supports the guard from underneath. The technician and the robot then jointly lower the guard and the technician guides the robot to the placing location. Further inspecting the conveyor system, the technician concludes that the drive



Fig. 11: ARMAR-6 assisting a technician in a mock-up maintenance task during one of many demonstrations held at the CEBIT'18 expo

system needs to be cleaned and uses a ladder to access the drive system. The robot infers that the cleaning agent is needed, fetches the spray bottle from the workbench and hands it over to the technician on the ladder. Once the technician is done cleaning it hands the spray bottle back to ARMAR-6 that places it back on the workbench.

This demonstration was repeatedly shown at the CEBIT 2018 expo in Hannover several dozens of times over the course of five days, also showcasing the reliability of the robot. A one-shot video of the demonstration is available on the H²T YouTube channel². In this scenario, ARMAR-6 recognizes the need of help via different sensor modalities:

1) *Recognition of the need of help based on visual information:* Given a description of the maintenance task with needed objects and actions, an interpretation of the current scene including objects and human activities, the robot is able to infer that it can provide help to the technician in the guard removal task. The robot is able to detect that 1) there is only one person unscrewing the guard and provides help by holding the guard, lowering it together with the human and 2) the human on the ladder is reaching out the arm to receive an object (the spray bottle) or to handover this object back to the robot.

2) *Recognition of the need of help based on haptics and force:* Force interaction takes place in collaborative human-robot bimanual manipulation tasks when jointly carrying the guard. Based on irregular force patterns and/or sudden changes of these pattern during the task execution, the robot is able to infer that the collaboration partner is struggling, and that regrasping may provide help as it will reduce the load on the human.

3) *Recognition of the need of help based on speech and dialogue:* Natural language commands are used to explicitly ask for help ("bring me the spray bottle") or the need of help is inferred from implicit human commands (Human: "We have to clean the drive train"; Robot: "I will bring you the spray bottle").

²<https://www.youtube.com/watch?v=6AFkrGkKI7g>

VI. CONCLUSION AND FUTURE WORK

We introduced ARMAR-6, the next generation of the ARMAR humanoid robot family with a high level of technology readiness for real world applications. The robot is being developed within the European project SecondHands to demonstrate the use of collaborative humanoid robots in maintenance and repair tasks in warehouse environments. Apart from its physical and sensorimotor performance, the high level of software and hardware integration, the robot is equipped with early cognitive abilities regarding the recognition of the need of help and providing such help in a proactive way. In our future work, we will extend and improve the different functional components towards recognition of the need of help and providing help in a wider range of maintenance and repair tasks We will investigate how such cognitive abilities can continuously be improved based on learning from experience. An important question which we will address in the future is the quantitative evaluation of the robot on system level.

ACKNOWLEDGMENT

The authors would like to acknowledge the help and support of all other members and students of the H²T lab as well as all members of the SecondHands project team, especially Aude Billard and Mahdi Khoramshahi from EPFL, Stefan Constantin and Alex Waibel from KIT, Fiora Pirri from Sapienza University of Rome, Lourdes Agapito from University College London, and Graham Deacon from Ocado Technology. Special thanks to Michael Suppa of Roboception for providing one of their earliest sensors to ARMAR-6.

REFERENCES

- [1] S. Rader, L. Kaul, P. Weiner, and T. Asfour, "Highly Integrated Sensor-Actuator-Controller Units for Modular Robot Design," in *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Munich, Germany, 2017, pp. 1160–1166.
- [2] S. Rader, L. Kaul, H. Fischbach, N. Vahrenkamp, and T. Asfour, "Design of a High-Performance Humanoid Dual Arm System with Inner Shoulder Joints," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, Cancn, Mexico, 2016, pp. 523–529.
- [3] N. Fukaya, T. Asfour, R. Dillmann, and S. Toyama, "Development of a five-finger dexterous hand without feedback control: The tuat/karlsruhe humanoid hand," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 2013, pp. 4533–4540.
- [4] N. Fukaya, S. Toyama, T. Asfour, and R. Dillmann, "Design of the TUAT/Karlsruhe Humanoid Hand," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Takamatsu, Japan, October 30 - November 5 2000, pp. 1754–1759.
- [5] N. Vahrenkamp, M. Wächter, M. Kröhnert, K. Welke, and T. Asfour, "The Robot Software Framework ArmarX," *Information Technology*, vol. 57, no. 2, pp. 99–111, 2015.
- [6] M. Henning, "A new approach to object-oriented middleware," *Internet Computing, IEEE*, vol. 8, no. 1, pp. 66–75, 2004.
- [7] Y. Zhou, M. Do, and T. Asfour, "Coordinate change dynamic movement primitives - a leader-follower approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea, October 2016, p. 54815488.
- [8] M. Khoramshahi and A. Billard, "A dynamical system approach to task-adaptation in physical human-robot interaction," *Autonomous Robots (under review)*, 2018.
- [9] J. R. Medina, F. Duvallet, M. Karnam, and A. Billard, "A human-inspired controller for fluid human-robot handovers," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 324–331.