

10th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '16

Coupling of a redundant manipulator with a virtual reality environment to enhance human-robot cooperation

Luigi Gammieri^a, Marco Schumann^b, Luigi Pelliccia^b, Giuseppe Di Gironimo^a, Philipp Klimant^{b,*}^a*Department of Industrial Engineering, University of Naples Federico II, Via Claudio, 21, 80125 Naples, Italy*^b*Institute for Machine Tools and Production Processes, Technische Universität Chemnitz, Reichenhainer Straße 70, 09126 Chemnitz, Germany** Corresponding author. Tel.: +49 371 531 36911; fax: +49 371 531 836911. E-mail address: philipp.klimant@mb.tu-chemnitz.de

Abstract

The current trend in manufacturing is to obtain a flexible work cell in which human and robot can safely interact and collaborate. Virtual Reality (VR) represents an effective tool capable of simulating such complex systems with a high level of immersion. In order to take advantage of VR technologies to study Human-Robot Cooperation (HRC), a digital model of a redundant manipulator (KUKA LBR iiwa) has been developed starting with kinematic modeling and then coupled with the real robot. This approach allows simulating HRC in several scenarios, to reproduce the safe behavior on the real robot, as well as to train operators.

© 2017 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the scientific committee of the 10th CIRP Conference on Intelligent Computation in Manufacturing Engineering

Keywords: Virtual reality; Simulation; Modelling; Human-robot collaboration

1. Introduction

Nowadays, flexibility in manufacturing has gained particular importance in order to face market demand fluctuations and to allow fast modifications in product characteristics. Human and robot cooperation in production lines seems suitable for allowing fast changes of production activity. Moreover, Human-Robot Cooperation (HRC) in the industrial sector allows to reduce physical efforts and to preserve the operator's health during the execution of heavy tasks. Furthermore, the arising fourth industrial revolution, Industry 4.0, with the introduction of Cyber-Physical Systems (CPS), offers new solutions to the problem of increasing flexibility in manufacturing processes. In this perspective, Virtual Reality (VR) and Augmented Reality (AR) can be used as extended interfaces between human operator and robots inside CPS [1]. Virtual Manufacturing refers to the use of VR technology in order to optimize both products and production processes [2]. VR and AR can represent strategic important tools also for the study of HRC. E.g. the coupling

of a 7-axis KUKA LBR 7 R800 with a digital reconstruction of a work cell in VR is discussed in [3].

Otherwise, in recent years, a great deal of research has been conducted in order to achieve a safe and dependable human-robot interaction in worker accessible environments [4]. To fulfill these requirements, the introduction of a different kind of industrial robot became necessary. Lightweight robots, designed with low inertia and equipped with more sensors (e.g. torque sensors), are capable of sharing a working environment with humans reducing the risk of severe injuries [5].

The aim of the conducted research activity has been the development of a VR-based simulation environment in which HRC can be effectively simulated. Once the safe behavior related to a specific task has been determined in VR, it will be replicated on the real robot. In such a way, it is possible to take advantages of VR technology to study HRC safely. In such an environment, the motion of the end effector of the virtual robot is controlled by means of an input device (e.g. space mouse). In order to allow real-time motion control of the robot in VR, forward kinematics (FK) and inverse

kinematics (IK) have been implemented. Thus, the connection between the real robot controller and the virtual scene allows real-time motion generation and the simulation of a human-robot workplace.

Simulation of the robotic system in VR is also suitable for fast and reliable testing of new products and production scenarios as well as for the training of operators thanks to its high level of immersion and interactivity [6].

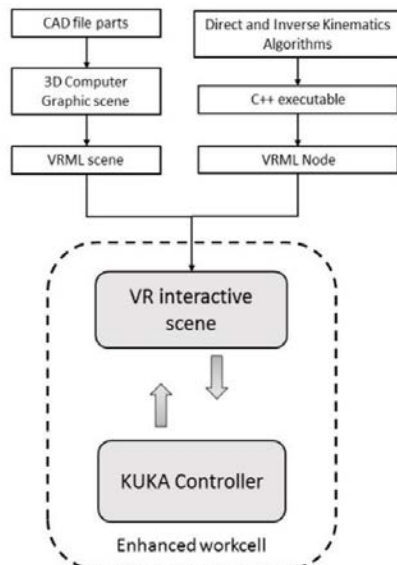


Fig. 1. Overview of the activity

2. State of the art

In recent years, HRC has gained an increasing attention from the research community. Safety and dependability are primary concerns whenever humans and robots share the same environment [7]. The basic paradigm used in order to guarantee humans' safety was the segregation between robots and humans, i.e. exclusion of humans from the workspace of the robot. The existing robotics safety regulations and standards [8],[9] are also based on such a philosophy. The high limitations imposed to the speed of the end effector of the robot are an example for such regulations. However, this approach fails when humans and robots have to share the same environment to fulfill a certain task.

Thus, a great effort has lately been devoted to the development of robots capable to achieve safe and dependable human-robot cooperation. Robots specifically designed to share human's space equipped with specific sensors, e.g. joint torque sensors, allowed to introduce the collision detection and fast reaction at control level. In [10] an end-user approach for collision detection and reaction is discussed, with the possibility to distinguish between accidental collisions and intended human-robot contacts for an industrial manipulator.

During the design of robots required to share humans' environment, it has to be taken into account that the environment is partially unknown. For this reason, such robots cannot simply move along predefined trajectories, but

must react to unexpected contact with the environment. In order to incorporate reactions to external forces, force control and impedance control have been widely used [11]. Together with the control aspects, mechanical design plays a fundamental role in safety. Specifically, lightweight design of the robots is an important requirement. Nowadays, VR technology is capable of effectively supporting researchers in studying complex systems. Thus, it may represent also a proper tool to study robotics and HRC. In [12] VR has been used to evaluate the performance of different manipulators mounted on a wheelchair. In [6] VR training system is shown that conceived in terms of serious game, developed with Unity, aimed at real-time simulation of the cooperation between industrial robotic manipulators and humans, in reference to simple manufacturing tasks. A Head-Mounted Display (HMD) provides immersion into the virtual scene while a Kinect sensor for tracking the operator skeletal also allows interaction. In [13] the inverse kinematics modeling of 7-axis manipulator based on Virtual Reality Modelling Language (VRML) is addressed. [14] describes the simulation of flexible automated manufacturing cells in VR using VRML. VR technologies are also suitable for programming of industrial robotic work cells. In [15] a procedure to program the KUKA KR16 in VR is presented. [16] addresses the development of a MATLAB toolbox aimed at controlling a 6-DOF robot by means of connection of a remote computer to the robot controller. The toolbox implements several functions, e.g. Forward and Inverse Kinematics, 3D animations with skeletal reconstruction of the robot. Such a toolbox allows also the connection with a haptic feedback device, in order to provide position-based force feedback. A survey on AR application in robotic work cell is provided in [17].

3. General approach

The main idea of the presented approach is to create a bidirectional coupling between VR and a real robot. This enables us to develop a platform for testing purposes for investigating HRC. Such a platform can be used to test any desired scenario by creating a virtual model of it for the VR environment and combining it with the real robot. The bidirectional communication enables us to use the real robot as input device for the VR scene. Likewise, it is possible to control the robot with (VR) input devices, e.g. a space mouse. Starting from the given CAD model of each component, a VR scene of the whole workspace must be created. Particularly, a consistent hierarchical 3D model of the robot must be developed. Later, the possibility to interact with the virtual environment by means of an input device (e.g. space mouse) must be provided. Furthermore, in order to achieve full interactivity with the virtual scene, it is necessary to provide the possibility of a real-time kinematic control of the virtual robot. To achieve this goal, FK and IK models must be implemented. Once the kinematic algorithms have been tested, it is necessary to implement such a model in VR for the real-time motion control. Finally, the robot controller must be coupled with the VR scene. The coupling interface of the robot can provide different data as output, e.g. measured joint

positions, measured joint torques, commanded joint positions, etc. For instance, the position and the orientation of the end effector (Tool Center Point, TCP) of the robot can be used as input for the IK algorithm in the VR scene. All steps together result in the proposed platform for the VR-robot coupling.

4. Implementation details

4.1. Test Setup

The experimental setup is composed of the following main components:

- KUKA LBR iiwa 7 R800 robot
- gripper station
- work desk
- projection wall with 2 beamers

where the gripper station is able to provide up to eight different kind of grippers (pneumatic and electric). The particular experimental set-up provides also an automated robot gripper changing system. A VR model of the setup was developed. Experiments were conducted at the Virtual Reality Center Production Engineering (VRCP) Lab of the Technische Universität Chemnitz.

4.2. Kinematic Modelling

This section addresses the FK and IK modelling of the 7-axis redundant manipulator KUKA LBR iiwa 7 R800. Kinematic redundancy allows the manipulator to achieve different postures in reference to a specified pose of the end-effector. Thus, it is possible to take advantage of the kinematic redundancy of the manipulator taking into account specific constraints while the end effector moves along a specified path (e.g. dexterity, distance from the joint limits as well as distance from obstacles in the workspace). Forward kinematic has been modeled through Denavit-Hartenberg (DH) convention and is expressed by means of a 4×4 homogeneous transformation matrix. The pose of the end effector can be easily determined as a function of the joint variables \mathbf{q}_i . Tab. 1 shows the specific DH values for the KUKA LBR 7 R800.

Table 1. Denavit-Hartenberg parameters.

Link	a_i (mm)	α_i (rad)	d_i (mm)	θ_i (rad)
1	0	$-\pi/2$	340	q_1
2	0	$+\pi/2$	0	q_2
3	0	$+\pi/2$	400	q_3
4	0	$-\pi/2$	0	q_4
5	0	$-\pi/2$	400	q_5
6	0	$+\pi/2$	0	q_6
7	0	0	126	q_7

Inverse kinematics has been implemented using a pseudo-inverse of Jacobian matrix starting from the differential equation:

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (4.1)$$

The equation (4.1) maps the joint space velocities $\dot{\mathbf{q}}$ onto the task space velocities $\dot{\mathbf{x}}$. The matrix $\mathbf{J}(\mathbf{q})$ is the (6×7) Jacobian matrix. The equation (4.1) can be inverted using the pseudo-inverse of the Jacobian matrix:

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger(\mathbf{q})\dot{\mathbf{x}} \quad (4.2)$$

where $\mathbf{J}^\dagger = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T)^{-1}$ is a (7×6) matrix obtained minimizing the norm of joint velocities $\dot{\mathbf{q}}$ [18]. It is possible to use several algorithms in order to determine the joint variables in reference to a planned trajectory of the end effector starting from the equation (4.1). Once the joints velocities have been computed from (4.2), the joint variables \mathbf{q}_i can be determined by means of numerical integration (Open-Loop Inverse Kinematics) as shown in Figure 2. However, this approach causes a drift of the error which can be avoided using a Closed-Loop Inverse Kinematics (CLIK) algorithm, which is capable of guaranteeing convergence to zero of the error [18]. In the proposed approach, both algorithms have been implemented and tested in MATLAB/Simulink. For our purposes, it was possible to adopt the Open Loop algorithm.

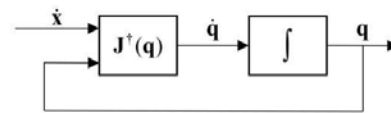


Fig. 2: Open-Loop Inverse Kinematic algorithm

4.3. VR Environment

The suggested approach needs the integration of several systems into the VR environment. Therefore, a flexible software system with possibilities for extension is needed. The VR framework Instant Player by Fraunhofer IGD [19] fulfills the needed requirements. Hence, the realized demonstrator application was implemented for this software system. Instant Player is capable to load the X3D format (successor of VRML [20]). This is an open standard format maintained by the W3C for describing 3D worlds. X3D offers several possibilities to influence the scene at runtime. One is scripting with ECMAScript. This allows changing node attributes in an easy manner. A second possibility is to use the External Authoring Interface (EAI). This is a network interface enabling external .NET and Java programs to influence the scene. Scripting takes place within the X3D software marked by a low latency but minor range of function. In contrast, EAI offers the full functionality of a programming language but lacks of a higher latency caused by the network interface.

Both interfaces are part of the X3D standard. Instant Player offers another (but proprietary) interface: the IO nodes (input output nodes). IO nodes behave like a new node type with defined input and output ports and can be used within the X3D routing mechanism. The programming takes place using C++. The result is an executable module which can be loaded and used by the Instant Player. Main purpose for this interface is to integrate new devices into the framework.

4.4. Coupling of VR with the Simulation Model

The algorithms for the inverse kinematic (see section 4.2) must be integrated into the VR application to calculate the current joint angles in real-time. The described IO node interface of the Instant Player is suitable for this case. Now, several approaches are possible to realize the mathematical model in VR. One solution is direct programming of the algorithm. This works, but is time-consuming and poor maintainable. Furthermore, mathematic models are often created with simulation software (e. g. MATLAB Simulink, Simulation X). Therefore, an automated way is needed to use the desired model within the VR environment. When using MATLAB/Simulink, the Simulink Coder (previously Real-time Workshop) can be used to achieve this.

This kind of approach is notably convenient considering the ease of manipulation of matrices and vectors in Simulink. In addition there is no need of further implementation in a new programming environment after testing.

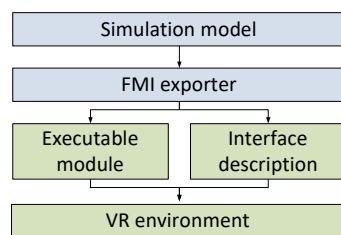


Fig. 3. Workflow with FMI

The Simulink Coder [21] takes a model and generates source code in C or C++ for it. The code is independent from the main software and compiles and runs for its own. So it can be used in the IO node to integrate the inverse kinematics algorithm.

This solution works fine for standard blocks, but not all are supported. Also it is a proprietary solution only working with MATLAB software. Another, more general way is to use FMI (Functional Mock-up Interface, [22]). This is a standardized interface to couple simulation software systems. It consists of an XML file describing the model interface and a binary file including the compiled math functions. It can also be used within the IO node to integrate the needed model. Summarizing, it can be noted that code generation enables an easy way of including models from the simulation software into a VR environment. As long as the model interface doesn't change, a new generation of the model code and a recompilation of the IO node source code are sufficient to maintain model changes into the VR.

4.5. Interaction in Virtual Scene

For the interaction, it is intended to move the manipulator in the joint space with FK or in the task space with IK. Both cases assume a well-modelled hierarchy of the scene nodes (parent-child order) depending on the kinematic structure of the robot joints. For the FK the joint values given by the user can be applied directly to the transformation matrixes of the

chosen joint scene nodes. This can be achieved by a separate GUI element in the VR scene or by picking and manipulating the scene nodes directly (e.g. with the *TouchSensor* in X3D).

Interacting in task space requires more efforts. The input in the task space can be done with a space mouse, a 6-DOF device which allows the translation along the 3 axes and rotations around them. This can be exploited to influence the position and orientation of the end effector (TCP). Other kinds of input devices are also possible, e.g. a tracking system with minor modifications to the interface. The new aligned TCP cannot be applied to the transformation matrixes. First, the data must be given to the simulation coupling which calculates the needed joint angles.

The described interaction approach was realized with X3D using script nodes. A graphical interface was developed similar to the physical controller that allows keeping sight of the current state of the robot also in the virtual scene, showing seven joint angles as well as end effector position and orientation via Euler angles.



Fig. 4. A visual model of the KUKA SmartPad

In this user interface (see Figure 4) seven sliders are present, one for each DOF (with reference to the joint limits and the actual value expressed in digits). Moreover, a calculation of the FK is present with end effector position and orientation values (Figure 4). Taking advantage of the script node in X3D, it is also possible to trace the path followed by the end effector in real-time (Figure 5a).

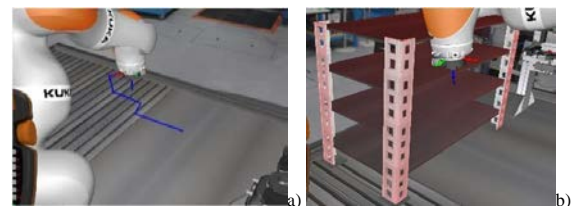


Fig. 5. a) Path drawing in the virtual scene;
b) Bounding box collision detection

The VR software also offers collision detection based on bounding boxes already implemented in an example scenario. When a collision between the robot and the virtual scene occurs, the object of the virtual scene involved in the collision is highlighted in red. This event warns the operator about the collision (Figure 5b).

4.6. Coupling of VR with the Real Robot

The connection between robot and VR can also be done within the IO nodes (see sections 4.2 and 4.3). The specific implementation depends on the used interface. E.g. the KUKA LBR iiwa offers the FRI (Fast Robot Interface) [23]. The FRI provides a direct low-level access to the KUKA Robot Controller at rates up to 1 kHz (the user can set the flexible cyclic time frame between 1 and 100 ms), while preserving all its industrial-strength features (such as, teaching/touchup, execution of motion primitives, fieldbus I/O and safety). It enables to read the current (measured) joint angles which then can be used to adapt the virtual robot model. To control the real robot arm from VR side, more efforts are necessary. First, the new desired position (TCP) and the current angle positions are needed. These are the input data for the kinematic simulation model which must be executed (see section 4.3). The results are (if no error case occurs) the new angle positions for every axis of the robot. These must be adapted to the virtual model and to the interface of the real robot arm. It must be kept in mind that only valid positions should be sent to the robot interface.



Fig. 6. Coupling between the virtual scene and the robot with path drawing.

5. Conclusions and Future Works

This paper discusses a framework for the virtualization and coupling of a redundant manipulator with the aim of an optimized interaction between human and robot. Main focus was an easy and concise workflow, cutting out all the considerable steps of code translations and subsequent testing due to the use of different platforms.

It was possible to obtain a coupling between the real robot and a complete new virtual scene. This VR model allows controlling the end effector in the task space via IK and the joints in the joint space via FK algorithm.

This project can be interpreted as an initial work-frame in the direction of further studies for a deep investigation in this field. In this perspective we can delineate more ways for the improvement and a better definition: the optimization of the algorithm for the inverse kinematic taking into account also the projection into the null space due to redundancy of the robot and a better management near the workspace limit or near singularities. This optimization can take advantage of the models already used in MATLAB/Simulink following the same workflow used for the existing algorithm.

Using that optimization, the opposite connection between the robot and the virtual model can be achieved, always using

the method discussed in the previous section. It could be possible to drive the real robot starting from the virtual scene, which can be really helpful for teleoperation and for the teaching of the robot.

Finally another possibility can be to develop a dynamic model of the robot with a reverse engineering approach [24] to better understand and refine the control of the robot in the virtual scene.

Finally, the VRCP Lab, as above mentioned, is already set up with a tracking camera system and a CAVE with another tracking system. So future research could be done on the robot control and on teleoperation within the CAVE.

References

- [1] D. Gorecky, M. Schmitt, M. Loskyll, and D. Zuhlke, "Human-machine interaction in the industry 4.0 era," *Proc. - 2014 12th IEEE Int. Conf. Ind. Informatics, INDIN 2014*, pp. 289–294, 2014.
- [2] P. Dépincé, D. Chablat, and P.-O. Woelk, "Virtual Manufacturing: Tools for improving Design and Production," *Dans Int. Des. Semin. - CIRP Int. Des. Semin.*, no. 1, pp. 1–12, Aug. 2007.
- [3] A. Y. C. Nee and S. K. Ong, "Virtual and Augmented Reality Applications in Manufacturing," *IFAC Proc. Vol.*, pp. 15–26, 2004.
- [4] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, "An atlas of physical human-robot interaction," *Mech. Mach. Theory*, vol. 43, no. 3, pp. 253–270, 2008.
- [5] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger, "The DLR lightweight robot: design and control concepts for robots in human environments," *Ind. Robot An Int. J.*, vol. 34, no. 5, pp. 376–385, 2007.
- [6] E. Matsas and G. C. Vosniakos, "Design of a virtual reality training system for human-robot collaboration in manufacturing tasks," *Int. J. Interact. Des. Manuf.*, no. April 2016, 2015.
- [7] M. Vasic and A. Billard, "Safety issues in human-robot interactions," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013, pp. 197–204.
- [8] ISO 10218: Manipulating Industrial Robots-Safety (International Organization for Standardization, Geneva 1992)
- [9] OSHA: Guidelines for Robotic Safety, OSHA Instruction PUB 8-1.3 (Occupational Safety and Health Administration, Washington 1987)
- [10] M. Geravand, F. Flacco, and A. De Luca, "Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture," *International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 4000–4007, 2013.
- [11] B. Siciliano, L. Villani, *Robot Force Control*, Kluwer Academic Publishers, Boston, MA, 1999.
- [12] A. De Santis, G. Di Gironimo, A. Marzano, B. Siciliano, and A. Tarallo, "A Virtual-Reality-based Evaluation Environment for Wheelchair-mounted Manipulators," pp. 35–42, 2008.
- [13] W. Shen, J. Gu, and M. Yide, "3D kinematic simulation for PA10-7C robot arm based on VRML," *Proc. IEEE Int. Conf. Autom. Logist. ICAL 2007*, no. 7, pp. 614–619, 2007.
- [14] G. C. Vosniakos and X. V. Gogouvis, "Structured design of flexibly automated manufacturing cells through semantic models and petri nets in a virtual reality environment," *Int. J. Interact. Des. Manuf.*, vol. 9, no. 1, pp. 45–63, 2014.
- [15] H. J. Yap, Z. Taha, S. Z. M. Dawal, and S. W. Chang, "Virtual reality based support system for layout planning and programming of an industrial robotic work cell," *PLoS One*, vol. 9, no. 10, 2014.
- [16] F. Chinello, S. Scheggi, F. Morbidi, and D. Prattichizzo, "The KUKA Control Toolbox: motion control of kuka robot manipulators with matlab," *IEEE Robot. Autom. Mag.*, pp. 1–10, 2010.
- [17] A. Y. C. Nee, S. K. Ong, G. Chrysosouris, and D. Mourtzis, "Augmented reality applications in design and manufacturing," *CIRP Ann. - Manuf. Technol.*, vol. 61, no. 2, pp. 657–679, 2012.
- [18] B. Siciliano, L. Sciacicco, L. Villani and G. Oriolo, "Robotics: Modelling, Planning and Control," Springer, 2010.

- [19] J. Behr, U. Bockholt, and D. Fellner, "Instantreality - a framework for industrial augmented and virtual reality applications," *Virtual Real. Augment. Real. Ind.*, pp. 91–99, 2009.
- [20] <http://www.web3d.org/standards>. Last visit: 28-04-2016
- [21] The MathWorks Inc., "Simulink Coder Reference," 2015.
- [22] O. Enge-Rosenblatt, C. Clauß, A. Schneider, and P. Schneider, "Functional Digital Mock-up and the Functional Mock-up Interface – Two Complementary Approaches for a Comprehensive Investigation of Heterogeneous Systems," 8th Int. Model. Conf. Dresden, pp. 748–755, 2011.
- [23] G. Schreiber, A. Stemmer, and R. Bischoff, "The fast research interface for the kuka lightweight robot," *IEEE Work. Innov. Robot Control Archit. Demanding Appl. How to Modify Enhanc. Commer. Control. (ICRA 2010)*, pp. 15–21, 2010.
- [24] C. Gaz and A. De Luca, "Identifying the Dynamic Model Used by the KUKA LWR : A Reverse Engineering Approach," pp. 1386–1392, 2014.