AS SEEN IN:
INNOVATIE NU | SEPTEMBER 2021

AI IN COLLABORATIVE ROBOTICS

Authors:

Christoph Nicksch

Research Fellow
Laboratory for Machine Tools and Production Engineering WZL of RWTH Aachen

Lars Leyendecker

Research Fellow
Fraunhofer Institute for Production Technology IPT

Guocai Ma

Engineer
State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering

Fei Li

Senior Engineer
State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering

Zhihong Cao

Engineer
State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering

Zhujuan Cal

Senior engineer
State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering

Tobias Claus Brandstätter

Research Fellow
Fraunhofer Institute for Production Technology IPT

Jonathan Krauß

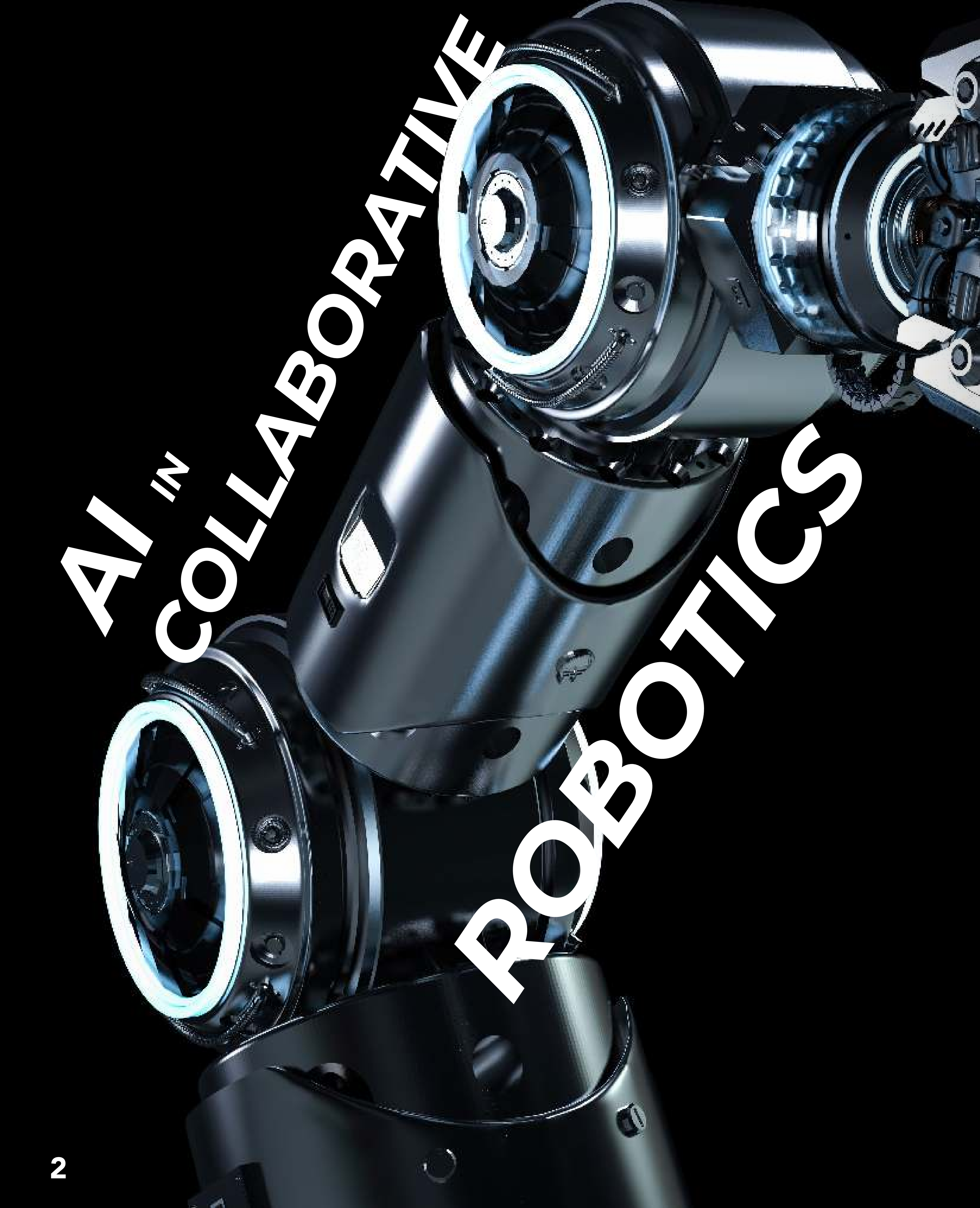
Head of Department
Production Quality
Fraunhofer Institute for Production Technology IPT

Robert H. Schmitt

Director
Laboratory for Machine Tools and Production Engineering WZL of RWTH Aachen and Fraunhofer Institute for Production Technology IPT

ADVANCED MANUFACTURING CENTER

ISSN 2772-428X



AI IN COLLABORATIVE ROBOTICS

[...] each robot can perceive its environment and can adapt to changes - an ability that conventionally programmed manipulators exhibit only to a very limited extent.

In the era of Industry 4.0, collaborative robots are one of the main pillars enabling flexible automation. In particular, dual-robot systems are increasingly seen as a promising trend of industrial automation for assembly and pick-and-place tasks. Dual-robot systems consist of two collaborating robot arms and offer advantageous characteristics. For instance, high redundancy can improve the flexibility in task manipulation, while synchronised manipulation can shorten the operation times.

To implement a dual-robot system, much of efforts are made on task planning. The two arms need to be synchronised to each other in time and space to avoid collisions and to improve the efficiency in terms of the operation time. Additionally, peripheral components as well as workpieces or workpiece positioners must be considered to define feasible trajectories, i.e. sequences of six-dimensional positions in Cartesian space, for both robot arms. To take advantage from dual-robot systems economically, the effort for task planning must be automated as well, in order to significantly decrease the set-up time. Following this idea, this work proposes an AI-based solution for a dual-robot system which enables autonomous trajectory planning for a given pick-and-place task by minimising the set-up time and by avoiding collisions.

A pick-and-place use case has been chosen as an illustrative example to verify the proposed algorithm. Within the use case, 16 cylindrical pins are palletised from a pallet to a turntable. The system consists of two KUKA KR 10 R1100 robots with six degree of freedom each. Both robots are equipped with two-finger grippers and share a common workspace. The system setup is shown in Figure 1.

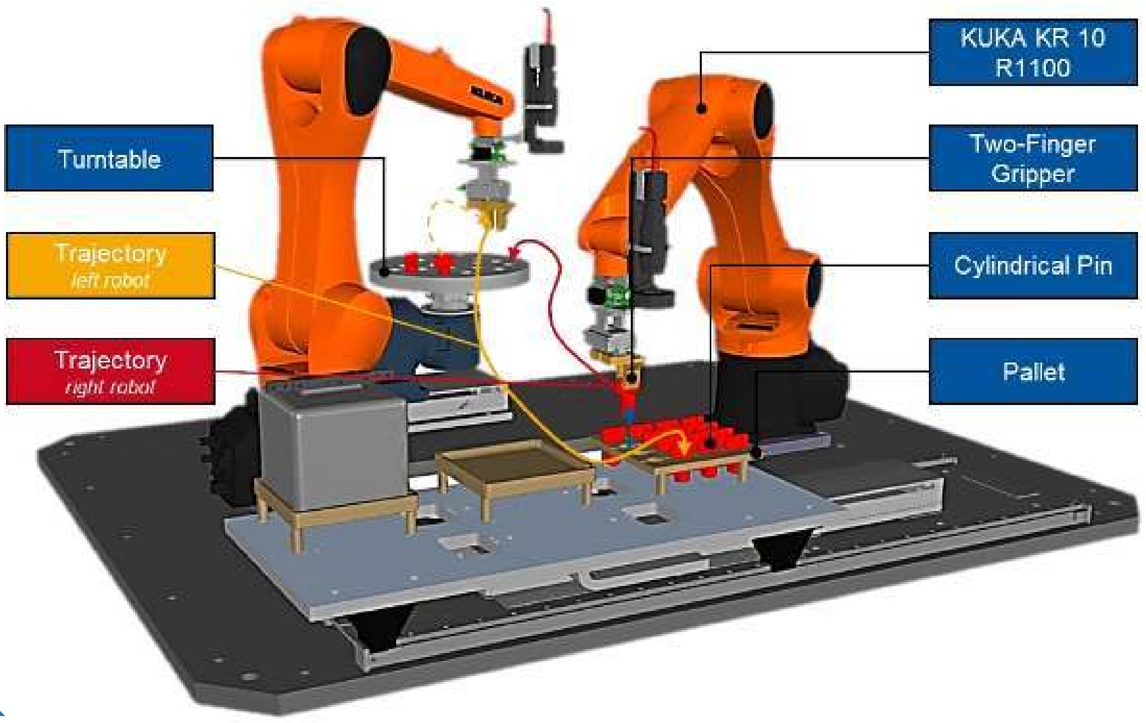


Figure 1: Components of the dual-robot system.

Problem Statement

The problem of robotic trajectory planning is the determination of six-dimensional pose sequences at a defined time step that the robot will approach with its tool center point (TCP) to perform a desired motion. On the one hand, robotic manipulation tasks generally exhibit a hierarchical structure. On the other hand, previous work demonstrates the suitability of AI-based trajectory planning for single robot setups. Our approach leverages and combines these two findings for multi-robot setups by designing a hierarchical control architecture that establishes a framework for multi-agent systems (see Figure 2 left): A superior manager logic supervises the fulfillment of the overall task and assigns subtasks to different specialised agents. Each individual agent is responsible for the trajectory planning of a particular robot at a particular time and is designed as a deep reinforcement learning (DRL) agent.

Approach

The goal is to train agents in dedicated simulation environment to acquire sub-task-specific skills utilising the reinforcement learning framework (see Figure 2 right).

Accordingly, an agent receives information about the state of the robot in its working environment and, based on this, selects an action to move the robot in space. In this way, each robot can perceive its environment and can adapt to changes - an ability that conventionally programmed manipulators exhibit only to a very limited extent. In addition to the state and action space, a reward function needs to be defined that evaluates the agent's actions in form of a numerical reward signal. The underlying algorithm uses this signal to adjust its internal policy – a mapping from states to actions – that is represented by an artificial neural network. Based on this continuous perceive-act-and-adjust loop, the agent gradually optimises its trajectory planning in order to complete the sub-task in the most optimal manner regarding well-defined performance and quality criteria.

Once the required agents have been successfully trained, they are integrated into the conditional control structure that represents the multi-agent system. The resulting modular architecture of interchangeable and specialised agents is intended to foster the reusability of task-specific learning environments. The resulting ever-growing library promises to significantly reduce the development efforts for future multi-robot use cases.

Starting on the system architecture level, one result of the development process is a client server architecture that allows the communication between the Python learning environment, in which the manager logic and the intelligent control policies are implemented, and the robot simulation (see Figure 2 left). The client-server communication is responsible for sending the control commands to the simulation and for receiving the corresponding simulation responses. For the given dual robot use-case, the control structure includes two agents, one for each robot, that simultaneously palletise the cylindrical pins from the pallet on to the turntable. The learning environments in which the trajectory planners are trained are structured in the same way: The observation for each robot includes the Cartesian position of its TCP and a target position that represents either the pick or the place position of the currently considered pin. The action vector that is predicted by the artificial neural network is a six-dimensional Cartesian position command. The internal controller takes this command, performs inverse kinematics and drives the joints of the manipulator accordingly. Both robots are trained simultaneously in the same simulation to allow them to collaborate. The reward function that feedbacks evaluations of the taken actions to the agent, is designed to reward decreasing distances between TCP and target position as well as decreasing trajectory durations. As a result, the corresponding target positions are approached very time-efficiently, but the robots are also likely to collide with each other or with static objects within their workspaces. To avoid this highly undesirable behavior, a collision detection is further implemented and the corresponding digital collision signal is added to the reward function. By penalising collisions, the agents are explicitly trained to prevent undesirable contacts.

Results

Figure 3 shows the evolving reward signal over the course of approximately twelve hours of training: In the beginning, the reward is comparatively low and shows strong peaks. This is an indication that the agent is not yet able to complete the task and a large number of collisions occur. As training progresses, the reward increases, shows fewer peaks, and converges to the maximum total reward of zero. The fact that peaks still occur at later times is because of the agent's internal exploratory behavior, which is being disabled during operation.

The final tests in the simulation show that the two robots perform collision-free motions within their workspaces and need 86.14 seconds for palletising the 16 pins. In comparison to a single, manually teached-in robot, this is a 46 % improvement in terms of time. Thereby, it is demonstrated that simulation-based autonomous trajectory planning using reinforcement learning proves to be a promising alternative to conventional teaching, which for such more complex

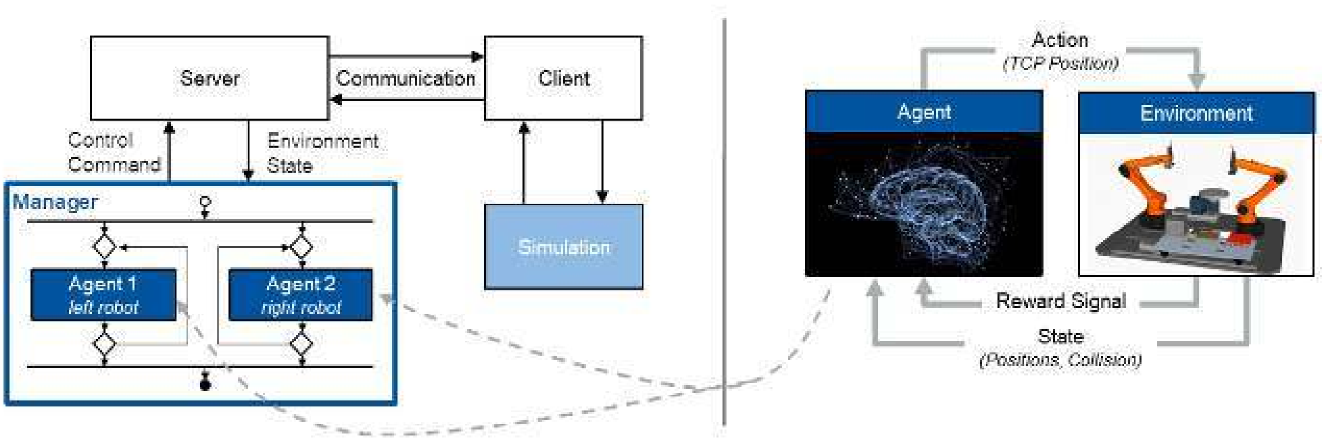


Figure 2: System architecture and manager-agent structure (left) and deep reinforcement learning control loop (right)

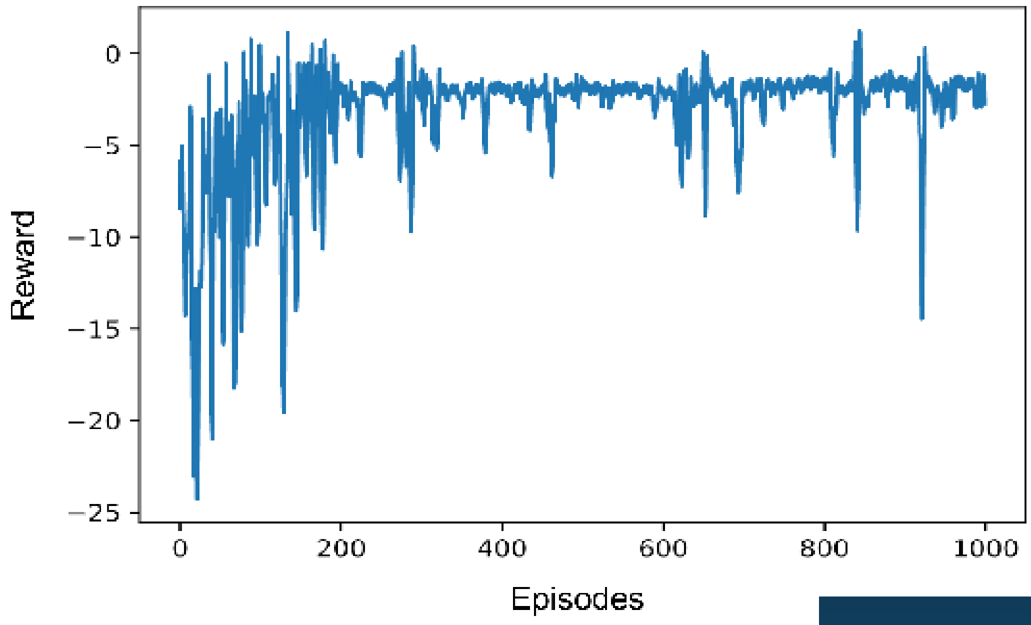


Figure 3: Evolvement of the reward signal received by one agent per episode over the entire training period of 1,000 episodes.

multi-robot systems requires extensive manual efforts resulting in long-lasting ramp-up and downtime phases. In addition, the hierarchical and modular approach proves to be particularly efficient when dealing with further use cases: By reusing existing learning environments, which may need to be slightly adapted to new tasks, development times progressively decrease. Accordingly, the efforts are mainly reduced to deriving the higher-level control structure from the structure of the manipulation task, monitoring the training processes and validating the final multi-agent solution.

Conclusion

The presented approach shows that multi-agent-systems based on deep reinforcement learning can enable efficient and full automated trajectory planning for dual-robot systems. Operation times can be decreased by 46 % in comparison to a single robot for the considered pick-and-place task. Future work will focus on a more complex dual mode manipulation task by handling a heavy component that exceeds the maximum payload of a single robot.

Lessons learned

- Simulations can be used for autonomous AI-based trajectory planning for dual robot systems
- While setting up the system architecture and especially the communication between simulation and learning environment, keep complexities low at first and focus on a working infrastructure
- Once the infrastructure functions, perform short adapt-train-evaluate cycles on the learning environments to obtain the desired agent skills
- Even if trajectory planning is conducted autonomously, it will require an unneglectable amount of computational time

Authors:

Christoph Nicksch

Research Fellow
Laboratory for Machine Tools and Production Engineering WZL of RWTH Aachen

Lars Leyendecker

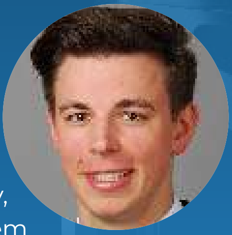
Research Fellow
Fraunhofer Institute for Production Technology IPT

Zhujuan Cal

Senior engineer
State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering

Guocai Ma

Engineer
State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering



Tobias Claus Brandstätter

Research Fellow
Fraunhofer Institute for Production Technology IPT

Fei Li

Senior Engineer
State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering

Jonathan Krauß

Head of Department
Production Quality
Fraunhofer Institute for Production Technology IPT

Zhihong Cao

Engineer
State Key Laboratory of Intelligent Manufacturing System Technology, Beijing Institute of Electronic System Engineering

Robert H. Schmitt

Director
Laboratory for Machine Tools and Production Engineering WZL of RWTH Aachen and Fraunhofer Institute for Production Technology IPT