



HAL
open science

ROS Based Safety Concept for Collaborative Robots in Industrial Applications

Stephan Kallweit, Robert Walenta, Michael Gottschalk

► **To cite this version:**

Stephan Kallweit, Robert Walenta, Michael Gottschalk. ROS Based Safety Concept for Collaborative Robots in Industrial Applications. 24th International Conference on Robotics in Alpe-Adria-Danube Region, May 2015, Bucarest, Romania. pp.27-35, 10.1007/978-3-319-21290-6_3 . hal-01298439

HAL Id: hal-01298439

<https://hal.science/hal-01298439>

Submitted on 6 Apr 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ROS Based Safety Concept for Collaborative Robots in Industrial Applications

Stephan Kallweit, Robert Walenta and Michael Gottschalk

Abstract The production and assembly of customized products increases the demand for flexible automation systems. One approach is to remove the safety fences that separate human and industrial robot to combine their skills. This collaboration possesses a certain risk for the human co-worker, leading to numerous safety concepts to protect him. The human needs to be monitored and tracked by a safety system using different sensors. The proposed system consists of a RGBD camera for surveillance of the common working area, an array of optical distance sensors to compensate shadowing effects of the RGBD camera and a laser range finder to detect the co-worker when approaching the work cell. The software for collision detection, path planning, robot control and predicting the behaviour of the co-worker is based on the Robot Operating System (ROS). A first prototype of the work cell shows that with advanced algorithms from the field of mobile robotics a very flexible safety concept can be realized: the robot not simply stops its movement when detecting a collision, but plans and executes an alternative path around the obstacle.

Keywords Collaborative robot · Human-Robot interaction · Safety concept · Workspace monitoring · Path planning · RGBD camera · ROS

S. Kallweit (✉) · R. Walenta
Institute for Mobile Autonomous Systems and Cognitive Robotics (MASCOR),
University of Applied Sciences Aachen, Aachen, Germany
e mail: kallweit@fh aachen.de

R. Walenta
e mail: robert.walenta@alumni.fh aachen.de

M. Gottschalk
ID Ingenieure & Dienstleistungen GmbH, Euskirchen, Germany
e mail: gottschalk.m@id engineerin.com

1 Introduction

Modern production of consumer goods will develop from mass production of identical items to customized products in the near future [1], a so called “mass customization”. The flexibility of a production line can be increased by using a combination of a flexible human worker and an industrial robot [2]. The robots task is the disposing of all repeatable jobs, which are too time consuming, too heavy for the human co-worker or need a relatively high positional accuracy. Several aspects need to be considered for a collaborative workspace, as shown in Fig. 1. The human co-worker is the flexible and fault tolerant partner, guiding the robot to his next mission. In our study a typical “Pick-and-Place” task is analysed. Although this is the basic scenario for a variety of assembling tasks, the drawback is the threat generated by a fast moving robot [3].

Various safety concepts exist in the industrial environment, which are even commercially available like camera based monitoring systems [4]. The system detects an intruder in the working area of the robot and stops the running procedure. This is not beneficial for a real collaborative approach: the process would be most of the time switched off. Collaboration between the human co-worker and the industrial robot is only possible if the robot perceives the dynamically changing environment and reacts to the new situation accordingly [5]. In the field of mobile robotics, this scenario became already a reality: the Google self-driving car is licensed to autonomously drive in several states in the US [6]. Another example from the field of logistics are Kiva Systems, recently acquired from Amazon [7], which deploy autonomous mobile robots for commissioning tasks. As a consequence, advanced algorithms from mobile robotics can be applied to standard industrial tasks, like e.g. assembly or pick-and-place jobs.

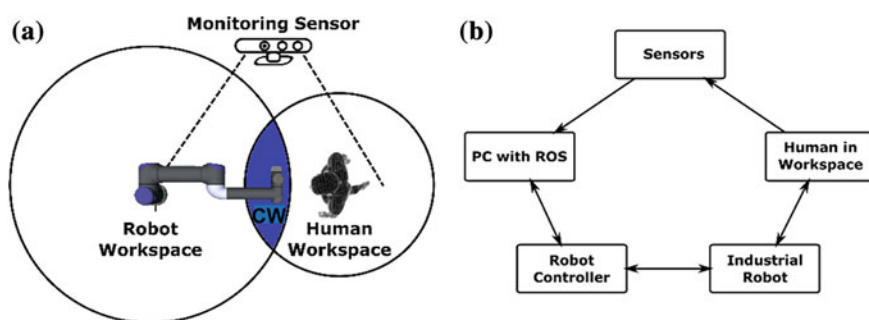


Fig. 1 a Collaborative workspace (CW) between human and robot monitored by multiple sensors; b interaction diagram showing the concept used for safe human robot collaboration

2 Safety Requirements

The operation of a collaborative robot system is strictly regulated. There are different regulations for each country: e.g. in Germany a license is required according to a combination of DIN-EN-ISO 10218 [8] and DIN-EN-12100 [9]. Here the fundamental risks are described which arise from the operation of a robot system without protection fences. Procedures are explained to discover and eliminate dangerous threats in order to protect the human co-worker. To operate a robot in a collaborative mode, the system needs to fulfil the following criteria:

- The robot needs to be realized in a light-weight design.
- An independent controller monitors the robot motion.
- Sensors are monitoring the position of the human co-worker.
- Safety classification must be done according to DIN EN ISO 13849 [10].

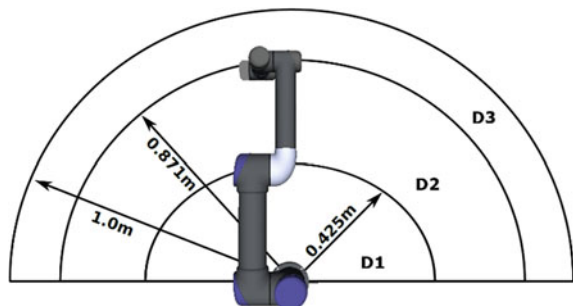
Figure 2 shows different “Hazard Zones” (D1–D3) which are generated considering the mentioned DIN standards, the speed of the robot, the latency time of the sensors and the distance between co-worker and robot base point. Zone D1 is the zone with highest hazard. The co-worker cannot leave quickly enough D1, if the robot makes an unpredicted movement. The only safe state for this situation is to stop completely the motion. Zone D2 is the collaborative zone. Here the interaction with the co-worker is allowed and all the sensor devices are monitoring continuously the human co-worker and the robot. If a threat is detected in D2, the system goes into a safety stop.

2.1 General Principles for the Design

An industrial robot is controlled by a user program which defines a specific task. Each part of this program, which initiates an action of the robot itself, has to be monitored by a safety function. This function needs to be classified as category 3 with “performance level d”, which means:

- Failures cannot lead to a loss of safety functions.
- Failures have to be recognized at an early state.

Fig. 2 Different “Hazard Zones” considering sensor latency and distance



- Every safety function leads to a safe state.
- The safe state is kept until the threat is relieved.

“Performance level d” indicates a probability of failures occurring when the system is used. This probability is below 10^{-6} for a critical failure/hour in the case of level d [10]. Safety functions are used for risk reduction. For a collaborative system the control of the robot motion is of great importance. This includes the ability to stop, to control the speed and to limit the free working area of the robot. The system needs an emergency stop and a safety stop. An emergency stop shuts the system down and cuts it from the energy source. A safety stop is only stopping the motion and waiting until a safe operation is guaranteed. The TCP velocity needs to be continuously controlled. It must be ensured that a maximum speed of 250 mm/s is not exceeded, when the co-worker is entering zone D3. The software must implement a limitation of the robot working range. When the robot reaches this limit, the robot has to stop. This simple method is an instrument for reducing the interaction area with the human co-worker and defines the area where the robot is allowed to move.

2.2 Collaboration Scenarios

Different collaboration tasks need different setups like described in [11]. The most interesting scenario for collaboration is a direct, joint interaction between human co-worker and robot. The other possible collaboration setups include e.g. forbidden areas, where a robot is integrated in a production line and some parts need to be loaded and unloaded to the production cell or heavy robot tools handled, where the human co-worker is just shortly exposed to the robot working area.

3 System Architecture

Several tasks can be associated to a system for robot collaboration. These tasks can be formulated quite similarly to the tasks a mobile robot has to solve; it is thus natural to use similar strategies. The main tasks for mobile robots are [12]: (a) Locomotion; (b) Perception; (c) Localization, and (d) Navigation.

Locomotion is realized by the robot standard hardware controller. When using an external control program which generates control commands for the joint positions, an external kinematic solver is needed. We use the Open Source library KDL [13] to solve the inverse kinematic iteratively for an UR5 light-weight industrial robot. The robot controller has a software interface to ROS [14] which simplifies the development.

The system controller needs to acquire enough data to generate a model of the current state. This perception of the environment is often done with 3D vision

systems; in our setup we use a simple RGBD camera, several proximity sensors and a laser range finder. These sensors are supported by ROS or simple interfaces are developed with the roserial package [14].

Another module is needed for collision detection, which is part of the localization task. Here the Open Source library FCL [15] is used, to allow a probabilistic collision check of static and dynamic objects in 3D environments. For collision avoidance a path planning algorithm is necessary for navigation. A common library for path planning within ROS is OMPL [16], which uses a local and a global planner to generate possible 3D paths of the kinematic structure and to check the closer surrounding of the robot. The path planning was simulated online in a virtual environment using MoveIt [17]. Here all the necessary geometrical constraints are provided as URDF [18] data sets. Figure 3 shows the main components of the generated system.

3.1 Robot Cell Setup

A laser range finder, mounted in front of the robot cell observes the plane orthogonal to the base coordinate plane of the robot and detects the human co-worker when entering D3. The RGBD camera is placed ~ 2 m above the working area for collision detection, i.e., detecting objects in all three Zones (D1 D3). Using depth images from the RGBD camera it is possible to monitor nearly the robot's complete working area.

Virtual fences limit the possible working area of the robot from three directions: in front of the robot cell an interaction with the robot is possible (D3 and D2); this area is partially monitored by the RGBD camera and some additional proximity

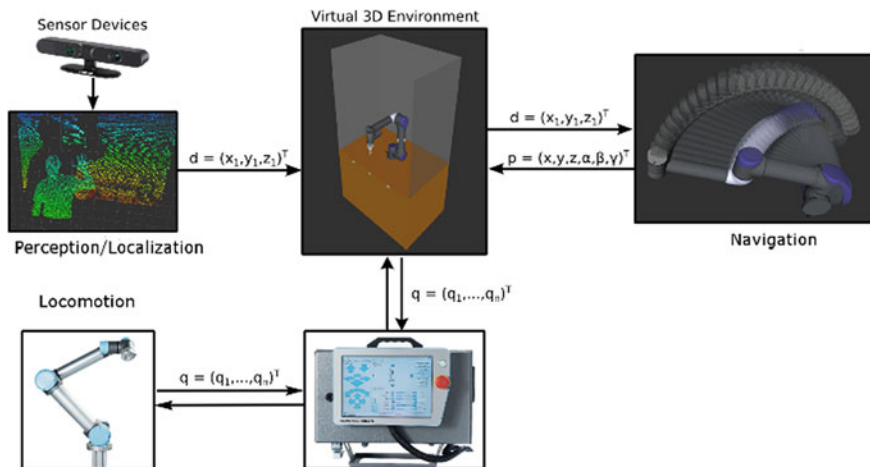


Fig. 3 Modules of the system architecture

sensors with a range of ~ 800 mm (Sharp GP2 Y0A21). The sensor signals S1 S3 are used for the safety state machines.

The robot itself is equipped with a sensor belt in order to detect objects near the kinematic chain. This belt consists of four of the already mentioned low cost proximity sensors (S4 S7), which have a lower latency time than the RGBD camera. Depending on the detected collision threat, the movement of the robot stops.

4 Safety Controller

The task of the safety controller is to perceive the working area and to monitor the robot and human co-worker. The implementation of object detection, collision avoidance, path planning and of the state machines was done in ROS using MoveIt. Due to the ROS communication engine, it is possible to exchange data for the UR5 robot via a simple TCP/IP connection. The robot hardware controller is expecting new joint positions, speed and acceleration values with a data rate of 125 Hz as a ROS client. The new data is FIFO transferred and directly sent to the servo motors.

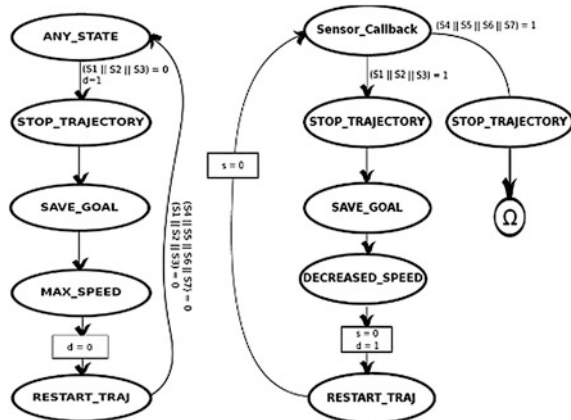
As a test case, we implemented a simple Pick and Place task with several PTP and LIN movements. It is worth mentioning that MoveIt doesn't support LIN movements, so the target points for a linear movement need to be calculated in advance and sent to the robot controller. A simple state machine is controlling the Pick and Place process. The transitions can only be triggered if non-blocking calls are used. That prevents from using the `move()` and `execute()` functions of the MoveGroup. One possibility to use the state machine is using MoveGroupAction, which starts the ActionServer and publishes an event when the action is triggered.

The safety functions are implemented as state machines. As soon as the human co-worker is detected in D3, the system reduces the speed to 250 mm/s. If the co-worker enters D2 and he is closer than 200 mm to the robots upper arm, the system is stopped. It is as well stopped, when entering D1. If a colliding movement of the robot in D2 or D3 is detected, the system stops and calculates an alternative path around the colliding object. If successful, the planned path is executed. Figure 4 depicts the safety state machines where the signals S1 S7 are used for additional intruder detection not covered by the RGBD and laser range finder data.

5 Results

A successful path planning with colliding objects is generated using the mentioned setup. When the human co-worker stretches his arm into the collaboration area D3-D2, the arm is detected as an obstacle but as the bounding box of the arm is not interfering with the planned trajectory, a re-planning is not necessary. The robot moves with reduced speed of 250 mm/s to the target position. If the arm is detected

Fig. 4 Safety functions as state machines



as a colliding object, the system is stopped for a short time for re-planning the path around the obstacle and—if successful—moves the robot using the new trajectory.

Other collaboration scenarios are as well possible, e.g. a pick-and-place application, where the robot should pick the object from the co-workers hand. Here the hand of the co-worker is not classified as a collision object, because he is in a special collaboration zone. If the hand would be defined as a collision object, the robot wouldn't be able to reach the target position. This collaboration zone has only a certain limited height ~ 50 mm, so that it is possible to interact with the robot. If the dimension of this zone is increased too much, it will be dangerous for the co-worker.

In order to get a first indication of the latency time of only the collision detection system, the time between a trigger from the laser range finder and the complete stop of the robot was measured using internal time stamps from ROS; proximity sensors were ignored for this test. These time stamps are indicators which should be interpreted with care, due to the limited real time possibilities of ROS. This data is a first estimate of a possible threat for the co-worker. Figure 5 shows the setup of the measurement: the robot is repeatedly moving from starting position A to target position D on a circular trajectory. The co-worker enters the zone D3 via B1 or B2, disturbing the trajectory. The time for a complete stop of the robot when detecting the collision object is measured. The laser range finder doesn't stop the robot; only the safety controller based on the collision detection without the proximity sensors is used.

Three different speeds for the robot were used: 10, 50 and 100 %. The mean values \bar{t} and the standard deviations s_d were calculated from 30 measurements (Table 1).

Obviously the latency time depends on the point where the co-worker enters the collaboration zone but does not depend on the robot's speed. The latency time between the two investigated points differs by ~ 0.2 s. During the test sequence the robot always stopped before the co-workers hand, so that the measured difference in latency time for B1 and B2 is due to different braking ramps of the robot hardware controller.

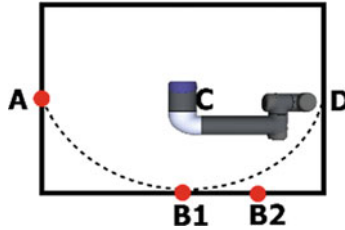


Fig. 5 Setup for measuring the latency time of the collision detection module

Table 1 Average latency times for different intrusion points and robot speeds

| Speed (%) | B1: \bar{t} in s | B2: \bar{t} in s | B1: s_d in s | B2: s_d in s |
|-----------|--------------------|--------------------|----------------|----------------|
| 100 | 0.7 | 0.96 | 0.2 | 0.17 |
| 50 | 0.63 | 0.9 | 0.19 | 0.1 |
| 10 | 0.74 | 0.97 | 0.16 | 0.1 |

6 Conclusion and Future Aspects

The successful setup of a collaborative robot system using the ROS framework shows that combining algorithms from mobile robotics with industrial tasks is suitable for future automation purposes. By using open source libraries a very flexible low cost collaborative system was developed. The used sensors are common, low cost devices which are not suitable for industrial safety applications, but the capabilities of combining 3D sensors with some additional proximity devices are considered.

The system is still not fast enough to detect quick movements of human co-workers due to the latency time of the RGBD camera and the high computational load. Here ROS is beneficial again: ROS scales quite well on multicore machines due to the fine-grained structure. Future multicore CPU power can be easily used.

Next steps in the research will be: the use of faster and more accurate RGBD cameras, e.g. some TOF cameras with higher frame rates and higher resolution. A decentralized approach with a cluster of powerful embedded systems will increase the overall performance. The use of a different kinematic solver can as well improve the global performance: KDL solves iteratively, so the algorithm spends more time in some kinematic positions compared to an analytic solver.

References

1. The Economist: The third industrial revolution, vol. 12, no. 16, <http://www.economist.com/node/21553017> (2012)
2. Bascetta, L., Ferretti, G., Rocco, P., Ardo, H., Bruyninckx, H., Demeester, E., Di Lello, E.: Towards safe human robot interaction in robotic cells: an approach based on visual tracking

- and intention estimation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2971-2978 (25-30 Sept 2011)
3. Bortot, D., Born, M., Bengler, K.: Directly or on detours? How should industrial robots approximate humans? In: Proceedings of 8th ACM/IEEE International Conference on Human Robot Interaction (HRI), pp. 89-90 (3-7 Mar 2013)
 4. Pilz, M.: SafetyEye, Internet <http://www.hft-stuttgart.de/Studienbereiche/Vermessung/Bachelor-Informationslogistik/Aktuell/Veranstaltungen/inflogtag2014/SafetyEYE-HFT-Stuttgart-09-04-14-genehmigt.pdf> (2014)
 5. Balan, L., Bone, G.M.: Real time 3D collision avoidance method for safe human and robot coexistence, intelligent robots and systems. In: IEEE/RSJ International Conference, pp. 276-282 (2006)
 6. CBS Detroit: New Law Allows Driverless Cars on Michigan Roads. 28 Dec 2013
 7. Kucera, D.: Amazon Acquires Kiva Systems in Second Biggest Takeover. Available at: <http://bloom.bg/Gzo6GU> (2012)
 8. DIN EN ISO 10218-1: Safety requirements for industrial robots, (Industrieroboter Sicherheitsanforderungen) (2012)
 9. DIN EN ISO 12100: Safety requirements for machines – common design guiding principles (Sicherheit von Maschinen – Allgemeine Gestaltungsleitsätze) (2011)
 10. DIN EN ISO 13849: Safety requirements for machines – safety related parts of control units (Sicherheit von Maschinen – Sicherheitsbezogene Teile von Steuerungen) (2010)
 11. Maschinen Markt: Neue Norm soll Sicherheit zwischen Mensch und Roboter gewährleisten, <http://www.maschinenmarkt.vogel.de>, Internet (2014)
 12. Siegwart, R., Nourbakhsh, I.: Introduction to Autonomous Mobile Robots. MIT Press, Cambridge (2004)
 13. Orocos Kinematics and Dynamics, <http://www.orocos.org/kdl>, Internet (2014)
 14. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.: ROS: an open source robot operating system. In: ICRA Workshop on Open Source Software (2009)
 15. Pan, J., Chitta, S., Manocha, D.: FCL: a general purpose library for collision and proximity queries. In: IEEE International Conference on robotics and automation (ICRA), 2012, pp. 3859-3866 (2012)
 16. Sucan, I., Moll, M., Kavraki, E.: The open motion planning library. IEEE Robot. Autom. Mag. **19**(4), 72-82 (2012)
 17. Chitta, S., Sucan, I., Cousins, S.: MoveIt!, IEEE Robot. Automat. Mag. **19**(1):18-19 (2012)
 18. Kunze, L., Roehm, T., Beetz, M.: Towards semantic robot description languages. In: IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 5589-5595, (2011)