Dynamic Obstacle Avoidance for Collaborative Robot Applications

Mukunda Bharatheesha
and Carlos Hernandez
and Martijn Wisse
TU Delft Robotics Institute
Delft University of Technology
Mekelweg 2, 2628CD Delft, The Netherlands
Email: {m.bharatheesha, c.h.corbato, m.wisse}@tudelft.nl

Nirmal Giftsun
LAAS-CNRS
7 avenue du Colonel Roche
31077 Toulouse Cedex 4, France
Email: Nirmal.Giftsun@laas.fr

Gautier Dumonteil
Siemens Industry Software SAS
Miniparc Bat. 2 478 Rue de la Découverte, 31670 Labége, France
Email: gautier.dumonteil@siemens.com

***Abstract*—Current collaborative robot arms allow more flexible work cells, where they safely collaborate with human operators augmenting productivity in tasks difficult for traditional automation. However, current solutions for safe interactions imply stopping the robot motion when a collision is detected. This reduces the productivity in an operational setup in which unintended, safe collisions can happen often. Active contact evasion by the robot arm is desirable so that the production process continues despite regular interferences and path obstructions. In the Factory-in-a-day project dynamic collision avoidance technologies have been developed, including a proximity-sensing robot skin, a motion control framework based on proximity-sensing and a reactive path-planning solution. This technologies have been integrated into a dynamic-obstacle avoidance framework successfully tested in simulation and laboratory set-ups. The goal of this paper is to present the obstacle avoidance solution that is currently being implemented with this framework for a collaborative pick and place application prototype. This prototype will be presented at the RoboBusiness Europe event in April 2017.**

I. Introduction

A desire for robotic solutions, particularly in the Small and Medium scale Enterprises (SMEs) is becoming increasingly prominent. Automation and robotics promise to deliver reduction on production costs and increase in productivity. However, traditional automation implies an investment prohibitive for SMEs, whose activities mainly involve small batches of production and high variety of products, for example, due to a seasonal nature of their operations. Concretely, tasks such as assembly, machine filling or packaging, can be automated with a robot in the workcell. However economic feasibility requires to reduce the robotization costs. The Factory-in-a-day project [1] tries to reduce the robotization cost by reducing the system integration cost and installation time. The key idea is that the robot solution is flexible so that it can be quickly re-installed and configured to another temporary product line.

To achieve this flexibility and maintain acceptable levels of productivity, in the Factory-in-a-day approach we propose to automate the easy 80% of the tasks and leave the hard 20% for human co-workers. Robot manipulators provide power, repeatability and extended work-space while the human operators provide flexibility and problem solving capacity. In addition, fenceless collaborative robots save space and installation cost. However, this approach requires a very high level of safety and agility; the robots should be aware of any obstacle, including dynamic obstacles such as its humans co-workers, and be able to move to avoid contact. Whereas current co-bots guarantee safe contacts, they degrade the performance of the work cell because of stopping the production. This is one of the breakthrough innovations of the Factory-in-a-day project, which is the focus of this paper: robot arms that are aware of all (dynamic) obstacles in their environment, and that respond by moving around these obstacles while still continuing their work.

The paper is organised as follows. Section II describes the solution for dynamic obstacle avoidance that makes use of a proximity-sensing robot skin. In Section III the robot motion control architecture to incorporate the collision information as safety constraints to dynamically adapt the trajectory is presented. Section IV presents the preliminary results obtained in two different robot setups. Finally, in Section V we present our concluding remarks and a discussion of the current work in progress.

II. Dynamic Obstacle Avoidance solution for collaborative manipulation

Current collaborative robot solutions guarantee safety, but they use obstacle detection to stop moving. Our dynamic obstacle avoidance solution is that of using obstacle detection to respond by moving around the obstacles while continuing to accomplish the desired tasks. The obstacles are detected by a proximity-sensing robot skin. Additionally, an integrated dynamic motion planning approach creates motion plans that fulfil various task specific constraints for typical industrial applications. For example the work cell 3D model is used to create a consistent model of the work environment, so that collision free trajectories are flexibly generated for different operations. The automatic consideration of these constraints drastically simplifies and speeds-up the deployment of the robot.

The solution presented in this paper relies on the innovative proximity-sensing Artificial Robot Skin (ARS) developed by the Institute of Cognitive Systems Systems in the Technical University of Munich [2]. This modular skin consist of identical 'cells' physically connected forming skin 'patches'. These patches can be applied to cover the robot's links and joints, while being electronically connected to work as a single, modular robot skin. Each cell in the skin produces 4 modalities of perception: 3D acceleration, force, temperature, and distance. The multi-modal signals from the Artificial Robot Skin can be used to control the dynamic behaviour of an industrial robot , for example to achieve compliant motions in a non-compliant robot manipulator [3]. These multi-modal signals can also be exploited to generate semantic representations [4] for teaching new task to the robot [5]. The ARS also features auto-calibration that allows to determine the kinematic chain of each cell to the robot base frame [6].

In this paper, the distance provided by an optical proximity sensor in the skin, is used to detect the obstacles around the robot.

An artist's illustration of the complete dynamic obstacle avoidance solution is shown in Fig. 1. The robot motion control component generates appropriate motion commands for the robot controller to follow the trajectories required for a given task. The proximity-sensing skin covers the links and joints of the manipulator, and produces information regarding potential collisions.

This information is used by the robot motion control module to adapt the robot motions on the fly to fulfil both constraints: following the current trajectory (with a certain tolerance) and avoid collisions. If the collision is unavoidable with local deformations of the current trajectory, the robot motion control module requests a (global) re-planning, which is performed on the fly by the reactive path-planner. The motion control then takes the end effector to the final goal pose using the alternative trajectory. The main functional modules of the system are discussed in the following sections of the paper.

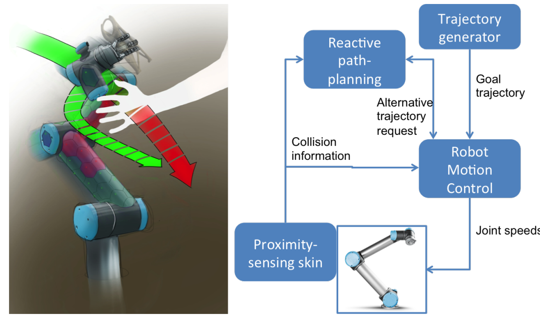

Fig. 1. An artist's schematization of the FiaD Dynamic obstacle avoidance concept is illustrated on the left side. On the right, an overview of the main components of the solution.

III. Robot Motion Control Based on Proximity Sensing

The motion control is achieved using the Stack of Tasks (SoT) controller framework [7] which employs a hierarchical jacobian control strategy eliminating the analytical inverse kinematics computation thus making it a generic controller for all robot platforms. The controller's hierarchical nature allows the robot to handle multiple kinematic tasks simultaneously exploiting the kinematic redundancy of the robot. The controller's real time capability comes from the high computational speed of the state of the art Hierarchical Quadratic Programming (HQP) solver backing it.

A *task* basically is a control law that achieves a specific objective which can be a free space task or just an inequality constraint that narrows down the workspace of the robot. The task function formalism is very well discussed in [8]. In the context of our work, tasks generally include robot joint posture task, collision avoidance task, joint limits task and so on. The SoT framework handles the task priorities hierarchically in the real time to ensure there are no conflicts among tasks which is used to achieve dynamic obstacle avoidance without compromising on the main goal.

For example, let us consider a pick and place application in a collaborative environment. The primary goal for this application is to enable a robot to move to a (set of) desired pick and place locations repetitively. The pick and place locations can be defined as posture tasks in SoT. However, a higher priority task considering the collaborative nature of the environment is to avoid collisions with obstacles that could be humans, for instance. Typically such a task is modelled as an "Inequality" task and an eventual feasible solution (if one exists) is computed by the solver by exploiting the kinematic redundancy of the robot. In the jargon of motion planning and control, this behaviour is similar to a *local planner*. However, it is likely that a feasible solution is not found due to the solver converging to a local minima[1] In such a scenario, SoT can also be used to leverage the services of a global planner (see Section III-A) from the current robot state to the goal so that an entirely new path is obtained which is free from collisions and consequently allowing all the specified tasks to be achieved in the order of their priorities. In Section IV, we present the experimental results of using the SoT controller on a practical setup and in simulation. The SoT controller has also been configured to work with the ROS-control interface. In all these setups, the proximity information from the artificial robot skin is used as an input to the collision avoidance task. In

[1]This is caused by the use of task Jacobians. For further details, please see [7].

the following part, we briefly present the global path planner software framework that is used when the SoT controller hits a local minima.

A. *Reactive Path-Planning*

The reactive path planning software framework is based on the industry grade KineoWorks™[2] path planning library from Siemens in order to provide fast and reliable robot paths. This framework has also been seamlessly integrated into the ROS-ecosystem via a ROS package called `kws_ros_interface` which provides the planner implementations of KineoWorks as shared objects that are readily usable in ROS-based software via the `kws_ros_planner` ROS node.

Robot kinematic models are provided to KineoWorks in the Unified Robot Description Format (URDF) which is a ROS standard. Furthermore, KineoWorks also accepts the standard ROS representation of a `PointCloud`[3] for creating collision models of dynamic obstacles in the environment. In our work, point clouds are generated in two ways. In one scenario the point clouds are generated by a standard Kinect 3D camera that is observing the immediate environment of the robot. In the other scenario, the point clouds are generated from the proximity data obtained from the Artificial Skin. Finally, the collision detection for dynamic obstacle avoidance is performed using the Kineo™Collision Detector (KCD)[4]. KCD performs 3D collision detection and minimal distance analysis between triangular mesh surfaces in assembly environments. KCD has been designed specifically to minimize memory usage and take advantage of parallel processing. The complete software architecture used in our paper for the Dynamic Collision Avoidance functionality is shown in Fig. 2.

In the following sections we present the current results we have of using the different functionalities described.

IV. Preliminary Results

The preliminary results of evaluating the different functionalities are presented in two main categories. The first category includes results from individual evaluation of the different functional components on practical applications. The second category involves simulation and partial practical results of integrated evaluation of the functional components.

1) Individual Components: The Artificial Robot Skin has already been deployed on a Universal Robots UR5 robot (see Fig. 3). For the moment, the ARS is being tested to provide proximity information related to obstacles in the immediate surroundings of the robot.

The Stack of Tasks (SoT) controller has already been deployed and tested for achieving different postures on the setup in Fig. 3. The authors are actively working on extending the behavior to path following and eventually integrate in accordance with the reactive collision avoidance architecture shown in Fig. 2.

[2]See Kineoworks.
[3]See http://wiki.ros.org/pcl
[4]See KCD.

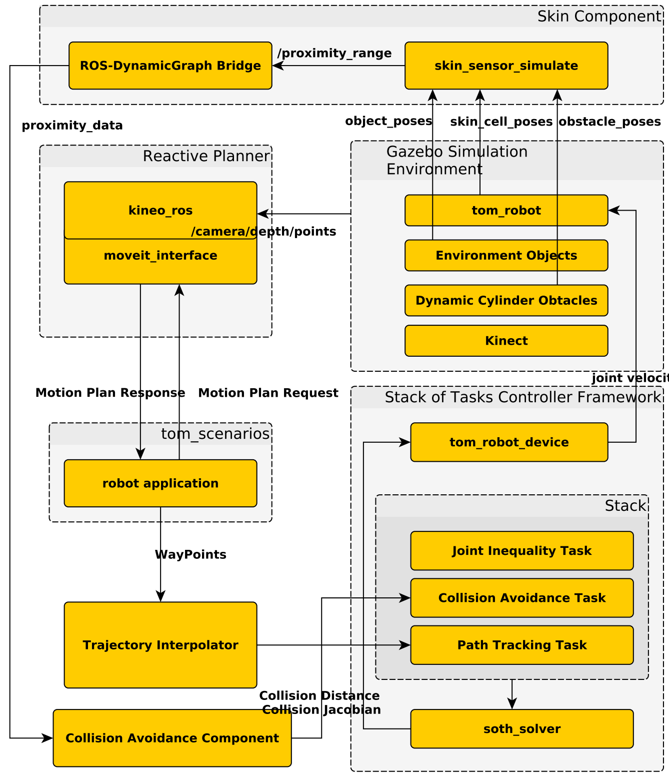

Fig. 2. Dynamic collision avoidance software architecture.

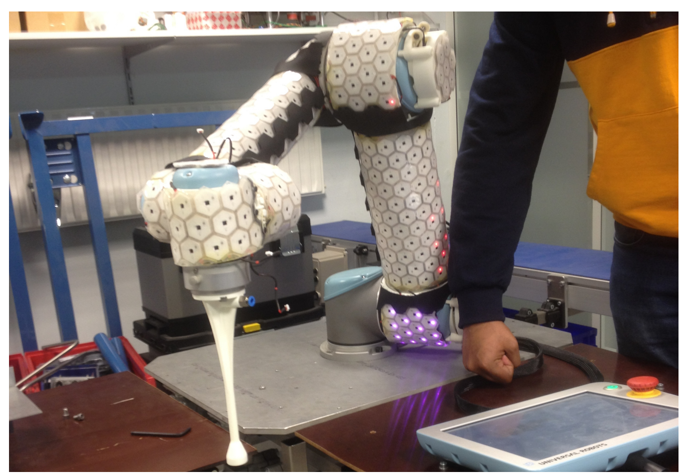

Fig. 3. UR5 setup with the Artificial Robot Skin by Mittendorfer *et al.* [2], showing some cells being activated (with red LEDs) by obstacles ($<= 6\,cm$).

A. *Integrated evaluation*

The integration of all the components has been evaluated on a simulation of the orange sorting robot by Dean *et al.* [9], [10] as shown in Fig. 4.

The evaluation is done in a ROS based gazebo environment with the skin sensors simulated using the flexible collision library to project the distance between objects to sensor range measurements. These measurements are mapped to signals compatible in dynamic graph framework using a bridge component to allow its use in the SoT controller. The collision

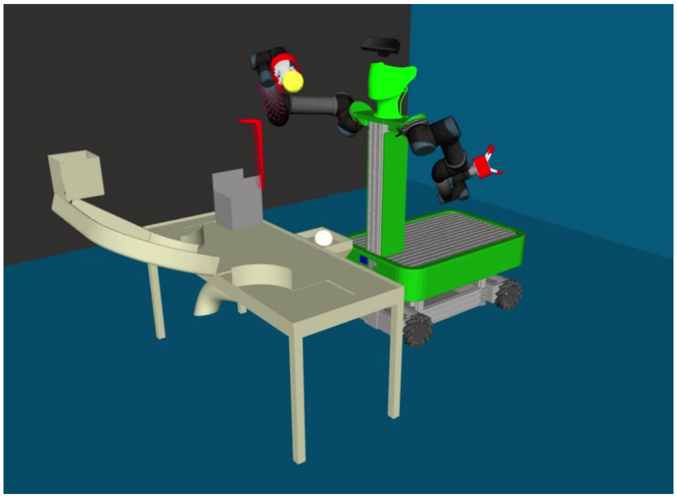

Fig. 4. Orange sorting scenario in simulation.The red point cloud is a simulated obstacle.

avoidance component computes the point distance and jacobian of each and every skin cell configured essential to feed as an inequality constraint to the solver which backs the SoT controller. The planning component having the capability to plan with point cloud data has a MoveIt! [11] python interface to query motion plan requests. The response is a set of way points which is then linearly interpolated to instantaneous joint position commands to a path tracking task in the SoT. The SoT controller also has a python interface which makes it easy to design application scenarios.

The combined use of a reactive motion planner and a hierarchical reactive SoT controller with skin data makes it a good candidate for applying dynamical obstacle avoidance in factory environments. Here is a video result of the same.

V. Concluding remarks and work in progress

This paper has presented the technologies that have been developed in the FiaD project to augment collaborative robot manipulators with dynamic obstacle avoidance. All these technologies: a proximity-sensing robot skin, a reactive path planning solution and a robot motion control strategy, have been validated in laboratory prototypes. Also, a preliminary prototype of an integrated solution based on these technologies has been tested in simulation. With the current promising results, we are currently working on a robotic system prototype (based on the setup in Fig. 3) that will be demonstrated in a real collaborative pick-and-place application (TRL 7 [12]) at the RoboBusiness Europe 2017 (RBE17) conference.

The integration and installation of advanced functionalities such as the dynamic obstacle avoidance solution presented poses three main challenges from the software point of view. The first is the integration of different components such as the skin driver, path planner and robot motion control. We address this challenge by adhering to the software development paradigm of the ROS-Industrial initiative. All the components discussed in this paper have been successfully integrated with ROS.

A second challenge is the quality assurance and robustness of the integrated robot software. This is crucial in production environments, and is specially important in collaborative applications, where safety needs to be guaranteed. For this purpose an Automated testing Framework (ATF) has been developed [13] as a part of the FiaD prohect, which allows for the systematic testing of robot software components, which includes unit testing, simulation-in-the-loop testing and eventually hardware-in-the-loop testing. The tests can be automated and integrated in a centralized continuous integration system. Preliminary test have already been conducted with the components of the robot software system of this work, and the integrated prototype applications will be tested with ATF.

Finally, the third challenge is the deployment of the software. One of the main barriers to transfer solutions based on robot frameworks such as ROS to industry, and specially SMEs, is how cumbersome it is to deploy. As a part of the FiaD project, a Robot deployment toolbox has been developed [14], based on ROS, which can also be integrated with ATF. The deployment tools will also be evaluated on the RBE17 prototype.

VI. Acknowledgements

The authors thank the Artificial Robot Skin research team at the Institute of Cognitive Systems in the Technical University of Munich, for providing the robot skin and for their continuous support in this work. The work leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 609206.

References

[1] M. Wisse, "Factory in a Day Project (FiaD)," http://www.factory-in-a-day.eu/, Oct. 2013.

[2] P. Mittendorfer, E. Yoshida, and G. Cheng, "Realizing whole-body tactile interactions with a self-organizing, multi-modal artificial skin on a humanoid robot." *Advanced Robotics*, vol. 29, no. 1, pp. 51–67, 2015.

[3] E. Dean-Leon, F. Bergner, K. Ramirez-Amaro, and G. Cheng, "From multi-modal tactile signals to a compliant control," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Nov 2016, pp. 892–898.

[4] Karinne Ramirez-Amaro, E. Dean-Leon, I. Dianov, F. Bergner, and Gordon Cheng, "General Recognition Models Capable of Integrating Multiple Sensors for Different Domains," in *IEEE-RAS International Conference on Humanoid Robots 2016*. Cancun, Mexico: IEEE, Nov 2016.

[5] llya Dianov, K. Ramirez-Amaro, P. Lanillos, E. Dean-Leon, F. Bergner, and G. Cheng, "Extracting general task structures to accelerate the learning of new tasks," in *IEEE-RAS International Conference on Humanoid Robots 2016*. Cancun, Mexico: IEEE, Nov 2016.

[6] P. Mittendorfer and G. Cheng, "3d surface reconstruction for robotic body parts with artificial skins," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 4505–4510.

[7] N. Mansard, O. Stasse, P. Evrard, and A. Kheddar, "A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks," in *International Conference on Advanced Robotics*, 2009.

[8] C.Samson, M. Borgne, and B.Espiau, *Robot Control: the Task Function Approach*. Clarendon Press, Oxford, United Kingdom, 1991.

[9] E. Dean-Leon, K. Ramirez-Amaro, F. Bergner, I. Dianov, P. Lanillos, and G. Cheng, "Robotic technologies for fast deployment of industrial robot systems," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2016, pp. 6900–6907.

[10] E. Dean-Leon, B. Pierce, P. Mittendorfer, F.Bergner, K. Ramirez-Amaro, W. Burger, and G. Cheng, “TOMM: Tactile Omnidirectional Mobile Manipulator,” in *2017 IEEE International Conference on Robotics and Automation (ICRA), Accepted*, May 2017.
[11] I. A. Sucan and S. Chitta, “Moveit! motion planning framework,” http://moveit.ros.org.
[12] E. Commission, “Technology readiness levels (TRL),” European Commission, Tech. Rep. HORIZON 2020 – WORK PROGRAMME 2014-2015 General Annexes, Extract from Part 19 - Commission Decision C(2014)4995.
[13] F. Weisshardt and F. Koehler, “Automatic testing framework for benchmarking applications,” in *Proceedings of ISR 2016: 47st International Symposium on Robotics*, June 2016, pp. 1–6.
[14] M. Lüdtke, “Integration platform and deployment environment,” https://github.com/ipa-mdl/ipde_utils, 2017.