DARKO
Dynamic Agile Production Robots That Learn and Optimise Knowledge and Operations

H2020-ICT-2020-2 Grant agreement no: 101017274

DELIVERABLE 4.3
Preliminary planning and control algorithms for safe manipulation

Dissemination Level: PUBLIC

Due date: month 36 (December 2023)

Deliverable type: Software

Lead beneficiary: TUM

1 Introduction

Although high automatization levels in industry are desirable, humans represent an integral and important part of every industrial process. Thus, some of the newer generations of robots are designed to share a workspace and work together with humans. Physical human-robot interaction (pHRI) has placed a spotlight on safety-certifiable robotic technologies, where the safe execution of collaborative tasks has become a major objective. Further, close physical collaboration necessitates sharing work environment with tighter interaction spaces between humans and robots to enhance manufacturing and logistics processes' flexibility and productivity. However, since such close proximity may bring the possibility of physical contact that can even be part of the interactive process, undesired and potentially dangerous situations including harmful collisions may occur. To reduce the related human injury risks, the robot task execution may have to be reduced to match the safety demands of current robot standards (e.g. ISO10218/TS15066 for industrial robots).

First development, a unified industry-validated approach for ensuring human safety during pHRI of stationary robot arms relies on an injury data-driven safety framework termed the Safe Motion Unit (SMU) [1]. Given relative human-robot distance and velocity information and based on a rich injury database for various impact curvatures [2], the SMU shapes the robot motion such that the impact energy of possible dynamic collisions cannot exceed the safety limits set by the standard or other safety mappings chosen to be respected.

In this part of the deliverable, we evaluate our SMU safety framework for limiting the arm velocity (theoretically similar to vSMU). We show how the proposed framework calculates the generalized reflected dynamics and implements the mobile manipulator SMU in real-time. The results have been partially validated on the DARKO robot platform/.

2 Theoretical Framework and Implementation Details

One of the key steps in the SMU computation is calculating the inertial properties of the arm. It's not always obvious which point could be the first contact point during the collision. Also, due to computational complexity, it is not feasible to inspect the whole robot mesh in each runtime iteration. Thus, we have decided to monitor a specially predefined set of dangerous points, as we refer to them further - points of interest (POIs)

Besides the mentioned key data inputs, such as Massmatrix and POIs, SMU implementation also aggregates several other data inputs. In particular, it needs robot frame transformations, dynamic and kinematic data, an injury database for characterized POIs, and human position data. A schematic representation is available in Figure 1. A more detailed procedural description of SMU can be found in Algorithm 1.

3 Integration with Motion Planning and Control Framework for the Manipulator

In Work Package 4 (WP4), a key task involves integrating the Safety Management Unit (SMU) with the motion planning and control layer within the DARKO platform. The SMU is responsible for providing velocity constraints to guarantee the safety of humans in proximity to the robot. Concurrently, the motion planning and control layer is tasked with calculating the instructions for the manipulator, enabling it to carry out the intended tasks. This integration is vital in the DARKO scenario, as it ensures a harmonious coexistence of robots and human workers within the same workspace. In this section, we start first

Algorithm 1 The safe motion unit (SMU) algorithm for ensuring human-safe task trajectory control for mobile manipulators

1: Assign a set of POIs encoding surface curvatures at key points on both the base vehicle and manipulator arm structures. Alongside position, directional normalized (unit) Cartesian vector u is defined perpendicular to the object surface in the Cartesian world frame (representing most dangerous direction).
2: Obtain the velocity of each POI from the requested mobile manipulator's end-effector velocity.
3: Calculate the reflected mass of the manipulator system at each POI, considering the impact direction u, oriented towards the human:

$$m_{POI} = 1/(c^{\intercal}\Lambda_{v,POI}^{-1}u) \tag{1}$$

4: Find the maximum permissible velocity $vmax$ that is biomechanically safe for the corresponding reflected mass of POI from the suitable safety curves of the pain/injury
5: Scale v_{max} with the projection of arm POI velocity in the impact u direction to get the new POI velocity $v_{POIsafe}$
6: Calculate the new, safe velocity limit of the platform v_{safe}. It is supplied to the control as the new desired task velocity.

giving a brief summary of the planning and control framework developed for the robotic manipulator mounted on the mobile platform. After that, we move to provide an overview of this first preliminary integration between these two layers. In the end, we discuss the next steps regarding this part of the project.

3.1 Planning and Control Framework

The planning and control framework designed for the robotic arm in DARKO offers an extensive array of solutions for issuing commands to the manipulator. This choice was made to take into account the various tasks that we have to accomplish inside the presented scenario (such as simple repositioning of the arm, object picking, throwing, etc...). For this first preliminary integration, we have decided to start integrating the SMU with the planning and control at the Cartesian level, which will be the most used modality to perform pick and place actions. For the sake of brevity, in this section, we will describe only the part related to the integration, while we refer the interested reader to Deliverable D4.1 "Preliminary planning and control software developments for efficient manipulation" for a deeper description of all the planning and control framework.

3.1.1 Control Algorithm

The control law chosen to follow a Cartesian reference is a classical Cartesian impedance controller. This is done to ensure a safer interaction with the environment than the classical position controllers which requires some policies to avoid the generation of high interaction forces. Given a desired Cartesian trajectory x_{des} and its time derivative $\dot{x}_{des}$, we can define the error between the actual and desired Cartesian position and the velocity of the manipulator as $e = x_{des} - x$ and $\dot{e} = \dot{x}_{des} - \dot{x}$. From these values, we can compute the joint torque necessary to follow the desired trajectory:

$$\tau = J^T(q)(K_p e + K_v \dot{e}) + G(q) + C(q,\dot{q})\dot{q} \tag{2}$$

where $J(q)$ is the Jacobian, $G(q)$ is the gravitational term, $C(q)\dot{q}$ is the Coriolis term, and K_p and K_v are the matrices that define the impedance behaviour of the controller.

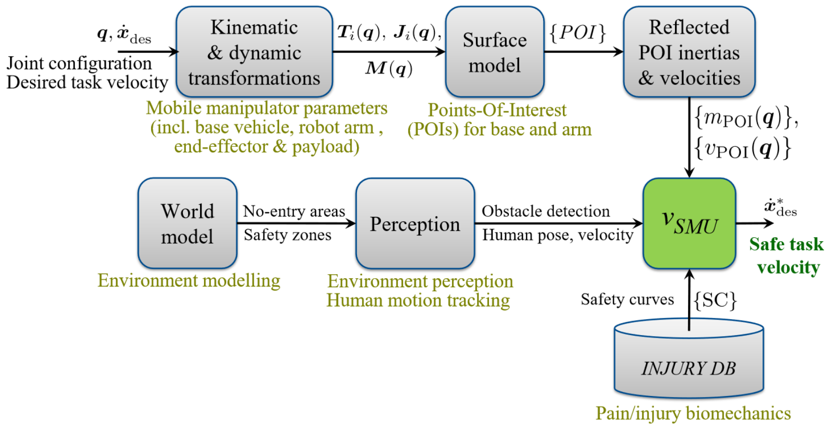

Figure 1: Schematic representation of SMU module integration and key working principles.

3.1.2 Planning Algorithm

Regarding the computation of the trajectory, the proposed technique is an algorithm able to generate motion similar to the ones produced by humans. The embedding of this behaviour in the robot motion is a key factor in increasing the predictability and the acceptability during human-robot interaction [3].

To achieve this goal, the proposed technique is to exploit the functional Principal Component Analysis (fPCA) to extract a representation of the human motion features using a reduced number of functional elements. Different works in the literature prove the effectiveness of this type of analysis in extracting a reduced basis of functions related to the human arm movements both in the joint [4] and the Cartesian domain [5], and also in using the output of this analysis to design a planning algorithm to generate Human-Like trajectories [6].

The main concept in exploiting fPCA to design a planning algorithm (referring the interested reader to [7] for further details) is that any movement can be represented as:

$$x(t) \simeq \bar{x} + S_0(t) + \sum_{i=1}^{s_{max}} \alpha_i \circ S_i(t) \tag{3}$$

where $\bar{x}$ is the average pose of the hand, $S_0(t)$ is the average trajectory across all the trajectories in the dataset, α_i is a vector of weights, s_{max} is the number of basis elements, $S_i(t)$ is the i^{th} basis element and the symbol $\circ$ represents the Hadamard product (i.e. the element-wise product).

To find the coefficients $\bar{x}$ and α_i given a set of constraints to be satisfied we can define an equation system to obtain the desired trajectory to be planned. For example, setting the initial and final position, velocity and acceleration, the following equation system is defined:

$$\begin{bmatrix} 1 & S_1(t_0) & \dots & S_5(t_0) \\ 1 & S_1(t_f) & \dots & S_5(t_f) \\ 0 & \dot{S}_1(t_0) & \dots & \dot{S}_5(t_0) \\ 0 & \dot{S}_1(t_f) & \dots & \dot{S}_5(t_f) \\ 0 & \ddot{S}_1(t_0) & \dots & \ddot{S}_5(t_0) \\ 0 & \ddot{S}_1(t_f) & \dots & \ddot{S}_5(t_f) \end{bmatrix} \begin{bmatrix} \bar{x} \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{bmatrix} = \begin{bmatrix} x(t_0) - S_0(t_0) \\ x(t_f) - S_0(t_f) \\ \dot{x}(t_0) - \dot{S}_0(t_0) \\ \dot{x}(t_f) - \dot{S}_0(t_f) \\ \ddot{x}(t_0) - \ddot{S}_0(t_0) \\ \ddot{x}(t_f) - \ddot{S}_0(t_f) \end{bmatrix} \tag{4}$$

by solving the system we can obtain the desired planned trajectory

$$x(t) = \bar{x} + S_0(t) + \sum_{i=1}^{5} \alpha_i S_i(t) \tag{5}$$

where, in our case, $x(t)$ is the end effector pose of the manipulator defined as position and orientation with respect to the base frame. The strength of this type of framework is that can compute in a closed form a Human-Like trajectory with arbitrary initial and final conditions.

The code regarding the Human-Like planning algorithm can be found in the DARKO project GitLab repository inside the project called "*darko_arm_planning*".

```
darko_arm_planning:
type: git;
url: https://gitsvn-nt.oru.se/darko/software/darko_arm_planning.git
version: master-unipi
```

3.2 Integration with SMU

As previously mentioned, the SMU determines the maximum permissible velocity to avoid severe injuries in the event of a collision between a robot and a human. Therefore, the arm motion planning algorithm must be crafted to modify its planned movements in line with the speed constraints set by the SMU. There are three methods to achieve this:

1. SMU can be a middleman between the Planner and Controller. In this case, the Planner can not foresee and avoid regions with lower speeds (it can not adjust for the speed limitation online). SMU rather just clamps the speed to ensure safe execution.

2. Similar to the previous case, but rather than limiting the speed itself, it constantly provides a maximum safe speed, taking into account the current robot configuration and human data.

3. SMU can work asynchronously (on request). Meaning that one can feed various arm configurations to be checked for safety. This can be useful for sampling-based planning approaches, to try to actively avoid low-speed regions.

For this first preliminary integration, as a proof of concept, the idea is to work towards approach 2. To simplify, we consider only the manipulator's end effector as a dangerous point (POI) and try to limit the speed based on the robot's reflected mass at that point. An example setup from the simulation can be seen in Figure 2. How SMU reacts to the Human proximity can be found in Figure 3. Complete software provided in [8]. For this integration, the idea was to provide an easy-to-use interface for the Cartesian planner developed by UNIPI.

From an architecture point of view, we decided to use a publisher/subscriber structure to pass the velocity limit from the SMU to the motion planning node (as indicated in approach 2). In this way, the SMU computes the safe velocity using the information regarding the actual state of the manipulator and its relative position with respect to the worker and makes it available to the other nodes publishing it in a topic.

Regarding the motion planning side of this infrastructure, the idea is to compute the trajectory and, while it sends the reference to the controller, it checks if the maximum planned velocity meets the constraint given by SMU and, if not, slows down the trajectory.

From a technical point of view, the node has a cycle inside that publishes each frame of the trajectory with the right sampling time to respect the desired behaviour. To integrate the information provided by SMU, at every cycle, the node listens for the maximum safe

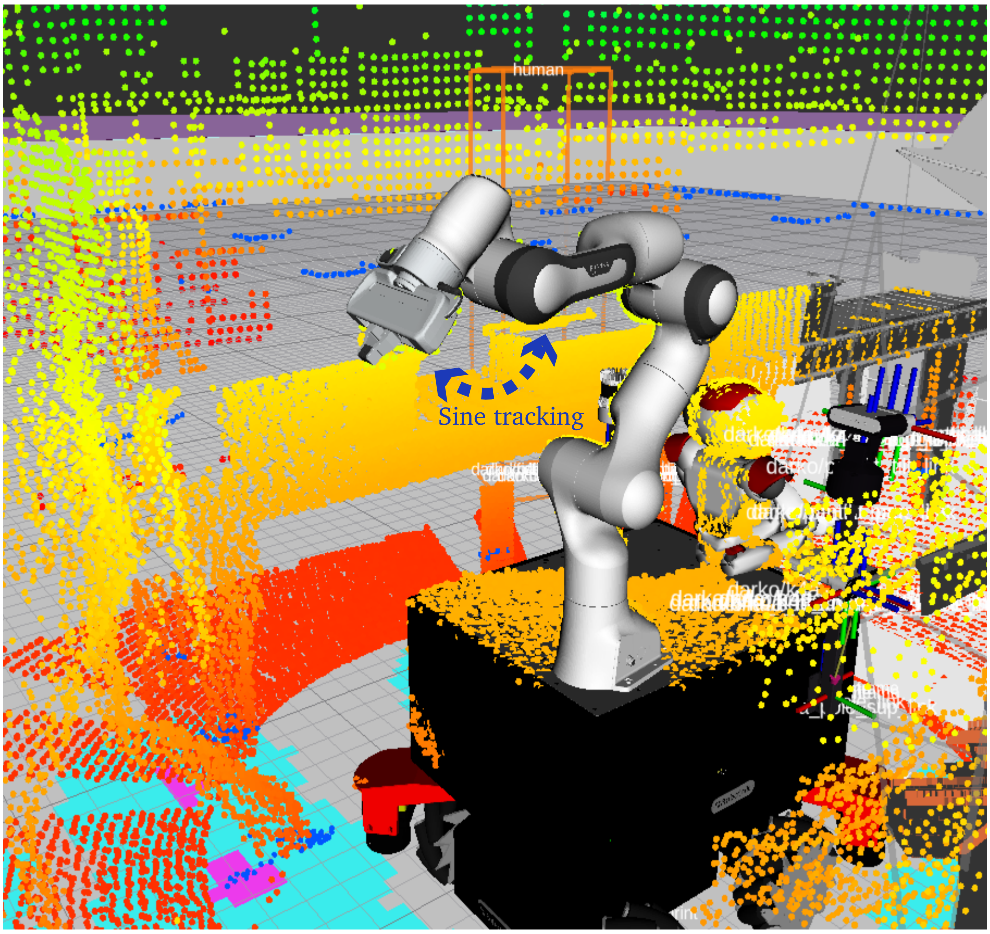

Figure 2: Simulation view of Robot and with sensor data for conducted SMU experiment. The Panda End-effector has been taken as a Point of Interest (POI). *Note: Human motion simulated by approaching dummy target.*

velocity, compares this value with the peak velocity of the remaining movement, and adjusts the sampling time of the trajectory to keep the peak velocity below the limit.

In this deliverable, we rather present the architectural-level descriptions alongside all the required building blocks. The plan is to integrate all the components in the next Integration, scheduled for April 2024 in Stuttgart.

3.3 Future steps

In this report, we present an architecture-level description and all individually tested building blocks for the integration of the SMU with the motion planning framework. The final result should be the ability of the DARKO platform to slow down the motion of the manipulator to ensure safety. Further improvements and implementation will be done during the next scheduled Stuttgart Integrations of the project.

Alongside the integration of currently developed components, we plan to work on approaches that should be able to foresee the speed limitations and avoid low-speed regions overall. This should keep the efficiency of the robot platform even in case of close proximity to humans.

Another part to be addressed is the integration of the SMU with the throwing part. While for picking tasks we do not have any constraint regarding the movement velocity to

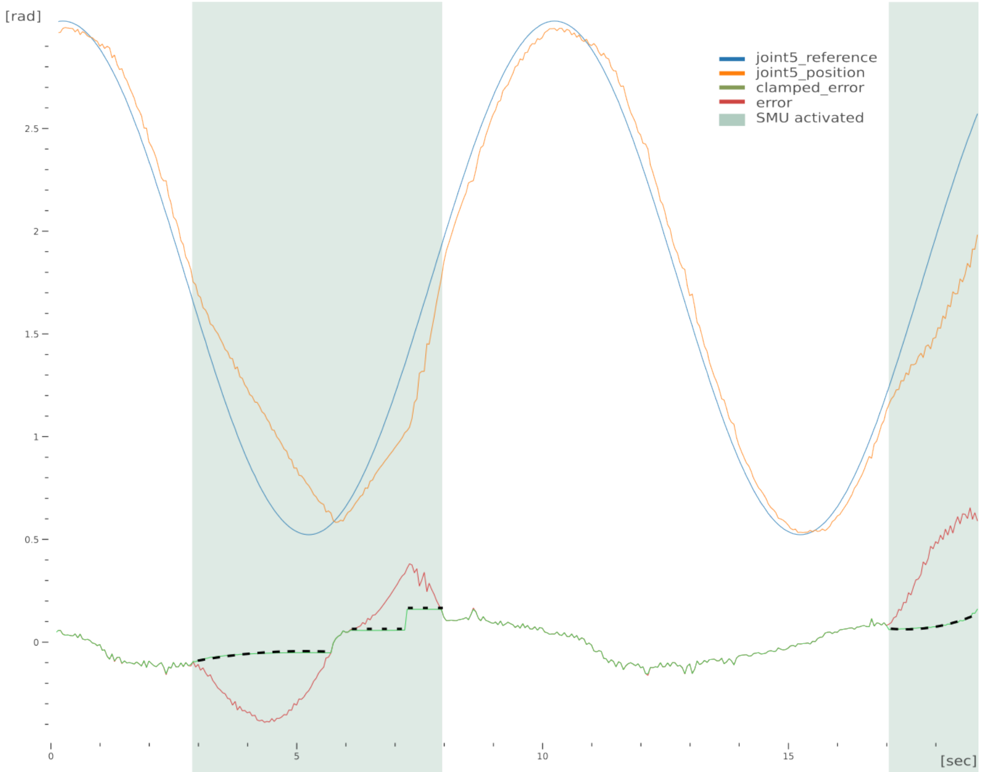

Figure 3: Tracking of sinusoidal trajectory for Panda Robot. SMU activation limits the speed of motion when required to ensure human safety.

accomplish it, for throwing tasks the velocity of the trajectory is a key factor in performing the task. In this sense, a suitable way to integrate these two parts has to be found.

References

[1] Mazin Hamad, Alexander Kurdas, Nico Mansfeld, Saeed Abdolshah, and Sami Haddadin. "Modularize-and-Conquer: A Generalized Impact Dynamics and Safe Precollision Control Framework for Floating-Base Tree-Like Robots". In: *IEEE Transactions on Robotics* 39.4 (2023), pp. 3200–3221.

[2] Sami Haddadin, Simon Haddadin, Augusto Khoury, Tim Rokahr, Sven Parusel, Rainer Burgkart, Antonio Bicchi, and Alin Albu-Schäffer. "A truly safely moving robot has to know what injury it may cause". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 5406–5413.

[3] Andrea Maria Zanchettin, Luca Bascetta, and Paolo Rocco. "Acceptability of robotic manipulators in shared working environments through human-like redundancy resolution". In: *Applied ergonomics* 44.6 (2013), pp. 982–989.

[4] Giuseppe Averta, Cosimo Della Santina, Edoardo Battaglia, Federica Felici, Matteo Bianchi, and Antonio Bicchi. "Unvealing the principal modes of human upper limb movements through functional analysis". In: *Frontiers in Robotics and AI* 4 (2017), p. 37.

[5] Marco Baracca, Paolo Bonifati, Ylenia Nisticò, Vincenzo Catrambone, Gaetano Valenza, A Bicchi, Giuseppe Averta, and Matteo Bianchi. "Functional Analysis of Upper-Limb Movements in the Cartesian Domain". In: *Converging Clinical and Engineering Research on Neurorehabilitation IV: Proceedings of the 5th International Conference on Neurorehabilitation (ICNR2020), October 13–16, 2020*. Springer. 2022, pp. 339–343.

[6] Giuseppe Averta, Danilo Caporale, Cosimo Della Santina, Antonio Bicchi, and Matteo Bianchi. "A technical framework for human-like motion generation with autonomous anthropomorphic redundant manipulators". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 3853–3859.

[7] Marco Baracca, Giuseppe Averta, and Matteo Bianchi. "A general approach for generating artificial human-like motions from functional components of human upper limb movements". In: *IEEE Transaction on Human-Machine Systems* (2023). (Under Review).

[8] Vasilije Rakcevic, Alessandro De Toni, Mazin Hamad, Saeed Abdolshah, Abdalla Swikir, and Sami Haddadin. *SMU(gSMU) software package*. https://gitsvn-nt.oru.se/darko/software/tum-g-smu. 2023.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017274