DARKO

Dynamic Agile Production Robots That Learn and Optimise Knowledge and Operations

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

DELIVERABLE 4.1

Preliminary planning and control software developments for efficient manipulation

Dissemination Level: PUBLIC

Due date: month 26 (February 2023)

Deliverable type: Software

Lead beneficiary: UNIPI

1 Introduction

This deliverable describes the initial software structure developed in tasks T4.1 and T4.3 to perform pick and place actions with the Franka Emika Panda arm installed on the DARKO platform. The contribution to this purpose can be divided into 3 main parts: 1) a MoveIt-based package to exploit the existing open-source code for motion planning; 2) a set of torque controllers for the robot arm to provide a lower level interface; and 3) a software package to plan Human-Like motion. All the packages described below were designed for ROS Melodic Morenia (the version actually in use on the DARKO platform), but they can be ported easily on the latest ROS 1 distribution (Noetic Ninjemys).

In the last part of the deliverable, in anticipation of the development of an elastic arm for the platform, we introduced some preliminary work regarding the planning and control of compliant manipulators.

2 MoveIt Package for Motion Planning

MoveIt is a software package commonly used for motion planning for robotic manipulators. It has an architecture which provides a set of tools necessary to solve planning problems (such as inverse kinematics, collision detection and planning algorithm) and permits the user to command his robot both using C++/Python packages or through a graphical interface.

In this section, we will focus on the explanation of the software architecture developed inside the project DARKO, while for more general information regarding the MoveIt package we refer the reader to [8, 24].

2.1 Package Description

All the code regarding motion planning and control with MoveIt described in this section can be found in the GitLab repo reported below:

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

The main ROS node responsible for motion planning and control is called "*hl_sequencer*". This node is created by instantiating an object belonging to the "'*human_like*" class, declared as a C++ library within the "*darko_arm_planning*" ROS package. The class's constructor creates and initializes publishers and subscribers to receive and publish messages to compute the motion planning and control phase for the Franka Emika robot. The basic idea besides the node is to listen to the five types of tasks:

- **Pick**: Plan a Cartesian trajectory using MoveIt and, at the end of the motion, close the gripper.
- **Place**: Plan a Cartesian trajectory using MoveIt and, at the end of the motion, open the gripper.
- **Throw**: Trigger the throwing part of the algorithm (described in D4.5).
- **Move**: Plan a Cartesian trajectory using the Human-like motion planning strategy described in Section 4.

- **Tool**: Trigger the motion sequence to perform pick and shoot with the pneumatic tool (described in D1.2 and D4.5)

Combining these tasks it is possible to perform more complex tasks such as picking and placing/throwing of a desired object in the desired position. This specific set of basic tasks is the one used for the demonstration performed during Milestone 2. However, tasks can be easily modified and/or added to integrate the new approaches developed until the end of the project.

A detailed description of the ROS topics, publishers and subscribers is reported here:

- "**/manipulation/action_type**": it is the subscriber's topic used by the ROS node "*hl_sequencer*" to get the type of tasks listed above. The topic's type is a *std_msgs/String* message.
- "**/manipulation/MovingPose**": it is the subscriber's topic used by the ROS node "*hl_sequencer*" to get the desired cartesian pose used by the human-like trajectory planning. The topic's type is a *geometry_msgs/PoseStamped* message.
- "**/robot/arm/cartesian_impedance_controller_softbots/equilibrium_pose**": it is the publisher's topic used by the ROS node "*hl_sequencer*" to publish the planned trajectory towards the goal pose received on the "**/manipulation/MovingPose**" topic. The topic's type is a *geometry_msgs/PoseStamped* message.
- "**/manipulation/desired_ee_pose**": it is the subscriber's topic used by the ROS node "*hl_sequencer*" to get the desired grasp pose set as pose target in the MoveIt trajectory planning. The topic's type is *geometry_msgs/PoseStamped* message.
- "**/robot/arm/position_joint_trajectory_controller/follow_joint_trajectory**": it the ROS action's topic used by the ROS node "*hl_sequencer*" to publish the MoveIt planned trajectory towards the goal pose received on the "**/manipulation/desired_ee_pose**" topic. The topic's type is a *trajectory_msgs/JointTrajectory*.

The ROS node "*hl_sequencer*" can be run using the launch file called "launchHumanLike.launch" contained in the folder "launch".

It is worth mentioning that the "*hl_sequencer*" ROS node can handle all the grippers used in the DARKO EU project: the DH3 gripper, the SoftHand 1 and the SoftHand 2. To do this, we can select what type of gripper we are using by simply modifying the input argument from the launch file ("launchHumanLike.launch").

3 Controller Packages for Franka Emika Panda

This section describes the controllers deployed on the DARKO platform to control Franka Emika Panda's arm.

3.1 Theorethical Framework

One of the significant advantages of Franka Emika Panda is the capability to accept torque commands, enabling the possibility to implement arbitrary controllers to follow different behaviours. Exploiting this possibility, we developed a set of impedance controllers to command the manipulator. The impedance behaviour is fundamental for a robot that has to interact physically with the environment in a safe and proficient manner. This class of controllers permits to avoid large impact forces while operating in an unstructured environment. To meet the needs of all partners of the consortium we implemented two

different controllers: 1) a Cartesian impedance controller and 2) a classical Computed Torque controller. This choice was made to be able to accept reference trajectories both in the joint space and in the task space.

3.1.1 Cartesian Impedance Controller

A Cartesian impedance controller is a type of robot controller that regulates the interaction between a robot and its environment by controlling the impedance of the robot's end-effector in Cartesian space. 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:

$$\tau = J^T(q)(K_p e + K_v \dot{e}) + G(q) + C(q,\dot{q})\dot{q} \tag{1}$$

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.

Usually, when we use manipulators with more than 6 degrees of freedom, an additional control law projected in the null of the Cartesian space is introduced to control the redundancy of the robotic arm. In our case we decided, as a secondary task, to maintain a fixed desired joint posture. To do this we defined the secondary torque command τ_{null} as:

$$\tau_{null} = P(q)(K_{pnull}(q_{des} - q) - K_{vnull}\dot{q}) \tag{2}$$

where $P(q) = (I - J^T(J^+)^T)$ is the Cartesian null space projector and J^+ is the pseudoinverse of J.

3.1.2 Computed Torque Controller

Computed Torque Control is a type of feedback control technique used in robotics and mechatronics applications. The aim of this control method is to control the motion of a robot or a mechatronic system with high precision, while also reducing the effect of external disturbances on the system. It comprises two main terms: the feedforward part, responsible for generating a desired torque based on the current system state and the desired trajectory, and the feedback part, responsible for making the motion stable. The torque to perform this type of control can be computed with the following equation:

$$\tau = M(q)(\ddot{q}_{des} + K_v \dot{e} + K_p e) + C(\dot{q},q)\dot{q} + G(q) \tag{3}$$

where $\ddot{q}_{des}$, $\dot{q}_{des}$ and q_{des} are the desired joints acceleration, velocity and trajectory, $e = q_{des} - q$ and $\dot{e} = \dot{q}_{des} - \dot{q}$.

3.2 Software Package

All the code regarding the controller used in the Franka Emika Panda arm is contained in the DARKO project GitLab repository called "*darko_arm_control*".

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

Both the Cartesian Impedance and Computed Torque controller can be loaded and spawned through the controller manager node recalled in the launch file called "*manipulation_complete.launch*" inside the "*darko_launch_system*" folder.

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

3.2.1 Cartesian Impedance Controller

To implement this controller on the DARKO platform we used the standard code structure provided by Franka to develop a ROS node to send torque commands to the robot joints. This node is written in C++ and it provides different topics to communicate with other nodes:

- "**/robot/arm/cartesian_impedance_controller_softbots/equilibrium_pose_pose**": it is a subscriber used by the node to get the desired end-effector pose x_{des} to be used to compute the torque in (1). This subscriber accepts a *geometry_msgs/Pose* message.
- "**/robot/arm/cartesian_impedance_controller_softbots/desired_stiffness**": it is a subscriber used to set online the parameters K_p and K_v in (1). This subscriber accepts a *geometry_msgs/Vector3* which represents the stiffness values for translation along axes x, y and z. The rotational stiffness is set as translational stiffness divided by 10. The diagonal values of the damping matrix K_v instead are set to ensure that the controller behaves as a critically damped system.
- "**/robot/arm/franka_state_controller/franka_states**": it is a publisher used by the controller to retrieve the actual end-effector pose as a *geometry_msgs/PoseStamped*.

3.3 Computed Torque Controller

To implement this controller on the DARKO platform we used the standard code structure provided by Franka to develop a ROS node to send torque commands to the robot joints. This node is written in C++ and it provides the following topics:

- "**/robot/arm/computed_torque_controller/command**": it is a subscriber used by the node to get the desired joint references in terms of position (q_{des}), velocity ($\dot{q}_{des}$) and acceleration ($\ddot{q}_{des}$). This subscriber accepts a *sensor_msgs/JointState* message.
- "**/robot/arm/computed_torque_controller/tracking_error**": it is a publisher used by the controller to make available to the other nodes the actual tracking error performed by the controller. It uses a ROS message of type *sensor_msgs/JointState*.

4 Human-Like Motion Planning Algorithm

4.1 State of the Art

Generating Human-Like (HL) movement with robotic manipulators is crucial because it can enhance safety, interaction, versatility, efficiency, and user experience. When robots move like humans, they are more predictable and safe to operate around humans. This reduces the risk of injury and allows for better communication and understanding between humans and robots.

Human motion has different key features which make it peculiar with respect to movements generated by classical planning algorithms. For example, in [10] the authors

analyzed human hand motion in 3D space during reaching tasks, with and without obstacles, finding that the movement paths are predominantly planar. In [14], the authors found that there is a relation between the curvature of a path and the velocity at which humans have to follow it. In [9] the authors observed that humans minimize jerk during movement execution.

Many researchers proposed strategies for generating Human-Like movements in different applications [11, 17]. One of the most popular solutions to achieve HL is to formalize an optimization problem whose functional cost is devised from neuroscientific observations. For example, in [18] the authors developed an optimization-based framework to generate minimum-jerk trajectories building on [9], while [12] exploited the minimization of the torque-change following the model proposed in [25]. However, such optimization approaches usually build upon hypotheses on motion generation, which can reduce the variability of the planned movement.

Another possible approach to generate Human-Like movements exploits learning and data-driven methods. This is a solution used very often in the field of animations and computer graphics where, after an extensive campaign of data recording via motion capture systems, the recorded datasets are used to train neural networks to animate the avatars [13, 16, 27]. For example, in [15] the authors used a recurrent neural network to act as a near-optimal feedback controller generating stable and realistic behaviour. Another example is [21], where the authors used Generative Adversarial Neural Networks for synthesizing gestures directly from speech. Some of these approaches are also applied in robotic applications for the generation of human-like movements with humanoid robots [23]. However, the common limitation of learning-based methods is related to the need for reliable datasets, whose dimensionality can be significant.

A possible solution to design an efficient Human-Like motion planning framework, overcoming the aforementioned issues, is to directly embed the main human motion characteristics in the algorithmic structure. Many works in literature addressed the analysis of human motion to extract movement patterns and obtain a reduced yet meaningful characterization of human kinematics [22]. Regarding the upper limb motion, in [3] we exploited functional Principal Component Analysis (fPCA) to identify a geometrical basis of functions whose elements can be combined to reconstruct the overall trajectory. These basis elements were also used to develop a planning algorithm in the joint space domain, which intrinsically embeds HL in the generated motion [4]. This planner, however, is strictly related to the kinematic description used to acquire human upper limb data, and a mapping strategy is needed to generalise the planning outcomes to manipulators with different kinematic structures. A solution to the latter problem was proposed in [2], where Cartesian impedance control was used to implement fPCA-based planning with manipulators with redundant anthropomorphic kinematic architectures - although dissimilar with respect to the human model used for functional mode extraction. However, these approaches in the joint space are associated with non-negligible computational time: for example, while obstacle-free planning can be solved in a closed form, devising a trajectory in the presence of obstacles requires solving an optimization problem, which can be required up to several seconds.

4.2 Theorethical Framework

To address both the problem of mapping and the reduction of the computational time of the aforementioned method, we propose a novel planning algorithm able to compute Human-Like trajectories of robotic manipulators directly in the Cartesian domain. To this aim, we built upon the results we presented in [6], where we showed that a geometrical representation of the human end-effector trajectory in terms of functional elements still holds in the Cartesian space, confirming the outcomes reported in [3] at the joint level. This

approach permits to obtain in a negligible time (less than 100 ms) a reference trajectory with an intrinsic Human-Like behaviour, which can be applied to any kinematic chain used for describing an artificial manipulator.

4.2.1 Functional Principal Component Analysis

Functional Principal Component Analysis (fPCA) is a statistical method to identify a geometrical basis of functions whose elements can be combined to reconstruct time series. In this section, we will provide a brief introduction to the underpinning theory and its application - without loss of generality - to the description of hand trajectories (i.e. the trajectories of the end-effector of the upper limb kinematic chain), while referring the interested reader to [20] for more details. Given a dataset of hand motions, the generic motion $x(t)$ can be represented as a weighted sum of a set of basis functions $S_i(t)$, or functional Principal Components (fPCs) extracted from the dataset, that is:

$$x(t) \simeq \bar{x} + S_0(t) + \sum_{i=1}^{s_{max}} \alpha_i \circ S_i(t) \tag{4}$$

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).

The first element of the functional basis or first fPC can be computed from the R motions of the dataset as:

$$\max_{S_1} \sum_{j=1}^{R} \left(\int S_1(t) x_j(t) dt \right)^2 \tag{5}$$

subject to

$$||S_1(t)||_2^2 = 1 \tag{6}$$

The other components $S_i(t)$ can be computed as:

$$\max_{S_i} \sum_{j=1}^{R} \left(\int S_i(t) x_j(t) dt \right)^2 \tag{7}$$

subject to

$$||S_i(t)||_2^2 = 1 \tag{8}$$

$$\int_0^{t_{end}} S_i(t) S_k(t) dt = 0, \forall k \in \{1, ..., i-1\} \tag{9}$$

In this manner, we can identify a basis of functional elements, ordered in terms of the explained variance that each element accounts for. The details on the results obtained applying fPCA to a dataset of human upper limb motions [1] can be found in [6].

4.2.2 Planning Algorithm

The fPCs extracted from a dataset that can be considered representative of the most common upper limb movements can be used to plan trajectories that intrinsically embed HL. In the following section, we provide a formalization of the planning problem starting with the no-obstacle case, and then we extend the approach to deal with the presence of an arbitrary number of fixed obstacles. Of note, fPCA is performed for each Degree of Freedom (DoF) of the kinematic chain separately. In the following, we report the equations

for a single DoF of the end effector, while the extension to multiple DoFs (e.g. the six DoFs describing the pose of the end effector) is trivial.

The reconstruction of the single DoF trajectory can be attained as:

$$x(t) \simeq \bar{x} + S_0(t) + \sum_{i=1}^{s_{max}} \alpha_i S_i(t) \tag{10}$$

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:

$$\begin{bmatrix} 1 & S_1(t_0) & \dots & S_5(t_0) \\ 1 & S_1(t_f) & \dots & S_5(t_f) \\ 0 & \dot{S}_1(t_0) & \dots & \dot{S}_5(t_0) \\ 0 & \dot{S}_1(t_f) & \dots & \dot{S}_5(t_f) \\ 0 & \ddot{S}_1(t_0) & \dots & \ddot{S}_5(t_0) \\ 0 & \ddot{S}_1(t_f) & \dots & \ddot{S}_5(t_f) \end{bmatrix} \begin{bmatrix} \bar{x} \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{bmatrix} = \begin{bmatrix} x(t_0) - S_0(t_0) \\ x(t_f) - S_0(t_f) \\ \dot{x}(t_0) - \dot{S}_0(t_0) \\ \dot{x}(t_f) - \dot{S}_0(t_f) \\ \ddot{x}(t_0) - \ddot{S}_0(t_0) \\ \ddot{x}(t_f) - \ddot{S}_0(t_f) \end{bmatrix} \tag{11}$$

by solving the system we can obtain the desired planned trajectory

$$x(t) = \bar{x} + S_0(t) + \sum_{i=1}^{5} \alpha_i S_i(t) \tag{12}$$

The strength of this type of framework is that is able to compute in a closed form a Human-Like trajectory with arbitrary initial and final conditions.

4.3 Future Developments

The closed-form structure of the framework permits to achieve an extremely low computational time in computing the single trajectory. This can be exploited in different ways inside the DARKO scenario.

4.3.1 Collision avoidance

One of the main aspects is collision avoidance. The idea is to find a set of viapoints to guide the trajectory around the obstacles and plan the trajectory in pieces, guaranteeing continuity in the junction points. Extensive testing on the planning strategy was performed, and the interested reader can find them in [5]. The next step in this direction will be to build a framework able to react in real time to unforeseen obstacles.

4.3.2 Grasping of moving objects

The other aspect is grasping moving objects. The idea is to plan toward a position where the object will be grasped by the SoftHand. However, to do this, one needs to predict the future position of the object, and this can be affected by the goodness of the motion state estimation of the object itself. The low computational time permits us to correct the trajectory in real time when the estimated grasping pose is updated. We performed some preliminary tests using an RGB-D camera (fixed on the environment) implementing an Extended Kalman Filter to estimate the motion state of the object. The results obtained are promising, and an example of this task can be observed in Figure 1. However, more trials to test the robustness of the approach have to be performed before implementing it in the DARKO platform. Moreover, the method should be adapted to the use of the onboard RGB-D cameras.

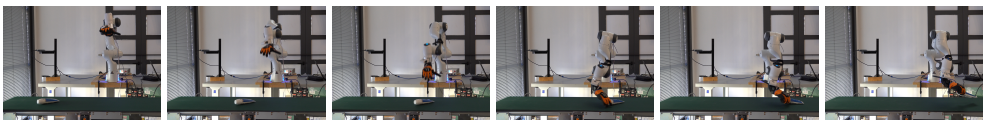

Figure 1: Screenshot from preliminary tests on grasping moving objects.

4.4 Software Package

In the actual state, only the obstacle-free version of the planner is implemented in the DARKO platform, while the complete version of the Human-Like planning algorithm will be introduced in the future.

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
```

The Human-like trajectory planning was implemented in ROS through a ROS service that takes as input the Cartesian target goal we would like to plan towards, and the current state of the robot in terms of cartesian pose. The computed trajectory is executed by a control loop that publishes the trajectory on the Cartesian impedance controller's topic.

5 Preliminary Work for Soft Manipulator

At the actual state, the DARKO platform has a classical rigid manipulator on top of it. However, as part of the project, an elastic manipulator will be developed to replace the Franka Emika Panda and exploit elasticity to enhance efficiency and performance. For this reason, we did different works regarding the planning and control of compliant kinematic chains to investigate possible solutions to be implemented on the DARKO platform.

Given that these works are still preliminary research in terms of application inside the DARKO project, none of them has code explicitly developed for the platform. For the code publicly available, we refer directly to the correspondent paper cited at the end of each specific section.

5.1 Optimal Control for Articulated Soft Robots

Soft robots can potentially execute tasks with forceful interactions. However, control techniques that can effectively exploit the systems' capabilities are still missing. Optimal control methods, especially Differential Dynamic Programming (DDP), have emerged as promising tools for achieving highly dynamic tasks. But most of the work in the literature tackles the application of DDP to articulated soft robots using numerical differentiation and only to perform explosive tasks with pure feed-forward control. Further, flexible link robots are known to be difficult to control and the use of DDP-based algorithms to control them is not yet addressed.

We propose an efficient DDP-based algorithm for trajectory optimization of articulated soft robots that can optimize the state trajectory, input torques and stiffness profile. We provide an efficient method to compute the forward dynamics and the analytical derivatives of SEA/VSA and flexible link robots. We present a state-feedback controller which uses the locally optimal feedback policies obtained from DDP. We show through simulations

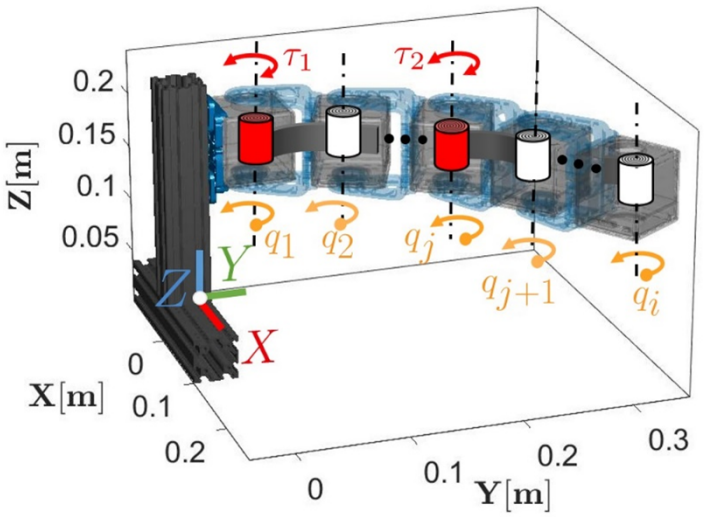

Figure 2: An underactuated compliant arm is a robot composed of several joints, of which only a few are actuated. The compliance is conferred to the arm by elastic elements lumped at the joints (both actuated and unactuated). It is worth noting that there is no limit on the number of unactuated joints. As an example, the figure depicts a robot with only two actuated elastic joints (red cylinders) and a generic number of passive ones (white cylinders).

and experiments that the use of feedback is crucial in improving the performance and stabilization properties in various tasks. We also show that the proposed method can be used to plan and control flexible link robots effectively. For further information regarding the work and the code released with the publication, the interested reader can refer to [7].

5.2 Iterative Learning Control for Compliant Underactuated Arms

Operations involving safe interactions in unstructured environments require robots with adapting behaviours. Compliant manipulators are a promising technology to achieve this goal. Despite that, some classical control problems such as following a trajectory are still open.

A typical solution is to compensate the system dynamics with feedback loops. However, this solution increases the effective robot stiffness and jeopardizes the safety property provided by the compliant design. On the other hand, purely feedforward approaches can achieve good tracking performance while preserving the robot intrinsic compliance. However, a feedforward control framework for robots with passive elastic joints is still missing.

To overcome these problems we have designed an iterative learning control algorithm for purely feedforward trajectory tracking for compliant underactuated arms. We tested the framework using an arm composed of active elastic joints and a generic number of passive ones connected through rigid links (depicted in Figure 2). We prove the convergence of the iterative method, also in the presence of uncertainties and bounded disturbances. Different output functions are analyzed providing conditions, based on the system inertial properties that ensure the algorithm's applicability. Additionally, an automatic selection of the learning gain is proposed. Finally, we extensively validate the theoretical results with simulations and experiments. For further detail the interested reader can refer to [19].

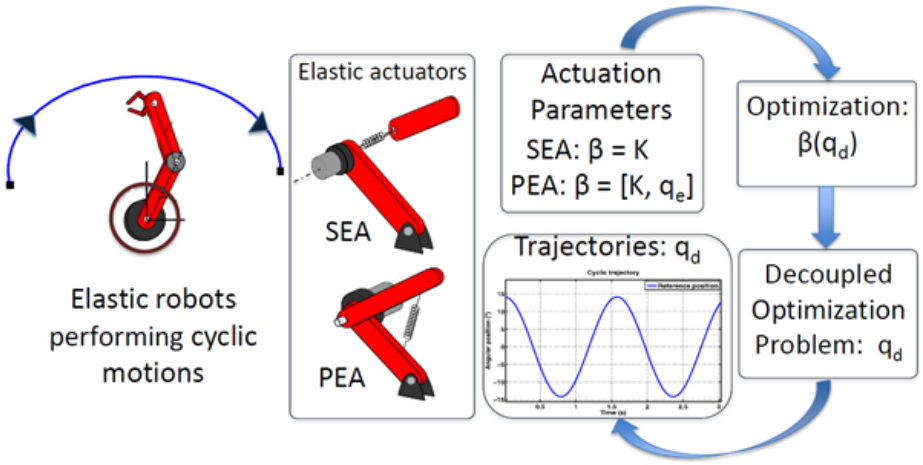

Figure 3: Schematic representation of the planning framework for minimization of energy consumption.

5.3 MInimizing Energy Consumption of Elastic Robots in Repetitive Tasks

Energy consumption is an important issue in robotics. We propose a solution to deal with the problem of reducing the energy consumption of compliant electro-mechanical systems while performing periodic tasks. After deriving performance indices to quantify the energy consumption of a mechanical system, we propose a method to determine both the optimal compliant actuation parameters and link trajectories to minimize energy consumption. A schematic framework is represented in Figure 3 We show how this problem can be cast in a simpler one where the optimization regards only parameters that define the shape of periodic trajectories to be subsequently determined by using numerical optimization tools. Indeed, in our framework, the optimal stiffness and spring pre-load can be analytically obtained as a function of the desired link trajectories.

We perform simulations and experimental validations on a two-link compliant manipulator platform which performs a repetitive pick-and-place task. Our experiments show that the use of compliant actuators instead of rigid ones and the optimization of their compliant parameters give rise to an energy saving up to 62% with respect to a rigid actuation. Moreover, the simultaneous optimization of the compliant parameters and link trajectories provides an additional energy saving up to 20%. For further detail the interested reader can refer to [26].

6 Conclusions

In this document, we reported an exhaustive description of the software infrastructure developed during the project. All the framework is already deployed in the DARKO platform and it was used to perform the demonstration during Milestone 2 in Stuttgart at Arena2036. It is worth mentioning that the entire framework has been built with a modular structure. This will allow in the future to easily insert new parts developed during the project, minimizing the changes to be made to the existing code.

We have also briefly described some preliminary results regarding the future steps to be tackled during the project. For these parts, we have provided mainly a theoretical

description, while the dissemination of the software is postponed until they are further developed from a DARKO perspective and implemented on the platform.

References

[1] Giuseppe Averta, Federica Barontini, Vincenzo Catrambone, Sami Haddadin, Giacomo Handjaras, Jeremia PO Held, Tingli Hu, Eike Jakubowitz, Christoph M Kanzler, Johannes Kühn, et al. U-limb: A multi-modal, multi-center database on arm motion control in healthy and post-stroke conditions. *GigaScience*, 10(6):giab043, 2021.

[2] 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)*, pages 3853–3859. IEEE, 2020.

[3] 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. *Frontiers in Robotics and AI*, 4:37, 2017.

[4] Giuseppe Averta, Cosimo Della Santina, Gaetano Valenza, Antonio Bicchi, and Matteo Bianchi. Exploiting upper-limb functional principal components for human-like motion generation of anthropomorphic robots. *Journal of NeuroEngineering and Rehabilitation*, 17(1):1–15, 2020.

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

[6] 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*, pages 339–343. Springer, 2022.

[7] Saroj Prasad Chhatoi, Michele Pierallini, Franco Angelini, Carlos Mastalli, and Manolo Garabini. Optimal control for articulated soft robots. *arXiv preprint arXiv:2306.01934*, 2023.

[8] David Coleman, Ioan Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *arXiv preprint arXiv:1404.3785*, 2014.

[9] Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, 5(7):1688–1703, 1985.

[10] Britta Grimme, John Lipinski, and Gregor Schöner. Naturalistic arm movements during obstacle avoidance in 3d and the identification of movement primitives. *Experimental brain research*, 222:185–200, 2012.

[11] Gianpaolo Gulletta, Wolfram Erlhagen, and Estela Bicho. Human-like arm motion generation: A review. *Robotics*, 9(4):102, 2020.

[12] Mary D Klein Breteler, Stan CAM Gielen, and Ruud GJ Meulenbroek. End-point constraints in aiming movements: effects of approach angle and speed. *Biological Cybernetics*, 85:65–75, 2001.

[13] Ariel Kwiatkowski, Eduardo Alvarado, Vicky Kalogeiton, C Karen Liu, Julien Pettré, Michiel van de Panne, and Marie-Paule Cani. A survey on reinforcement learning methods in character animation. In *Computer Graphics Forum*, volume 41, pages 613–639. Wiley Online Library, 2022.

[14] Francesco Lacquaniti, Carlo Terzuolo, and Paolo Viviani. The law relating the kinematic and figural aspects of drawing movements. *Acta psychologica*, 54(1-3):115–130, 1983.

[15] Igor Mordatch, Kendall Lowrey, Galen Andrew, Zoran Popovic, and Emanuel V Todorov. Interactive control of diverse complex characters with neural networks. *Advances in neural information processing systems*, 28, 2015.

[16] Lucas Mourot, Ludovic Hoyet, François Le Clerc, François Schnitzler, and Pierre Hellier. A survey on deep learning for skeleton-based human animation. In *Computer Graphics Forum*, volume 41, pages 122–157. Wiley Online Library, 2022.

[17] Clautilde Nguiadem, Maxime Raison, and Sofiane Achiche. Motion planning of upper-limb exoskeleton robots: a review. *Applied Sciences*, 10(21):7626, 2020.

[18] Aurelio Piazzi and Antonio Visioli. Global minimum-jerk trajectory planning of robot manipulators. *IEEE transactions on industrial electronics*, 47(1):140–149, 2000.

[19] Michele Pierallini, Franco Angelini, Riccardo Mengacci, Alessandro Palleschi, Antonio Bicchi, and Manolo Garabini. Iterative learning control for compliant underactuated arms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.

[20] James O Ramsay, Giles Hooker, and Spencer Graves. *Functional data analysis with R and MATLAB*. Springer Science & Business Media, 2009.

[21] Manuel Rebol, Christian Gütl, and Krzysztof Pietroszek. Real-time gesture animation generation from speech for virtual human interaction. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–4, 2021.

[22] Marco Santello, Matteo Bianchi, Marco Gabiccini, Emiliano Ricciardi, Gionata Salvietti, Domenico Prattichizzo, Marc Ernst, Alessandro Moscatelli, Henrik Jörntell, Astrid M.L. Kappers, Kostas Kyriakopoulos, Alin Albu-Schäffer, Claudio Castellini, and Antonio Bicchi. Hand synergies: Integration of robotics and neuroscience for understanding the control of biological and artificial hands. *Physics of Life Reviews*, 17:1–23, 2016.

[23] Trenton Schulz, Jim Torresen, and Jo Herstad. Animation techniques in human-robot interaction user studies: A systematic literature review. *ACM Transactions on Human-Robot Interaction (THRI)*, 8(2):1–22, 2019.

[24] Ioan Sucan and Sachin Chitta. Moveit, 2013.

[25] Yoji Uno, Mitsuo Kawato, and Rika Suzuki. Formation and control of optimal trajectory in human multijoint arm movement. *Biological cybernetics*, 61(2):89–101, 1989.

[26] Alexandra Velasco Vivas, Antonello Cherubini, Manolo Garabini, Paolo Salaris, and Antonio Bicchi. Minimizing energy consumption of elastic robots in repetitive tasks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.

[27] Wenjie Yin, Hang Yin, Danica Kragic, and Mårten Björkman. Graph-based normalizing flow for human motion generation and reconstruction. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, pages 641–648. IEEE, 2021.

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