Operational Space Control with Reinforcement Learning Redundancy Resolution for Agricultural Manipulators*

Héctor Garcés[1], Miguel Torres-Torriti[1] and Fernando Auat Cheein[2]

***Abstract*— In this study we present a neural controller for agricultural manipulators that combines a Reinforcement Learning (RL) agent with a low-level Operational Space Controller (OSC). The agent is responsible for motion planning, as well as redundancy resolution through null space torque compensation. Preliminary results show that the method proposed achieves better energy efficiency, a higher task success rate, decreasing task time and collision rates.**

I. INTRODUCTION

The advancement of agricultural technology, particularly agricultural robots, is an area of increasing research interest [6]. The integration of robotic manipulators in agriculture faces challenges due to the less structured nature and predictability of environment [2]. To achieve the greater manipulation dexterity and flexibility required to carry out agricultural tasks, such as harvesting and pruning, robotic arms employed often possess more degrees of freedom (DOF), even if in task space the required degrees of freedom are less [1]. Redundant DOFs allow the robot to approach fruits from multiple configurations [8]. On the other hand, redundancy may also be useful for obstacle avoidance [7], which is necessary when the target is behind leaves or branches.

Redundancy has been traditionally solved with analytical approaches to optimize secondary objectives that do not interfere with the end-effector's desired motion [7]. However, these approaches assume perfect knowledge of the underlying models. Unfortunately, this assumption is frequently not satisfied due to computational approximations and model simplifications, decreasing the system's performance. To overcome this limitation, reinforcement learning (RL) has gained attention because it optimizes cost functions using data from the robot's interactions with its environment [5].

We propose an RL agent that separately acts as a task space motion planner and is also responsible for the redundancy resolution. The RL agent exploits the null space to maximize the given reward function, which in this case considers factors such as energy consumption, obstacle avoidance, time, and task success. To assess the effectiveness of the proposed strategy, the training employs a simulated version of a 7-DOF Franka Emika Research 3 robotic arm in MuJoCo to solve the motion planning and control for harvesting fruits that appear within the robot's workspace. The trained algorithm is tested in the laboratory using the robots of Fig. 1.

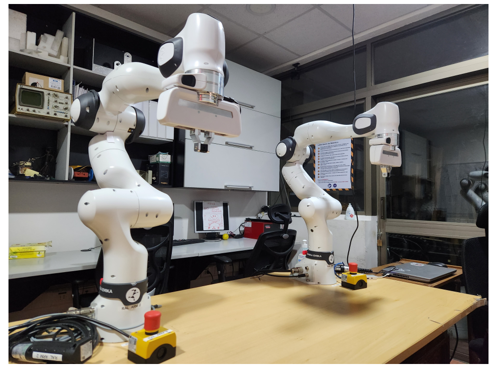

Fig. 1. Robotic arms Franka Emika Research 3 setup in laboratory

II. PROPOSED METHOD

The low-level operational space controller (OSC) employs the robot's dynamic model to define the control torques:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \tag{1}$$

where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^n$ is the Coriolis/centripetal vector, $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ is the gravity vector, and $\boldsymbol{\tau} \in \mathbb{R}^n$ is the joint torque vector.

Torques in equation (1) can be separated into those that generate force in the end-effector from those that do not [7]. Additionally, some changes can be done, neglecting the Coriolis force [4], making a full compensation of the gravitational forces [7], and removing terms that contain the differentiation of the Jacobian due to computational complexity, thus reducing equation (1) to:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\bar{\mathbf{J}}(\ddot{\mathbf{x}}_{ref}) + \mathbf{g}(\mathbf{q}) + (\mathbf{I} - \mathbf{J}^T\bar{\mathbf{J}}^T)\boldsymbol{\tau}_{null} \tag{2}$$

where $\boldsymbol{\tau}_{null}$ represents the null space torques that do not influence the Cartesian motion of the end-effector, and $\bar{\mathbf{J}}$ denotes the inertia-weighted pseudoinverse. The reference acceleration $\ddot{\mathbf{x}}_{ref}$ is defined as:

$$\ddot{\mathbf{x}}_{ref} = \ddot{\mathbf{x}}_d + k_d(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + k_p(\mathbf{x}_d - \mathbf{x}) \tag{3}$$

where $\ddot{\mathbf{x}}_d$ and $\dot{\mathbf{x}}_d$ are respectively the desired acceleration (set to zero) and velocity, while $\boldsymbol{\tau}_{null}$ and $\mathbf{x}_d$ are the outputs computed by the reinforcement learning agent. In other words, the applied robot joint torques combine in (2) the OSC with the task space acceleration $\mathbf{x}_d$ and the null space torque $\boldsymbol{\tau}_{null}$ recommended by the RL agent.

*This project has been supported by the National Agency of Research and Development (ANID) under grants Fondecyt 1220140 and ANID FB0008 Basal Project.

[1]Héctor Garcés and Miguel Torres-Torriti are with the Department of Electrical Engineering, Pontificia Universidad Católica de Chile, Av. Vicuña Mackenna 4860, Chile hagarces@uc.cl, mtorrest@ing.puc.cl

[2]Fernando Auat Cheein is with the Department of Electronics, Universidad Tecnica Federico Santa María, Av. España 1680, Valparaíso, Chile fernando.auat@usm.cl

The RL is based on the Soft Actor Critic (SAC) algorithm, which is robust and has good performance on continuous action spaces, and is thus well-suited for motion control of robotic arms [3] or grasping [9]. Other RL algorithms try to maximize the expected future reward, but SAC also has an entropy component in its objective function. This gives the agent an incentive to explore new actions. The objective function is defined as follows:

$$J(\pi) = E_{(s_t,a_t)\sim\rho_\pi} \sum_{t=0}^{T} \gamma^t \left[r(s_t, a_t) + \alpha H(\pi(*, s_t))\right] \quad (4)$$

where $r(s_t, a_t)$ is the reward obtained by the agent at time t and $H(\pi(*, s_t))$ is the entropy of the policy at state s and time t.

The state representation of the environment consists of a 22-dimensional vector:

$$s = [\sin(\mathbf{q}), \cos(\mathbf{q}), \dot{\mathbf{q}}, \Delta x_{target}, \theta_{target}] \quad (5)$$

where Δx_{target} is the Cartesian distance between the effector and the target, and θ_{target} is the angle between a projected line of the effector and the target.

The agent's action vector is defined as:

$$a = [\mathbf{x}_{desired}, \boldsymbol{\tau}_{null}] \quad (6)$$

Each episode ends after 10 seconds or when the robot harvests its objective. The controller sampling time is 1 millisecond like the sampling time of the real Franka Emika Research 3 robot. The agent outputs new signals every 0.5 seconds, thus in every episode there up to 20 interactions.

The goal of the robot is to learn how to harvest fast, with colliding, and employing the least possible amount of energy. To this end, the reward for each episode is computed as:

$$R_{total} = \Delta t \sum_{t=1}^{20} \left[-k_t - k_e \sum_{n=1}^{N} \tau_n(t)\dot{q}_n(t)\right] - r_c + r_s \quad (7)$$

where k_t represents a constant penalty for each time interval that the robot has not reached its objective, k_e indicates the penalty associated with energy consumption, while r_c is the penalty when the robot crashes with itself or the environment. Finally, r_s corresponds to a positive reward assigned upon successful completion of the robot's task.

III. RESULTS

The Franka Emika Research 3 robot and its environment were first simulated with MuJoCo; see Fig. 2. The agent was programmed with Python 3 and the PyTorch library. The results summarized in Table I indicate that the proposed strategy has better performance in terms of energy efficiency, completion time, and fewer collisions with the environment compared to traditional OSC, or the RL motion planner without null space torque compensation (an implementation of an RL agent that only computes an action $a = \mathbf{x}_{desired}$ and does not compute $\boldsymbol{\tau}_{null}$).

IV. CONCLUSIONS

The OSC with RL-SAC motion planner and null space torque compensation achieves better performance than traditional OSC implemented as motion controller for harvesting tasks. Ongoing research considers the extension of the proposed strategy to dual-arm mobile manipulators.

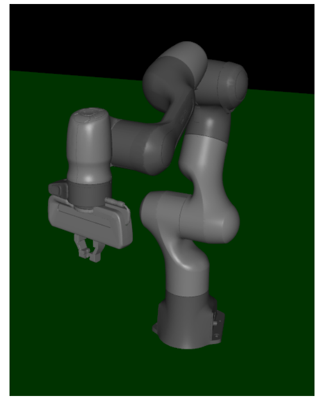

Fig. 2. Simulated Franka Emika Research 3 robot arm in MuJoCo.

TABLE I

PERFORMANCE COMPARISON OF DIFFERENT HARVESTING MOTION CONTROL STRATEGIES

Methods	**Score**	**Energy** [J]	**Time** [s]	**Success** rate	**Collision** rate
OSC	-2.561	47.874	7.735	64%	2%
OSC + RL	-0.517	31.474	5.705	98.0%	0.0%
OSC + RL without $\boldsymbol{\tau}_{null}$	-1.641	40.365	6.268	86%	8%
$\boldsymbol{\tau} = \mathbf{g}(\mathbf{q})$	-3	0	-	0%	0%

ACKNOWLEDGMENT

This project has been supported by the National Agency of Research and Development (ANID) under Doctoral grant 21212303. Grant acknowledgments Fondecyt 1220140 and ANID FB0008 Basal Project.

REFERENCES

[1] Mihai Crenganis and Octavian Bologa. Efficient method for redundancy resolution of a 7 dof manipulator. 2015.

[2] Leonidas Droukas, Zoe Doulgeri, Nikolaos L. Tsakiridis, Dimitra Triantafyllou, Ioannis Kleitsiotis, Ioannis Mariolis, Dimitrios Giakoumis, Dimitrios Tzovaras, Dimitrios Kateris, and Dionysis Bochtis. A survey of robotic harvesting systems and enabling technologies. *Journal of Intelligent Robotic Systems*, 107(2), jan 2023.

[3] Zeyuan Huang, Gang Chen, Yue Shen, Yu Liu, Hong You, and Tong Li. Lnoa: A real-time obstacle avoidance motion planning method for redundant manipulator based on reinforcement learning. In *2022 International Conference on Service Robotics (ICoSR)*, pages 1–6, June 2022.

[4] Manuel Kaspar, Juan D. Muñoz Osorio, and Juergen Bock. Sim2real transfer for reinforcement learning without dynamics randomization. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4383–4388, Oct 2020.

[5] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[6] Vasso Marinoudi, Claus G. Sørensen, Simon Pearson, and Dionysis Bochtis. Robotics and labour in agriculture. a context consideration. *Biosystems Engineering*, 184:111–121, 2019.

[7] Jun Nakanishi, Rick Cory, Michael Mistry, Jan Peters, and Stefan Schaal. Operational space control: A theoretical and empirical comparison. *The International Journal of Robotics Research*, 27(6):737–757, 2008.

[8] Tien Thanh Nguyen, Erdal Kayacan, Josse De Baedemaeker, and Wouter Saeys. Task and motion planning for apple harvesting robot*. *IFAC Proceedings Volumes*, 46(18):247–252, 2013. 4th IFAC Conference on Modelling and Control in Agriculture, Horticulture and Post Harvest Industry.

[9] Zixin Tang, Xin Xu, and Yifei Shi. Grasp planning based on deep reinforcement learning: A brief survey. In *2021 China Automation Congress (CAC)*, pages 7293–7299, Oct 2021.