2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

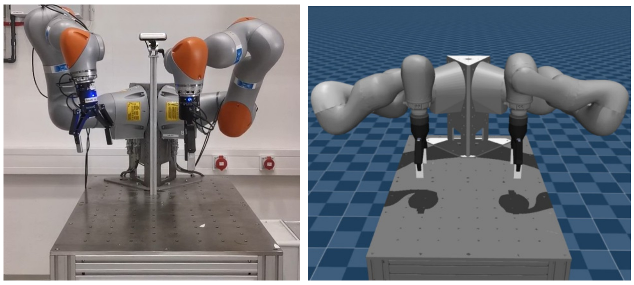
# Look Ahead Optimization for Managing Nullspace in Cartesian Impedance Control of Dual-Arm Robots

Vamsi Krishna Origanti[1], Adrian Danzglock[1] and Frank Kirchner[1,2]

***Abstract*— This paper presents a method for handling nullspace challenges in Cartesian impedance control of a dual-arm KUKA IIWA robot by employing a Look ahead Controller(LAC) in nullspace. Ambidexterity is crucial for dual-arm robots to perform complex tasks that require coordinated use of both arms. Cartesian impedance control provides significant advantages in dual-arm manipulation tasks, especially in imitation learning for reproducing learned compliant interactions and precise control of end-effector poses. This approach enables the learned tasks to be robot-agnostic, facilitating transfer to other robotic systems. However, the nullspace handling of cartesian impedance control is very challenging. In this paper, we address this issue to handle kinematic constraints and facilitate avoiding singularities, joint limits, and collision in nullspace or redundant space of dual arms with each other with the help of a LAC. The proposed approach utilizes Sequential QP in the optimization loop of LAC for estimating optimal joint configurations for a horizon in redundant space, this provides the safe and efficient operation. Results are provided in this paper for two trajectories and compared with and without optimization, results demonstrate the method's effectiveness in maintaining desired end-effector poses while avoiding kinematic constraints and nullspace collisions.**

## I. INTRODUCTION

Ambidexterity, the ability to use both hands with equal proficiency, is an essential feature for dual-arm robots to reproduce complex tasks learned from imitation learning that require coordinated and simultaneous use of both arms as shown in Fig.1. Achieving this ambidexterity poses significant challenges, including avoiding singularities, joint limits, and collisions between the arms. Cartesian impedance control is a widely used technique for robotic manipulation, providing compliant behavior and precise control of end-effector poses, particularly in imitation learning. This control approach allows robots to adaptively reproduce tasks by adjusting to dynamic environments, making it highly beneficial for transferring learned behaviors across different robotic systems [1], [4]. However, handling the nullspace in Cartesian impedance control presents significant challenges, such as avoiding singularities and avoiding joint limits while maintaining the desired end-effector trajectory [2], [3]. In addition to the above challenges, avoiding nullspace collisions of the dual arms with each other is also an important factor to be considered in dual-arm robots. These complexities can be addressed with sophisticated optimization techniques to make use of the robot's redundant degrees of freedom effectively.



(a) Robot in Real World (b) Robot in Mujoco Simulation

Fig. 1: Dual Arm KUKA IIWA robot.

This paper focuses on addressing these challenges by employing Sequential Least Squares Quadratic Programming (SLSQP) optimization. This method integrates with a look-ahead controller (LAC) framework, which acts as a foresee controller estimating the optimal trajectory of the nullspace configuration for future horizons by utilizing the kinematic model of the robot. The proposed solution computes optimal joint configurations while maintaining safety and efficiency during dual-arm operations.

The paper is organized as follows: Section II reviews related work, Section III describes the methodology which includes the nullspace projection and the main objective function of the optimization problem, Section IV details the implementation of the Cartesian impedance controller with nullspace joint impedance control, optimization module, and interface through ROS2 and, Section V presents the results and evaluation for the experiments conducted in simulation, and Section VI concludes the study with potential future work.

## II. RELATED WORK

Previous research has focused on various methods for Cartesian impedance control and collision avoidance in robotic systems. Nullspace optimization techniques have been explored for single-arm robots, but extending these methods to dual-arm systems introduces additional complexities due to inter-arm interactions. Ambidextrous capabilities in humanoid robots have been a topic of growing interest, emphasizing the need for effective control strategies to manage the intricate dynamics and coordination required for such tasks. In recent years, Cartesian impedance control has become a focal point in robotic manipulation, particularly for collaborative and compliant interactions [1]. Various studies

[1]German Research Center for Artificial Intelligence (DFKI GmbH), Robotics Innovation Center, 28359 Bremen, Germany. firstname.lastname@dfki.de
[2]Department of Mathematics and Informatics, University of Bremen, 28359 Bremen, Germany. frank.kirchner@uni-bremen.de

have explored different approaches to enhance robot performance, including the integration of nullspace optimization to manage redundant degrees of freedom. Khatib's operational space formulation laid the groundwork for combining motion and force control [2]. Siciliano and Khatib further investigated kinematic control of redundant robots, emphasizing the importance of nullspace handling to avoid singularities and improve joint limit management [3].

Mansfeld et al. [12] proposed an approach to improve the performance of auxiliary null space tasks via time scaling-based relaxation of the primary task. This work highlights the significance of balancing primary and secondary tasks to achieve better performance in robotic systems with redundant degrees of freedom. However, the primary task is in joint space in contrast to our requirement where the primary task is in Cartesian space.

Regarding collision avoidance Lei et al. [10] proposed a real-time kinematics-based self-collision avoidance algorithm for dual-arm robots. While their approach is effective for self-collision avoidance, our work extends beyond this by integrating it in a multi-objective optimization framework that not only addresses collision avoidance but also optimizes joint configurations for manipulability, joint limit avoidance, and jerk minimization within the nullspace of the Cartesian impedance controller.

Recent advancements have incorporated optimization algorithms, such as quadratic programming, to refine nullspace projections and ensure task-priority frameworks [5]. Hoffman et al. [11] presented a multi-priority Cartesian impedance control method based on quadratic programming optimization. Their approach emphasizes managing multiple task priorities using quadratic programming to achieve optimal control in task space. Kuindersma et al. [6] presented an optimization-based approach for whole-body control of humanoid robots, focusing on dynamic stability and force optimization. Kanoun et al. [7] explored kinematic control with inequality constraints to address joint limit and collision avoidance. Additionally, Zhou et al. [8] investigated multi-objective optimization in robotic manipulators, emphasizing adaptability in complex environments. Furthermore, Wu et al. [9] utilized reinforcement learning to enhance impedance control strategies, demonstrating improved performance in unstructured settings.

Model Predictive Control (MPC) with Cartesian Impedance control has been extensively applied to manage variable stiffness and damping. Thelenberg et al. [25] proposed an MPC-based strategy for dynamically adjusting stiffness and damping to ensure compliance during trajectory tracking. Similarly, Anand et al. [26] introduced a Deep Model Predictive Variable Impedance Controller, integrating MPC with learned Cartesian impedance models to adapt impedance parameters dynamically for diverse manipulation tasks.

While MPC has demonstrated robust capabilities in primary task control, its reliance on precise system models and significant computational overhead can present challenges, particularly in real-time applications. In this study, LAC is employed specifically for nullspace optimization in Cartesian impedance control, enabling effective management of redundant degrees of freedom while maintaining compliance. Unlike previous works, where MPC governs primary task dynamics, LAC is applied solely in the nullspace, optimizing secondary objectives. This decoupling of primary task control from nullspace optimization ensures computational efficiency while addressing constraints unique to redundant systems.

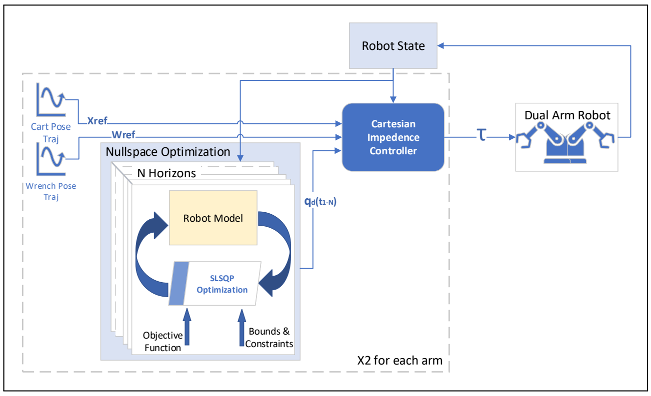



Fig. 2: Overview of the Nullspace Optimization of Cartesian Impedance Controller using Look Ahead Control, computes optimal nullspace configuration for N Horizons.

Our method leverages a scalarized multi-objective function within the LAC framework to predict and interpolate optimal nullspace configurations over future horizons. This approach ensures balanced optimization across constraints without the computational burden typically associated with MPC for primary tasks, making LAC a suitable choice for managing nullspace configurations in dual-arm systems.

## III. Methodology

In this section, we elucidate our approach to optimize nullspace of the cartesian impedance controller to address the challenges without disturbing the task space. The overview of the implementation is illustrated in the Fig.2.

### A. *Cartesian Impedance Control Law*

Cartesian impedance control is a strategy for robotic manipulators that establishes a dynamic relationship between a robot's pose and interaction forces, making it ideal for collaborative robots. It allows the robot to adjust its position based on contact forces, enhancing safety and compliance in shared environments. The control law for Cartesian impedance [1] can be expressed as:

$$\mathbf{F} = \mathbf{K}_p(\mathbf{x}_d - \mathbf{x}) + \mathbf{D}_p(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) \tag{1}$$

Where $\mathbf{F}$ is the wrench vector applied at the end-effector, $\mathbf{K}_p$ is the stiffness matrix, $\mathbf{D}_p$ is the damping matrix, $\mathbf{x}_d$ and $\mathbf{x}$ are the desired and actual positions, respectively, and $\dot{\mathbf{x}}_d$ and $\dot{\mathbf{x}}$ are the desired and actual velocities.

The conversion of wrenches to joint torques, considering external Cartesian wrenches $\mathbf{F}_{\text{ext}}$, is given by:

$$\tau = \mathbf{J}^T(\mathbf{F} + \mathbf{F}_{\text{ext}}) \tag{2}$$

where $\tau$ represents the joint torque vector, $\mathbf{J}^T$ is the transpose of the Jacobian matrix, and $\mathbf{F}_{\text{ext}}$ denotes the desired external Cartesian wrenches at the end-effector.

Incorporating nullspace control allows for the utilization of redundant degrees of freedom, optimizing joint configurations. The nullspace projection ensures that the robot's primary task is prioritized, while secondary tasks (e.g., joint configurations) are handled without affecting the main task [2], [3]. The control law in the presence of nullspace optimization is given by:

$$\tau = \mathbf{J}^T(\mathbf{F} + \mathbf{F}_{\text{ext}}) + (\mathbf{I} - \mathbf{J}^T \left(\mathbf{J}^T\right)^{\dagger})\tau_{\text{null}} \quad (3)$$

where $\tau$ is the joint torque vector, $\mathbf{J}$ is the Jacobian matrix, $(\mathbf{J}^T)^{\dagger}$ is the pseudo-inverse of the Jacobian Transpose, and $\tau_{\text{null}}$ represents the torque components in the nullspace, which is our main point of interest that we make use of this term to achieve our objectives.

Our idea is to implement a joint impedance control within the nullspace of the Cartesian impedance control. This enables the control of the desired nullspace configuration, which is the solution acquired from the optimization problem detailed in the following sections.

The joint impedance control law in the nullspace can be expressed as [13]:

$$\tau_{\text{null}} = \mathbf{K}_{null}(\mathbf{q}_d - \mathbf{q}) + \mathbf{D}_{null}(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) \quad (4)$$

where $\mathbf{K}_{null}$ is the joint stiffness matrix, $\mathbf{D}_{null}$ is the joint damping matrix, $\mathbf{q}_d$ and $\mathbf{q}$ are the desired nullspace and actual joint configurations, and $\dot{\mathbf{q}}_d$ and $\dot{\mathbf{q}}$ are the desired and actual joint velocities. However, in order to simplify the problem, we only give current joint velocities $\dot{\mathbf{q}}$.

### *B. Nullspace Optimization*

Nullspace optimization is the main objective of this work to make use of the redundant degrees of freedom of a redundant robot. This technique allows the robot to prioritize primary tasks, maintaining the desired end-effector position, orientation, and wrench, while simultaneously optimizing secondary objectives within the nullspace. We employ Sequential Least SQuares Programming (SLSQP) optimizer inside LAC to achieve our objectives. The optimization process computes the desired joint configurations over a period of horizon that achieves these goals without compromising the main task.

**Look ahead control:** In Look Ahead Control (LAC), optimization is performed at each step within a predefined horizon, as formulated in Algorithm 1. This allows the system to predict and generate an optimized trajectory for each robotic arm across future states within the horizon. The process begins by using the current joint positions as the initial guess for optimization. For subsequent steps within the horizon, the initial guess for each next time step is derived from the optimal nullspace configuration obtained in the previous time step.

During each step of the optimization process, the robot's system model is utilized in the objective function to search for an optimal nullspace configuration. This model-based optimization adapts the nullspace configuration dynamically for future states, ensuring that the trajectory respects the robot's kinematic constraints.

At each horizon, the optimization process generates a trajectory that balances multiple objectives in the nullspace while maintaining the primary control objectives of the Cartesian impedance controller. The resulting optimized trajectory is then streamed to the nullspace of the controller, ensuring efficient management of the nullspace dynamics. However, the length of the prediction horizon heavily influences the system accuracy, hence needs to be chosen wisely for generating optimal trajectories.

**Algorithm 1** Look Ahead Control (LAC) with Nullspace Optimization

**Input:** Current joint states $\mathbf{q}_0$, horizon length $N$, objectives $f_1, f_2, \dots, f_n$
**Output:** Optimal trajectory $\mathbf{q}_{\text{traj}}$ for a length of horizon $N$
**for** each time step $t$ **do**
  Initialize current initial guess to current joint states $\mathbf{q}_t$
  Initialize $\mathbf{q}_{\text{traj},t} = [\ ]$ ▷ Initialize the optimal trajectory for the current horizon
  **for** each step $k = 0$ to $N - 1$ **do**
    Update the robot dynamic model for $\mathbf{q}_{t+k}$
    Formulate scalarized multi-objective function:

$$J = \sum_{i=1}^{n} w_i f_i(\mathbf{q}_{t+k})$$

    Minimize the objective with an initial guess:

$$\mathbf{q}_{\text{opt},t+k} = \arg\min_{\mathbf{q}} J$$

$$\text{subject to } \mathbf{q}_{\text{min}} \leq \mathbf{q} \leq \mathbf{q}_{\text{max}}$$

    Update current initial guess to $\mathbf{q}_{\text{opt},t+k}$
    Append $\mathbf{q}_{\text{opt},t+k}$ to $\mathbf{q}_{\text{traj},t}$
  **end for**
  Command $\mathbf{q}_{\text{traj},t}$ to the nullspace of the Cartesian impedance controller
  Update $\mathbf{q}_t$ based on feedback
**end for**

We try to optimize the null space to achieve five objectives ensuring smooth and efficient operation of the robot.

**Objective Function:** It is clear that the optimization is a multi objective optimization problem, due to the computational complexity, in contrast to the multi objective optimization techniques such as Pareto based optimization, here we minimize a scalarized multi-objective function that combines the multi objective optimization problem to a single objective function by a weighted sum of the objective functions. where $f_i(\mathbf{q})$ represents the objectives and $w_i$ are the weights assigned to each objective. By assigning weights to each objective, the function allows us to prioritize specific goals based on the task requirements. We consider to minimize the below objective functions:

*1) **Cartesian Pose Error**:* The first objective is the Cartesian pose error, which is crucial for ensuring that the nullspace configuration corresponds accurately to the desired Cartesian position. If the nullspace configuration does not align with the Cartesian position, it can result in instability and unpredictable behavior of the robot. The Cartesian pose error is the difference between the current and desired end-effector poses, including both positional and orientation errors. The Cartesian pose error is defined as:

$$f_1(\mathbf{q}) = \|\mathbf{x}_{\text{desired}} - \mathbf{x}_{\text{current}}\| + \mathbf{Q}_{\text{dist}}(\mathbf{q}_{\text{desired}}, \mathbf{q}_{\text{current}}) \quad (5)$$

Where $\mathbf{x}_{\text{current}}$ and $\mathbf{x}_{\text{desired}}$ are the current and desired Cartesian positions, $\mathbf{q}_{\text{current}}$ and $\mathbf{q}_{\text{desired}}$ are the current and desired orientations. The orientation error is computed by the quaternion absolute distance [19]:

$$\mathbf{Q}_{\text{dist}}(\mathbf{q}_{\text{desired}}, \mathbf{q}_{\text{current}}) = 1 - (\mathbf{q}_{\text{desired}} \cdot \mathbf{q}_{\text{current}})^2 \quad (6)$$

By minimizing this error, we ensure that the nullspace configuration is aligned with the Cartesian position, hence this objective is considered high priority.

*2) **Manipulability Cost:*** The manipulability cost is derived from the singular values of the Jacobian matrix. High manipulability indicates that the robot can move freely in its workspace, while low manipulability suggests that the robot is near a singular configuration.

Manipulability measures were first introduced by Yoshikawa [14], who proposed using the determinant and singular values of the Jacobian matrix to assess the robot's capability to perform tasks. This measure has been widely adopted and extended in various works. For instance, Tsai and Morgan [15] and Ang et al. [16] utilized manipulability for path planning and control of robotic manipulators. From the above reference, the manipulability of a redundant robot can be defined as:

$$\mu = \frac{\sigma_{\min}}{\sigma_{\max}} \quad (7)$$

The second objective, manipulability cost, which can be inversely related to the manipulability measure of the robot can be defined as:

$$f_2(\mathbf{q}) = \frac{1}{\mu^2} \quad (8)$$

where $\sigma_{\min}$ and $\sigma_{\max}$ are the minimum and maximum singular values of the Jacobian matrix, respectively.

This formulation ensures that as the robot approaches a singular configuration, the cost increases significantly, thereby guiding the optimization to avoid such configurations.

*3) **Joint Limit Cost:*** In order to ensure that the optimal joint angles remain within their specified limits, we have formulated a cost function that penalizes joint configurations as they approach these limits. This joint limit cost function is crucial for maintaining the integrity and safety of the robot's movements, as it prevents the joints from exceeding their mechanical boundaries. The joint limit cost function is defined as:

$$f_3(\mathbf{q}) = \sum_i \begin{cases} \left(\frac{q_i - (q_{\min,i} + \epsilon)}{\epsilon}\right)^2 & \text{if } q_i < q_{\min,i} + \epsilon \\ \left(\frac{q_i - (q_{\max,i} - \epsilon)}{\epsilon}\right)^2 & \text{if } q_i > q_{\max,i} - \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Where, $q_i$ is the joint angle,$q_{\min,i}$ is the lower limit of the joint,$q_{\max,i}$ is the upper limit of the joint, $\epsilon$ is a small threshold that defines a boundary region near the joint limits.

This cost function increases the penalty as the joint angles approach their limits, effectively guiding the optimization process to avoid configurations that are too close to these boundaries. By doing so, we ensure that the robot operates within a safe and efficient range.

*4) **Collision Avoidance Cost:*** To ensure the safe operation of the dual-arm robot, it is crucial to avoid nullspace collisions between the two arms. Nullspace collisions occur when the redundant degrees of freedom, managed within the nullspace, lead to configurations where the links of the two arms come too close or intersect each other. This situation is particularly problematic as it can happen without affecting the primary task performance, making it less obvious to detect and prevent. The collision avoidance cost is designed to minimize the risk of such collisions by considering the minimum collision distances between the collision geometries of the links of both arms. The computations exclude the pairs that are fixed and adjacent to each other. These collision geometries are defined in the URDF of the robot and can be either mesh files or geometric primitives. The cost function is defined as:

$$f_4(\mathbf{q}) = \frac{K}{\sum_i \left(\min_j \text{dist}(\text{geom}_i^{\text{left}}, \text{geom}_j^{\text{right}})\right)^2} \quad (10)$$

where, $\mathbf{q}$ represents the joint angles of the robot. $\text{geom}_i^{\text{left}}$ denotes the collision geometry of the $i$-th link of the left arm not in the excluded pairs. $\text{geom}_j^{\text{right}}$ denotes the collision geometry of the $j$-th link of the right arm not in the excluded pairs. $\text{dist}(\text{geom}_i^{\text{left}}, \text{geom}_j^{\text{right}})$ represents the distance between the collision geometries of the $i$-th link of the left arm and the $j$-th link of the right arm. And $\min_j dist$ represents the minimum collision distance to the $j$-th link of the right arm

This formulation ensures that the cost increases significantly as the minimum collision distances between the selected links of the two arms decrease, thereby encouraging the optimization process to find configurations that maximize these distances and reduce the risk of collisions. The Fig. 3 illustrates an example of the minimum collision distance calculation of a link in the right arm with respect to the selected links of the left arm.

The minimum collision distance between the selected link pairs is computed in real time using the Flexible Collision Library (FCL) [20] and the Trimesh library [21] from the collision geometries defined in the URDF for each link, which can be either mesh or geometric primitives. This approach provides a more accurate representation of the robot's physical structure compared to using the link centers, resulting in effective collision cost estimation.

### *C. Nullspace Adaptive Joint Limit Avoidance*

In our approach, although the optimization in the nullspace includes the bounds to achieve optimal configurations within

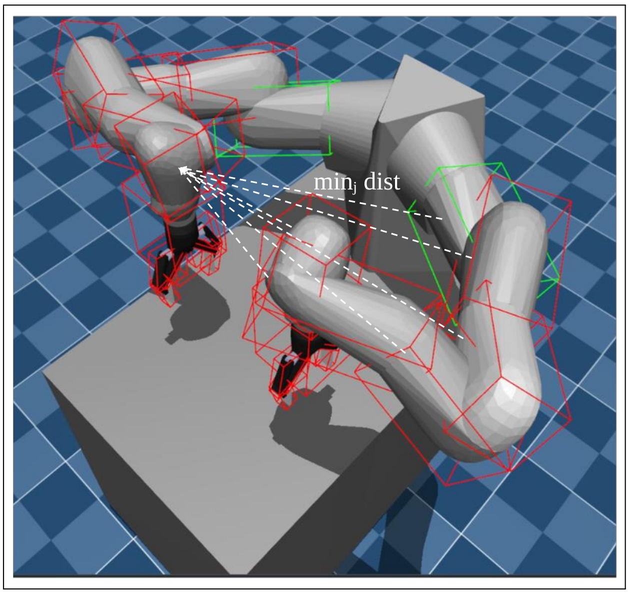



Fig. 3: Visualization of Minimum Collision Distances (white dashed lines) of a link in the right arm to the selected links in the left arm.

joint limits, task space domination can occasionally lead to configurations where the nullspace approaches these limits. To address this issue, we implemented an adaptive stiffness and damping mechanism in the nullspace to compute additional torques that push the robot away from the joint limits. This is achieved by incorporating an additional joint impedance control in the nullspace, where the error is the difference between the current joint positions and the midpoint of the joint limits. However, it is not necessary to apply these torques constantly when the joint positions are not at the midpoints; therefore, stiffness is adapted and smoothly increases as the joint positions approach the limits.

The adaptive nullspace stiffness $K_{\text{null\_adapt}}$ is defined by the following formula:

$$K_{\text{null\_adapt}} = |K_m \left[\tanh(b(q - c_1)) + \tanh(b(q - c_2))\right]| \tag{11}$$

where:

- $K_m$ is a scaling factor that determines the maximum stiffness increase,
- $b$ controls the steepness of the stiffness increase,
- $q$ represents the joint angle,
- $c_1$ and $c_2$ are the lower and upper thresholds, respectively, beyond which stiffness begins to increase.

The corresponding adaptive damping can be defined as:

$$D_{\text{null\_adapt}} = 2\sqrt{|\mathbf{K}_{\text{null\_adapt}}|} \tag{12}$$

Finally, the joint limit torques in the nullspace can be computed as:

$$\boldsymbol{\tau}_{\text{joint\_limits}} = \mathbf{K}_{\text{null\_adapt}} \cdot (-\mathbf{e}_{\text{limit}}) - \mathbf{D}_{\text{null\_adapt}} \cdot \dot{\mathbf{q}} \tag{13}$$

Whereas, $\mathbf{e}_{\text{limit}}$ is the error input which is the difference from the current position to the midpoints of the joints.

$$\mathbf{e}_{\text{limit}} = q - \frac{\mathbf{q}_{\text{min}} + \mathbf{q}_{\text{max}}}{2}$$

This formulation allows for a smooth increase in stiffness as the joint angles approach their limits, effectively preventing violations of joint constraints while maintaining flexibility within the safe operating range.

## IV. Implementation

The proposed approach was implemented on a dual-arm KUKA IIWA 14 robot as shown in Fig.1 using ROS2 [24] interface. The Cartesian impedance controller was adapted to ROS2 and modified as required from the ROS1 implementation as mentioned in [22]. This control law leverages its compatibility with collaborative robots and its effectiveness in force control tasks.

The entire implementation was carried out on an Intel PC equipped with an i9 processor, 64 GB RAM, and 32 cores, ensuring sufficient computational power for real-time processing. We utilized the SciPy [23] optimize library to perform optimization tasks, focusing on achieving a balanced solution across multiple objectives, including end-effector accuracy, manipulability, and collision avoidance. The PyKDL library was employed for modeling the robot's kinematics, allowing precise calculations within the optimization loop.

The implementation involves several key components and interactions between different ROS2 nodes and libraries. The Fig.4 outlines the structure and data flow within the system, as detailed below:

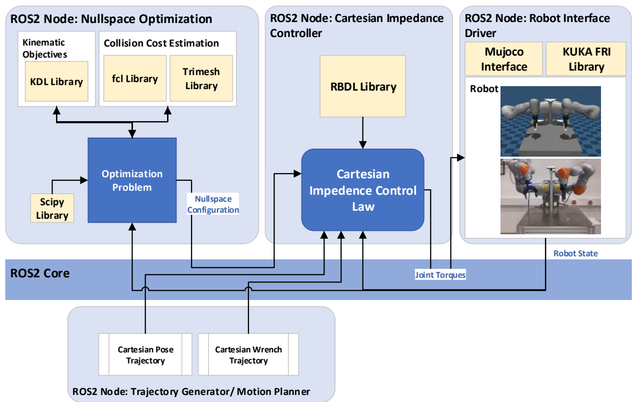
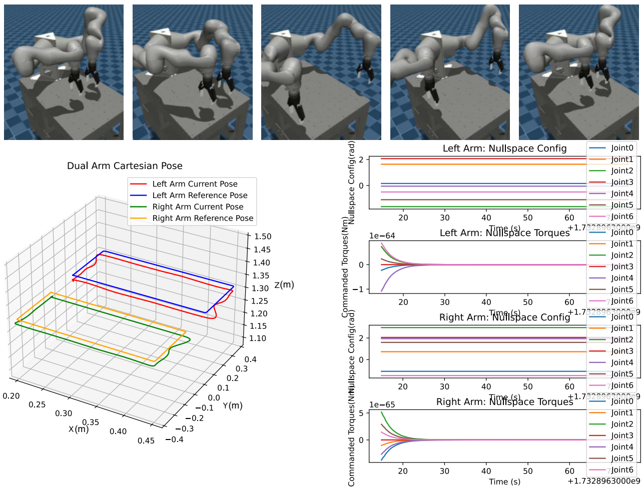
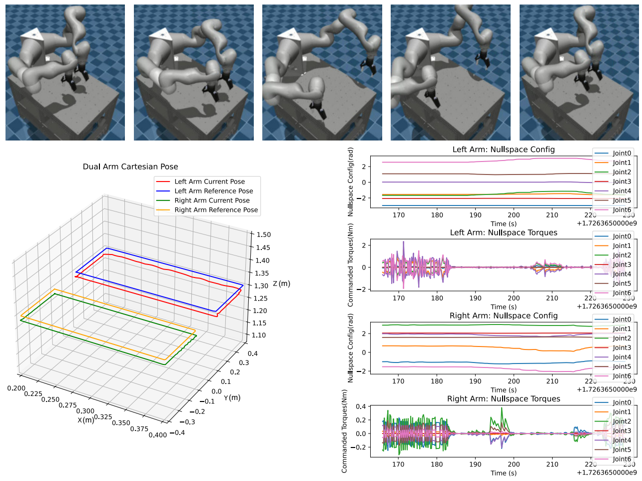



Fig. 4: Implementation Overview.

- **ROS2 Core:** The central communication framework that connects all ROS2 nodes and facilitates data exchange.
- **ROS2 Nodes:**
  - **Cartesian Impedance Controller** This node is responsible for calculating the Cartesian impedance control law and generating the required joint torques. It interacts with the robot library to send commands to the robot.

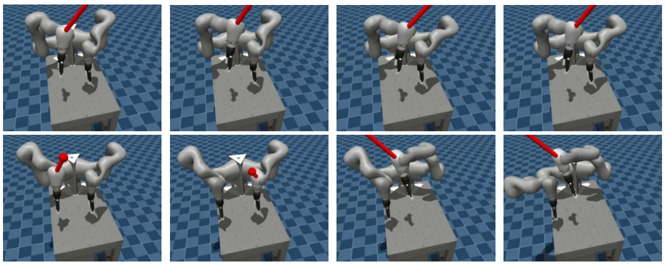



(a) Without Optimization

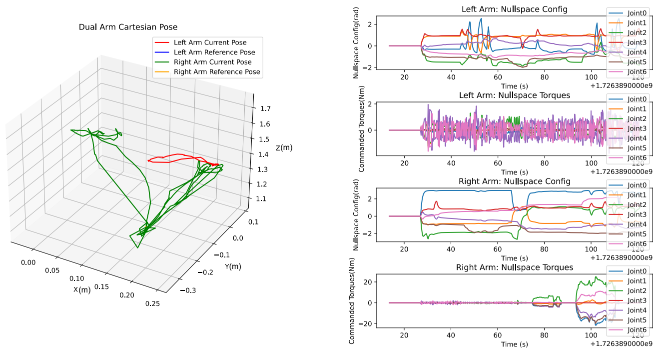



(b) With Optimization

Fig. 5: Visualization of a Trajectory execution with and without Optimization.



(a) Visualization of Nullspace Collision Avoidance in each arm.




(b) Trajectory execution and Optimal Nullspace Configuration.

Fig. 6: Nullspace Collision avoidance with optimization.

  - **Robot Interface Driver** Interfaces with the physical robot hardware or simulation robot, providing real-time interface for feedback and control.
  - **Nullspace Optimization** Handles the optimization of the nullspace configuration to avoid joint limits, singularities, and collisions. It utilizes the Scipy library for optimization and interacts with kinematic and collision cost estimation libraries.
  - **Trajectory Generator/Motion Planner** Generates the Cartesian pose and wrench trajectories required for task execution.

- **RBDL Library:** Used for rigid body dynamics computations required for the Cartesian Impedance Controller.
- **Scipy Library:** Library used for solving the nullspace optimization problem.
- **Kinematic Objectives:** Define the goals for the robot's joint positions and movements within the nullspace.
- **Optimization Problem:** Multi-objective optimization problem incorporating kinematic objectives, joint limit avoidance, collision cost estimation, etc,.
- **Collision Cost Estimation:** Utilizes the FCL, Trimesh, and KDL libraries.
- **FCL, Trimesh, and KDL Libraries:** Provide tools for collision detection and kinematic calculations.
- **Cartesian Impedance Control Law:** The control law is implemented to achieve the desired Cartesian pose and wrench by calculating the necessary joint torques.
- **Robot State:** The current state of the robot, including cartesian positions, joint positions, torques, etc,.
- **Nullspace Configuration:** The optimized configuration of the robot's joints within the nullspace.

## V. Experiments and Results

In this section, we evaluate the performance of the Cartesian Impedance Controller with nullspace optimization. The experiments were conducted on Mujoco simulation.

The results in Fig. 5 clearly demonstrate the improvement in trajectory tracking with nullspace optimization. Without optimization, the robot failed to reach certain points due to an invalid nullspace configuration. Nullspace optimization resolves these issues, allowing the robot to avoid joint limits, singularities, and collisions effectively. The plots on the right side show the real time computation of optimal nullspace configuration and the corresponding torque computed from nullspace joint impedance controller.

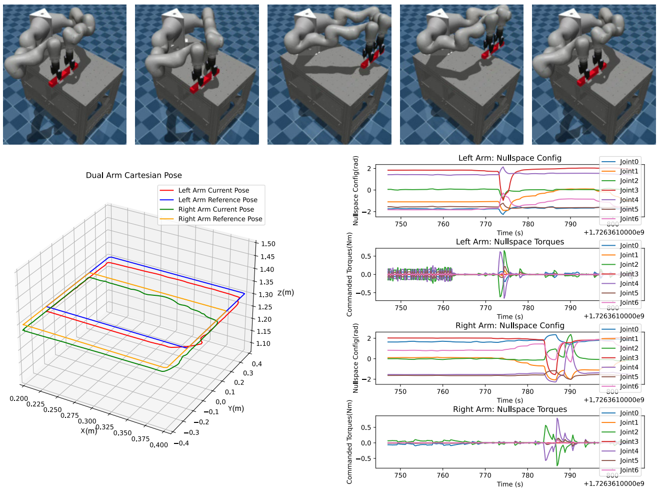



Fig. 7: Visualization of an example Dual-arm coordinated motion execution with Optimization.

Figure 6 demonstrates the proposed nullspace optimization in avoiding inter-arm collisions in nullspace during dual-arm trajectory execution. Subfigure (a) shows the sequence of execution of the robot maintaining safe distances between the arms, while subfigure (b) visualizes the optimized nullspace configurations. The collision avoidance cost dynamically adjusts the nullspace configuration, to maintain safe distances between selected collision geometries without compromising primary task performance.

An example of coordinated dual-arm motion with optimization is visualized in Fig. 7. It shows the effectiveness of the proposed method, where compliance is in place between the arms holding a rectangular bar and maintaining desired end-effector trajectories while optimizing the nullspace configurations.

In our experiment, we used the trajectory to challenge the robot by pushing it near its kinematic reach. However, the optimization worked effectively by dynamically adjusting the nullspace configuration, as shown in Fig. 7. While perfect tracking was not possible due to the limits of robot reach, the optimization prioritized objectives, allowing the robot to attempt trajectory tracking even when encountering singularities and joint limits.

## VI. CONCLUSION

This paper presents an approach for nullspace optimization in Cartesian impedance control of dual-arm robots to achieve ambidextrous capabilities. By employing a scalarized multi-objective function within a Look Ahead Control (LAC) framework, our method addresses the complexities associated with nullspace configurations, including avoiding singularities, maintaining joint limits, minimizing jerk, and preventing nullspace collisions between the arms.

Experimental results validate the effectiveness of our approach. The inclusion of nullspace optimization significantly improves performance such as joint limit avoidance, and manipulability, demonstrating the method's capability for efficient operation. The integration of nullspace collision avoidance ensures that the robot maintains safe distances in nullspace during operation. The collision cost function, embedded within the optimization framework, dynamically penalizes configurations that risk collisions. This mechanism allows the robot to execute trajectories that respect all constraints, maintaining task-space accuracy and compliance while optimizing nullspace configurations.

The dual-arm coordinated motion experiments demonstrated the effectiveness of nullspace optimization, most importantly, the primary compliance behavior of the Cartesian impedance controller is not affected. The robot's compliance in responding to external forces by each other remained unaffected.

Future work will explore the implementation of our method on more complex robotic systems and investigate further enhancements to the optimization framework to accommodate a wider range of tasks and applications. The potential of integrating advanced learning techniques, such as reinforcement learning, to adapt the optimization parameters in real time will also be considered to further improve the robot's adaptability and performance.

## ACKNOWLEDGMENT

This research work was conducted as part of the HARTU Project.This project has received funding from the European Union's research and innovation program Horizon Europe under grant agreement No. 101092100.

## REFERENCES

[1] Hogan, N. (1985). Impedance Control: An Approach to Manipulation. In Proceedings of the American Control Conference.

[2] Khatib, O. (1987). A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation. IEEE Journal on Robotics and Automation.

[3] Siciliano, B., & Khatib, O. (1991). Kinematic Control of Redundant Robot Manipulators. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems.

[4] Origanti, V.K., Eiband, T., & Lee, D. (2022). Automatic Parameterization of Motion and Force Controlled Robot Skills. Robot Intelligence Technology and Applications 6. RiTA 2021. Lecture Notes in Networks and Systems, vol 429. Springer, Cham. https://doi.org/10.1007/978-3-030-97672-9$_7$

[5] Buschman, R., & al. (2011). Quadratic Programming-based Control of Redundant Robots. In Proceedings of the International Conference on Robotics and Automation.

[6] Kuindersma, S., Permenter, F., & Tedrake, R. (2016). An optimization-based approach to whole-body control of humanoid robots. Journal of Field Robotics, 32(2), 305-321.

[7] Kanoun, O., Lamiraux, F., & Wieber, P. B. (2011). Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task. IEEE Transactions on Robotics, 27(4), 785-792.

[8] Zhou, J., & others. (2020). Multi-objective optimization in robotic manipulators. Robotics and Autonomous Systems.

[9] Wu, Y., & others. (2019). Reinforcement learning for impedance control in unstructured environments. International Journal of Robotics Research.

[10] Lei, M., Wang, T., Yao, C., Liu, H., Wang, Z., & Deng, Y. (2020). Real-Time Kinematics-Based Self-Collision Avoidance Algorithm for Dual-Arm Robots. Appl. Sci., 10, 5893. https://doi.org/10.3390/app10175893

[11] Hoffman, E. M., Laurenzi, A., Muratore, L., Tsagarakis, N. G., & Caldwell, D. G. (2018). Multi-Priority Cartesian Impedance Control Based on Quadratic Programming Optimization. 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, pp. 309-315, doi: 10.1109/ICRA.2018.8462877.

[12] Mansfeld, N., Michel, Y., Bruckmann, T., & Haddadin, S. (2019). Improving the Performance of Auxiliary Null Space Tasks via Time Scaling-Based Relaxation of the Primary Task. 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, pp. 9342-9348, doi: 10.1109/ICRA.2019.8794225.

[13] C. Ott (2008), Cartesian Impedance Control of Redundant and Flexible-Joint Robots, Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-69255-3.

[14] Yoshikawa, T. Manipulability of Robotic Mechanisms. *The International Journal Of Robotics Research*. **4**, 3-9 (1985)

[15] Tsai, M. & Morgan, A. Solving the Kinematics of the Most General Six- and Five-Degree-of-Freedom Manipulators by Continuation Methods. *Journal Of Mechanical Design*. **109**, 191-199 (1987)

[16] Ang, W., Khosla, P. & Kanade, T. Determination of Manipulator Attributes using a Weighted Jacobian. *Proceedings Of IEEE International Conference On Robotics And Automation*. pp. 414-420 (1994)

[17] Flacco, F., De Luca, A. & Khatib, O. Prioritized Multi-Task Motion Control of Redundant Robots Under Hard Joint Constraints. *Proceedings Of The IEEE International Conference On Intelligent Robots And Systems (IROS)*. pp. 3970-3977 (2012)

[18] Lee, J. Representing rotations and orientations in geometric computing. *IEEE Computer Graphics And Applications*, **28**(2), 75-83 (2008).

[19] Huynh, D.Q. Metrics for 3D Rotations: Comparison and Analysis. *Journal Of Mathematical Imaging And Vision*, **35**, 155-164 (2009). https://doi.org/10.1007/s10851-009-0161-2.

[20] J. Pan, S. Chitta and D. Manocha, "FCL: A general purpose library for collision and proximity queries," 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 2012, pp. 3859-3866, doi: 10.1109/ICRA.2012.6225337.

[21] Dawson-Haggerty et al., "trimesh," Version 3.2.0, 2019. [Online]. Available: `https://trimesh.org/`. [Accessed: 08-Dec-2019].

[22] M. Mayr and J. M. Salt-Ducaju, "A C++ Implementation of a Cartesian Impedance Controller for Robotic Manipulators," arXiv preprint arXiv:2212.11215, 2022. [Online]. Available: `https://arxiv.org/abs/2212.11215`.

[23] P. Virtanen et al., "Fundamental Algorithms for Scientific Computing in Python," Nature Methods, vol. 17, no. 3, pp. 261-272, 2020.

[24] S. Macenski et al., "Robot Operating System 2: Design, architecture, and uses in the wild," Science Robotics, vol. 7, no. 66, pp. eabm6074, 2022, doi: 10.1126/scirobotics.abm6074.

[25] N. Thelenberg and C. Ott, "On Handling Variable Stiffness Parameters in Compliance Control via MPC," 2024 European Control Conference (ECC), Stockholm, Sweden, 2024, pp. 615-620, doi: 10.23919/ECC64448.2024.10591005.

[26] Anand, A.S., Abu-Dakka, F.J. and Gravdahl, J.T., 2022. Deep Model Predictive Variable Impedance Control. arXiv preprint arXiv:2209.09614.