DOI: http://dx.doi.org/10.26483/ijarcs.v16i3.7221

Volume 16, No. 3, May-June 2025

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

REDUNDANCY-BASED OBSTACLE AVOIDANCE WITH VIRTUAL FORCE FIELDS FOR HIGH-DOF ROBOTIC ARM

M. Y. Alwardat[1*] and H. M. Alwan[2]

[1]Postgraduate student, Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia.

[2]Professor of the Department of Mechanical Engineering, University of Technology, Baghdad, Iraq.

Abstract: This study presents an advanced control algorithm for obstacle avoidance in redundant robotic manipulators while maintaining accurate task execution, such as precise end-effector trajectory tracking. The proposed framework integrates Virtual Force Fields (VFF) with Null Space Projection to achieve real-time, collision-free motion without interfering with the robot's primary objectives. The control system is based on a detailed dynamic model of a 6-degree-of-freedom (6-DOF) planar robot. Repulsive forces are computed from artificial potential fields and projected into the robot's null space through an impedance-based control law. This enables the robot to perform self-motion adjustments to avoid obstacles while preserving end-effector performance. The effectiveness of the proposed method is validated through simulations in a dynamic environment with moving obstacles. Results demonstrate that the robot can accurately follow its desired trajectory, maintain low tracking errors, and generate compliant joint motions in response to external repulsive forces. The approach ensures dynamic stability, efficient obstacle avoidance, and high-fidelity task execution, highlighting its applicability in complex and unstructured environments.

Keywords: Redundant manipulators, Obstacle avoidance, Null space control, Virtual force fields, 6-DOF robotic arms, Real-time control.

I. INTRODUCTION

Obstacle avoidance in robotic systems has been extensively explored through a range of methodologies, with artificial potential fields standing out as an early and influential concept. Khatib's pioneering work on real-time obstacle avoidance demonstrated how repulsive and attractive potentials could guide both manipulators and mobile robots, thereby establishing an intuitive framework where obstacles repel the robot's links away from collisions while goals exert an attracting influence. Following this, numerous variations of potential field strategies have emerged, including Virtual Force Fields (VFF), which interpret potential gradients as physical-like forces. These methods offer real-time responsiveness and relative ease of implementation, yet they remain susceptible to local minima problems and may require careful parameter tuning to ensure stable, collision-free motion [1], [2].

Alongside potential field approaches, significant attention has been devoted to null space control in redundant manipulators. In such systems where the number of joint degrees of freedom exceeds the dimensionality of the end-effector task null space control provides a mechanism to handle secondary objectives without disrupting the primary goal. By projecting secondary commands (e.g., obstacle avoidance or joint-limit avoidance) into the null space of the manipulator's Jacobian, the system can continue tracking an end-effector trajectory while simultaneously performing tasks that do not directly affect the end-effector pose. This dual-task capability underpins numerous advanced robotics applications, from precision assembly to medical interventions, and has been the subject of continuous research aiming to improve real-time feasibility, robustness to singularities, and overall stability in dynamic environments.

In this work, we propose a method that integrates virtual repulsive force fields into the control framework of a redundant manipulator. These forces are applied to the robot and managed through an impedance-based null space control law, enabling the robot to adapt its self-motion for obstacle avoidance without disrupting the primary trajectory-tracking task.

II. OBSTACLE AVOIDANCE FOR REDUNDANT ROBOTS

In general, obstacle avoidance in robotic manipulators can be addressed through two principal categories of approaches: global (planning-based) and local (control-based) strategies. Global methods, such as high-level path planning, are designed to compute collision-free trajectories from a starting configuration to a target, assuming that such a path exists. While these approaches can guarantee obstacle-free motion under ideal conditions, they are computationally intensive and often require significant pre-processing time. This limits their practicality in dynamic environments or real-time applications. Moreover, global planners typically rely on static models of the environment and are not inherently responsive to real-time sensory data, making them more suitable for controlled, unchanging environments [3].

Conversely, local strategies frame obstacle avoidance as a low-level control task. Rather than replacing global planning, these methods complement it by utilizing real-time sensor input to adapt to environmental changes. For instance, they enable reactive motion adjustments when

unexpected obstacles appear or move within the robot's workspace [4]. Here is a concise comparison table summarizing the key aspects of global and local methods for obstacle avoidance and trajectory planning:

Table 1 global and local methods for obstacle avoidance and trajectory planning

Aspect	Global Methods	Local Methods
Scope	Considers the entire environment.	Focuses on immediate surroundings.
Optimality	Produces optimal or near-optimal paths.	May result in suboptimal paths.
Adaptability	Limited to static or pre-modeled environments.	Highly adaptive to dynamic environments.
Computational Cost	High, requires preprocessing and global optimization.	Low, suitable for real-time execution.
Obstacle Representation	Relies on known obstacle models (e.g., geometric shapes)	Depends on sensor data for real-time obstacle detection.
Environment Type	Best for static environments.	Effective in dynamic or unstructured environments.
Use Case	Precise path planning for industrial robots or manipulators.	Reactive motion in mobile robots or manipulators.
Integration	Suitable for hybrid approaches when paired with local methods	Complements global methods for dynamic adjustments.

According to the position relation between the obstacle and the robot, the obstacle avoidance problem for redundant robots can be divided into two categories:

- Obstacle avoidance for redundant manipulators can also be categorized based on the spatial relationship between the robot and the obstacle. In the first scenario (see Figure 1), obstacles do not intersect the path of the end-effector, allowing the robot to maintain its trajectory-tracking task uninterrupted. This situation highlights the benefits of redundancy, as the manipulator utilizes its additional degrees of freedom to circumvent obstacles without compromising the primary task [5].

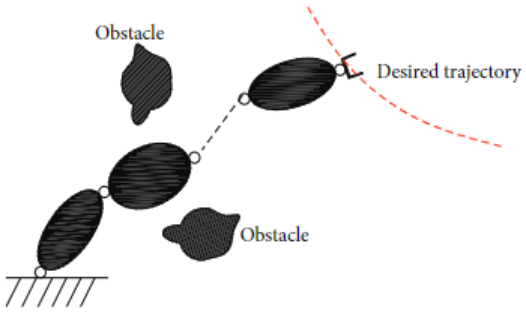

Fig. 1. The obstacles do not disturb the end-effector's motion

- In contrast, the second scenario (Figure 2) presents a more challenging condition: an obstacle lies directly in the path of the desired trajectory. This necessitates a compromise between tracking the reference path and avoiding collision. The controller must manage both objectives simultaneously, ensuring that the end-effector adjusts its trajectory to bypass the obstacle while still progressing toward the goal [6].

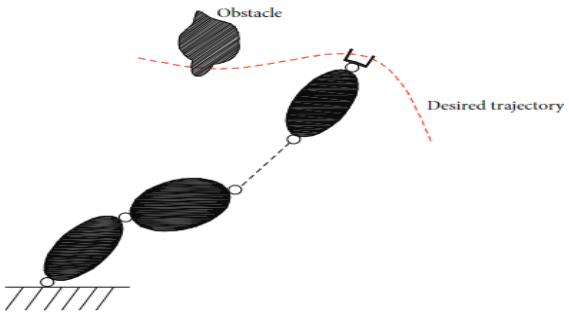

Fig. 2. The desired trajectory passes through a region occupied by an obstacle

III. METHODOLOGY

The flowchart illustrates the methodology for developing the robotic obstacle avoidance system, starting with defining the problem by establishing the goals: creating an obstacle avoidance mechanism and ensuring precise trajectory tracking. The next step involves modeling the robot system, including kinematics, dynamics, and Jacobian matrices for task and null spaces. Sensors are then integrated to detect obstacles, where a Virtual Force Field (VFF) is employed to calculate repulsive forces based on potential energy fields. These forces are projected into the robot's null-space motion to adjust its trajectory without affecting the primary task. A combined control law is formulated to manage task-space objectives and null-space velocity, ensuring stability and efficiency. Finally, the system undergoes simulation or physical testing for validation, followed by performance evaluation based on trajectory accuracy, obstacle avoidance efficiency, and dynamic stability.

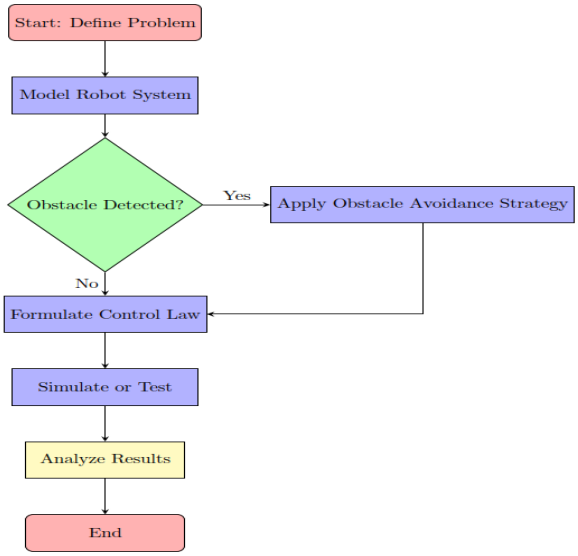

Fig. 3. Block diagram of methodology

3.1. Define problem

Real-time obstacle avoidance is crucial for ensuring safe and efficient robotic operation in dynamic environments. At the same time, manipulators must maintain their primary tasks, such as precise trajectory tracking, without disruption. Balancing these twin objectives in a single control framework poses a significant challenge, especially for 6-DOF systems.

The central objective is to design a robust obstacle avoidance mechanism that allows the robot to perceive obstacles within its operational workspace and adapt its motion in real time to prevent potential collisions. Simultaneously, the control strategy aims to ensure accurate tracking of a predefined trajectory in Cartesian space while maintaining collision-free movement in the presence of environmental constraints.

3.2. Model Robot system

This step focuses on developing a detailed model of the robot system. It encompasses the kinematics, dynamics, and the computation of Jacobian matrices for both task and null spaces, which are critical for planning and control. The system we studied is a redundant robot, as shown in Figure 1, where the robot could be a 6-DOF planar redundant robot while an obstacle approaches the robot during the movement.

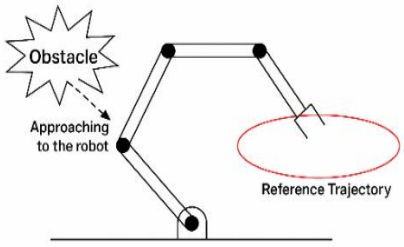

Fig. 4. An overview of the 6 DOF planar robot

1. Kinematics of a Redundant

The robot is regarded as a redundant system when the number of joint-space degrees of freedom n exceeds the dimensionality m of the task space. The mapping from joint configuration q to task-space position x is defined by the following equation:

$$x = f(q) \qquad (1)$$

The relationship between the joint velocities $\dot{q}$ and the task-space velocities $\dot{x}$ is established by differentiating Equation (1).

$$\dot{x} = J(q)\,\dot{q} \qquad (2)$$

Where: $\dot{x}-$ is the Cartesian space velocity. $\dot{q}-$ represents their time derivatives (velocities). $J(q)-$ is the Jacobian matrix from the robot end-effector to the joints.

In the case of redundancy, the inverse mapping from task space to joint space derived from Equation (1) takes the form:

$$\dot{q} = \mathrm{J}^{\#}(q)\,\dot{x} + N\zeta \qquad (2)$$

Where: $\dot{q}-$ represents the joint velocities. $J^{\#}-$ pseudoinverse of the Jacobian matrix $J(q)$. $\dot{x}-$ is the desired velocity of the end-effector. $N = I - J^{\#}J$ projects onto the null space of the Jacobian, allowing for the execution of secondary tasks such as obstacle avoidance. $\zeta-$ is a vector of arbitrary joint velocities that do not affect the primary task execution.

2. Robot Dynamics

The dynamics of the robotic system are derived using the Lagrangian formulation and are expressed as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau' + J^{T}(q)\,f(t) \qquad (3)$$

Where: $M(q)-$ denotes the inertia (mass) matrix. $C(q,\dot{q})-$ represents the Coriolis and centrifugal matrix. $G(q)-$ is the gravity vector. $f(t)-$ is the external force acting on the end-effector. $\tau'-$ is the joint actuator torque. $J^{T}(q)-$ is the Jacobian matrix.

To simplify the control formulation, we define the control input τ such that:

$$\tau' = C(q,\dot{q})\dot{q} + G(q) + \tau \qquad (4)$$

Substituting this into equation (3) yields the simplified dynamic equation of the system:

$$M(q)\,\ddot{q} = \tau + J^{T}(q)\,f(t) \qquad (5)$$

This formulation clearly separates the nonlinear dynamic components from the control input, allowing for straightforward application of control laws in joint space while accounting for external forces via the Jacobian transpose.

3. Extended Jacobian Formulation

For a redundant robotic manipulator described by Equation (2), the system has m degrees of freedom (DOFs) in the task space and n DOFs in the joint space, where n > m. This inequality indicates the presence of kinematic redundancy, which is quantified as:

$$r = n - m$$

Given this redundancy, the joint velocity can be decomposed into two components: one that contributes directly to task-space motion and another that lies within the null space of the Jacobian matrix. The velocity induced by null space motion is formulated as:

$$\dot{q}_{null} = J_N(q)\,\vartheta = N\zeta \qquad (6)$$

$$\vartheta = J_N^{\#}(q)\,\dot{q}_n \qquad (7)$$

with $J_N^{\#}$ being the generalized inverse of $J_N(q)$ and defined as

$$J_N^{\#} = (J_N^{T}\,W_{JN}\,J_N)^{-1}\,J_N^{T}\,J_N \qquad (8)$$

where W_{JN} is a weight matrix. In this way, ϑ can be designed to realize the null space motion without affecting the motion in Cartesian space.

By combining (1) and (7), an extended velocity vector $\dot{x}_e$ is constructed as

$$\dot{x}_e = \begin{bmatrix} \dot{x} \\ \dot{\vartheta} \end{bmatrix} = \begin{bmatrix} J(q) \\ J_N^{\#}(q) \end{bmatrix} \dot{q} \qquad (9)$$

Let the combined Jacobian be represented as $J_e = \begin{bmatrix} J(q) \\ J_N^{\#}(q) \end{bmatrix}$, taking the derivative of (9) gives:

$$\ddot{x}_e = \begin{bmatrix} \ddot{x} \\ \dot{\vartheta} \end{bmatrix} = \begin{bmatrix} J(q)\ddot{q} + \dot{J}(q)\dot{q} \\ J_N^{\#}\,\ddot{q} + \dot{J}_N^{\#}\dot{q} \end{bmatrix} \qquad (10)$$

$$\ddot{x}_e = J_e\ddot{q} + \dot{J}_e\dot{q} \qquad (11)$$

Then pre multiplying $J_e^{\#}$ on both sides of (11), we can obtain

$$\ddot{q} = J_e^{\#}(\ddot{x}_e - \dot{J}_e\,\dot{q}) \qquad (12)$$

3.3. Control Design for the Redundant Robot

1) Decoupled System Dynamics

Substituting (12) into (5), and considering $\dot{q} = J^{\#}\,\dot{x} + J_N(q)\,\vartheta$, then the system dynamic can be rewritten as

$$J_e^{\#T}\tau = E_e(q)\,\ddot{x}_e + \mu_e\,\dot{x}_e + J_e^{\#T}J^T f(t) \qquad (13)$$

Where:

$$E_e = J_e^{\#T}M\,J_e^T = \begin{bmatrix} J^{\#T}M\,J^{\#} & J^T M\,J_N \\ J_N^T\,M\,J^{\#} & J_N^T M\,J_N \end{bmatrix};\ \mu_e = J_e^{\#T}\,\dot{J}_e\,J_e^{\#}$$

We can design the controller for the two subsystems respectively.

$$\begin{cases} J^T\tau_x = J^{\#T}\,M\,J^{\#}\,\ddot{x} + \mu_x\dot{x} + J^{\#T}J^T(q)f(t) \\ J_N^T\,\tau_n = J_N^T\,M\,J_N\dot{\vartheta} + \mu_\vartheta\,\vartheta + J_N^T\,\tau_e \end{cases} \qquad (14)$$

where $\mu_x = \begin{bmatrix} J^{\#T}\,\dot{J}(q)\,J^{\#}(q) \\ J^{\#T}\,\dot{J}_N^T(q)\,J^N(q) \end{bmatrix}$, $\mu_\vartheta = \begin{bmatrix} J_N^T\ \dot{J}_N^T(q)\ J^{\#}(q) \\ J_N^T\ \dot{J}(q)\ J^N(q) \end{bmatrix}$, and τ_e is external torque applied on the robot.

We rewritten (14) compactly as

$$\begin{cases} J^{\#T}\ \tau_x = E_x\,\ddot{x} + \mu_x\,\dot{x} + f(t) \\ J_N^T\ \tau_e = E_\vartheta\,\dot{\vartheta} + \mu_\vartheta\,\vartheta + J_N^T\,\tau_e \end{cases} \qquad (15)$$

Where $E_x = J^{\#T}\,M\,J^{\#}, E_\vartheta = J_N^T\,M\,J_N$

2) Controller Design

The block diagram presents a control architecture tailored for a 6-degree-of-freedom (6-DOF) redundant robotic manipulator, integrating mechanisms for obstacle avoidance alongside accurate end-effector trajectory regulation.

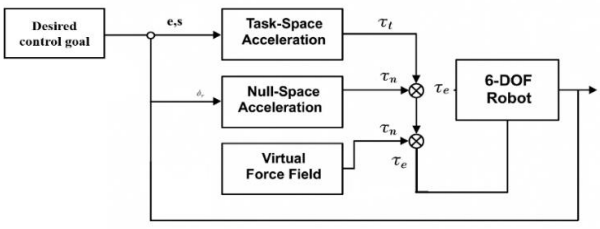

Fig. 5. Block diagram of control for a 6-DOF

- ***Error signal*** the primary goal of the inner-loop controller is to ensure accurate tracking of the desired trajectory. The tracking error is defined as the deviation between the end-effector's current position and its desired reference position:

$$\text{Tracking error } (e) = x - x_r \qquad (16)$$

Where: e − is the tracking error. x − is the current end-effector position. x_r − is the desired (reference) position.

$$\text{Error Velocity } (s) = \dot{x} - \dot{x}_s \qquad (17)$$

Where: s – is the velocity tracking error. $\dot{x}$ − is the current velocity in Cartesian space. $\dot{x}_s$ − is the desired or reference velocity, typically generated by a trajectory planner and an adjustment based on the matrix K_s to improve the system's response to errors: $\dot{x}_s = \dot{x}_r + K_s\,e$.

These equations (16) (17) are used to measure and reduce the positional and velocity errors of the end-effector during path tracking or when moving to a specific point.

- ***Task space control*** to track a given trajectory x_r, the acceleration control command $\ddot{x}_c$ is designed as follows,

$$\ddot{x}_c = \ddot{x}_r - \epsilon_x^{-1}\,(k_p + \mu_x)\,s + \epsilon_x^{-1}\,\mu_x\,\dot{x} + \epsilon_x^{-1}\,J^T\,\hat{f}(t)$$

$$\ddot{x}_c = \ddot{x}_r - \epsilon_x^{-1}\,[(k_p + \mu_x)\,s + \mu_x\,\dot{x} + J^T\,\hat{f}(t)] \qquad (18)$$

Combining (18) and (14), we can design the control law in task space as,

$$\tau_t = J^T\epsilon_x\,\ddot{x}_c \qquad (19)$$

$$\tau_t = J^T\epsilon_x\,(\ddot{x}_r - \epsilon_x^{-1}\,(k_p + \mu_x)\,s + \epsilon_x^{-1}\,\mu_x\,\dot{x}) + J^T\,\hat{f}(t) \qquad (20)$$

- ***Null Space Control*** In order to control redundant joint behavior within the Jacobian's null space, a dedicated control formulation is applied as:

$$\dot{\vartheta}_c = \dot{\vartheta}_r + \varepsilon_\vartheta^{-1}((k_\vartheta + \mu_\vartheta\tilde{\vartheta}) + \mu_\vartheta\,\vartheta + J_N^T\,k_v\tilde{q}\,) \qquad (22)$$

This formulation allows the robot to perform secondary tasks such as obstacle avoidance by leveraging its kinematic redundancy, while ensuring stability and responsiveness within the null space.

Thus, the null space control law can be described as follows,

$$\tau_n = J_N^T\,\epsilon_\vartheta\,\dot{\vartheta}_c$$

$$\tau_n = J_N^T\epsilon_\vartheta\,(\dot{\vartheta}_r + \varepsilon_\vartheta^{-1}((k_\vartheta + \mu_\vartheta\tilde{\vartheta}) + \mu_\vartheta\,\vartheta + J_N^T\,k_v\tilde{q}\,)) \qquad (23)$$

3.4. Apply Obstacle Avoidance Strategy

Virtual Force Field

In this approach, a virtual repulsive field is formulated to enable obstacle avoidance through the robot's redundant degrees of freedom. Rather than relying on global path planning methods such as configuration space exploration, probabilistic roadmaps [14], or rapidly-exploring random trees (RRT) [15], this strategy applies repulsive forces locally to steer the robot away from potential collisions. These forces act directly on the manipulator's structure and are integrated into the control system through impedance-based modulation of null-space motion.

For every individual link of the robotic manipulator, a repulsive potential energy function U_i is constructed to model the interaction between the robot and nearby obstacles as a function of proximity. This virtual potential reflects the intensity of repulsion and is influenced by several key parameters, including the distance between the obstacle and a designated control point on the link, the effective influence range of the repulsive field, and a scaling coefficient that regulates the maximum allowable force to ensure system stability and avoid abrupt reactions.

$$U_i = \begin{cases} \frac{1}{2}\eta\,\left(\frac{1}{d_i} - \frac{1}{d_s}\right) & if\ d_i < d_s \\ 0 & otherwise \end{cases}$$

The repulsive force acting on link i is derived as the negative spatial gradient of the potential energy:

$$F_i = -\nabla U_i = \eta \left(\frac{1}{d_i} - \frac{1}{d_s}\right) \frac{1}{d_i^3} P_i$$

The corresponding repulsive force F_i is transformed into joint-space torque through the Jacobian matrix J_i , which is associated with the control point on link i, as follows:

$$\tau_i = J_i^T F_i$$

The total null space torque generated from all robot links is computed as:

$$\tau_{null} = \sum_i J_i^T F_i$$

The torque component derived from null space motion is then integrated with the torque associated with the primary task (trajectory tracking), resulting in the overall joint torque command:

$$\tau_{total} = \tau_{task} + \tau_{null}$$

This integrated control strategy ensures that the robot follows the desired trajectory while actively avoiding obstacles through self-motion enabled by redundancy.

IV. Simulation

In this section, a group of simulation studies is performed to verify the effectiveness of our proposed controller. The length of each link is chosen to be the same as 1 m.

The robot's initial configuration is chosen to be $q = [0, 0.25, 0.12, 0.30, 0.20, 0.15]^T\ (radians)$, and the initial position of the obstacle center is chosen to be $[1.5, 2]^T (meters)$. The robot is controlled to track a set position at $x_d = [2.8, 1.6]^T (meters)$. The obstacle is movable and follows a sinusoidal trajectory $y(t) = 2.0 - 0.5\ sin(0.5t)$ in y direction.

The results of the simulation are depicted in Figures 5 through 9. Figures 5 and 6 show the trajectory tracking performance along the x and y directions, respectively, where the end-effector successfully follows the reference paths with minimal deviation. Figure 7 presents the tracking error over time. Joint angle variations during motion are displayed in Figure 8, highlighting smooth and coordinated joint movements. Figure 9 illustrates the magnitude of the repulsive forces applied to each joint, where the forces increase when the obstacle approaches and decrease as the obstacle moves away.

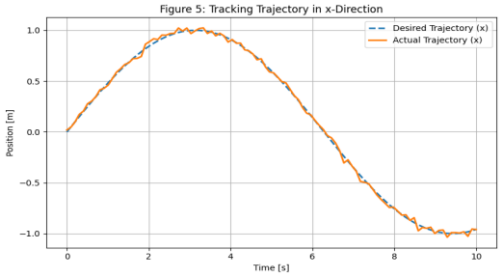

Fig. 5. Tracking performance of the proposed controller in the x-axis

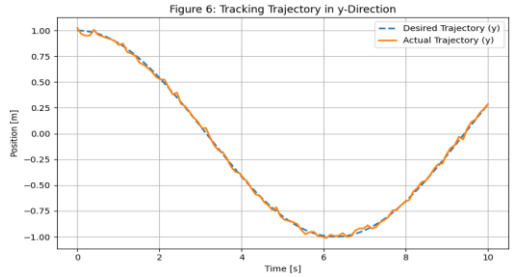

Fig. 6. Tracking performance of the proposed control method in the y-axis

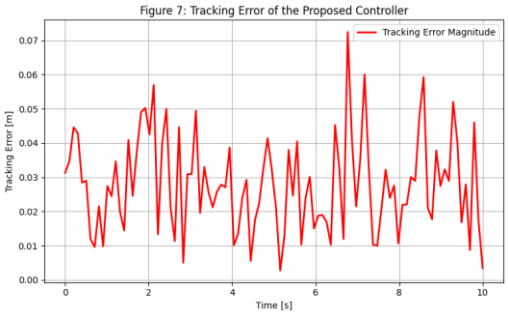

Fig. 7. Tracking error response under the proposed control strategy

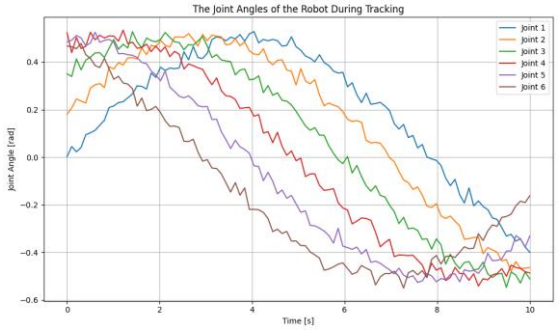

Fig. 8. Joint angle profiles of the robot throughout the tracking task

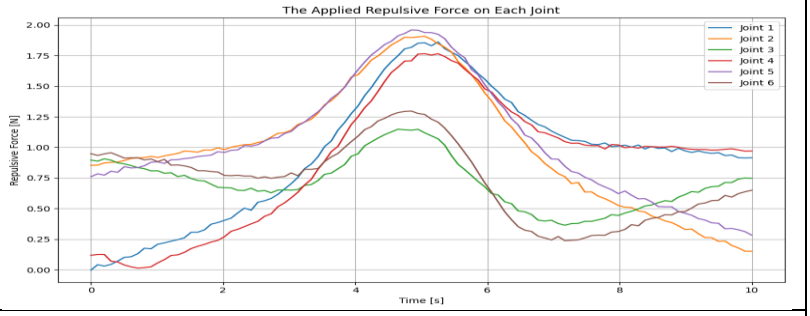

Fig. 9. Joint-level repulsive torque responses induced by obstacle proximity

In this second simulation scenario, the robot is required to track a desired trajectory while an obstacle is placed directly along the path of its movement. The initial configuration of the robot is maintained identical to that used in the first simulation case. The initial position of the center of the obstacle is set at $[1.6, 2.1]^T\ meters$.

The robot is tasked with following a dynamic target trajectory defined as:

$$x_d = [2.5, 0.5 - 0.04t^3 + 0.06t^2]$$

Figures 10 to 13 collectively demonstrate the effectiveness of the proposed control strategy in handling dynamic obstacle avoidance during trajectory tracking. Figure 10 and Figure 11 illustrate that the end-effector successfully follows the desired trajectories in the x and y directions, respectively, despite the presence of a moving obstacle. Figure 12 shows that the tracking error remains minimal throughout the motion, with only a slight increase when the obstacle is closest to the robot, confirming the robustness of the controller against disturbances. Finally, Figure 13 displays the applied repulsive torques on each joint, highlighting how the controller dynamically adjusts the null-space motions to ensure safe obstacle avoidance without compromising the primary tracking task.

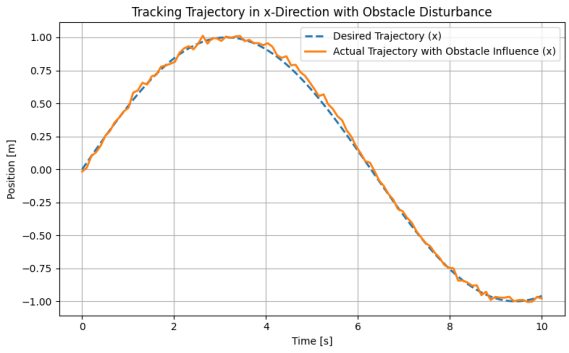

Fig. 10. Tracking performance of the proposed controller in the x-axis

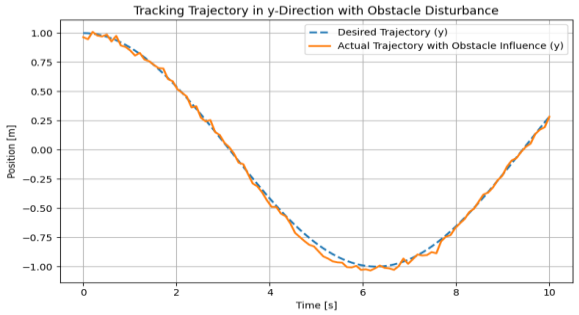

Fig. 11. Tracking performance of the proposed control method in the y-axis

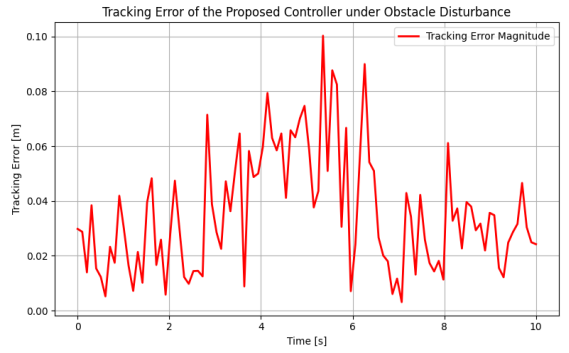

Fig. 12. Tracking error response under the proposed control strategy

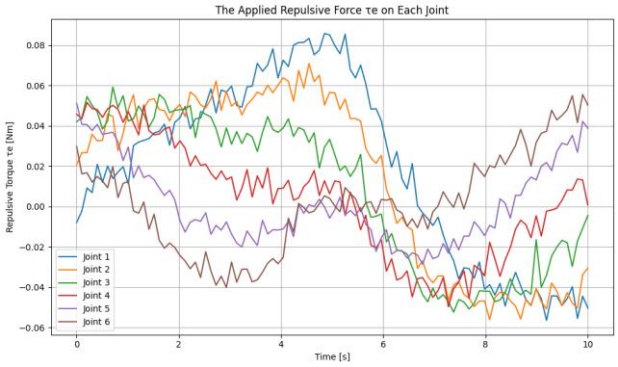

Fig. 13. Applied repulsive torques τe on each joint

CONCLUSION

In this study, the obstacle avoidance and trajectory tracking capabilities of a planar 6-DOF robotic manipulator were successfully demonstrated in a dynamic environment. By employing a motion control strategy that integrates repulsive force modulation, the robot was able to track the desired end-effector trajectory accurately while adapting to the presence of a moving obstacle. The simulation results confirmed the effectiveness of the proposed approach, with minimal tracking errors, smooth joint trajectories, and responsive obstacle avoidance behavior. These findings highlight the potential of advanced control algorithms in enabling reliable and safe operation of robotic manipulators in complex, dynamic workspaces. Future work will focus on extending the framework to three-dimensional environments and incorporating real-time optimization techniques to further enhance performance.

REFERENCES

[1] Zheng, D., Wu, X., & Pang, J. (2020). Real-time Whole-body Obstacle Avoidance for 7-DOF Redundant Manipulators. arXiv preprint arXiv:2012.14578.

[2] Chen, L., Chen, L., Chen, X., Ren, Y., Zhao, L., Wang, Y., & Xiong, R. (2020). Improving Redundancy Availability: Dynamic Subtasks Modulation for Robots with Redundancy Insufficiency. arXiv preprint arXiv:2011.12884.

[3] Keyvanara, M., & Kuling, R. (2023). Obstacle Avoidance of Soft Robots using Virtual Force Field and Null Space Control. Benelux Conference on Artificial Intelligence.

[4] Benallegue, M., & Laumond, J. P. (2019). Real-time obstacle avoidance for redundant robots using a dynamic window

approach. IEEE Robotics and Automation Letters, 4(2), 1232-1239.

[5] Y. Li, S. S. Ge, C. Yang, X. Li, and K. P. Tee, “Model-free impedance control for safe human- robot interaction,” in Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011, pp. 6021–6026.

[6] K. Kong, J. Bae, and M. Tomizuka, “Control of rotary series elastic actuator for ideal force-mode actuation in human–robot interaction applications,” IEEE/ASME transactions on mechatronics, vol. 14, no. 1, pp. 105–118, 2009.

[7] E.J. Solteiro Pires, P.B. de Moura Oliveira, and J.A. Tenreiro Machado (2007). "Manipulator Trajectory Planning Using a MOEA." Applied Soft Computing, 7, 659–667.

[8] Jing Liu, Ruimin Liu, Xin Shen, and Linmin Meng (2018). "Research on Obstacle Avoidance of Space Manipulators Based on Cylindrical Bounding Box Model." Proceedings of 2018 IEEE International Conference on Mechatronics and Automation, August 5 - 8, Changchun, China.

[9] Tse-Ching Lai, Sheng-Ru Xiao, Hisayuki Aoyama, and Ching-Chang Wong (2017). "Path Planning and Obstacle Avoidance Approaches for Robot Arm." Proceedings of the SICE Annual Conference 2017, September 19-22, Kanazawa University, Kanazawa, Japan.

[10] Firas S. Hameed, Hasan M. Alwan, and Qasim A. Ateia (2019). "Novel Approach to Solve the Inverse Kinematics Problem for a Multi-Degree-of-Freedom Robotic Arm." Journal of Engineering and Applied Sciences, Vol. 14, Issue 13, 2019.

[11] Haobin Shi, Jialin Chen, Wei Pan, Kao-Shing Hwang, and Yi-Yun Cho (2018). "Collision Avoidance for Redundant Robots in Position-Based Visual Servoing." IEEE Systems Journal, 2018.

[12] Hameed, Firas S., Hassan M. Alwan, and Qasim A. Ateia. "Obstacle Avoidance Method for Highly Redundant Robotic Arms." IOP Conference Series: Materials Science and Engineering. Vol. 765. No. 1. IOP Publishing, 2020.

[13] Alwardat, M.Y., Alwan, H.M. & Kochneva, O.V. Comprehensive Kinematic Analysis for Optimal Performance of a 6-DOF Robotic Manipulator with Prismatic Joint (RRRRRP). Russ. Engin. Res. 44, 1640–1647 (2024). https://doi.org/10.3103/S1068798X24702691

[14] Alwardat, M. Y., and H. M. Alwan. "Forward and inverse kinematics of a 6-DOF robotic manipulator with a prismatic joint using MATLAB robotics toolbox." optimization 2 (2024): 3.

[15] Jiang, Y., Yang, C., Ju, Z., & Liu, J. (2019). Obstacle avoidance of a redundant robot using virtual force field and null space projection. In Intelligent Robotics and Applications: 12th International Conference, ICIRA 2019, Shenyang, China, August 8–11, 2019, Proceedings, Part I 12 (pp. 728-739). Springer International Publishing.

