Revista Facultad de Ingeniería (Rev. Fac. Ing.) Vol. 33, No. 69.
L-ISSN: 0121-1129, e-ISSN: 2357-5328.

DOI: 10.19503/01211129.v33.n69.2024.18055

INVERSE KINEMATICS FOR SYNCHRONIZATION OF THREE DEGREES OF FREEDOM ROBOTS: TECHNIQUES AND APPLICATIONS

Cinemática inversa para la sincronización de robots con tres grados de libertad: Técnicas y aplicaciones

Griselle Salazar
Universidad Autónoma de Chile (Santiago, Chile).
griselle.salazar@cloud.uautonoma.cl

Oscar Loyola
Ph.D. Autónoma de Chile (Santiago, Chile).
oscar.loyola@uautonoma.cl

Eduardo Villaroel
Ph.D. Universidad Autónoma de Chile (Santiago, Chile).
eduardo.villaroel@uautonoma.cl

Claudia Sandoval-Yañez
Ph.D. Universidad Autónoma de Chile (Santiago, Chile).
claudia.sandoval@uautonoma.cl

Received / Recibido: 13-06-2024
Accepted / Aceptado: 29-09-2024

ABSTRACT

The coordination and synchronization of two physical robots with two degrees of freedom (3-DOF) is a critical challenge in collaborative robotics, particularly in applications where precise, simultaneous movement is required. This paper addresses the limitations of traditional inverse kinematics (IK) methods, which, while effective in controlled environments, lack flexibility and adaptability in dynamic settings. We propose a hybrid approach combining IK with Model Predictive Control (MPC) to improve synchronization and trajectory accuracy in a dynamic environment. Our methodology involves analyzing the performance of both elbow-up and elbow-down configurations in terms of synchronization error, trajectory deviation, and arrival times. The results demonstrate that the elbow-up configuration, particularly when enhanced with MPC, provides superior synchronization and reduced trajectory error, making it a preferable option for complex, coordinated tasks in robotics. This study contributes to the ongoing development of adaptive, robust synchronization techniques for multi-robot systems, with implications for various industrial and research applications.

Keywords: inverse kinematics; MPC; robot manipulator.

Esta edición se financió con recursos del Patrimonio Autónomo Fondo Nacional de Financiamiento para la Ciencia, la Tecnología y la Innovación, Francisco José de Caldas, Minciencias

How to cite: G. Salazar, O. Loyola, E. Villaroel, C. Sandoval-Yañez, "Inverse Kinematics for Synchronization of Three Degrees of Freedom Robots: Techniques and Applications", *Revista Facultad de Ingeniería*, vol. 33, no. 69, e18055, 2024. https://doi.org/10.19053/01211129.v33.n69.2024.18055

RESUMEN

La coordinación y sincronización de dos robots físicos con dos grados de libertad (3-DOF) es un desafío crítico en la robótica colaborativa, especialmente en aplicaciones donde se requiere un movimiento preciso y simultáneo. Este artículo aborda las limitaciones de los métodos tradicionales de cinemática inversa (IK), que, si bien son efectivos en entornos controlados, carecen de flexibilidad y adaptabilidad en contextos dinámicos. Proponemos un enfoque híbrido que combina IK con control predictivo basado en modelo (MPC) para mejorar la sincronización y la precisión de la trayectoria en un entorno dinámico. Nuestra metodología implica analizar el rendimiento de las configuraciones "elbow-up" y "elbow-down" en términos de error de sincronización, desviación de trayectoria y tiempos de llegada. Los resultados demuestran que la configuración "elbow-up", particularmente cuando se mejora con MPC, proporciona una mejor sincronización y reduce el error de trayectoria, lo que la convierte en una opción preferible para tareas complejas y coordinadas en robótica. Este estudio contribuye al desarrollo continuo de técnicas de sincronización adaptativas y robustas para sistemas multirobot, con implicaciones en diversas aplicaciones industriales y de investigación.

Palabras clave: Cinemática inversa; manipulador robótico; MPC.

1. INTRODUCTION

Synchronized coordination between two physical robots is a fundamental aspect of various applications in modern robotics, e.g., manufacturing, assembly, and other critical areas of collaborative robotics [1]-[2]. Synchronization capability not only involves robots moving simultaneously but also requires these movements to be precise and coordinated, so that robots are able to complete complex tasks efficiently and without interference. Inverse kinematics (IK) plays a crucial role in this synchronization process because it enables the determination of the joint configurations necessary to reach the desired positions and orientations of the end effectors [3]. However, traditional IK methods, while effective in controlled scenarios, often present significant limitations in terms of flexibility and adaptability, especially when faced with dynamic environments where conditions can change unpredictably.

Inverse kinematics has been an essential component in robotics since its inception because it provides a robust mathematical framework for solving complex problems related to robot movement and positioning [4]-[5]. Traditionally, IK techniques have relied on analytical solutions that, while offering precision in structured environments, lack the necessary adaptability in situations where operational conditions may vary. These analytical solutions are particularly useful when working with robots with simple architecture, where the equations can be solved directly. However, as robotic applications have evolved and become more complex, there has been a growing need for more flexible and robust methods that can handle the uncertainty and variability inherent in unstructured environments.

These advances have enabled robots not only to perform precise movements in controlled environments but also to adapt more effectively to changes in their operating environment. For example, optimization-based methods have proven to be particularly useful in situations where analytical solutions are difficult or impossible to obtain due to the complexity of the robot's kinematics [6]-[7]. These methods employ iterative algorithms to find joint configurations that minimize an objective function, such as the error between the desired position and that reached by the end effector. Although these methods may require more computational resources, they offer flexibility, which is crucial for operation in dynamic environments.

Additionally, the integration of artificial intelligence techniques, such as neural networks and evolutionary algorithms, has opened new possibilities for inverse kinematics [8]. Neural networks, for example, can be trained to approximate IK solutions in real-time, thus allowing robots to quickly adapt to changes in the environment or the tasks to be performed [9].

The synchronization of multiple robots is another critical challenge in modern robotics and it has been addressed with various methodologies over the past decades [10]. Precise coordination of movements between multiple robots is essential for completing complex collaborative tasks, such as manipulating large objects or performing tasks that require multiple stages of assembly. Traditional methods for robot synchronization often depend on rigid kinematic models and centralized algorithms that, while effective in some contexts, present significant limitations in terms of adaptability and scalability.

Centralized models for robot synchronization typically require a central controller to manage the coordination of all robots in a system. While this approach can ensure high precision in coordination, it also introduces significant vulnerabilities. For example, any failure in the central controller can compromise the operation of the entire system, and the ability to adapt to environmental changes is limited due to the inherent rigidity of the model. Additionally, as the number of robots in the system increases, the complexity of centralized control grows exponentially, which can make the system impractical for large-scale applications [11].

In response to these limitations, recent research has explored decentralized approaches to robot synchronization [12]. In these approaches, each robot acts more autonomously using its local perception of the environment and its interaction with other robots to adjust its behavior. This decentralized approach enhances the system's resilience to individual failures, as robots can continue operating even if one experiences problems. Furthermore, it allows for greater flexibility in task coordination, as robots can autonomously adapt to changes in the environment without the need for complete system reconfiguration.

One approach that has gained popularity in the context of decentralized synchronization is the use of intelligent agents, where each robot is considered an autonomous agent that interacts with other agents and the environment to achieve a common goal. This approach allows robots to make decisions locally, based on their current knowledge and the information shared with other robots. The distributed intelligence that emerges from this interaction can result in more robust and efficient synchronization, especially in dynamic environments where conditions can change unpredictably [13]-[14].

Adaptive learning, particularly reinforcement learning, has emerged as a powerful solution for improving robots' ability to learn and adapt to new situations, especially in the context of multiple robot synchronization [15]. Reinforcement learning is a machine learning technique that allows robots to develop control policies that optimize their performance over time based on feedback received from their environment. Instead of relying on a static, predefined model of how robots should behave, reinforcement learning enables robots to explore different synchronization strategies and adjust their behaviors based on the outcomes achieved.

This adaptability is particularly valuable in situations where environmental conditions are highly dynamic or unpredictable. For example, in a manufacturing environment where object placement may change or unexpected obstacles may arise, robots equipped with reinforcement learning can adjust their coordination in real-time, avoiding collisions and ensuring that tasks are completed efficiently. Moreover, reinforcement learning allows robots to learn from their mistakes, which is crucial in applications where precision is essential. As robots accumulate experience, their control policies become more refined, and they perform better in future tasks [16].

A key aspect of reinforcement learning in robot synchronization is the definition of the reward function, which guides the learning process. The reward function must be carefully designed to incentivize desirable behaviors, such as minimizing interference between robots or maximizing efficiency in task completion. Through continuous iteration and optimization of the control policy based on the feedback obtained, robots can develop synchronization strategies that are not only efficient but also highly adaptable to changes in the environment or tasks to be performed [6].

The use of reinforcement learning in robotics [17] is not limited to robot synchronization but has broader applications in collaborative and autonomous robotics. In these contexts, the ability of robots to learn and adapt is crucial to their success in a wide range of tasks. For example, in search and rescue scenarios, where conditions can change rapidly and collaboration between robots is essential, reinforcement learning enables robots to adjust their strategies in real-time, improving their chances of success.

This work contributes to the field of multi-robot synchronization by proposing a hybrid approach that combines Inverse Kinematics (IK) techniques with Model Predictive Control (MPC). This approach enhances the flexibility and adaptability of robots in dynamic environments by addressing the limitations of traditional methods that, while effective in controlled scenarios, often fail under changing conditions. Research demonstrates that the combination of IK and MPC not only optimizes synchronization but also significantly reduces trajectory errors, which is crucial for collaborative tasks in robotics.

The document is structured as follows: the first section introduces the problem of synchronization in multi-robot systems and the motivation behind the proposed approach and a review of existing literature, covering traditional IK methods and recent applications of MPC in robotics. The second section details the methodology used, including system parameters and the configurations evaluated in the experiments. The third section presents the results, offering a comparative analysis of synchronization errors, trajectory deviations, and arrival times for both the elbow-up and elbow-down configurations. Finally, the fourth section discusses the implications of the findings, explores potential industrial applications, and offers recommendations for future research.

2. MATERIALS AND METHODS

A. System Parameter Description

This study focuses on the synchronization of two robotic arms (hereinafter Arm A and Arm B), each with three degrees of freedom (3-DOF). In this context, inverse kinematics techniques are used to determine the joint configurations required for the end effectors of both arms to reach specific positions within a two-dimensional workspace. The geometric and kinematic parameters of the arms are described below:

Arm A:

Length of the first link (L_1): 10 units.

Length of the second link (L_2): 15 units.

Length of the third link (L_3): 5 units.

Desired end-effector position (x_e , y_e) (20, 10).

Desired end-effector orientation (γ): 30°.

Arm B:

Length of the first link (L_1): 8 units.

Length of the second link (L_2): 12 units.

Length of the third link (L_3): 6 units.

Desired end-effector position (x_e , y_e): (18, 15).

Desired end-effector orientation (γ): 45°.

Both arms have constant angular velocities of 30°/s for the motors controlling the joints.

B. Inverse Kinematics Calculation

Inverse kinematics is implemented to calculate the joint angles that allow the end effectors of both arms to reach the desired positions with specified orientations. This involves solving the system of equations that defines the end-effector position based on link lengths and joint angles.

C. Third Joint Position Calculation

The position of the third joint (x_3, y_3) of each arm is first calculated considering the end-effector orientation (γ) and the length of the third link (L_3), as described in Equation (1) :

$$\begin{aligned} x_3 &= x_e - L_3 cos(\gamma) \\ y_3 &= y_e - L_3 sin(\gamma) \end{aligned} \tag{1}$$

Using these values, the distance C between the base of the arm and the third joint is determined as Equation (2):

$$C = \sqrt{x_3^2 + y_3^2} \tag{2}$$

D. Joint Angle Determination

Knowing C and the lengths of the first two links (L_1 and L_2), joint angles are calculated using the following equations:

Angle a between links L_1 and L_2:

$$a = \cos^{-1}\left(\frac{L_1^2 + L_2^2 - C^2}{2L_1L_2}\right) \tag{3}$$

Angle B between the line connecting the base to the third joint and the first link L_1:

$$B = \cos^{-1}\left(\frac{L_1^2 + C^2 - L_2^2}{2L_1C}\right) \tag{4}$$

Using these values, two possible joint configurations (elbow-up and elbow-down) for both arms are obtained:

Elbow-down configuration:

$$\begin{aligned} J1a &= \tan^{-1}\left(\frac{y_3}{x_3}\right) - B \\ J2a &= 180^\circ - a \\ J3a &= \gamma - J1a - J2a \end{aligned} \tag{5}$$

Elbow-up configuration:

$$\begin{aligned} J1b &= \tan^{-1}\left(\frac{y_3}{x_3}\right) + B \\ J2b &= -(180^\circ - a) \\ J3b &= \gamma - J1b - J2b \end{aligned} \tag{6}$$

E. Synchronization and Error Calculation

1) Arrival Time Calculation.

The arrival time for each configuration (elbow-up and elbow-down) is determined by dividing each joint angle by the constant angular velocity of the motors. The maximum time among the joints is considered to define the arrival time for each arm:

Elbow-down configuration:

$$\text{Arrival time} = \frac{\max(|J1a|, |J2a|, |J3a|)}{\text{angular velocity}} \tag{7}$$

Elbow-up configuration:

$$\text{Arrival time} = \frac{\max(|J1b|, |J2b|, |J3b|)}{\text{angular velocity}} \tag{8}$$

2) Synchronization Error.

The synchronization error between the two arms is defined as the absolute difference between the calculated arrival times for each arm:

$$\text{Synchronization error} = |\text{Arrival time Arm A} - \text{Arrival time Arm B}| \tag{9}$$

Errors are calculated for both configurations (elbow-down and elbow-up).

3) Simulated Synchronized Motion and Dynamic Error Calculation.

A simulation is conducted to represent the synchronized movement of both arms over 100 frames. In each frame, the current positions of the joints and end effectors of each arm are calculated for both elbow-down and elbow-up configurations.

The dynamic error is calculated at each frame as the Euclidean distance between the current end-effector position and the desired position:

$$\text{Dynamic error} = \sqrt{(x_e^{\text{current}} - x_e^{\text{desired}})^2 + (y_e^{\text{current}} - y_e^{\text{desired}})^2} \tag{10}$$

Additionally, the dynamic synchronization error is calculated at each frame:

$$\text{Dynamic synchro error} = |\text{Current arrival time ArmA} - \text{Current arrival time ArmB}| \tag{11}$$

F. IMPLEMENTATION OF MODEL PREDICTIVE CONTROL (MPC)

1) Description of the MPC Algorithm.

To enhance synchronization between the two robotic arms and minimize trajectory tracking errors, a Model Predictive Control (MPC) strategy is implemented. It is an advanced control technique that uses a dynamic model of the system to predict the future behavior of controlled variables and optimize system performance based on a cost criterion.

MPC is applied to adjust the angular velocities of each arm's motors to minimize the difference in arrival times at the final positions while simultaneously maintaining minimal trajectory error.

2) Control Problem Formulation.

The primary objective of the MPC is to minimize a cost function defined as the weighted sum of the synchronization error and the trajectory error. The cost function J to be minimized is expressed as:

$$J = \sum_{i=1}^{N} \left[\lambda_1 \cdot (\text{Synchronization Error})_i^2 + \lambda_2 \cdot (\text{Trajectory Error})_i^2 \right] \tag{12}$$

where N is the number of steps in the prediction horizon, λ_1 and λ_2 are the weights associated with the synchronization error and trajectory error, respectively.

3) System Modeling.

The model used in the MPC is a kinematic model of robotic arms, assuming that the motor angular velocities determine the joint angles, which in turn determine the end-effector position.

The system is modeled based on the direct kinematic equations that relate the joint angles to the end-effector coordinates:

$$\begin{aligned} x_e \quad & f(\theta_1, \theta_2, \theta_3) \\ y_e \quad & g(\theta_1, \theta_2, \theta_3) \end{aligned} \tag{13}$$

where $\theta_1, \theta_2, \theta_3$ are joint angles controlled by the motor angular velocities $\omega_1, \omega_2, \omega_3$.

4) Prediction and Optimization.

At each control instance, MPC predicts the future behavior of the system over a prediction horizon using the kinematic model. It then solves an optimization problem that minimizes the cost function J under system constraints, which include limits on angular velocities and possible collisions between the arms.

The solution to the optimization problem provides the optimal angular velocities $\omega_1^*, \omega_2^*, \omega_3^*$ for each arm, which are applied in the next-time step.

3. RESULTS AND DISCUSSION

A. Position Error Analysis

The comparative graphs of position errors over time for the elbow-up and elbow-down configurations for both Arm A and Arm B reveal significant differences in synchronization and movement accuracy depending on the configuration adopted.

In elbow-down configuration, Arm A shows a lower initial position error compared to Arm B. However, both arms converge to the target within the specified time interval, with Arm A reaching the target before Arm B. The synchronization error in this configuration is approximately 1.15 seconds, thus indicating that the lack of synchronization is a significant factor in this mode.

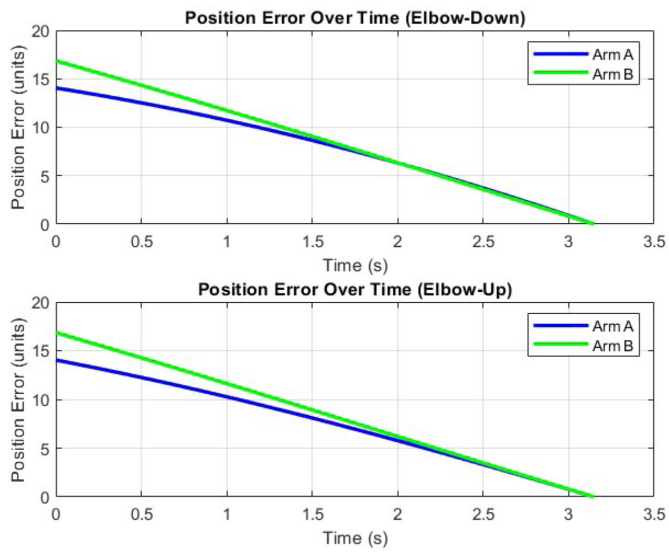

Fig 1. *Position error over time inverse kinematics.*

In elbow-up configuration, a similar behavior to the elbow-down configuration is observed, although with a slightly lower synchronization error of 0.67 seconds. This result suggests that the elbow-up configuration offers better synchronization between the two arms. Although Arm B has a higher initial position error than Arm A, the difference in arrival times is less pronounced.

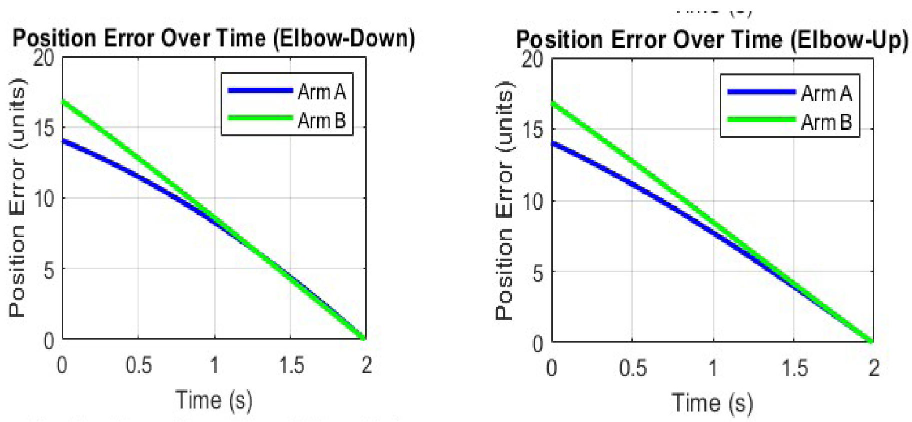

Fig 2. *Position Error Over Time MPC.*

B. Arrival Times and Synchronization Error

The arrival times for both configurations and arms indicate that Arm A always reaches the target before Arm B, regardless of the configuration adopted. In particular, the arrival time in the elbow-down configuration is significantly longer for Arm B than for Arm A, thus resulting in a larger synchronization error.

Elbow-Down Configuration: Arm A has an arrival time of 3.15 seconds, while Arm B arrives in 1.98 seconds, resulting in a considerable synchronization error of 1.15 seconds.

Elbow-Up Configuration: Arm A maintains a similar arrival time as in the elbow-down configuration (3.15 seconds), but Arm B improves its arrival time to 2.48 seconds, reducing the synchronization error to 0.67 seconds.

These results demonstrate that the elbow-up configuration not only improves synchronization but also contributes to a faster convergence of Arm B to the target.

C. Movement Trajectories

The trajectories obtained from inverse kinematics (IK) and model predictive control (MPC) for both configurations reveal differences in accuracy and smoothness of the movements.

Inverse Kinematics (IK): In the IK trajectories, both Arm A and Arm B show consistent convergence to the target with minimal trajectory error. The differences in trajectory between elbow-up and elbow-down are marginal, although the elbow-up configuration shows a slight advantage in terms of final precision.

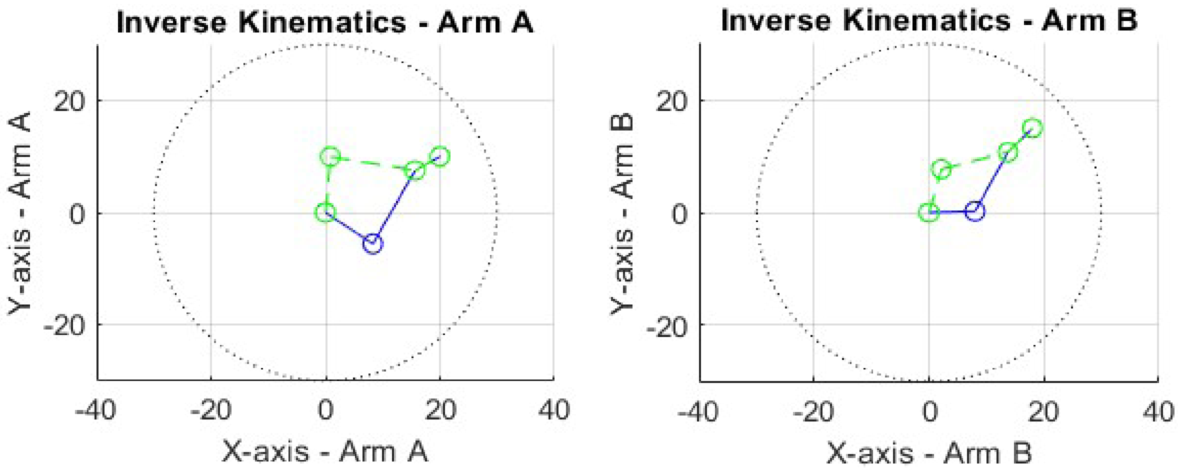

Fig 3. *Inverse Kinematics Trajectories.*

Model Predictive Control (MPC): The predictive simulations show that MPC provides greater accuracy in the trajectory, especially in the elbow-up configuration, and trajectory deviation is minimal. However, it is observed that MPC generates more abrupt trajectories, which could indicate greater complexity in practical implementation.

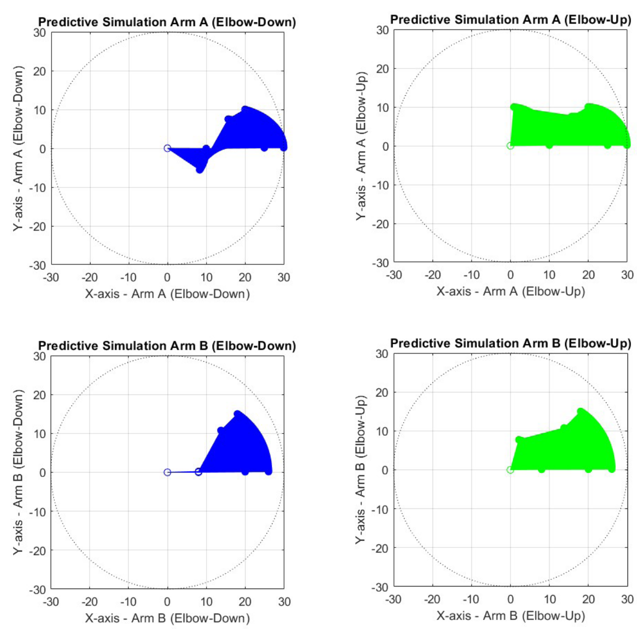

Fig. 4. *Predictive Simulation Trajectories (MPC).*

Table 1. summarizes the key results, including arrival times, synchronization errors, and trajectory deviations for both elbow-up and elbow-down configurations, using both IK and MPC methods.

This table makes a clear comparison of the performance across different configurations and methods, supporting the discussion and highlighting the strengths and weaknesses of each approach.

Table 1. *Key results.*

Configuration	Method	Arrival Time(s) – Arm A	Arrival Time (s) – Arm B	Synchronization Error (s)	Trajectory Deviation
Elbow-Down	IK	3.15	1.98	1.15	Low
Elbow-Down	MPC	1.57	1.99	0.42	Moderate
Elbow-Up	IK	3.15	2.48	0.67	Low
Elbow-Up	MPC	1.57	2.48	0.91	Moderate

D. Discussion of Results

The results obtained allow us to conclude that the elbow-up configuration presents notable advantages in both synchronization and movement accuracy when applying inverse kinematics (IK) and model predictive control (MPC) techniques. Specifically, the reduction in synchronization error and trajectory deviation observed in this configuration highlights its suitability for applications requiring precise coordination among multiple robotic arms, such as in manufacturing or assembly tasks.

In greater detail, the comparative analysis between IK and MPC reveals that, although MPC achieves higher trajectory accuracy, its implementation introduces challenges due to the abrupt nature of the generated trajectories. To mitigate this, fine-tuning of the controller parameters is suggested to smooth movements without compromising precision, thereby enhancing the viability of MPC for practical applications.

These findings suggest that the elbow-up configuration, particularly when coupled with optimized MPC settings, provides an optimal balance between synchronization and precision, and makes it a robust solution for industrial settings and collaborative robotics research where task coordination is critical. For instance, in industrial assembly lines, where synchronization directly impacts production efficiency and safety, the proposed approach could significantly improve the coordination between robotic units.

In summary, the advantages demonstrated by the elbow-up configuration and MPC integration have meaningful implications for the development of synchronized robotic systems in both industrial and research contexts and contributes to the advancement of adaptive, high-precision control strategies for complex multi-robot tasks.

4. CONCLUSIONS

This study demonstrates that the elbow-up configuration, when combined with Model Predictive Control (MPC) techniques, provides enhanced synchronization and improved trajectory accuracy over the elbow-down configuration. While MPC introduces complexity in implementation, it offers substantial benefits in trajectory precision and synchronization error reduction; therefore, it is highly suitable for applications demanding precise multi-robot coordination.

These findings carry significant implications for the design and control of robotic systems, particularly in industrial settings where synchronized operations are critical to performance and safety. For instance, the proposed approach could be directly applied in assembly lines and collaborative robotic systems in manufacturing, where synchronized movements ensure efficient task execution and minimize collision risks. Additionally, in research environments, this method provides a foundation for developing adaptable multi-robot systems that can perform in dynamic and unpredictable scenarios.

Future research should focus on fine-tuning MPC parameters to achieve smoother trajectories without compromising precision, a crucial step to facilitate implementation in real-world applications. Further exploration of alternative joint configurations and additional degrees of freedom is also recommended to evaluate the potential advantages of this hybrid approach across various robotic tasks and industries. Addressing these aspects will not only expand the applicability of the methodology but also provide valuable insights into overcoming synchronization challenges in large-scale robotic systems.

AUTHOR CONTRIBUTIONS

Griselle Salazar: Research, Writing-Original Draft, Writing-Review & Editing, Software.

Oscar Loyola: Research, Writing-Original Draft, Writing-Review & Editing, Resources, Software.

Eduardo Villarroel: Writing-Original Draft, Writing-Review & Editing.

Claudia Sandoval: Writing-Original Draft, Writing-Review & Editing.

ACKNOWLEDGEMENTS

This work was developed with the support of the Vice-Rectorate for Research and Doctoral Programs at Universidad Autónoma de Chile, through the DIUA 296-2024 grant.

REFERENCES

[1] C. S. V. Gutiérrez, L. U. S. Juan, I. Z. Ugarte, I. M. Goenaga, L. A. Kirschgens, V. M. Vilches, “Time Synchronization in modular collaborative robots,” *Cornell University*, 2018. https://doi.org/10.48550/arXiv.1809.07295

[2] T. Borangiu, F. D. Anton, and S. Anton, “Robot Motion Synchronization in Cooperative Tasks,” *Solid State Phenomena*, vol. 166-167, pp. 63-68, 2010. https://doi.org/10.4028/www.scientific.net/SSP.166-167.63

[3] E. Yime, J. Roldán Mckenley, J. L. Villa Ramirez, “Computed torque control of a 2-RR planar parallel robot // Control por par calculado de un robot paralelo planar 2-RR,” *Prospectiva*, vol. 15, no. 2, pp. 85-95, 2017. https://doi.org/10.15665/rp.v15i2.1111

[4] J. Guzmán Giménez, Á. Valera, V. Mata-Amela, M. Á. Díaz Rodríguez, “Obtención del modelo cinemático inverso de sistemas robotizados de cadena cinemática abierta empleando bases de Groebner: aplicación a un robot hexápodo,” in *XL Jornadas de Automática*, Universidade da Coruña, 2020, pp. 726-734. https://doi.org/10.17979/spudc.9788497497169.726

[5] T. Ribeiro, A. Paiva, “Expressive Inverse Kinematics Solving in Real-time for Virtual and Robotic Interactive Characters,” *Cornell University*, 2019. https://doi.org/10.48550/arXiv.1909.13875

[6] F. Benzi, C. Secchi, "An Optimization Approach for a Robust and Flexible Control in Collaborative Applications," *Cornell University*, 2021. https://doi.org/10.48550/arXiv.2103.03082

[7] D. Cardona-Ortiz, A. Paz, G. Arechavaleta, "Exploiting sparsity in robot trajectory optimization with direct collocation and geometric algorithms," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 469-475. https://doi.org/10.1109/ICRA40945.2020.9196668

[8] A. Bahani, M. E. houssine Ech-Chhibat, H. Samri, H. A. Elattar, "The Inverse Kinematics Evaluation of 6-DOF Robots in Cooperative Tasks Using Virtual Modeling Design and Artificial Intelligence Tools," *International Journal of Mechanical Engineering and Robotics Research*, pp. 121-130, 2023. https://doi.org/10.18178/ijmerr.12.2.121-130

[9] J.-G. Habekost, E. Strahl, P. Allgeuer, M. Kerzel, S. Wermter, "CycleIK: Neuro-inspired Inverse Kinematics," *Cornell University*, 2023. https://doi.org/10.48550/arXiv.2307.11554

[10] F. Zhang, J. Qu, H. Liu, Y. Fu, "A Multi-Priority Control of Asymmetric Coordination for Redundant Dual-Arm Robot," *International Journal of Humanoid Robotics*, vol. 16, no. 2, e1950008, 2019. https://doi.org/10.1142/S0219843619500087

[11] J. Manuel, E. Zulueta, B. Fernndez, M. Gr, "Multi-Robot Systems Control Implementation," in *Robot Learning*, Sciyo, 2010. https://doi.org/10.5772/10254

[12] S. Garcia, C. Menghi, P. Pelliccione, T. Berger, R. Wohlrab, "An Architecture for Decentralized, Collaborative, and Autonomous Robots," in *IEEE International Conference on Software Architecture (ICSA)*, IEEE, 2018, pp. 75-79. https://doi.org/10.1109/ICSA.2018.00017

[13] E. Sebastian, T. Duong, N. Atanasov, E. Montijano, C. Sagues, "Physics-Informed Multi-Agent Reinforcement Learning for Distributed Multi-Robot Problems," *Cornell University*, 2023. https://doi.org/10.48550/arXiv.2401.00212

[14] M. Afrin, J. Jin, A. Rahman, A. Rahman, J. Wan, E. Hossain, "Resource Allocation and Service Provisioning in Multi-Agent Cloud Robotics: A Comprehensive Survey," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 2, pp. 842-870, 2021. https://doi.org/10.1109/COMST.2021.3061435

[15] J. S. Toquica, P. S. Oliveira, W. S. R. Souza, J. M. S. T. Motta, D. L. Borges, "An analytical and a Deep Learning model for solving the inverse kinematic problem of an industrial parallel robot," *Computers and Industrial Engineering*, vol. 151, e106682, 2021. https://doi.org/10.1016/j.cie.2020.106682

[16] G. I. Parisi, C. Kanan, "Rethinking Continual Learning for Autonomous Agents and Robots," *Cornell University*, 2019. https://doi.org/10.48550/arXiv.1907.01929

[17] R. Julian, B. Swanson, G. S. Sukhatme, S. Levine, C. Finn, K. Hausman, "Never Stop Learning: The Effectiveness of Fine-Tuning in Robotic Reinforcement Learning," *Cornell University*, 2020. https://doi.org/10.48550/arXiv.2004.10190