COBEM 2023 27TH INTERNATIONAL CONGRESS OF MECHANICAL ENGENEERING

ABCM Brazilian Society of Mechanical Sciences and Engineering
27th ABCM International Congress of Mechanical Engineering
December 04-08, 2023, Florianópolis-SC, Brazil

COB-2023-0857
Adaptive Trajectories for Robotic Manipulators

João Victor Zanoni*
jvzanoni111@gmail.com
Fernanda de Oliveira Ferreira*
fernanda_o_ferreira@hotmail.com
Caroline de Lima Pereira**
carolinelimma@gmail.com
Leonardo Mejia Rincon**
leonardo.mejia.rincon@ufsc.br
*Undergraduate program in Automation, Control, and Computation Engineering (CAC) Federal University of Santa Catarina - Campus Blumenau. Santa Catarina, Brazil.
**Graduate Program in Textile Engineering (PGETEX) Federal University of Santa Catarina - Campus Blumenau. Blumenau, Santa Catarina, Brazil

Abstract.
In response to continuous advancements in engineering and the rise of Industry 4.0, robotics has become a central point of interest, drawing widespread attention on a global scale. In this context, the problem of generating trajectories emerges as a significant challenge in the industry. In general, handling objects may require precision and stability to avoid damage and ensure the quality of the final product. Among many other challenges, the reliability of robots must be extremely high to allow and guarantee good safety. In light of these considerations, this paper proposes a new methodology for the automatic generation of optimized trajectories in robotic manipulators, anticipating and avoiding collisions with objects and obstacles. Grounded in foundational theories of industrial robotics and trajectory planning, our methodology employs a real-time spatial analysis algorithm to dynamically identify points of interest and obstacles within the robot manipulator's operational workspace. This enables us to generate an optimized trajectory that obviates the need for object removal or complete cessation of manipulator movement. As the central aim of this research project, we have devised an algorithmic methodology specifically tailored for future implementation in a Cartesian pick-and-place manipulator system. The proposed methodology allows for the execution of intelligent trajectory planning that not only avoids collision scenarios but also optimizes the robot's energy efficiency during task execution. The findings of this study substantiate the feasibility of offline intelligent trajectory planning. Moreover, the proposed methodology exhibits extensibility, suggesting its potential applicability to a broader spectrum of complex scenarios and heterogeneous manipulator architectures.

Keywords: *Adaptative Trajectories, Optimization, Robotics, Robotic Manipulation.*

1. INTRODUCTION

The deployment of adaptive trajectory algorithms in industrial robotic manipulators introduces multifaceted challenges. A primary concern stems from the inherently dynamic nature of manufacturing workflows, which are subject to frequent alterations. Such variability necessitates that the robotic system perpetually refines its motion planning to ensure precision in object manipulation while mitigating the risk of unintended damage (Javaid *et al.*, 2021). Concurrently, in environments where collaboration with other entities (be it humans or auxiliary machinery) is prevalent, real-time trajectory adaptation becomes essential to preemptively avoid collisions. Additionally, the exigencies of industrial-scale production call for highly efficient algorithms capable of governing robotic manipulator movements without sacrificing perception accuracy and control fidelity. Addressing these challenges is pivotal for enhancing the overall efficacy of automated manufacturing systems (Kiyokawa *et al.*, 2023). Based on this perspective, this work proposes a novel methodology for the automatic generation of optimized trajectories in robotic manipulators, predicting and avoiding collisions with objects and obstacles.

Future manufacturing applications will require an increasing involvement of robots performing tasks such as *pick-and-place* (Li *et al.*, 2020). These robots are used to automate the process of picking up and placing objects in industrial environments. This enables a reduction in production time and an increase in process efficiency. Additionally, these robots can be employed in repetitive and hazardous tasks or in situations where human presence is impossible or undesirable (Khurshid *et al.*, 2017). For these reasons, *pick-and-place* robots provide an efficient solution to enhance safety and mitigate the risk of accidents in industrial processes.

The functionality of a *pick-and-place* robot can be conceptualized into three primary stages. The initial stage involves the end effector grasping the object. Following this, the robot aligns and positions the item at the intended location. The final stage sees the object placed securely in the desired position. To facilitate these operations, the robot must possess a minimum of four degrees of freedom (DoF), comprised of three DoFs for translation and one for rotation (Zhang *et al.*, 2010). These pick-and-place operations constitute the majority of current industrial robotic applications and have significant potential for expansion (Mnyusiwalla *et al.*, 2020).

In industrial settings, object manipulation presents considerable challenges that demand both precision and stability to mitigate the risk of damage and maintain the quality of the finished product. To address these needs, pick-and-place robotic manipulators offer a compelling solution. These robots can be configured with adaptive trajectory algorithms, designed based on comprehensive three-dimensional models of the target objects (Pagano *et al.*, 2020). These models incorporate not only object geometry but also potential environmental obstructions within the manipulator's operational workspace. Armed with detailed geometric and positional data, the trajectory planning algorithms are capable of delineating an optimal path, thereby minimizing collision risks and ensuring proficient object handling. This approach significantly diminishes the necessity for manual trajectory adjustments, consequently enhancing the efficiency and accuracy of manipulation tasks.

2. THEORETICAL FOUNDATIONS

The task of formulating automated trajectories for industrial robotic manipulators necessitates the amalgamation of several key computational and mathematical tools. This section offers a critical survey of the pertinent theoretical basis, with an emphasis on the core principles underpinning optimization algorithms and adaptive trajectory planning methods. The objective is to harness these tools cohesively to establish a robust and high-performance system. Such a system aims to excel in optimizing trajectory planning while dynamically adapting to the ever-changing nuances of the manufacturing environment. A comprehensive exposition of these foundational tools and their synergistic integration serves as the conceptual bedrock for the subsequent empirical analysis and evaluation of our proposed methodology.

2.1 Optimization and the Storn and Price Algotithm

Optimization is a fundamental tool used in numerous scientific and engineering disciplines, providing a systematic approach to finding optimal solutions in complex systems. It involves maximizing or minimizing a given objective function while considering a set of constraints.

Optimization methods encompass a wide range of techniques, each suited to specific problem characteristics and requirements. Traditional approaches include gradient-based methods that rely on derivative information, such as the well-known gradient descent. These methods are effective for smooth and differentiable functions but may struggle with non-convex, multimodal, or discrete problems. For these three categories, Evolutionary algorithms (EAs) have gained significant attention due to their ability to solve complex optimization problems.

EAs mimic natural evolution processes, combining survival of the fittest principles with random variation and recombination. They explore the search space through a population of potential solutions and iteratively improve upon them. EAs, including Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and differential evolution (DE). Among the DE algorithms, the algorithm introduced by Rainer Storn and Kenneth Price in 1995, has been successful in solving complex problems across various domains and has emerged as a powerful and widely used optimization technique.

The Storn and Price algorithm utilizes a combination of mutation, crossover, and selection operators to explore the search space efficiently, handling both, continuous and discrete variables and making it suitable for a broad range of optimization problems. In essence, the Storn and Price algorithm presents different strategies for obtaining the solution (Mejia *et al.*, 2014a,b), with the main ones highlighted in Table 1. All techniques employ differential evolution (DE), but some vary in the use of the base vector, either randomly or selecting the best from the population, and may involve exponential or binary distribution. Its simplicity, global search capabilities, and ability to converge to near-optimal solutions have made it a popular choice for researchers and practitioners (Pierezan *et al.*, 2017; Weihmann *et al.*, 2012). This algorithm was used in this research as the main tool to optimize the trajectories and all the strategies shown in Table 1 where considered.

Table 1: Storn and Price algorithm strategies

Strategy	Definition
1	DE/best/1/exp
2	DE/rand/1/exp
6	DE/best/1/bin
7	DE/rand/1/bin

2.2 Bezier curves

The Bezier curves, introduced by Pierre Bézier in the 1960s, has historically had a profound impact on computer graphics, geometric modeling, and path generation for different applications in engineering (Li *et al.*, 2022). The Bézier curve is a polynomial representation expressed as a linear interpolation between some representative points, called control points. The Bezier curve relies on Newton's Binomial for the calculation of its coefficients and is solved through the equation:

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{i} x^{n-k} y^k. \quad x = t, \quad y = 1-t \tag{1}$$

where the t-index is a parameterization value to traverse the curve (values between zero and one can be used), n is the degree of the Binomial, so we use $n+1$ control points for each curve we want to draw, and $\binom{n}{i}$ are the binomial coefficients. The control points B_n can be randomly chosen, and each one should be multiplied by one of the terms of the resolved binomial. The n-th coefficient of the interpolation is obtained through Newton's Binomial and is a polynomial of the form:

$$P_{in}(t) = \binom{n}{i} (1-t)^{n-i} t^i \tag{2}$$

Finally, In a formal way, it is possible to obtain a Bezier curve as a parameterized curve in terms of t as shown below:

$$B(t) = \sum_{i=0}^{n} P_{in}(t) \cdot B_i = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i \cdot B_i \tag{3}$$

The Bezier equation expressed by Eq. (3) represents the fundamental mathematical model used in this paper to dynamically change the trajectories for different scenarios. The implemented Bezier curves dynamically modify their control points depending on the task to be executed and the number of obstacles present in the environment. The process of dynamic trajectory adaptation is shown in more detail in Section (3).

3. PROPOSED METHODOLOGY

This section introduces an innovative proposed approach to generate collision-free trajectories based on Bezier curves, wherein the number of control points is dynamically determined by the number of obstacles within the environment. This methodological approach is structured in four sequential steps: a) Identification and characterization of obstacles, b) Calculation of the required control points for the Bezier curve, c) Derivation of the equation for the parametrized Bezier curve, and d) Optimization of the Bezier curve. Below, the methodological steps are shown in a more detailed way.

a) Identification and Characterization of Obstacles: Initially, the system accepts the pre-defined positions of the objects as input parameters. This presupposition facilitates the subsequent tasks of quantifying, measuring, and pinpointing the locations of obstacles within the operational workspace. The input data for this phase is formulated as a vector of n elements, each element encapsulating the center-of-mass coordinates and dimensional attributes of an individual obstacle. Although this obstacle data is currently supplied parametrically through a computational function, future work aims to develop an automated algorithm capable of dynamically extracting these parameters from optical sensors.

b) Calculation of Control Points for the Bezier Curve: In an obstacle-free environment, a straight line may intuitively seem the most efficient trajectory for the manipulator, as it would guarantee the shortest possible path. In such cases, the trajectory is a parametric straight line as those shown in Eq. 4 whose length can be described through Equation 5. For Equations 4 e 5 the terms x_{min} , y_{min}, and z_{min} represents the coordinate of the starting point, the terms x_{max} , y_{max}, and z_{max} represents the coordinate of the destination point and, the term k represents the parameter of the parametric straight line.

$$s(k) = (s_x; s_y; s_z) = ((x_{min} + (x_{max} - x_{min}) * k); (y_{min} + (y_{max} - y_{min}) * k); (z_{min} + (z_{max} - z_{min}) * k)) \tag{4}$$

$$d = \sqrt{(x_{min} - x_{max})^2 + (y_{min} - y_{max})^2 + (z_{min} - z_{max})^2} \tag{5}$$

However, the introduction of obstacles necessitates consideration of potential collisions, obliging the creation of alternate trajectories. In the proposed methodology, it was supposed to use a Bezier curve having a number of control points as shown in Equation 6.

$$Pc = 4 * n - (n - 1) \qquad \forall\, n \geq 1 \tag{6}$$

This equation takes into account that for each obstacle (n) a simple curve will be constructed consisting of an initial control point, a final control point, and two intermediate control points. If multiple obstacles exist, the final control point of one curve segment will serve as the initial control point for the next. Thus, a single obstacle necessitates four control points, while two obstacles require seven control points, and so on.

c) Derivation of Equation for Parametrized Bezier Curve: In this step, a simple implementation of the Bezier curve is essential, which can be achieved using the approach presented in Section 2.2 and employing Equation 3.

d) Optimization of Bezier Curve: In this stage, an optimization process is introduced using the Storn & Price algorithm. It is important to note that the primary objective of this optimization process is to determine the position of all control points, excluding the first and last ones, as they signify the beginning and end of the trajectory, respectively.

Given that the objective function of the Storn & Price algorithm is the heart of the optimization, in this case, it aims to achieve the shortest possible trajectory length by avoiding collisions with obstacles. In other words, during each iteration, it examines whether the Bézier curve generated with the control points intersects with the limits that define the obstacles. If a collision is detected, the control points are re-evaluated, if no collision occurs, the trajectory is optimized to minimize the distance traveled.

The objective function is defined through Eq. 7a, which represents the curve length function, while the constraints for collisions are defined by Eqs. 7b, 7c, 7d, 7e, 7f, and 7g.

$$\text{minimize} \quad \int_0^1 \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2}\, dt \tag{7a}$$
$$\text{subject to} \quad f(x_i) \leq L_i, \tag{7b}$$
$$f(y_i) \leq W_i, \tag{7c}$$
$$f(z_i) \leq H_i, \tag{7d}$$
$$f(x_i) \geq L_i + a_i, \tag{7e}$$
$$f(y_i) \geq W_i + b_i, \tag{7f}$$
$$f(z_i) \geq H_i + c_i \tag{7g}$$

Where $f(x_i)$, $f(y_i)$, and $f(z_i)$ represent the current coordinates in the X, Y, and Z axes, respectively. Meanwhile, L_i, W_i, and H_i denote the corresponding coordinates in the X, Y, and Z axes of the object's initial point, and a_i, b_i, and c_i represent the lengths in the X, Y, and Z axes, respectively.

4. RESULTS

To evaluate the performance of the Storn & Price algorithm in trajectory estimation, we conducted simulations by varying its algorithmic strategies, namely 1, 2, 6, and 7. Furthermore, certain simulations factored in a potential deviation in the minimum distance between the initial and target points. Specifically, we considered scenarios where this minimum distance was increased by 1%, meaning it was set at 101% of the true minimum distance.

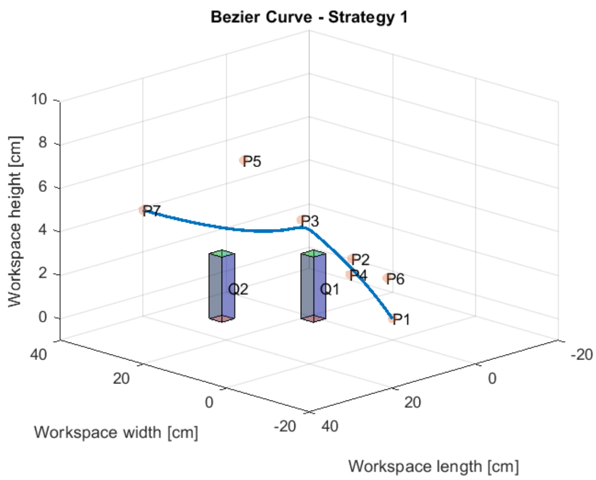
4.1 Adaptive Trajectories for 2 Objects with 101% of the Minimum Distance

In the initial series of simulations, two pre-identified obstacles, each measuring 3x3x3 cm, are present within the workspace. Here, the end effector's origin point is denoted as $P_I = [0\ 0\ 0]$, while $P_E = [30\ 30\ 5]$ represents the target endpoint. As elucidated in Eq. 4, the minimum distance between these two points can be expressed as:

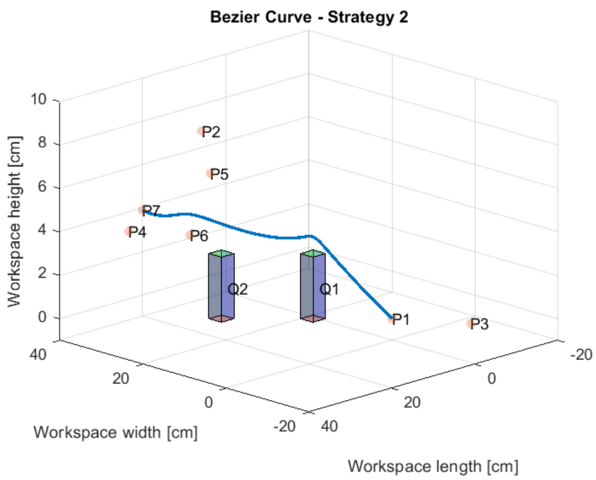
$$d = \sqrt{(0-30)^2 + (0-30)^2 + (0-5)^2} = 42.7200cm \tag{8}$$

Since this is the minimum possible distance and it hardly can be achieved, because it should be in a perfect straight line, a slightly higher value is adopted, in this case, 1%, to allow more freedom in optimization. Therefore, in this simulation, a value of 101% of such a distance as a stop criterion was used, that is $d = 1.01 \cdot 42.7200 = 43.1472cm$.

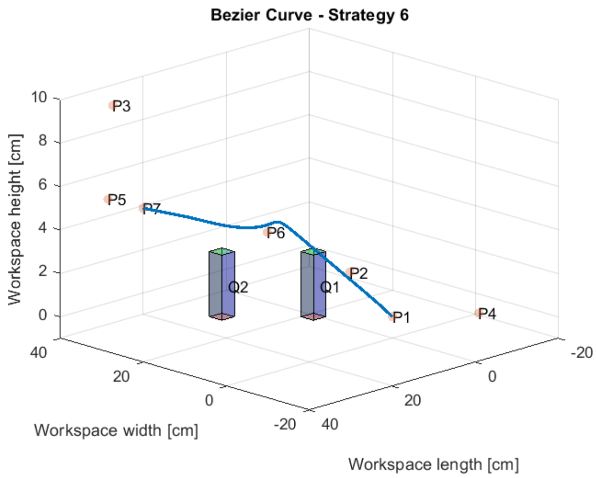
Based on these definitions, Figure 1a shows the obtained result for the trajectory using strategy 1 of the Storn & Price algorithm. Furthermore, Figure 1b depicts the obtained result for the trajectory using strategy 2 of the Storn & Price algorithm and, Figure 1c illustrates the result obtained for the trajectory using strategy 6 of the algorithm. Similarly, Figure 1d shows the result obtained for the trajectory using strategy 7 of the Storn & Price algorithm.



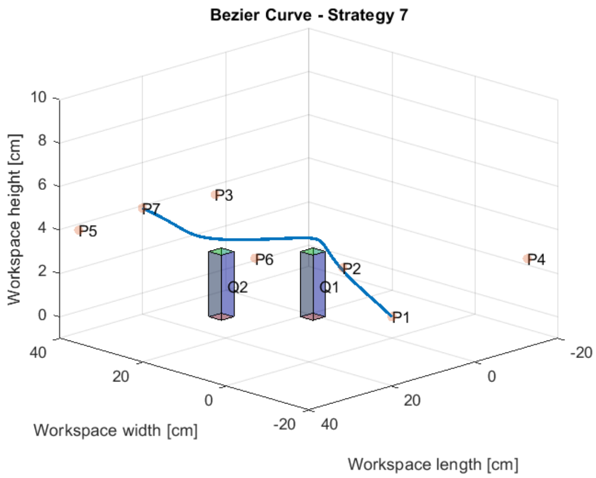
(a) Bézier Curve with Strategy 1.



(b) Bézier Curve with Strategy 2.



(c) Bézier Curve with Strategy 6.



(d) Bézier Curve with Strategy 7.

Figure 1: Bézier Curve of Degree 6 Obtained with 101% of the Minimum Distance for 2 Objects.

For the simulations, the input parameters considered are the strategy, the number of objects, and the minimum distance. As a result, the desired trajectory is obtained. Additionally, the iteration in which the solution was found, the corresponding distance, and the execution time were recorded. These data are presented in Table 2.

Table 2: Analysis Results of the different strategies of the Storn & Price algorithm

Strategy	Operations	Objects	Minimum Distance (cm)	Iteration Found	Distance Found (cm)	Time
1	20000	2	43.1472	6000	43.147220	41min
2	20000	2	43.1472	8000	43.147218	1h34min
6	20000	2	43.1472	6000	43.147220	1h17min
7	20000	2	43.1472	7000	43.147220	2h2min

Based on the analysis of the results obtained and presented in Table 2, it is evident that all strategies were subjected to the same input data, which included 20,000 iterations, two objects, and a minimum distance of 101%. From these input data, it can be observed that all simulations achieved a similar distance, but with different execution times and numbers of iterations.

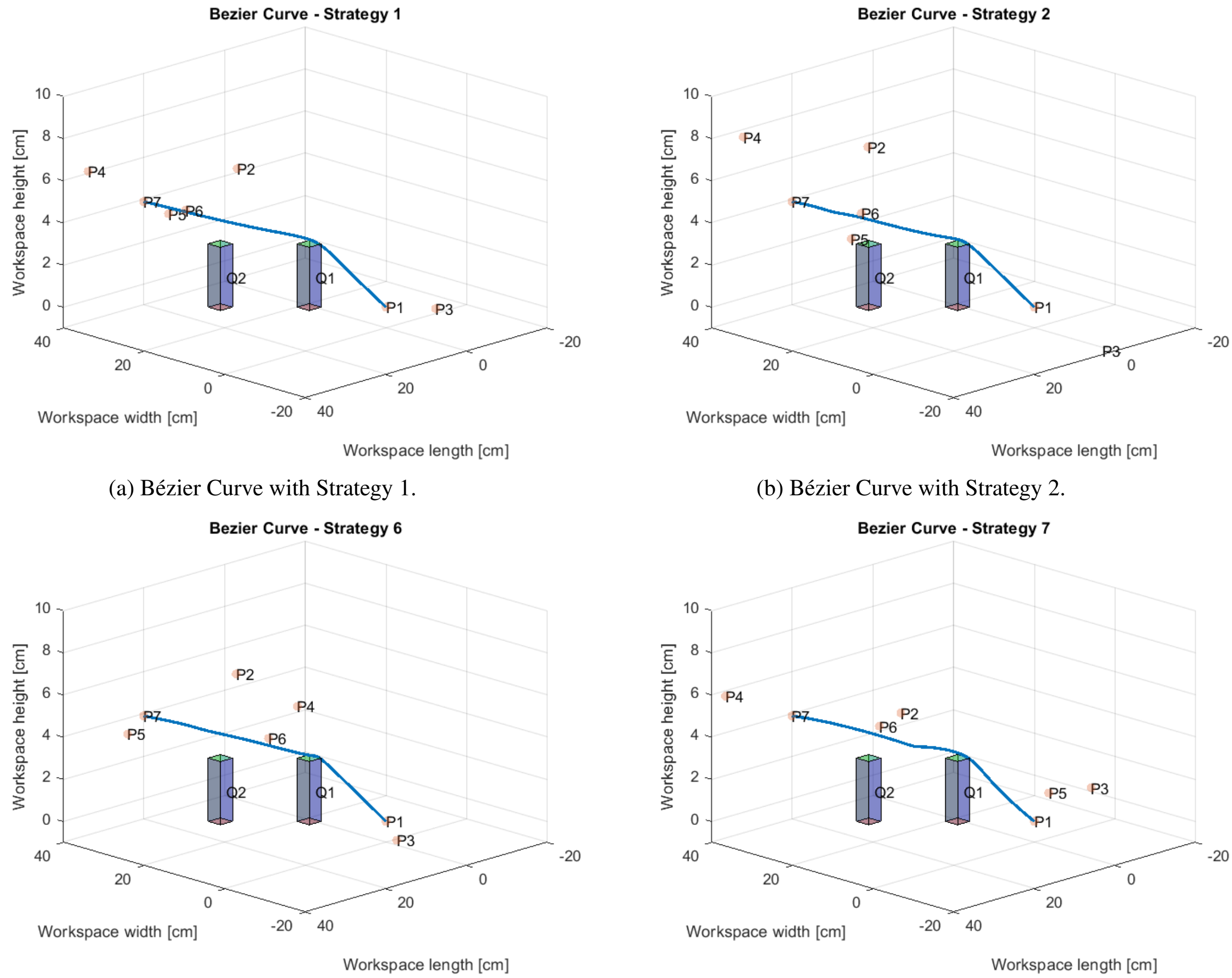
In this case, strategy 1 demonstrated higher efficiency, optimizing the trajectory in 6,000 iterations with a time of 41 minutes. Following that, strategy 6 stands out with 6,000 iterations and a duration of 1 hour and 17 minutes. On the other hand, among the strategies with poorer performance are strategies 2 and 7. The former required 8,000 iterations and took 1 hour and 34 minutes, while the latter needed 7,000 iterations and took 2 hours and 2 minutes.

In addition to Table 2, it is worth noting that the estimated curves for the trajectories depicted in Figures 1a, 1b, 1c, and 1d still can be optimized.

4.2 Adaptive Trajectories for 2 Objects with Minimum Distance

To perform an optimization that yields more accurate trajectories, a new simulation scenario is carried out, following the same principle presented in subsection 4.1 However, in this case, only the minimum distance is sought, that is, $d = \sqrt{(0-30)^2 + (0-30)^2 + (0-5)^2} = 42.7200cm.$

Given these definitions, Figure 2a illustrates the achieved result for the trajectory using Strategy 1 of the Storn & Price algorithm and, Figure 2b presents the obtained result for the trajectory using Strategy 2 of algorithm. Similarly, Figure 2c depicts the achieved result for the trajectory employing Strategy 6 of the Storn & Price algorithm and, the result obtained for the trajectory using Strategy 7 of the Storn & Price algorithm is demonstrated in Figure 2d.



(a) Bézier Curve with Strategy 1. (b) Bézier Curve with Strategy 2.

(c) Bézier Curve with Strategy 6. (d) Bézier Curve with Strategy 7.

Figure 2: Bézier Curve of Degree 6 Obtained with Minimum Distance for 2 Objects.

In Table 3, the input and output parameters of the simulation are presented. By analyzing those results, it is observed that all strategies were subjected to the same initial parameters, which are 20,000 iterations, two objects, a minimum distance of 47.7200cm, and a robot step of 0.01. In this case, Strategy 1 showed the highest efficiency, optimizing the trajectory in 3 hours and 49 minutes, followed by Strategy 2 with 3 hours and 52 minutes. Next, Strategy 6 took 4 hours and 2 minutes, and finally, Strategy 7 took 4 hours and 14 minutes. Overall, it is noted that the estimated curves for the trajectories presented in Figures 2a, 2b, and 2c for Strategies 1, 2, and 6 are similar and provide a good estimation. On the other hand, it is noticed that Strategy 7, represented by Figure 2d, may still have room for improvement in the trajectory.

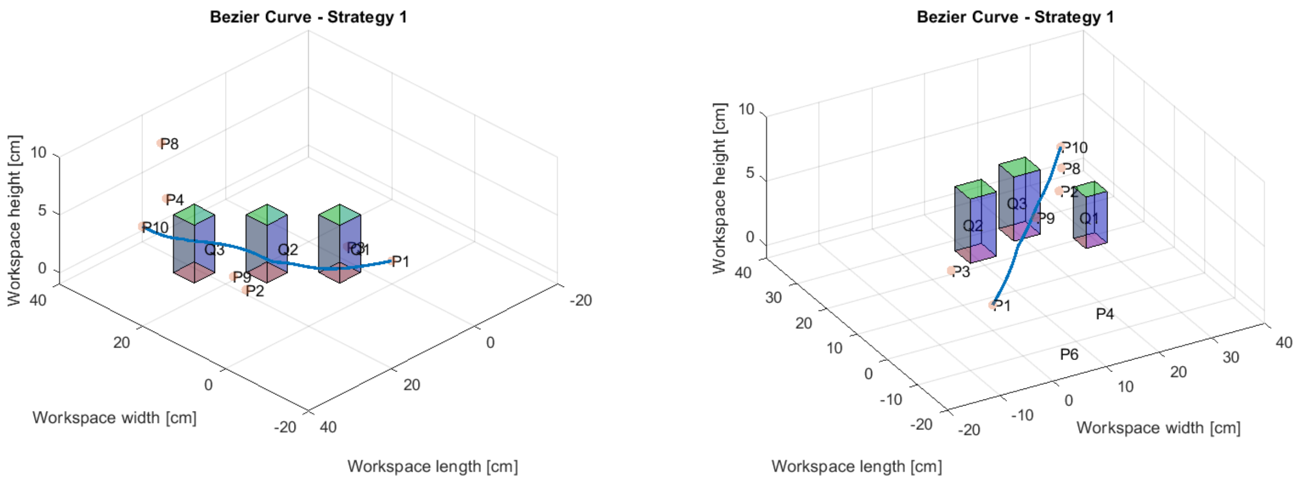
Considering the results obtained so far, the next step in the analysis is to consider the interaction of three objects while maintaining the minimum distance. This approach can reveal new insights and provide valuable information for future decision-making in the project.

Table 3: Analysis Results of the different strategies of the Storn and Price algorithm

Strategy	Operations	Objects	Minimum Distance (cm)	Iteration Found	Distance Found (cm)	Time
1	20000	2	47.7200	20000	42.88	3h49min
2	20000	2	47.7200	20000	42.88	3h52min
6	20000	2	47.7200	20000	42.88	4h2min
7	20000	2	47.7200	20000	42.90	4h14min

4.3 Adaptive Trajectories for 3 Objects with Minimum Distance

In this case, we consider point $P_I = [0\ 0\ 1]$ as the origin of the end effector and $P_E = [30\ 30\ 4]$ as the final point. Furthermore, as demonstrated earlier, the minimum distance possible between these two points is given by $d = \sqrt{(0-30)^2+(0-30)^2+(1-4)^2} = 42.532$. Furthermore, it is important to highlight that based on the previously presented data, Strategy 1 has shown to be the most efficient in optimizing the trajectory. Therefore, in this simulation scenario, only Strategy 1 is used. Thus, in Figures 3a and 3b, the results obtained for two distinct simulations are presented. In each scenario, three 5x5x5 cm objects were used spread across different places on the work area.



(a) Bézier Curve with Strategy 1. (b) Bézier Curve with Strategy 1.

Figure 3: Bézier Curve of Degree 9 Obtained with Strategy 1 and Minimum Distance for 3 Objects.

Similarly as shown before, in Table 4, the input and output parameters of the simulation are presented. With the analysis of the results obtained and displayed in this table, it is noted that the simulations were subjected to the same initial parameters, which are 20,000 iterations, three objects, and a minimum distance of 45.532 cm. In this case, it is noticed that the optimization time was longer than for two objects. Additionally, when observing Figures 3a and 3b, it can be seen that the found trajectories are well-optimized and do not collide with the obstacles present in the workspace.

Table 4: Analysis Results of the different strategies of the Storn and Price algorithm

Strategy	Operations	Objects	Minimum Distance (cm)	Iteration Found	Distance Found (cm)	Time
1	20000	3	42.532	20000	42.804594	7h19min
1	20000	3	42.532	20000	42.768302	9h4min

At this point, it is important highlighting that the hardware used consists of an Intel Core i7 processor of 7th generation 2.7GHz with Turbo Boost up to 3.5GHz, dual-core. And when observing the high processing times, it is noted that this approach presents greater viability in creating adaptive trajectories offline, where the trajectories are created and thought out first and are then sent to the manipulator.

5. FUTURE WORKS

To validate the efficacy of our proposed methodology, we constructed a 4-DoF Cartesian manipulator optimized for pick-and-place operations in which we intend to implement our proposed methodology. This manipulator supports prismatic movements along the primary axes (X, Y, and Z) and introduces a rotational movement about the Z-axis. In addition, we custom-engineered and fabricated an end effector, which is affixed at the terminal point of the kinematic chain, to handle lightweight, small-sized objects adeptly. Plans are also underway to integrate a computer vision algorithm, which would facilitate the quantification, measurement, and precise localization of obstacles in the workspace. Such integration would enable the automatic identification of objects and pave the way for testing our methodology in generating adaptive trajectories for an actual physical manipulator. Figure 4 provides a visual representation of our current progress.

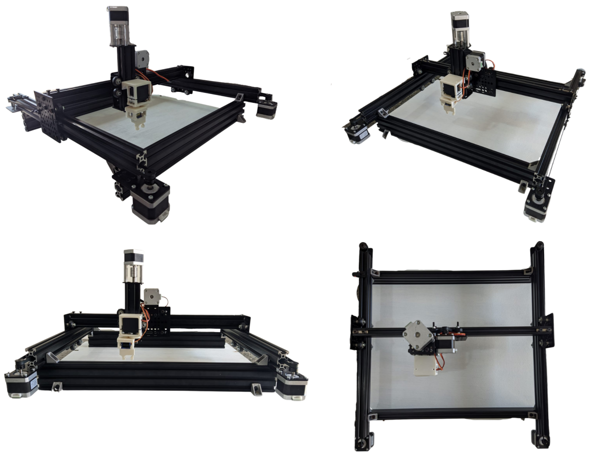


Figure 4: Built *Pick-and-place* manipulator.

6. CONCLUSIONS

This study aimed to propose a new methodology for the automatic and intelligent generation of optimized trajectories, anticipating and avoiding collisions with objects and obstacles. To make this possible, industrial robotics techniques were successfully used to increase the reliability and efficiency of handling such objects, proposing better quality and optimization in the production of trajectories. In this paper, a series of tests were carried out to evaluate the functionality and effectiveness of the generated adaptive trajectories. The results obtained showcased the proficiency in predicting and avoiding collisions in offline mode, thus safeguarding the workspace.

The introduction of an automatic and intelligent approach to generating optimized offline trajectories has demonstrated its feasibility and promising potential in the industry. With this, the union of the Storn & Price optimization algorithm with the Bézier curves to generate trajectories made it possible to navigate effectively without the need to remove objects present in the path. Consequently, this led to a considerable increase in operational efficiency and trajectory optimization. Finally, the possibility of continuing the project with the physical construction of a manipulator also proved to be viable and of great relevance for validating the trajectories.

7. ACKNOWLEDGEMENTS

The authors would like to extend their sincere gratitude to the Graduate Program in Textile Engineering (PGETEX) and the undergraduate program in Automation, Control, and Computation Engineering (CAC), both at the Federal University of Santa Catarina - Campus Blumenau for their invaluable support and collaboration in this project. We are especially grateful for their commitment to training and nurturing future masters and engineering professionals, which has greatly contributed to the success of this endeavor.

8. REFERENCES

Javaid, M., Haleem, A., Singh, R.P. and Suman, R., 2021. “Substantial capabilities of robotics in enhancing industry 4.0 implementation”. *Cognitive Robotics*, Vol. 1, pp. 58–75. ISSN 2667-2413. doi:https://doi.org/10.1016/j.cogr.2021.06.001. URL `https://www.sciencedirect.com/science/article/pii/S2667241321000057`.

Khurshid, R.P., Fitter, N.T., Fedalei, E.A. and Kuchenbecker, K.J., 2017. “Effects of grip-force, contact, and acceleration feedback on a teleoperated pick-and-place task”. *IEEE Transactions on Haptics*, Vol. 10, No. 1, pp. 40–53. doi:10.1109/TOH.2016.2573301. URL `https://ieeexplore.ieee.org/document/7479565`.

Kiyokawa, T., Shirakura, N., Wang, Z., Yamanobe, N., Ramirez-Alpizar, I.G., Wan, W. and Harada, K., 2023. “Difficulty and complexity definitions for assembly task allocation and assignment in human–robot collaborations: A review”. *Robotics and Computer-Integrated Manufacturing*, Vol. 84, p. 102598. ISSN 0736-5845. doi:https://doi.org/10.1016/j.rcim.2023.102598. URL `https://www.sciencedirect.com/science/article/pii/S0736584523000741`.

Li, F., Jiang, Y., Li, T., Feng, Y. and Chen, S., 2020. "Design of a robot end effector with measurement system for precise pick-and-place of square objects". *Procedia Manufacturing*, Vol. 48, pp. 172–180. ISSN 2351-9789. doi:https://doi.org/10.1016/j.promfg.2020.05.035. URL https://www.sciencedirect.com/science/article/pii/S2351978920314864. 48th SME North American Manufacturing Research Conference, NAMRC 48.

Li, X., Gao, X., Zhang, W. and Hao, L., 2022. "Smooth and collision-free trajectory generation in cluttered environments using cubic b-spline form". *Mechanism and Machine Theory*, Vol. 169, p. 104606. ISSN 0094-114X. doi:https://doi.org/10.1016/j.mechmachtheory.2021.104606. URL https://www.sciencedirect.com/science/article/pii/S0094114X21003451.

Mejia, L., Simas, H. and Martins, D., 2014a. "Force capability polytope of a 3rrr planar parallel manipulator". In M. Ceccarelli and V.A. Glazunov, eds., *Advances on Theory and Practice of Robots and Manipulators*. Springer International Publishing, Cham, pp. 537–545. ISBN 978-3-319-07058-2. doi:10.1007/978-3-319-07058-2_60. URL https://doi.org/10.1007/978-3-319-07058-2_60.

Mejia, L., Simas, H. and Martins, D., 2014b. *Force Capability Polytope of a 4RRR Redundant Planar Parallel Manipulator*, Springer International Publishing, Cham, pp. 87–94. ISBN 978-3-319-06698-1. doi:10.1007/978-3-319-06698-1_10. URL https://doi.org/10.1007/978-3-319-06698-1_10.

Mnyusiwalla, H., Triantafyllou, P., Sotiropoulos, P., Roa, M.A., Friedl, W., Sundaram, A.M., Russell, D. and Deacon, G., 2020. "A bin-picking benchmark for systematic evaluation of robotic pick-and-place systems". *IEEE Robotics and Automation Letters*, Vol. 5, No. 2, pp. 1389–1396. doi:10.1109/LRA.2020.2965076. URL https://ieeexplore.ieee.org/document/8954746.

Pagano, S., Russo, R. and Savino, S., 2020. "A vision guided robotic system for flexible gluing process in the footwear industry". *Robotics and Computer-Integrated Manufacturing*, Vol. 65, p. 101965. ISSN 0736-5845. doi:https://doi.org/10.1016/j.rcim.2020.101965. URL https://www.sciencedirect.com/science/article/pii/S0736584519301097.

Pierezan, J., Freire, R.Z., Weihmann, L., Reynoso-Meza, G. and dos Santos Coelho, L., 2017. "Static force capability optimization of humanoids robots based on modified self-adaptive differential evolution". *Computers Operations Research*, Vol. 84, pp. 205–215. ISSN 0305-0548. doi:https://doi.org/10.1016/j.cor.2016.10.011. URL https://www.sciencedirect.com/science/article/pii/S0305054816302593.

Weihmann, L., Martins, D. and dos Santos Coelho, L., 2012. "Modified differential evolution approach for optimization of planar parallel manipulators force capabilities". *Expert Systems with Applications*, Vol. 39, No. 6, pp. 6150–6156. ISSN 0957-4174. doi:https://doi.org/10.1016/j.eswa.2011.11.099. URL https://www.sciencedirect.com/science/article/pii/S0957417411016459.

Zhang, Y., Chen, B.K., Liu, X. and Sun, Y., 2010. "Autonomous robotic pick-and-place of microobjects". *IEEE Transactions on Robotics*, Vol. 26, No. 1, pp. 200–207. doi:10.1109/TRO.2009.2034831. URL https://ieeexplore.ieee.org/document/5342467.

9. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.