Analysis of Trajectory Optimization for Solving Tower of Hanoi Problem using Industrial Manipulator

Ahmed Iqdymat
Department of Automation and Industrial Informatics
University Politehnica of Bucharest
Bucharest, Romania
ahmed.iqdymat@stud.acs.upb.ro

Grigore Stamatescu
Department of Automation and Industrial Informatics
University Politehnica of Bucharest
Bucharest, Romania
grigore.stamatescu@upb.ro

***Abstract*—This paper presents implementation and analysis of three different trajectories of an industrial manipulator for solving the Tower of Hanoi problem. ABB's robot IRB120 was simulated in RobotStudio. MATLAB code was used to plan the steps to solve the Tower of Hanoi problem. MATLAB Simulink was then used to solve the robot inverse kinematics to find the joint angles for achieving the desired position for the end effector. Finalized trajectory was then forwarded to RobotStudio that simulated the whole trajectory. Distance travelled by end effector was calculated for each trajectory. Time taken by the robot to complete the task was estimated using RobotStudio simulation. Based on these distances and time durations, it has been concluded that circular trajectories are more efficient than linear or combination of linear and circular trajectories.**

***Index Terms*—artificial intelligence, machine learning, robot studio, tower of hanoi, trajectory planning**

I. Introduction

Dynamic business environment and highly changing demands have put many challenges to the modern industry. Quick production and its advertisement in the market promise modest advantage in the form of monetary benefit and customer satisfaction [1] Computer and robots with cutting edge technologies have played pivotal role in industrial automation for efficient and effective handling of various processes [2] This change in industry can save much human resource that can be utilized in other processes.

It is evident that human beings made mistakes even in processes that are repetitive in nature; however robots can perform these tasks accurately [3]. That's the reason robots are replacing humans everywhere in the industry. Specially for the tasks that cannot be performed by the humans due to safety concerns. Sometime robots are used for tasks where trained manpower is not available. Unlike human beings, robots can be programmed for specific tasks and they keep on repeating that process for unlimited time without any loss of accuracy.

Industrial robots or industrial manipulators are used for tasks like pick and place objects from any start point to another goal point. There are infinite possibilities for a robot to move from one state to another. Efforts are made to design trajectories from industrial manipulators to minimize time and cost. Sahar et al. [4] presented a graph search and a dynamic time scaling algorithm to plan trajectory with minimum time for industrial manipulator.

Trajectory planning becomes more challenging in dynamic environment where environment and constraints are changing with time. Industrial robot performing pick and place tasks are often very large and handling heavy objects. For such robots, it is required to design trajectories with minimum jerks and accelerations. Broquere et al. [5] present a soft motion trajectory planner that avoids sudden or rapid change in velocity and acceleration. Gasparetto et al. [6] presented that minimizing a weighted sum of integral squared jerk and execution time may be used to plan an optimum trajectory. Wang, He, et al. [7] considered acceleration, constant velocity and deceleration for each joint of an industrial manipulator to generate smooth point to point trajectory.

Researchers are putting much efforts to automate industrial robots using artificial intelligence [8], machine learning and deep reinforcement learning [9], [10] as well as new Industrial Internet of Things (IIoT) technologies [11] Various simulation platforms are used for trajectory planning and analysis for industrial manipulators. RobotStudio is an efficient and user friendly plate form for simulating ABB's robots [12]. Liqiu, Zhou, et al. [13] used MATLAB and RobotStudio for analysis of straight and arc trajectory for industrial manipulators. We also used same simulation environment for analysis and research. Tower of Hanoi is an interesting and challenging problem for designing optimum trajectories for industrial manipulators in constraint environment. Wang, Lin, et al. [14] solved Tower of Hanoi problem for designing optimum trajectories using motion planning algorithms.

This paper also shows the comparison of three different trajectories for solving Tower of Honai problem using RobotStudio as simulation environment. Robot kinematics were solved in MATLAB/Simulink using model of IRB120 industrial manipulator. Analysis and comparison of trajectory optimization for multiple trajectories in such integrated simulation environment is the novelty of this paper.

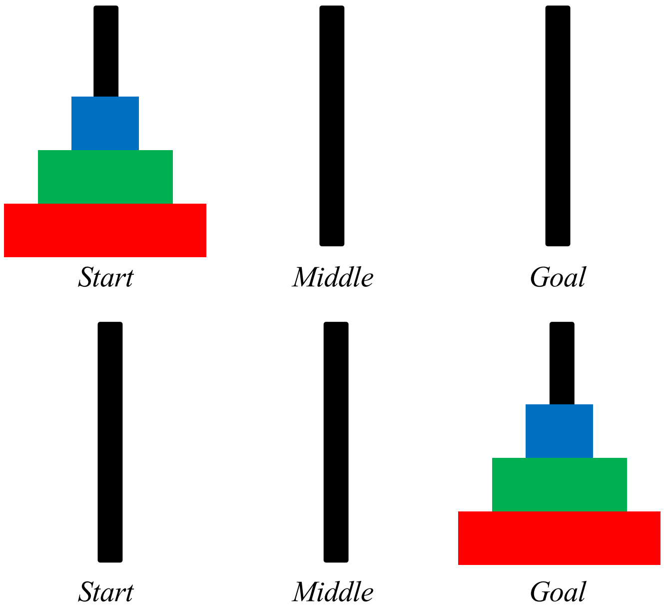

Fig. 1. Tower of Hanoi

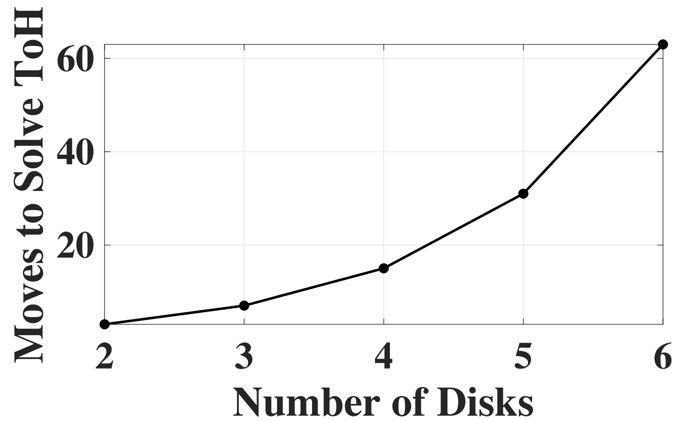

Fig. 2. Moves to solve ToH problem

II. Tower of Hanoi Problem

The tower of Hanoi (ToH) is a mathematical problem. ToH consists of three rods and several disks of different sizes. Initially, all the disks are placed on one rod, one over the other in ascending order of size similar to a cone-shaped tower, as in Figure 1.

The objective of the puzzle is to move all disks in last rod obeying the following rules:

- Only one disk can be moved at a time.
- Each move consists of shifting the disk from one rod to another.
- No larger disk cannot be placed on smaller disk in any case.

With 3 disks, the puzzle can be solved in 7 moves. The minimal number of moves required to solve a Tower of Hanoi puzzle is $2^n - 1$, where n is the number of disks. Figure 1 shows the start and goal state of the puzzle for the disks while Figure 2 shows the relation between number of moves to solve ToH for different number of disks.

III. System Description

RAPID programming language has been used to simulate ABB's robot (IRB120) in RobotStudio to solve Tower of Hanoi problem using three different trajectories.

A. Industrial Manipulator

Industrial manipulators are robots that perform specific tasks in specific environment in industry. There are various standard industrial manipulators that have been designed as standard hardware available for variety of tasks to be performed in industry.

We used IRB120 robot that is a six axis industrial manipulator used for assembly applications. IRB120 provides an

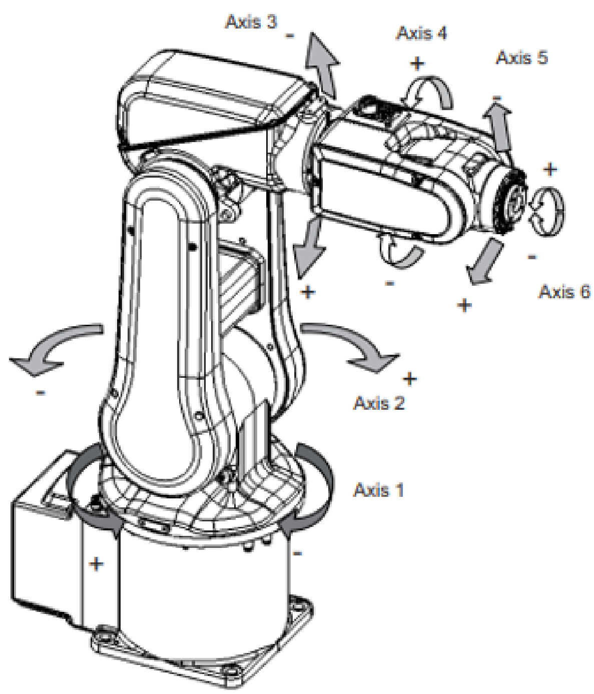

Fig. 3. ABB IRB120 Industrial Manipulator

TABLE I
ABB IRB120 Specifications

S. No.	Parameter	Value
1	Payload	3 [Kg]
2	Base	180 x 180 [mm]
3	Rated Power	3 [KVA]
4	Arm Reach	982 [mm]
5	Weight	24 [Kg]
6	Robot Height	700 [mm]
7	Application	Pick and Place

TABLE II
ABB IRB120 AXIS CONSTRAINTS

Axis	Type of Motion	Range of Movement
1	Rotation Motion	-165°to +165°
2	Arm Motion	-110°to +110°
3	Arm Motion	-110°to +70°
4	Wrist Motion	-160°to +160°
5	Bend Motion	-120°to +120°
6	Turn Motion	-400°to +400°

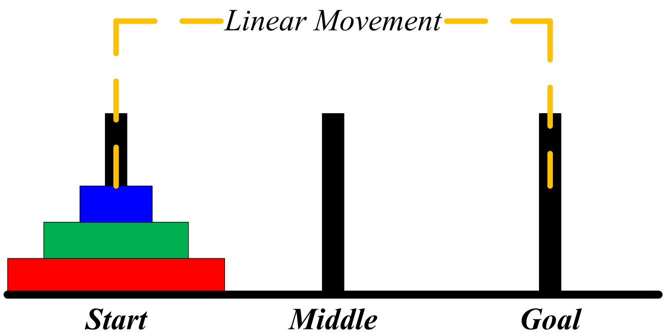

Fig. 4. Linear Trajectory

agile solution for flexible and combat handling. IRB120 is a product of ABB robots. Figure 3 shows the six axis degrees of freedoms of IRB120 while Table 1 and Table 2 show the specifications and the range of rotation angle for each axis respectively.

B. Trajectory Planning

Trajectory planning is the movement of joints of an industrial manipulator from one state to another. In domain of industrial manipulators, trajectory planning is of immense importance because in real world scenarios, no robot can move everywhere. Every manipulator has to work in an environment full of constraints. Considering these constraints, trajectories are optimized so that manipulator may move from any initial state to its desired final state covering minimum distance.

Keeping in view the ToH problem, IRB20 has to pick the disks from one rod and put it to another rod. For this purpose, robot end effector has to move from one rod to another. This can be achieved in three different types of motions:

- Linear Trajectory
- Circular Trajectory
- Non-Linear Trajectory

1) Linear Trajectory: In linear trajectory, robot end effector moves linearly i.e. vertical or horizontal. This is the most simplified trajectory. Figure 4 shows the linear trajectory for IRB120 while solving ToH problem.

2) Circular Trajectory: In circular trajectory, robot end effector moves in circular path as shown in Figure 10. This trajectory is more efficient as will be shown in last section of this paper. But in real world problems, circular trajectory is not always feasible due to real world environment constraints.

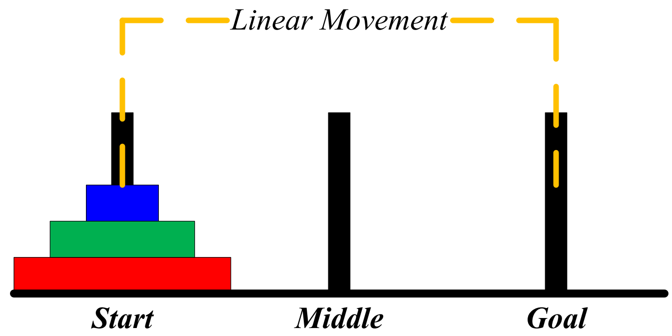

Fig. 5. Circular Trajectory

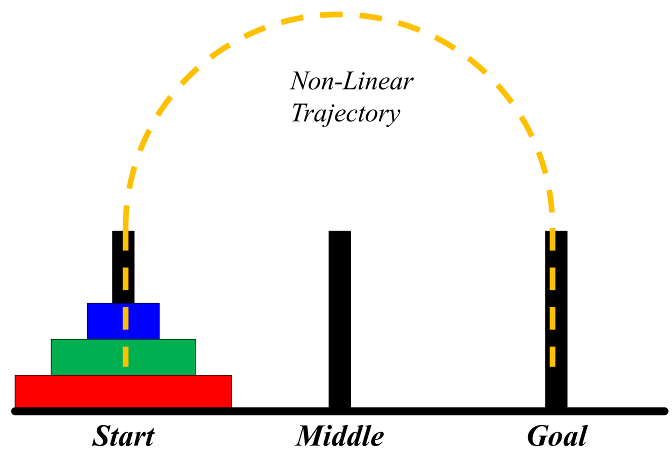

Fig. 6. Non-Linear Trajectory

3) Non-linear Trajectory: Non-Linear trajectory is a combination of linear and circular trajectory (Figure 6. In case of ToH problem, if robot has to move largest disk from one rod to another, it cannot move in circular path beginning from start as rod is vertically straight. Therefore, robot will have to move linearly till the disk is removed from the rod and then move in circular path till next rod. Finally, robot end effector will again move linearly vertical to place largest disk in destination rod. That's how non-linear trajectories are designed in combination with linear and circular trajectories.

IV. PROBLEM FORMULATION

Three disk ToH (Tower of Hanoi) has been used in this study to compare three different trajectories of IRB120. $2^n - 1 = 2^3 - 1 = 7$ steps are required to solve the problem.

d_1, d_2 and d_3 are the three disks, where d_1 is the smallest disk. r_1, r_2 and r_3 are the three rods, where r_1 is the start point and r_3 is the goal point. $r_1 \longrightarrow d_3, d_2, d_1$ indicates that first rod contains all the three disks in sequence shown i.e. d_3 (largest) first and then d_2 (middle) and then d_1 (smallest). With these notations, Table 3 shows the steps to solve the ToH problem.

TABLE III
TOH SOLUTION STEPS

Steps	r_1	r_2	r_3
Start	d_3, d_2, d_1	—	—
1	d_3, d_2	—	d_1
2	d_3	d_2	d_1
3	d_3	d_2, d_1	—
4	—	d_2, d_1	d_3
5	d_1	d_2	d_3
6	d_1	—	d_3, d_2
7 (Goal)	—	—	d_3, d_2, d_1

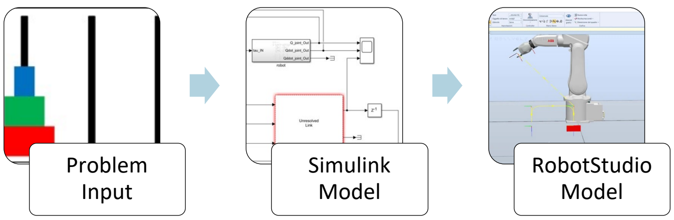

Fig. 7. Simulation Environment

V. SIMULATION ENVIRONMENT

RobotStudio is widely used to visualize ABB's robot working to reduce commissioning time. RobotStudio was used to simulate IRB120 robot for solving ToH problem. MATLAB Simulink was used to solve inverse kinematics to calculate intermediate states of industrial manipulator during trajectory tracking. Problem was input to Simulink that solved it and then Simulink send trajectory data to RobotStudio for simulation as shown in Figure 7.

Figure 8 shows Simulink block diagram of the torque controller that takes desired torque value as input and computes commands for each joint of the robot. These commands further act as input to the robot joints model as shown in Figure 9 i.e. Simulink block diagram of robot model.

VI. RESULTS AND DISCUSSION

Three disk Tower of Hanoi problem was solved with three different trajectories as explained in previous sections. Figure 10 shows the circular motions of end effector during circular trajectory tracking. For all the three trajectories, distance travelled by end effector was calculated and time taken to travel this distance in order to complete the task was recorded in simulation environment as shown in Table 4. These results show that circular trajectories are optimized as compared to linear and non-linear trajectories for such type of industrial manipulators.

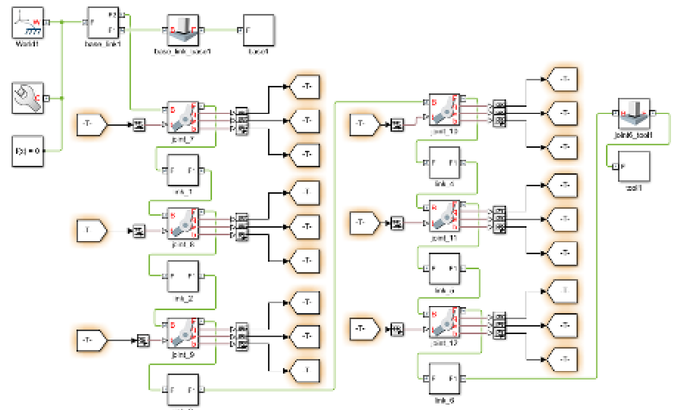

Fig. 9. Robot Model

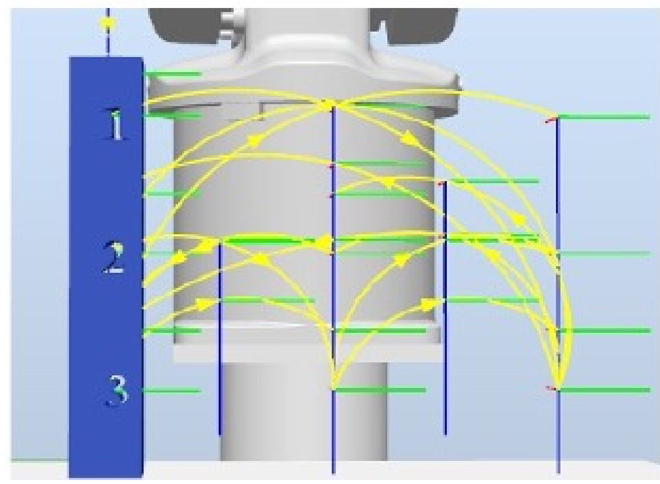

Fig. 10. Circular Trajectories

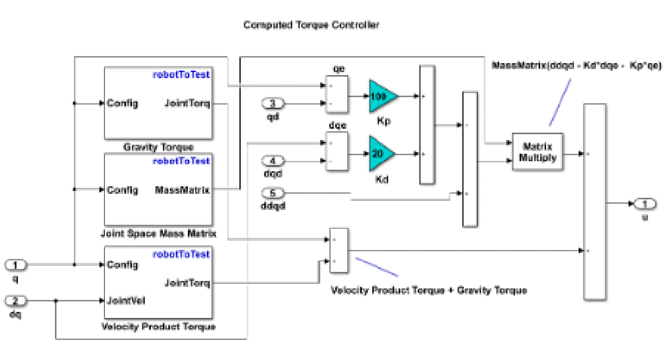

Fig. 8. Computed Torque Controller

TABLE IV
SIMULATION RESULTS

S. No.	Trajectory	Dist. [mm]	Time [sec]
1	Linear Motion	3683	33
2	Circular Motion	2616	22
3	Linear and Circular Motion	3397	41.9

VII. CONCLUSION

This work demonstrates the comparison of three different trajectories for an industrial manipulator. ABB's IRB120 robot was modeled in Simulink and integrated with RobotStudio for simulation purpose. Tower of Hanoi problem was then solved in simulation environment. Three different trajectories were simulated i.e. circular trajectory, linear trajectory and combination of linear and circular trajectories. Time taken by the robot to complete each trajectory was recorded. Simulation results proved that circular trajectory is 1.5 times faster than linear trajectory while 1.9 times faster than linear and circular motion. Thus circular trajectory is more efficient than other two trajectories.

REFERENCES

[1] M. Andreasen and L. Hein, *Integrated Product Development*. IPU, 2000.

[2] M. P. Groover, *Automation, production systems, and computer-integrated manufacturing*. Pearson Education India, 2016.

[3] T. R. Kurfess *et al.*, *Robotics and automation handbook*. CRC press Boca Raton, FL, 2005, vol. 414.

[4] G. Sahar and J. M. Hollerbach, "Planning of minimum-time trajectories for robot arms," *The International journal of robotics research*, vol. 5, no. 3, pp. 90–100, 1986.

[5] X. Broquere, D. Sidobre, and I. Herrera-Aguilar, "Soft motion trajectory planner for service manipulator robot," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 2808–2813.

[6] A. Gasparetto and V. Zanotto, "Optimal trajectory planning for industrial robots," *Advances in Engineering Software*, vol. 41, no. 4, pp. 548–556, 2010.

[7] H. Wang, H. Wang, J. Huang, B. Zhao, and L. Quan, "Smooth point-to-point trajectory planning for industrial robots with kinematical constraints based on high-order polynomial curve," *Mechanism and Machine Theory*, vol. 139, pp. 284–293, 2019.

[8] Z. H. Ang, C. K. Ang, W. H. Lim, L. J. Yu, and M. I. Solihin, "Development of an artificial intelligent approach in adapting the characteristic of polynomial trajectory planning for robot manipulator," *International Journal of Mechanical Engineering and Robotics Research*, vol. 9, no. 3, pp. 408–414, 2020.

[9] J. Zhong, T. Wang, and L. Cheng, "Collision-free path planning for welding manipulator via hybrid algorithm of deep reinforcement learning and inverse kinematics," *Complex & Intelligent Systems*, pp. 1–14, 2021.

[10] M.-D. Pavel, S. Roșioru, N. Arghira, and G. Stamatescu, "Control of open mobile robotic platform using deep reinforcement learning," in *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future*, T. Borangiu, D. Trentesaux, and P. Leitão, Eds. Cham: Springer International Publishing, 2023, pp. 368–379.

[11] R.-A. Luchian, G. Stamatescu, I. Stamatescu, I. Fagarasan, and D. Popescu, "Iiot decentralized system monitoring for smart industry applications," in *2021 29th Mediterranean Conference on Control and Automation (MED)*, 2021, pp. 1161–1166.

[12] S. Rosioru, G. Stamatescu, I. Stamatescu, I. Fagarasan, and D. Popescu, "Deep learning based parts classification in a cognitive robotic cell system," in *2022 26th International Conference on System Theory, Control and Computing (ICSTCC)*, 2022, pp. 403–408.

[13] Z. Liqiu, A. Juan, Z. Ronghao, and M. Hairong, "Trajectory planning and simulation of industrial robot based on matlab and robotstudio," in *2021 IEEE 4th International Conference on Electronics Technology (ICET)*. IEEE, 2021, pp. 910–914.

[14] L. Wang, X. Yue, and J. Peng, "Modeling and simulation of a tower-of-hanoi-playing robot station based on robotstudio," in *Advances in Mechanical Design: Proceedings of the 2021 International Conference on Mechanical Design (2021 ICMD)*. Springer, 2022, pp. 1507–1521.