Natural Gradient Boosting Collision Detection in Robot Manipulators

Hesam Jafarian, Marzieh Zare, Roel Pieters, Saeed Mehrang, Ari Visa. Paper-ID [9]

Abstract—**Lately, obtaining autonomous solutions is attaining remarkable momentum due to the recent rise in demand and the necessity for higher production rates. The importance of this focus manifests when there are global crises or economic aspirations. Notwithstanding this, the current factory solutions require hand-operated technicians to program the robots on factory sites. It is explicit, the main barrier to obtaining fully autonomous solutions arises from collision-free path planning. In the planning process, the principal bottleneck is perception and geometrical information such as the orientation and location of robot structures versus an environment. In fact, the planners require this information at any time to obtain a collision-free path. Since this information requires a constant update, the existing solutions are computationally expensive and deceptive. In this regard, we propose a data-driven approach that utilizes machine learning to model and approximates the collision space for a static industrial environment. The solution proves a reduction of the inference time and is able to approximate a large state space with a certain amount of precision compensation. The results of this work are suitable for accelerating any path planning algorithms such as probabilistic sampling approaches.**

I. INTRODUCTION

The human agents and their operation time are too precious to dissipate by the tasks which are not required or are replaceable with autonomous intelligent solutions. However, a real-time design of such agents is arduous due to the bottleneck that originates from geometrical perception and navigation. Numerous studies attempted to eliminate the barriers, yet factors exist that dwindle the gradual production cycle-time. This enigma usually correlates to path planning and Collision Detection (CD). These processes are the principal sources of delay for reaching a real-time process. In this work, we proposed a data-driven learning approach to approximate the collision checking behavior and calculations. Subsequently, the trained model approximates the collision status for further application in the path-finding and planning algorithms. In addition, we investigated the industrial cases that the geometrical topology of the environment is static while robots operate dynamically in the designated safe zones. It is notable that the existing collision engines address the complexity of the scenes for training or subdividing a complex condition. But this imposes computation time that we also approximate it using a learning technique. Finally, the information on false positives or false negatives is accessible in a model on our simulation website for further analysis (https://simulbotics.com/publications.html).

II. RELATED WORKS

It is conceivable to categorize the CD process efforts into two subcategories. Classical approaches that most studies and collision engines utilize with standard algorithms and assumptions. Ericson [4] and Bourg [1] explain most of these methods and cases with primitive geometrical assumptions. As an example, Wu et al. [7] researched the hybrid bounding volume hierarchy (BVH) to detect collision of virtual manipulators. All of these works utilize mathematical geometry computation for CD. In contrast, Learning-based methods intend to learn the CD process and to model it for future checks. For instance, Zhao and Li [8] and Ramírez et al. [6] utilized bio-inspired Genetic Algorithm (GA) for CD or García et al. [5] utilized artificial neural networks.

III. PROBLEM FORMULATION

A. Space State Operation

The process of collision checking requires the computation of geometrical information at any time. The existing methods either utilize Inverse Kinematic (IK) or Forward Kinematics (FK). In industry, the joint space operation is more appreciated rather than cartesian state space. The principal reason is that the IK has problems such as computation delay time, multiple or no solutions due to singularity. Nonetheless, a non-avoidable computation is FK or the transformation matrix of collision geometries with every movement.

B. Geometrical Collision Computations

We allocate collision geometries to the environment and robot manipulators. For a robot manipulator of n Degree of Freedom (DoF), each new geometry ending configuration is computed using Eq.1. The T matrix is the transformation matrix described based on Denavit and Hartenberg (DH) as Corke [2] illustrates in his work. Movement of objects require a new transformation computation and similarly the robot's location and orientation of collision geometries.

$$\xi_E = {}^{0}_{1}T \cdot {}^{1}_{2}T \cdot .. \cdot {}^{n-1}_{n}T \tag{1}$$

IV. METHODOLOGY AND CASE STUDY

We described our strategy on how to simplify the mathematical computations in this section, as illustrated in Fig.1.

A. Data Collection and Use Of Geometrical Topologies

We collected the collision data by utilizing the joint's state space of a UR5 robot. The operation in this space is reliable since the FK solution function is unique and certainly leads to a solution. Besides, in the data collection process, we assigned collision geometries to the robot and environment. We utilized mesh colliders for the robot as shown in Fig.2 and allocated

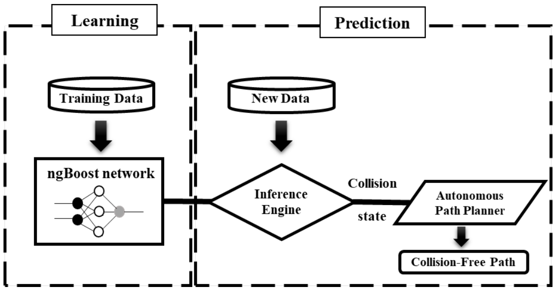

Fig. 1. The diagram illustrating the pipeline of CD using a ngBoost network.

bounding volumes to environment objects. The physics engine subsequently calculated the movement of the geometries with a transformation matrix described in earlier sections and provide a collision status. We labeled the collision status with True and False values. For this case study, we obtained 300,000 data samples.

B. Training

For the training process, we have utilized the Stanford ML group algorithm for the objective learning model. In their article Duan et al. [3] presents Natural Gradient Boosting (NGBoost) algorithm for probabilistic prediction via gradient boosting. The gradient boosting algorithms present novel estimation models especially applicable for CD. The main objective to select these models is the quality of performance in comparison to other learning approaches. Finally, we designated 50 percent of the data for training and 50 percent for tests in the learning stage.

V. RESULTS

After the training process, we provide 150,000 configurations for testing the performance of the network. The classical mathematical CD computation time is 625 checks per minute. However, after the training process, the model computes approximately 938,477 collision checks per minute. With this model, the computation speed increases approximately by 1000 times. Fig.3 illustrates the ROC curve for the predicting model.

VI. CONCLUSION

In this work, we illustrated the approximation of the collision behavior of an environment. This learning model is utilizable by other path planning algorithms to obtain a collision-free path. The main contribution of this work is bypassing the collision checking computations using a learning approach within a joint space state similar to what is customary in the industry. The obtained model abridges the CD process by reducing the inference time and facilitating real-time applications. Nonetheless, we welcome investigating results with optimizing algorithms or other innovative techniques, especially with more dynamic case studies.

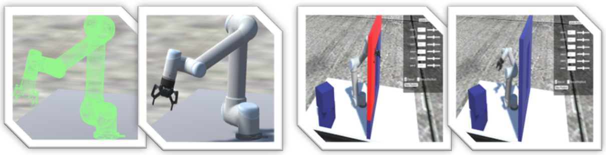

Fig. 2. UR5 Collision Geometries (left images) versus Robot and Environment (right images).

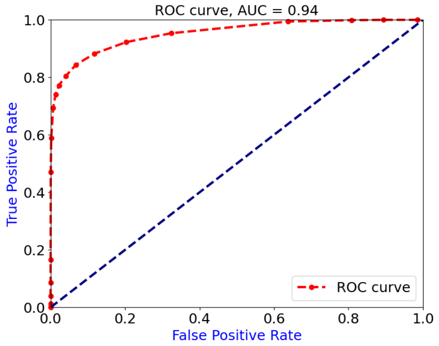

Fig. 3. Illustration of the receiver operating curve for the ngBoost network model.

REFERENCES

[1] Bryan Bourg, Bywalec. *Physics for Game Developers: Science, math, and code for realistic effects.* ” O'Reilly Media, Inc.”, 2013.

[2] Peter Corke. *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised*, volume 118. Springer, 2017.

[3] Tony Duan, Avati Anand, Daisy Yi Ding, Khanh K Thai, Sanjay Basu, Andrew Ng, and Alejandro Schuler. Ngboost: Natural gradient boosting for probabilistic prediction. In *International Conference on Machine Learning*, pages 2690–2700. PMLR, 2020.

[4] Christer Ericson. *Real-time collision detection*. CRC Press, 2004.

[5] Ignacio García, José D Martín-Guerrero, Emilio Soria-Olivas, Rafael J Martínez, Silvia Rueda, and Rafael Magdalena. A neural network approach for real-time collision detection. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 5, pages 5–pp. IEEE, 2002.

[6] Esmitt Ramírez, Héctor Navarro, Rhadamés Carmona, and José Dos Ramos. Optimizing collision detection based on obb trees generated with genetic algorithm. In *IV Iberoamerican Symposium in Computer Graphics-SIACG*, pages 1–7. Citeseer, 2009.

[7] Hong Yu Wu, Zhi Meng Shu, and Yong Guang Liu. Study based on hybrid bounding volume hierarchy for collision detection in the virtual manipulator. In *Applied Mechanics and Materials*, volume 454, pages 74–77. Trans Tech Publ, 2014.

[8] Jing Zhao and Zhongke Li. A collision detection algorithm based on ga. In *2010 WASE International Conference on Information Engineering*, volume 1, pages 234–236. IEEE, 2010.