Robotic Soccer Goalie: Evaluation of Learning and Physics-Based Controllers for Dynamic Tasks

Daniel Prakah-Asante
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, USA
doprakah@mit.edu

Aileen Liao
Department of Electrical Engineering and Computer Science
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, USA
ai0liao@mit.edu

***Abstract*—Various robotic controllers are currently employed in manipulation tasks. In highly dynamic systems, the limitations of the robot's bandwidth and external factors, such as visual impairment, impede its ability to acquire real-time information from the environment. Therefore, robust predictive systems are crucial for effective decision-making. Our work examines multiple predictive controllers on Boston Dynamic's Spot® Robot acting as a goalkeeper in a simulated soccer environment. Soccer represents a highly dynamic environment where accurately predicting unknown factors, such as ball speed, spin, and location, is essential for the goalkeeper to successfully block the ball. We evaluated two classes of predictive controllers: (1) physics-based models such as pure physics and sampling controllers, and (2) neural network-based controllers, including Recurrent Neural Networks (RNN), Gated Recurrent Neural Networks (GRU), and Neural Ordinary Differential Equations (ODE-LSTM). The physics-based sampling controller exhibited the highest performance, achieving a block rate of 36%. The ODE-LSTM, which combines neural network approaches with physics assumptions, performed comparably to our physics-based methods. It achieved a block rate of 28%, while also providing a more generalized approach to dynamics modeling. This work can help guide the selection of controllers in comparable dynamic environments.**

***Index Terms*—Neural Networks, Predictive Models, Ordinary Differential Equations, Robots**

I. Introduction

Soccer poses a distinct challenge for goalkeepers, who must demonstrate exceptional reaction speeds, accurate ball tracking, and sophisticated predictive abilities. In professional soccer matches, goalkeepers face shots from distances exceeding 11 meters. The speed of these shots for professional players typically averages around 112 km/h (70 mph) [1]. This scenario leaves goalkeepers with a brief reaction time of around 400 milliseconds to make a crucial save. In our work, we replicated these challenging conditions in a robotic environment.

Robotics professionals utilize various controller strategies to tackle diverse manipulation challenges. Among these, dynamic and accurate manipulation emerges as a particularly complex task. To achieve high levels of accuracy in robotic control, engineers often need to compromise and sacrifice adaptability for precision.

Controllers designed for precision and speed operate to maximize the exactness of the robot's movements. The industry standard for precise and fast controllers in factory robots remains centered around position or impedance-based controllers. These controllers exhibit rigidity, failing to adapt to minor environmental changes such as slight object displacements. Their lack of sensing and robustness against environmental changes hinders factory robots from undertaking more dynamic, mobility-based tasks and collaborating effectively with humans.

Learning-based controllers prioritizing adaptability excel in environments with constant changes. Engineers of these systems focus on enabling diverse decision-making with generalized approaches, allowing robots to adapt to new situations in real-time. However, such adaptability often compromises precision and speed, rendering these controllers less suitable for tasks necessitating exact movements.

In this work, we examine the capabilities of robotic systems using predictive controllers in dynamic environments, aiming to maintain a balance between speed and precision. We developed various controllers for the Boston Dynamics Spot® Robot, equipping it to anticipate and intercept high-speed soccer shots. This endeavor explores the limits of robotic agility and precision and investigates potential applications in other high-speed interception scenarios, including autonomous vehicles, object-tracking systems, and industrial automation.

II. Prior Works

Current methodologies in robotic control encompass both physics/model-based and learning-based approaches.

Physics-based approaches, such as the sampling-based model-predictive control (MPC) discussed in STORM [2], heavily depend on high bandwidth. In the MPC framework, the system rapidly solves simplified optimization problems at each iteration. These optimizations incorporate assumptions about dynamics and aim to minimize a cost function. This function typically includes considerations for the end goal, avoidance of joint collisions, and other relevant rewards. The strength of this approach lies in its ability to make real-time adjustments based on a predefined physical model, offering a robust solution for tasks requiring high precision. However, they are not able to adapt as well to uncontrolled or new environments.

Learning-based approaches are gaining prominence due to their adaptability and potential for handling complex tasks. In

[3], advancements in recurrent neural networks (RNN) showcase their application in controlling end effector positions. This paper emphasizes minimizing positional errors during repetitive motions, significantly enhancing accuracy for potential factory applications. This approach is particularly useful in environments with variability and unforeseen challenges, where traditional model-based methods may fall short. Overall, learning-based approaches can be less precise and slower due to its more generalized nature.

Some hybrid approaches have successfully incorporated elements from physics-based and learning-based controls, aiming to leverage the strengths of each. A notable example is the work of Hasini et al., who introduced the Liquid Time-Constant Networks (LTC) [4]. LTC's are a specialized form of ordinary differential equation-based controllers. They integrate the adaptability capabilities of neural networks with the structured, physics-informed approach of differential equations.

LTC's allows for more dynamic adjustments and learning in real-time, moving beyond the constraints of discretized predefined models or extensive training data. The LTC controllers are particularly adept at handling scenarios where the environment or the task dynamics are continuously changing. By utilizing a continuous-time framework, they can process temporal information more efficiently and with greater precision than traditional discrete-time models. In addition, the integration of differential equations, commonly used in physics-based models, not only enhances the controller's performance in complex environments but also improves its interpretability and reliability.

Our work investigated the strengths of the physics and learning approaches in an unpredictable, dynamic setting.

III. Approach

A. Simulation

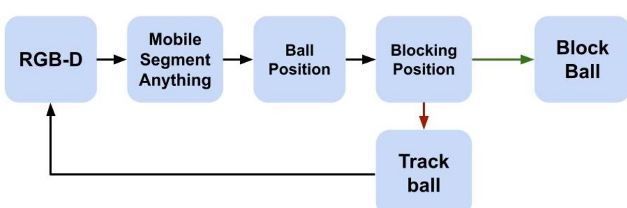

Fig. 1. Workflow. The RGB-D image received from the hand and segmented with mobile segment anything. From this, the ball and consequently Spot's blocking position are calculated. If spot is close to the ball, it moves to block. Otherwise, Spot continues to track the ball and receive camera outputs.

To evaluate each controller, we created a simulation environment within the Drake framework [5]. This environment encompasses attacking and defending players, a ball, Spot® robot, and the goal. Each simulation episode begins with three attackers, colored red, positioned 17 to 20 meters from the goal. These attackers progressively advance toward the goal, executing random lateral movements. The attacker with the ball dribbles randomly within a half-meter radius of their feet. Furthermore, the attackers pass the ball among themselves at random intervals.

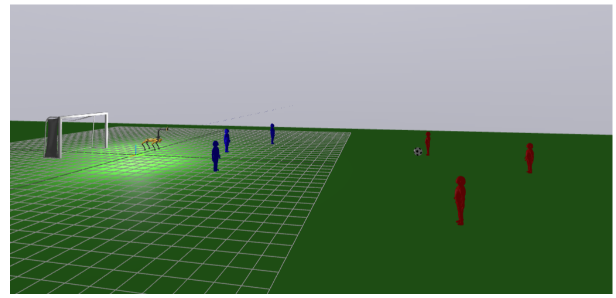

Fig. 2. Attacker Shooting. The field setup includes red attackers and blue defenders. The attacking player is posed to shoot at the goal.

When the attackers [Fig. 2] reach a distance randomly selected between 8 and 13 meters from the goal, they shoot by imparting an impulse to the ball. This action propels the ball at speeds varying between 20 m/s to 32 m/s (approximately 44 mph to 71 mph). Additionally, we manipulate the trajectory by applying forces that simulate the effect of a player adding spin to the ball. These forces curve the ball's path and accelerate it to its highest speed as it nears the goal. The unpredictability of this scenario and the limited reaction window increase the predictive challenge for the controller. The defenders, positioned 6 to 8 meters from the goal, act as visual obstacles, obstructing Spot's view without making contact with the ball.

The states within each episode, passing, dribbling, and shooting under conditions of occluded vision and external spin, enable us to evaluate the controller's ability to detect and track the ball, its responsiveness in movement, and its resilience to external influences.

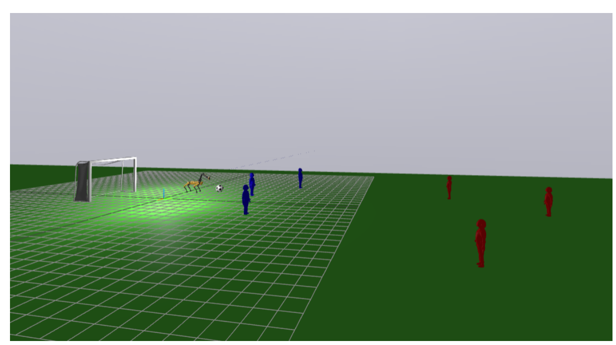

Fig. 3. Spot blocking ball with hand. This phase of the simulation depicts Spot computing its optimal block location after the attackers have shot.

Spot begins each episode positioned at the center of the goal, which is 8 meters wide and 2.5 meters tall. Its movement is restricted within the goal width boundaries and up to 3 meters in front of the goal. Initially, Spot operates in 'tracking mode,' utilizing an RGB-depth camera in its hand and the Mobile Segment Anything algorithm [6] to estimate the ball's position [Fig. 1]. In this process, we select two white or black points from the RGB image to input into Mobile Segment Anything, which segments the ball. From the segmented image, we calculate the ball's center. By combining data from the depth camera with Spot's internal estimations, we use the pinhole camera model to predict the ball's position in the real world.

Fig. 4. Spot blocking ball in Simulation. The close up image depicts Spot with the camera visible in its open hand.

Once Spot estimates the ball's position, it adjusts its position [Fig. 3] to align with the ball's lateral movement. Tracking mode is maintained until the predictive controller, utilizing the history of the ball's locations, identifies a suitable target within Spot's operational range to block. Upon receiving this directive, Spot actively maneuvers its hand towards the target, aiming to block the goal. Spot executes the hand movement at a maximum speed of 2 m/s. We define a successful block as Spot either stopping the ball or deflecting it out of bounds [Fig. 4].

We calculate Spot's movement commands using a differential inverse kinematics controller. This controller ensures precise and efficient movements by calculating the necessary joint angles to achieve the desired hand position and orientation. The differential inverse kinematics approach enables Spot to respond quickly and accurately.

B. Controllers

We developed multiple controllers for Spot® to detect and block high-speed soccer balls in real-time, using a single RGB-depth camera located in Spot's hand. Through exploration and evaluation of various controllers, we identified approaches that optimize performance across these three critical facets. For our environment setup, we confined ourselves to using only the portable camera located in Spot's hand.

We evaluated 5 controllers **Physics Based, Sampling Based, ODE-LSTM, GRU, RNN** and examined 2 baselines **Tracking, Random**.

TABLE I
CONTROLLER CATEGORIES

Approach	Controllers
Baseline	Random, Tracking
Physics-Based	Pure Physics, Sampling
Neural Network	RNN, GRU
Hybrid	ODE-LSTM

TABLE II
CONTROLLER OVERVIEW

Controller	Description
Random	Baseline performance. Spot moves randomly within goal bounds.
Tracking	Baseline performance. Spot is constantly tracking and moving laterally toward the last ball position.
Pure Physics	Predicts a trajectory of the soccer ball using classical mechanics and the last 2 ball locations computed from the hand camera. Uses the trajectory to predict the next location of the ball.
Sampling	Samples over a range of trajectories given different potential spin conditions. This approach requires the last 3 ball locations computed from the hand camera. Trajectories are computed using the 3rd to last and 2nd to last ball positions. An optimal trajectory is chosen by selecting the trajectory with the minimum distance from the last ball position.
ODE-LSTM	Given the last 3 ball locations and times, the neural net estimates dy/dt of the ball at the last time point. Using Euler approximation, a trajectory is calculated by integrating from the last known position to .5 seconds in the future.
GRU	Given the last 3 ball locations, the Neural Net predicts a blocking position.
RNN	Given the last 3 ball locations, the Neural Net predicts a blocking position.

Baseline

Tracking The tracking controller commands Spot to laterally move toward the ball's last known position, leading to delayed tracking. In the beginning, all other controllers also operate in this mode. This approach, however, faces limitations due to bandwidth speed and obstructing defenders, potentially rendering the ball's last known positions inaccurate.

Random The random controller detects the state to be shooting and selects a random target using a uniform Gaussian distribution scaled to the size of the goal constraints.

Physics Based

Pure Physics The physics-based controller operates by utilizing the last two known coordinates of the ball, determined through the analysis of camera images. It computes the ball's velocity based on the positions of these points and the time delay between camera frames. To predict the ball's next location, we employ the Euler approximation method over a small time interval (dt=0.05 seconds), spanning a maximum prediction horizon of three frames. This approach is based on the principle equation for the parabolic trajectories of flying objects.

$$p_f = p_i + v_0 * t + 1/2 * g * t^2$$

where p_f=final position, p_i= initial position, v_0= velocity, g= gravity, and t=time. The predicted position is commanded to Spot if the position is within a Spot's operational range.

Sampling The sampling physics-based controller processes the ball's last three coordinates, determined through camera images. It uses the second and third points to compute physics-based trajectories, and the most recent point to select the optimal trajectory from the generated samples, taking spin forces into account. From five samples, the controller calculates velocities and accelerations for given target positions, incorporating Gaussian random noise scaled to the goal size. We compute the trajectory similarly to the standard physics-based approach.

From the five samples, the optimal trajectory is selected as the trajectory predicting the minimum distance to the most recent computed coordinate of the ball. This minimizing objective can be written as:

$$\min_{i=1...5} \|f_i(x) - x_r\|$$

where x_r is the most recently computed coordinate of the ball and f_i is the trajectory function.

The sampling method distinguishes itself from the physics-based approach by exploring a variety of potential trajectories. By generating multiple trajectories, this method assesses each one and selects the most optimal path for the given situation due to varying spin directions and magnitudes.

Neural Nets

Data from Spot, operating in tracking mode, was collected over 400 episodes. This collection simulates the actual information Spot will receive from its cameras. We generated a supervised learning model data set with 1500 data points by incorporating Spot's movement limitations and accurately known ball trajectories to give the "best" blocking position.

RNN The Recurrent Neural Network (RNN) processes the last three known positions, and the Neural Network predicts an optimal blocking position. This RNN consists of three layers with 50 hidden units each. Its final state is then input into a feed-forward neural network comprising 10 and 3 units for output prediction. The model employs a mean squared error (MSE) loss function that utilizes an Adam optimizer with standard parameters (β_1= 0.9 and β_2 = 0.999) and was trained individually until convergence.

GRU The Gated Recurrent Neural Network [7] (GRU) was trained using the same methodology as the RNN, but utilized gated recurrent unit cells. This design choice differentiates the GRU from the standard RNN, as GRUs incorporate gates to regulate the flow of information. These gates help the model retain important information over longer periods and forget the unnecessary data, making it particularly effective for tasks that require understanding and maintaining context over time. The use of GRU cells enables more sophisticated handling of temporal dependencies, potentially leading to more accurate predictions in scenarios where the sequence of data is crucial.

ODE-LSTM The Neural Ordinary Differential Equations [8] Long Short-Term Memory [9] (ODE-LSTM) network underwent a unique training process compared to other neural networks. This LSTM processes the last three known positions of an object, along with the timestamps of these positions. It is designed to predict the rate of change of position, denoted as $\frac{dy}{dt}$, at the last known time, using this prediction to integrate and forecast the object's subsequent locations.

Mathematically, this process is represented as:

$$\hat{y}_{t+\Delta t} = y_t + \int_t^{t+\Delta t} f(\theta, y, t)\, dt$$

In this equation, $\hat{y}_{t+\Delta t}$ symbolizes the anticipated final position of the object at time Δt from the most recently computed location, y_t. The function $f(\theta, y, t)$ within the integral is the core of the LSTM. We employ an Euler approximation for practical computation of these integrations, simplifying the calculation of the gradients of the neural network parameters.

Despite mirroring the structure of GRU and RNN models in terms of overall architecture, the ODE-LSTM distinguishes itself by using LSTM cells, effective for handling long-term dependencies in data sequences. This structural choice equips the ODE-LSTM to proficiently process and predict sequential data with complex temporal dynamics.

By comparing varying controllers in the same simulation environment, we create a comprehensive understanding of the trade-offs associated with both physics-based and ML-driven approaches, ultimately guiding us in determining the most efficient and effective controller for Spot's real-time soccer ball blocking capabilities.

IV. Evaluation

Each controller was evaluated over 50 episodes. The percentage of successful blocks were recorded.

Overall, the sampling controller exhibited significantly better performance, achieving a block rate of 36%, compared to all other controllers. The ODE controller was on par with the other physics-based controllers, attaining a block rate of 28%, which was better than the pure physics controller (24% block rate). This suggests that integrating physics principles, such as motion dynamics and force interactions, enhances the predictive accuracy and effectiveness of the controllers in dynamical systems.

The neural network-based approaches were unable to outperform the baseline controller in tracking tasks. Moreover, the RNN performed even worse than random tracking. This outcome suggests that the neural network models, particularly

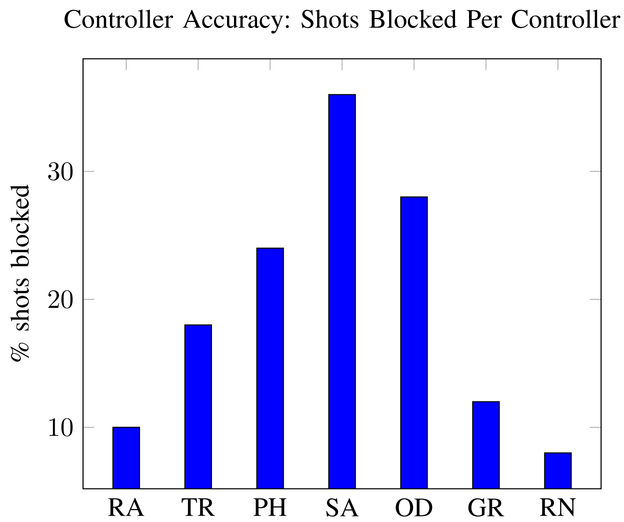

Fig. 5. Comparison of controller accuracy in terms of percentage of shots blocked per controller. Includes RA for Random, TR for Tracking, PH for Physics, SA for Sampling, OD for ODE-LSTM, GR for GRU, RN for RNN

the RNN, may have struggled with the complexity and unpredictability inherent in the tracking tasks. One potential reason could be the difficulty neural networks face in capturing the dynamic and rapidly changing conditions of the environment, especially when compared to more specialized or simple controllers.

Our data suggests a notable enhancement in the predictive accuracy and effectiveness of controllers when integrating physics principles, such as motion dynamics and force interactions. Physics-based methods, which explicitly account for forces and apply classical mechanics principles, contrast starkly with neural networks. Neural networks, not bound by concrete physical assumptions, often explore a range of possible ball trajectories, some of which may not align with physical reality. In dynamic and fast-paced environments, the grounded assumptions of physics-based methods enable quicker determinations of blocking positions, leading to more successful goal blocks. Physics models balance the solid theoretical underpinnings of physics with the flexibility of algorithmic predictions, yielding more efficient and precise outcomes in dynamic tasks like blocking soccer balls.

Additionally, while the sampling-based method proved superior, the performance of the ODE suggests that it could be an effective controller for dynamic situations where the physics of the environment is not well understood. Neural ODEs, being more generalized, are particularly suited for tasks in which the inherent physics of the environment is less known, offering a robust alternative to traditional physics-based controllers.

V. Conclusion

Physics based approaches performed better in dynamic situations due to its abilities to predict realistic trajectories. Neural ODEs performed similarly to the pure physics controller and may be useful for situations with more unknown dynamics.

This investigation provided insights into the comparative advantages and disadvantages of physics-based methods, which rely on predicting accurate movements based on discrete shot frames, in contrast to the continuous tracking approaches dependent on machine learning (ML).

Our findings are promising in guiding controller selection for dynamic tasks that require high degrees of accuracy, such as in search and rescue operations. These insights can be generalized across various robot platforms, including legged and mobile-based robots. More broadly, this research advocates for the development of more dynamic and adaptable robots, paving the way for closer collaboration with humans. Controllers capable of faster and more dynamic reactions, higher precision, and greater adaptability will make robots safer for interaction in uncontrolled environments like restaurants and homes.

The limitations of this study are that all controller evaluations were assessed in simulation only. Factors such as ground friction, complex spin patterns, and inertia were estimated and may not fully reflect hardware performance. Block checking may also not account for edge case ball blocking scenarios. Lastly, game simplifications such as the ball being allowed to pass through defensive players during shooting do not reflect real-world scenarios.

In future research, we aim to delve deeper into Continuous Neural Networks for estimating dynamics across various time samplings, with a particular focus on Liquid Time-constant Networks. We also plan to explore the potential of combined methodologies, specifically investigating multimodal-sampling-based approaches. Lastly, we intend to conduct evaluations of the controller results, comparing their performance in both simulated and real-world settings. This will provide valuable insights into the practical applicability and effectiveness of our findings.

Acknowledgment

We thank Professor Russell Tedrake, Ethan Yang, and the 6.4210 staff for their guidance with project formulation and help with implementing simulations in Drake.

References

[1] B. Clayfield, “How fast is a soccer ball kicked?,” Your Soccer Home, https://yoursoccerhome.com/how-fast-is-a-soccer-ball-kicked/ (accessed Dec. 11, 2023).

[2] M. Bhardwaj et al., ‘STORM: An Integrated Framework for Fast Joint-Space Model-Predictive Control for Reactive Manipulation’, in Proceedings of the 5th Conference on Robot Learning, 2022, vol. 164, pp. 750–759.

[3] Z. Xie, L. Jin, X. Luo, Z. Sun and M. Liu, ”RNN for Repetitive Motion Generation of Redundant Robot Manipulators: An Orthogonal Projection-Based Scheme,” in IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 2, pp. 615-628, Feb. 2022, doi: 10.1109/TNNLS.2020.3028304.

[4] R. Hasani, M. Lechner, A. Amini, D. Rus, and R. Grosu, “Liquid Time-constant Networks”, AAAI, vol. 35, no. 9, pp. 7657-7666, May 2021.

[5] R. Tedrake and T. D. D. Team, ‘Drake: Model-based design and verification for robotics’. 2019.

[6] C. Zhang et al., ‘Faster Segment Anything: Towards Lightweight SAM for Mobile Applications’, arXiv preprint arXiv:2306. 14289, 2023.

[7] K. Cho et al., ‘Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation’, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1724–1734.

[8] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, ‘Neural Ordinary Differential Equations’, in Advances in Neural Information Processing Systems, 2018, vol. 31.

[9] S. Hochreiter and J. Schmidhuber, ‘Long short-term memory’, Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.