A Case Study on Inverse Optimal Control and Model Predictive Control for Autonomous Decision Making Under Timely Distributional Shift

Fredson Silva de Souza Aguiar

November 4, 2024

Abstract

Optimal Control (OC) Theory is an area of optimization responsible for finding a control strategy under a dynamical system, and has shown good results in many fields. However classical OC approaches rely on knowledge of the system and costs, and can become intractable under uncertain or long term scenarios, rendering its not being ideal for some highly uncertain real world tasks, such as piloting or driving. In this work, we propose the study of an Autonomous Moving agent under environmental Timely Distributional Shift from an Inverse Optimal Control (IOC) and Model Predictive Control (MPC) perspective. We discuss how these techniques can be used in order to tackle such problem, offering good generalization properties while being an adaptable and computationally tractable framework.

Keywords: Autonomous Decision; Inverse Optimal Control; Model Predictive Control.

1 Introduction

In the last years, society has seen a steep advance in the use of new machine learning techniques in a variety of fields, from image recognition, chatbots such as ChatGPT [Ray23], and even playing complex games, such as chess and go [Ber22]. Success examples like these explicit the utility of such techniques, and it's potential to replace humans in even more challenging activities.

While some of those seem innocuous, there are many cases where a decision involves harmful or unexpected results, specially when involving human-machine interaction and real world deployment, including autonomous driv-

ing, drone piloting and other complex tasks [Lab24, Ami24]. In this sense, the study of reliably robust methods for such tasks becomes paramount. As an example, in 2017 [LMK+17] proposed a suite of environments illustrating a set of desirable agent properties, such as adversarial and ill-specification robustness, safe interruptibility, safe exploration, among others.

In this direction, researchers have advocated for the importance of bridging technical and practical aspects of Machine Learning and mathematical tools such as Optimal Control Theory [BP07, Lib11], to discuss emerging topics involving machine learning integration in society [Zua, Ber19, Ber23].

In fact, the study of Optimal Control Theory has shown positive outcomes in ensuring properties and safe guarantees regarding decision making but, sometimes, lack computational and theoretical tractability, specially in the cases of long-term problems or uncertain scenarios. On the other side, learning-based techniques can introduce adaptability and generality to model-based control [BGH+21].

In light of this discussion, we propose the study of conjoint techniques from the field of Optimal Control Theory focused on the problem of **robustness under distributional shift**: ensuring an agent behaves robustly when the test and training environments differ, inspired in [LMK+17]. We stress the previous problem by proposing it's time-dependent extension of **robustness under timely distributional shift**: ensuring an agent trained in static scenarios is robust when deployed in a continuously updated scenario.

More precisely, we discuss how Inverse Optimal Control (IOC) [ASD20] can be an efficient alternative for generalizable learning foundation, while Model Predictive Control (MPC) [HWMZ20] introduces online adaptability to environmental changes.

1.1 Organization

Section 2 introduces the problem being studied, our implementation of a simulation environment for such problem, and comments on alternative implementations that pose more realistic, complex implementations that could be useful in future studies for the similar problem.

Section 3 introduces learning Learning From Demonstrations (LfD) frameworks, formulates the Inverse Optimal Control (IOC) problem, how this model is a good alternative for generalizable learning from samples, previous approaches to such problem, and proposes a gradient-based method for IOC. The Model Predictive Control (MPC) problem is formulated in Section 4, where we discuss how this method poses an alternative for adaptive motion

planning, and discuss pros and cons, specially due to it's short-sightedness.

Finally, in Section 5, we discuss the end-to-end results, and possible future works.

2 Robustness Under Distributional Shift

The learning environments from [LMK+17] propose problems reflecting aspects of real-world safety problems, such as Reward gaming and Distributional shift. In these examples, each problem is static, which does not always reflect real world scenarios.

In that sense, distributional shift happens during interaction, and sometimes it reflects real-world interaction better than the static case. Agents such as cleaning robots or autonomous driving vehicles should be faced with environments that change over time: an individual might introduce a new breakable object while the robot cleans the room; or an unexpected vehicle may cut the line in front of the self-driving car.

We, therefore, propose the study of an extension of the previous problem by proposing it's time-dependent version: an agent learns under a time-distributed environment should be able to reliably act when the scenario it's deployed is updated under a different distribution. For this study, we focus on the extreme case of this **robustness under timely distributional shift**: ensuring an agent trained in static scenarios is robust when deployed in a continuously updated scenario.

Such proposition also imposes a data efficiency challenge: since the dynamics under time shift occur in a much bigger space, static data becomes sparse representations of the real dynamics to be faced; on the other direction, if an agent is able to learn, under a methodology, from static time samples, similar rationale suggests it's a data efficient methodology.

2.1 The Moving Agent Problem

An interesting example can be found in [Lab24]: in one of their studies, a flying drone has to move while avoiding collision with another drone that passes by. Inspired in this situation, we propose the following study problem: a **moving agent** has to move from it's starting position to a **moving target point** while avoiding a **moving hazard point** before a fixed time limit.

Interestingly enough, such simple problem imposes a considerable set of specification questions: *how much should the hazard be avoided? should the agent remain at a certain distance? should the agent opt to collide to reach the target in time?* Also, due to it's simplicity, results coming from

this study should easily generalize to more complex scenarios, such as those stated before.

2.2 Safe Learning Environment

To support such study, we implemented a Safe Learning Environment (see Figure 1). It's purpose is to be a simple implementation, easy do use and configure, compatible with the classical Reinforcement Learning environment suit Gymnasium[1]. In fact, our concept is a simplified version of the ongoing work Safety-Gymnasium[2], specially the Safe Navigation suit. Due to it's incomplete documentation, we decided to implement our own environment.

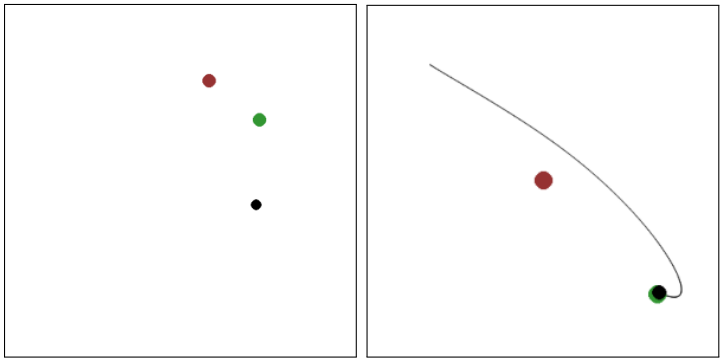

Figure 1: The Safe Learning Environment. From left to right: no trajectory traces and random initial positions; optimal trajectory for fixed target and hazard.

Currently, the implementation is publicly available. For a more complete documentation on installation an running examples, one may refer to our repository[3]. The suit features a single environment and consists of the simple task described as before, where the objects can move under directional acceleration:

- **Observation Space:** the observation consists of 6 (six) $2d$ vectors, representing respectively: agent position; agent velocity; target position; target velocity; hazard position; and hazard velocity. By standard, the positions and velocities are limited to the interval $[-1, 1]$.

[1] for reference, see: `https://gymnasium.farama.org/index.html`
[2] for reference, see: `https://safety-gymnasium.readthedocs.io/en/stable/`
[3] repository: `https://github.com/fredsonaguiar/safe-learning-environments/`

- **Action Space:** the possible actions consist o 3 (tree) $2d$ vectors, representing respectively: agent acceleration; target acceleration; and hazard acceleration. By standard, the accelerations are limited to the interval $[-1, 1]$.

- **Reward:** The standard reward is $-dt$ per step, where dt is the inverse of time discretization, meaning the agent receives a total negative loss of the time it took to complete the task.

It's important to notice that the standard **Reward** proposed above is only a baseline: in fact, the *ideal reward* is the central problem to be studied since defining the reward means defining the agent's beahaviour.

Finally, it's noticeable that such suit can be easily extended to include more complex tasks, such as variable number of objects, and more complex dynamics. For now, the main purpose of the implementation is to be simple and accessible.

3 IOC for Learning from Demonstrations

As previously stated, specifying how an agent *should* behave trough loss functions can become a prohibitively complex task. In fact, even for simple tasks, such that of our study, many specification question arise, and, in case there are no such questions, it can hard to *encode* the desired behavior trough parametrization [HMMA$^+$20].

In such context, it becomes more reliable and efficient to transmit intended behavior or intuitive expertise by providing a learning agent with (optimal) samples of the task, trajectories or state-control measures. This approach has given rise to different formulations of the same leaning problem that became know generally as Aprenticeship Learning (AL) or Learning from Demonstrations (LfD) [ASD20, RPCB20, LKWP24]. This way, increasingly more complex tasks have been achieved, such as autonomous helicopter aerobatics [ACN10], trajectory prediction for car racing [RMS$^+$22], and playing complex games [HVP$^+$17], among others.

3.1 Inverse Optimal Control

Inverse Optimal Control (IOC) in the problem of reconstructing an underlying Lagrangian function or, equivalently, learning the fundamental cost functions trough optimal trajectory samples. In general, such problem is formulated as finding a set of weights $(w_i)_{i=1}^n$ for a loss basis $(\phi_i)_{i=1}^n$ in order

to approximate the real loss f being optimized:

$$f(t,x,u,y) \approx \hat{f}(t,x,u,y,\hat{w}) = \sum_{i=1}^{n} \hat{w}_i \phi_i(t,x,u,y),$$

where $x(t)$ represents the current state, $u(t)$ represents the control, and $y(t)$ represents the environment states. One may refer to [ASD20] for a historical overview of the field and motivations, from classical inverse optimal control to more modern approaches. A deeper theoretical formulation for the problem can be found in [Mas18], including classical IOC as an stabilization problem, formulation trough geodesics, the discussion of injectivity and surjectivity of costs, and more.

The general IOC formulation has applications in many fields including biology, medicine and economics, and has shown special positive outcomes in engineering and robotics in fields such as autonomous robotics and human-robot interaction [RCR+19, RMS+22].

3.2 Related IOC Approaches

Different methodologies have been proposed in order to approach IOC problem, assuming different hypothesis about the associated optimal control problem.

In [MTL10] the authors propose bilevel formulation, by minimizing the deviation of computed optimal control paths, given a set of parameters, from measurements. They propose the technique to better understand human locomotion and generate more natural robotic humanoid movement.

As an interesting alternative approach in given in [JAB13], where the authors propose efficient parameter estimation by minimizing the amount to wich necessary conditions from *Pontriagyn's Maximum Principle* are violated. In a similar stile, the work [EVT17] proposed a more general approach that includes restrictions, by minimizing the violation from the KKT conditions. In both cases, the authors reformulate the problems under the form of well known Linear Quadratic Regulators (LRQ) problems, and reach good outcomes. Following from the previous ones, the work [CBO+23] presents a method to also recover state and control box constraint, under incomplete observations, from the Lagrange method.

In a more theoretical founded stile, [MZ18] studies convex formulations for IOC problems under an LQR law, offering uniqueness results as well as highlighting robustness under noisy data and computational efficiency properties.

In an alternative, authors have proposed a deep learning based method [FLA16] to solve an IOC problem in high dimensional and unknown dynamics, while avoiding extensive feature extraction.

3.3 Gradient-Based Formulation

Stochastic Gradient Descent (SGD) is a well-known iterative optimization method[4]. In SGD, one approximates the intended optimal parameter $w*$ by iteratively taking

$$w_{k+1} := w_k - \gamma \nabla L(w_k),$$

where γ is known *stepsize* or *learning rate*, and defines the how much the current step will follow the descending direction. This method usually works well for functions under suitable hypotheses and is frequently used in training neural networks and other weight adjustments.

Our proposition follows a similar approach from previous works, considering a *total observed trajectory error*, $L(w)$, given by

$$L(w) := \sum_{i=0}^{k} l_i(w), \quad \text{where} \quad l_i(w) := ||x_i^{(w)} - x_i^*||^2,$$

that is, $l_i(w)$ is the i-th *observed trajectory error*, x_i^* is the i-th optimal trajectory observation and $x_i^{(w)}$ is the i-th optimal trajectory under the loss defined by w starting from the same initial condition as x_i^*. Ideally, under the same initial conditions and real parameters $\bar{w}$, the optimal trajectories $x_i^{(\bar{w})}$ should coincide with the observation x_i^*, therefore reaching the global minimum $L(\bar{w}) = 0$. This way, our method consists of finding the optimal parameter w^* that minimizes L.

We assume the mapping L to be (sub)differentiable. The idea is that we should be able to differentiate the observation errors with respect to the parameters in order to find the global optimal point from a gradient-based method. More precisely, we propose the use of the SGD method to iteratively minimize $L(w)$.

As an important observation, Figure 2 shows how the errors are structured around the real weights. In a real scenario, one would not have access to the real weights prior to the investigation, but this suggests the aforementioned hypotheses are reasonable.

A main issue in our proposition is that we don't have access to the gradients ∇L and, in reality, this function may be prohibitively complex

[4]for reference, see: `https://en.wikipedia.org/wiki/Stochastic_gradient_descent`

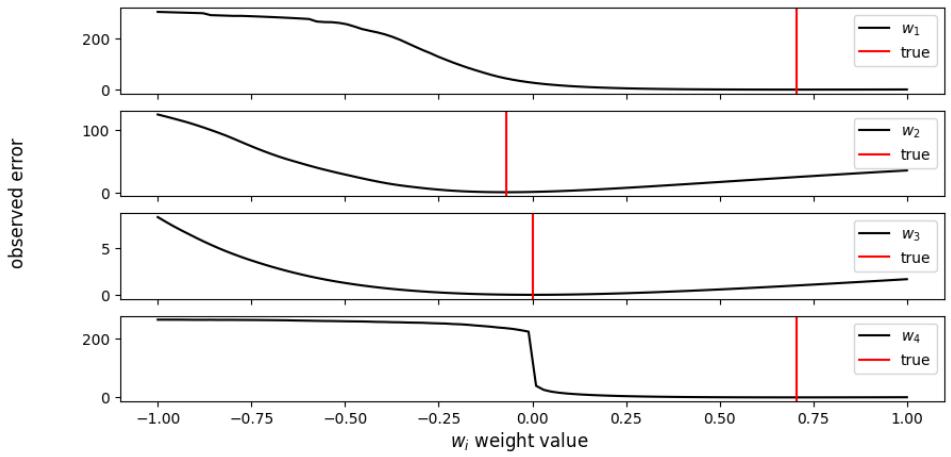

Figure 2: The observation error as the weights vary in the interval $[-1, 1]$. For this exemplification, each of the other weights assume their true values.

or impossible to represent in an analytical form. An interpretation to this function follows from

$$\nabla L(w) = \sum_{i=0}^{k} \nabla ||x_i^{(w)} - x_i^*||^2 = \sum_{i=0}^{k} 2(x_i^{(w)} - x_i^*)^T \nabla x_i^{(w)},$$

meaning that one would have, effectively, access to the variation of the optimal solutions $x_i^{(w)}(t)$ with respect to variations in the weight parameters w or, equivalently, know how the weights locally define the solutions to the optimal control problem.

Our way to solve this issue is by considering $\nabla L(w) = \left(\frac{\partial L(w)}{\partial w_1}, ..., \frac{\partial L(w)}{\partial w_n}\right)$. This way, we can approximate each of the partial derivatives from a symmetric difference quotient, so we take

$$\frac{\partial L(w)}{\partial w_i} := \frac{L(w + \epsilon e_i) - L(w - \epsilon e_i)}{2\epsilon}$$

where $\epsilon > 0$ is a small variation size, and e_i is the i-th canonical vector, representing a variation added to the i-th weight w_i while maitaining the others fixed. Of course, this means we need to solve the direct optimal control problems to find each of the optimal solutions $x_i^{(w \pm \epsilon e_i)}$ in order to calculate the losses.

3.4 Cost Learning for Moving Agents

In our example, as described in Section 2, we are interested in learning a cost function that governs the movement of an **agent** moving towards a **target** while avoiding a **hazard**; the **agent** it could, as well, avoid or incentivize other features such as limited acceleration due to fuel costs. Following [EVT17], we assume the cost basis terms given by the sum of weighted squared features. We chose the relevant features as follows: ϕ_1 - distance from **agent** to **target**; ϕ_2 - distance from **agent** to **hazard**; ϕ_3 - **agent**'s velocity; and ϕ_4 - **agent**'s acceleration.

For the learning experiment, we chose the weights (w_1, w_2, w_3, w_4) to represent the intended behavior, and generate optimal trajectory observations from random initial positions, maintaining **target** and **hazard** stationary as proposed.[5]

In some LfD problems, the trajectories may not be measurable or compatible with the learning agent. For instance, if a robot learns to imitate human walking from examples, the relevant observation and costs may include only the compatible measures (e.g, leg and feet positions and velocities) leaving out unmeasurable quantities (e.g, humans inner muscular tension levels) and incompatible control mechanics (e.g, articulation motor torques and velocities).

In the investigation, this means that the observed trajectories or, equivalently, the observation losses, x_i may include the **agent**'s position and velocity, but also it's control function, the acceleration. As being so, we compare results obtained by the gradient-based method in the two cases. Figure 3 shows the training logs: the observation loss and weights reaches the desired tolerance earlier (taking around 900 training steps) when including the accelerations in comparison to the other case (taking around 2000 training steps). This may have to do with the fact that, in the first case, more information is used to the adjusted loss function. In both cases, we reach an error between the real and learned weights of the order 10^{-3}, a good approximation.

The good parameter approximation, as well as good losses, obtained after training suggests that the agent was able to learn the intended behavior from observations; this means that the actions taken by the trained agent reflect (or approximate to a good extent) those from the optimal trajectory samples.

However, we must recall the learned actions and trajectories have been taken and trained under static environments, what does not reflect our in-

[5] real weights considered: $w = (10, -1, 0, 10)$, before normalization.

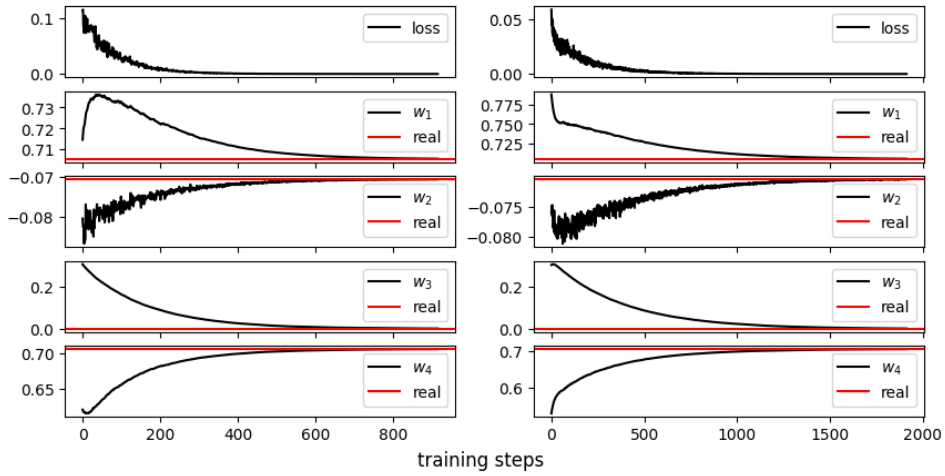

Figure 3: Training from same random initial conditions w_0 until error tolerance of 10^{-6}. 1000 samples, $5s$ duration, batchsizes 20, $\delta = 0.01$, $\gamma = 0.05$. Losses include acceleration (left); losses leave out acceleration (right).

tended problem. In next session, we discuss how to extend the learned static-time behavior to timely adaptive motion.

4 MPC for Adaptive Autonomous Motion

Model Predictive Control (MPC) is an optimal control technique that allows one to replace an analytically or computationally potentially intractable Optimal Control Problem by a simpler version. It provides a more tractable version of an Optimal Control Problem by iteratively solving a simplified version of the control problem over a shorter horizon [HWMZ20].

Effectively, in MPC one takes the current optimal control strategy as being the first step of an optimal control problem starting from the current state to a predefined time horizon:

$$\begin{aligned} \min_{U} \quad & \sum_{i=0}^{N} l_{mpc}(x_{i|k}, u_{i|k}, i+k) \\ \text{s.t.} \quad & x_{i+1|k} = f_{mpc}(x_{i|k}, u_{i|k}, i+k), \\ & U = \left[u_{0|k}, ..., u_{N|k}\right] \in \mathcal{U}_j, j = 1, ..., n_{c_u}, \\ & X = \left[x_{0|k}, ..., x_{N|k}\right] \in \mathcal{X}_j, j = 1, ..., n_{c_x}. \end{aligned}$$

Then $\pi_{mpc}(x_k, u_k) = u^*_{0|k}$, the k-th first step of the optimal solution to the k-th receding problem provides an adaptive feedback control strategy. Those methods have recently been used to ensure safer learning, where some security restriction are enforced in the learning process [BGH$^+$21].

4.1 Related MPC Applications

Many applications on MPC discussing adaptability or Safe Learning and robotics can be found on literature, and rather justify our next discussion on it's application for autonomous motion under dynamic environmental update. One may refer to [HWMZ20] for a greater discussion on MPC and safety.

Interesting examples of MPC in engineering and robotics are discussed in [Lab24, Ami24], including legged robots, autonomous driving, collision avoidance and technical discussions, such as controllability and safety verification from reachable sets, among others.

An approach for safety is Model Predictive Safety Certification (MPSC), explored in [WZ19, WZ21, GHdRD24]. In MPSC, proposed trajectories (e.g., from an optimal controller or reinforcement learning agent) are checked online and filtered or modified in order to meet security restrictions.

Approaches focused on adaptability are discussed in [BZTB21, KKS$^+$20, LCA19] by proposing online parametric adaptation through set membership methods or polytopic approximation. These techniques have shown positive outcomes under scenario uncertainty.

Other approaches such as [ME23] are focused on offering strong theoretical guarantees for scalability and robustness under distributional shift, besides convergence results and feasibility. Usually, such approaches rely on a model-based hypothesis in order to obtain strong results.

4.2 Environmental Adaptability

In our study, we're interested in extending actions learned from static environments to dynamic ones. We claim that the adaptability property of an MPC-based feedback control is, itself, enough to solve the problem with minor adaptations due to computational tractability.

Our strategy is to consider the optimal control feedback given by MPC assuming the environment becomes static in the current state for the predictive horizon. After applying the following the chosen strategy step, the environment updates and the agent repeats the method for the next step. See Figure 4.

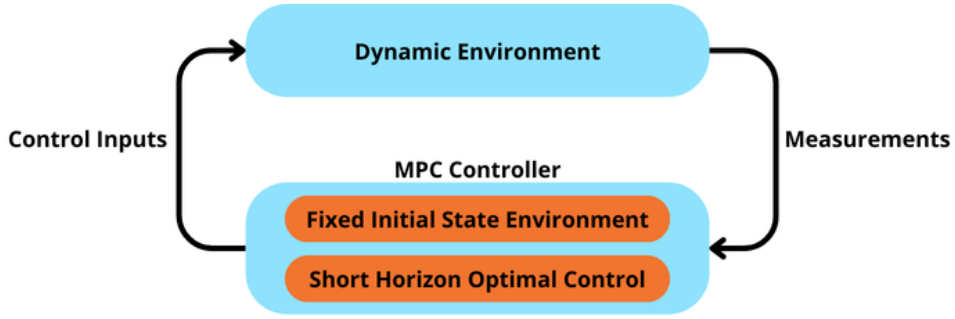

Figure 4: Proposed strategy: solve a short horizon optimal control assuming the environment remains static at current state; inputs control to dynamic environment; and refeeds the updated measures to the MPC controller.

However, in the problem proposed, two relevant issues arise: first, we need to adapt the final condition (the agent must reach the target in a given time) from longer time horizon to a series of short-horizon problems; and second, the agent must be able to recalculate it's route on real time in order to react to the environment, meaning we are only allowed to use a short prediction horizon. These issues are dealt in the following Subsections.

4.3 Forward Motion Planning

For the first issue, we follow the discussion on *Nominal Stability of NMPC* from [Die11] to relax the terminal constraints by replacing them with a sequence of terminal costs or inequalities that imply the intended stabilization at the terminal time.

For instance, one could include quadratic costs to incentivize the stabilization at the intended point, however, this does not guarantee finite time stabilization, and could directly interfere with the costs learned previously. Instead, we propose a sequence of finite time inequalities.

Precisely, without loss of generality, let $t_0 = 0$ be the task initial time, T be the terminal time for the complete problem, t be the current task time and h be the MPC time horizon; suppose also $t + h \leq T$ since the MPC horizon should not extend beyond the final time, otherwise take $h = T - t$. We must grant the distance $||a(T) - p(T))||$ from **agent** to **target** to be 0 at terminal time T. We, then, include the following inequality restriction to be satisfied by the MPC controller:

$$||a(t+h) - p(t+h)|| \leq \frac{T-(t+h)}{T}||a(t) - p(t)||.$$

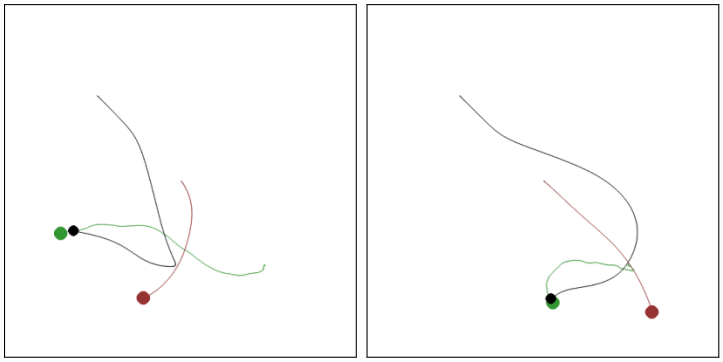

Figure 5: Trajectory results using MPC feedback controller for a $5s$ target reaching task, using a $1s$ time horizon.

Such inequality states that the motion is planned in order to make sure the **agent** approximates the **target** proportionally to the amount of time taken by the MPC task in relation to the amount of time left. Also, it's easily verified the the terminal time constraint is eventually satisfied, since near the terminal time we get $T = t + h$, thus

$$||a(T) - p(T)|| = ||a(t+h) - p(t+h)|| \leq \frac{T-(t+h)}{T}||a(t) - p(t)|| = 0,$$

satisfying the final time constraint as stated initially.

As Figure 5 shows, the implementation of such strategy works well even under dynamic environmental changes.

4.4 MPC Short-Sightedness

As issued before, in a real world scenario, the **agent** must be able to react to the environmental changes on time, generally without any computations. Also, the agent itself does not have access to extensive computational power but, rather, basic embedded hardware [Die11]. This motivates the need for efficient on-the-go control computation, and leads to a variety of computationally efficient algorithms.

Considering MPC methodology, this leads to an immediate trade-off on defining the time horizon: ideally, we should consider the maximum possible horizon in order to approximate the optimal stabilizing control; on the other side, the longer the horizon, more complex and time consuming is to solve the associated control; finally, the smaller the horizon, the less the feedback control approximates the intended optimal control in reality, in an effect called *short-sightedness*.

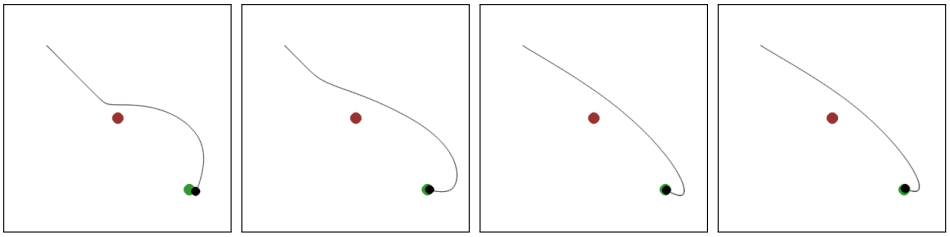

Figure 6: The moving agent problem fro the same initial conditions, static environment, total time $T = 5$. From left to right: MPC controllers with time horizons $h_1 = 0.5s$, $h_2 = 1s$, and $h_3 = 2s$; optimal controller (right).

Such effect suggests to the fact that the control accounts only for too immediate results, leaving aside long-term returns. Figure 6 exemplifies this effect: for the same static environment, the shorter horizon MPC controller (left) only accounts and reacts to the **hazard** when near collision; on the other side, the larger horizon MPC controller (3rd from left to right) approximated well the globally optimal control (right).

Finally, for our examples in Figure 5, the intermediate time horizon $h = 1s$ was chosen. In practice, we observed this horizon to represent well the longer horizon behavior while being computationally tractable on real time.

5 Results and Discussion

We were able to discuss how techniques of IOC and MPC can be used in approaching generalizable and adaptable control even under updating environments while maintaining the intended behavior. In our discussion, results presented show positive outcomes in solving the intended proposed problem of an autonomous **agent** moving while looking for a **target** and avoiding a **hazard**. Codes and simulation files generated in the discussion are are publicly[6].

It is important to mention that the techniques and environment employed in the discussion are fundamentally simple representations of much larger fields, what raises the importance of more profound discussions and opens the possibility of employing alternative methods. Some possible alternative works and techniques are presented below.

[6]see repository: `https://github.com/fredsonaguiar/ioc_mpc_timely_dist_shift`

5.1 Online Learning Techniques

Even when ensuring good adaptability properties to a controller, these can be limited by the uncertainty of real world environments, reducing it's performance. Therefore, learning for adaptability should, as well, occur online: for instance, a drone that must remain stable might learn to account for unforeseen wind direction during flight.

Works applying similar approaches have already been mentioned, such as [RMS+22, LCA19, BZTB21], among others. This kind of technique allows for more reliable control and better performance by improving it's inner predictive model.

5.2 Control Barrier Functions

When dealing with problems of safe exploration, safe online learning or *safety-critical control*, an area that has received increasing attention relies on the use of Control Barrier Functions (CBFs) [ACE+19, OST21]. The formalism of CBFs is inspired in Lyapunov stabilization, but extends this concept to *safe set* invariance, meaning the system tends to stay in a set. The advantage of this formalism is that it offers a clear definition of *safety* while also implying strong theoretical safety guarantees.

An application of this technique is presented in [COMB19], where the authors propose a combination of techniques brought from CBFs and RL to propose a framework that guarantees safety with high probability during the learning process while maintaining efficient exploration.

5.3 Alternative Environments

More complex environments might be discussed in order to better represent reality, including more complex dynamics and numerous features. A possible environment specific for safety tasks has already been introduced, the Safety-Gymnasium.

Alternatively, environments for *multi-agent* problems may be suitably discussed in order to make agents reliable under adversarial or cooperative games. For intance, the PettingZoo[7] interface offer a set of general multi-agent problems in a variety of environments with implementation for python.

On a different direction, PyFlyt[8] is a library publishing a set of UAV Simulation Environments, built on a powerful physics engine, and allowing

[7] for reference, see: `https://pettingzoo.farama.org/`.
[8] for reference, see: `https://taijunjet.com/PyFlyt/`

flexible personalization and configuration of the environment (e.g., wind velocities) and elements (e.g., drones, motors, camera). As standard, the library publishes a set of complex tasks, such as landing a falling rocket, and balancing a pole or catching a falling ball on a flying drone.

References

[ACE+19] Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications, 2019.

[ACN10] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13):1608–1639, 2010.

[Ami24] Amii. Ai seminar series: Mo chen, optimal control and machine learning in robotics (jan 8). `https://www.youtube.com/watch?v=fzvVN572OyI`, jun. 2024.

[ASD20] Nematollah Ab Azar, Aref Shahmansoorian, and Mohsen Davoudi. From inverse optimal control to inverse reinforcement learning: A historical review. *Annual Reviews in Control*, 50:119–138, 2020.

[Ber19] Dimitri P. Bertsekas. *Reinforcement Learning and Optimal Control*. Athena Scientific, 2019.

[Ber22] D. Bertsekas. *Lessons from AlphaZero for Optimal, Model Predictive, and Adaptive Control*. Athena Scientific optimization and computation series. Athena Scientific, 2022.

[Ber23] D. Bertsekas. *A Course in Reinforcement Learning*. Athena Scientific, 2023.

[BGH+21] Lukas Brunke, Melissa Greeff, Adam W. Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P. Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *CoRR*, abs/2108.06266, 2021.

[BP07] Alberto Bressan and Benedetto Piccoli. *Introduction to the mathematical theory of control*. American Institute of Mathematical Sciences, 2007.

[BZTB21] Monimoy Bujarbaruah, Xiaojing Zhang, Marko Tanaskovic, and Francesco Borrelli. Adaptive mpc under time varying uncertainty: Robust and stochastic, 2021.

[CBO+23] Z. Chen, T. Baček, D. Oetomo, Y. Tan, and D. Kulić. Inverse optimal control for dynamic systems with inequality constraints. *IFAC-PapersOnLine*, 56(2):10601–10607, 2023. 22nd IFAC World Congress.

[COMB19] Richard Cheng, Gabor Orosz, Richard M. Murray, and Joel W. Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks, 2019.

[Die11] Moritz Diehl. Numerical optimal control, 07 2011.

[EVT17] Peter Englert, Ngo Anh Vien, and Marc Toussaint. Inverse kkt: Learning cost functions of manipulation tasks from demonstrations. *The International Journal of Robotics Research*, 36(13-14):1474–1488, 2017.

[FLA16] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization, 2016.

[GHdRD24] Sven Gronauer, Tom Haider, Felippe Schmoeller da Roza, and Klaus Diepold. Reinforcement learning with ensemble model predictive safety certification, 2024.

[HMMA+20] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart Russell, and Anca Dragan. Inverse reward design, 2020.

[HVP+17] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Gabriel Dulac-Arnold, Ian Osband, John Agapiou, Joel Z. Leibo, and Audrunas Gruslys. Deep q-learning from demonstrations, 2017.

[HWMZ20] Lukas Hewing, Kim P. Wabersich, Marcel Menner, and Melanie N. Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):269–296, 2020.

[JAB13] Miles Johnson, Navid Aghasadeghi, and Timothy Bretl. Inverse optimal control for deterministic continuous-time nonlinear systems. In *52nd IEEE Conference on Decision and Control*, pages 2906–2913, 2013.

[KKS+20] Johannes Köhler, Peter Kötting, Raffaele Soloperto, Frank Allgöwer, and Matthias A. Müller. A robust adaptive model predictive control framework for nonlinear uncertain systems. *International Journal of Robust and Nonlinear Control*, 31(18):8725–8749, August 2020.

[Lab24] CMU Robotic Exploration Lab. Optimal control 2024. `https://www.youtube.com/playlist?list=PLZnJoM76RM6Jv4f7E7RnzW4rijTUTPI4u`, jun. 2024.

[LCA19] Matthias Lorenzen, Mark Cannon, and Frank Allgöwer. Robust mpc with recursive model update. *Automatica*, 103:461–471, 2019.

[Lib11] Daniel Liberzon. *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press, 2011.

[LKWP24] Taeyoon Lee, Jaewoon Kwon, Patrick M. Wensing, and Frank C. Park. Robot model identification and learning: A modern perspective. *Annual Review of Control, Robotics, and Autonomous Systems*, 7(Volume 7, 2024):311–334, 2024.

[LMK+17] Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A. Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. AI safety gridworlds. *CoRR*, abs/1711.09883, 2017.

[Mas18] Sofya Maslovskaya. *Inverse Optimal Control : theoretical study*. Theses, Université Paris Saclay (COmUE), October 2018.

[ME23] Robert D. McAllister and Peyman Mohajerin Esfahani. Distributionally robust model predictive control: Closed-loop guarantees and scalable algorithms, 2023.

[MTL10] Katja Mombaur, Anh Truong, and Jean-Paul Laumond. From human to humanoid locomotion-an inverse optimal control approach. *Auton. Robots*, 28:369–383, 04 2010.

[MZ18] Marcel Menner and Melanie N. Zeilinger. Convex formulations and algebraic solutions for linear quadratic inverse optimal control problems. In *2018 European Control Conference (ECC)*, pages 2107–2112, 2018.

[OST21] Matthew Osborne, Hyo-Sang Shin, and Antonios Tsourdos. A review of safe online learning for nonlinear control systems. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 794–803, 2021.

[Ray23] Partha Pratim Ray. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, 3:121–154, 2023.

[RCR+19] Ahmed Ramadan, Jongeun Choi, Clark J. Radcliffe, John M. Popovich, and N. Peter Reeves. Inferring control intent during seated balance using inverse model predictive control. *IEEE Robotics and Automation Letters*, 4(2):224–230, 2019.

[RMS+22] Rudolf Reiter, Florian Messerer, Markus Schratter, Daniel Watzenig, and Moritz Diehl. An inverse optimal control approach for trajectory prediction of autonomous race cars. In *2022 European Control Conference (ECC)*, volume abs 2011 8152, page 146–153. IEEE, July 2022.

[RPCB20] Harish Ravichandar, Athanasios S. Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(Volume 3, 2020):297–330, 2020.

[WZ19] Kim P. Wabersich and Melanie N. Zeilinger. Linear model predictive safety certification for learning-based control, 2019.

[WZ21] Kim P. Wabersich and Melanie N. Zeilinger. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems, 2021.

[Zua] Enrique Zuazua. Control and machine learning. `https://sinews.siam.org/Details-Page/control-and-machine-learning`. Accessed: 2024-05-22.