Safe Learning-Enabled Decision-Making for Autonomous Navigation in Unknown Environments

Somil Bansal
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

I. Motivation

The era of autonomous Unmanned Aerial Vehicles (UAVs) will transform our lives in more ways than one– UAVs will deliver our Amazon packages, inspect our bridges, search for and rescue distressed swimmers, and the list goes on and on. Tasks like these will require UAVs to navigate in *a priori* unknown environments, where decision making processes can be very challenging. Moreover, with UAVs already flying nearby commercial aircrafts and humans, it is also important to perform a diligent safety analysis before deploying them in the real-world.

Safe autonomous navigation is a challenging pursuit for real-world autonomous systems due to unavoidable uncertainties that exist in the real-world. First, there is uncertainty around how the system itself might evolve over time. Generally, a model of the system is used for control and safety analysis; however, unavoidable model-mismatches make it harder to ensure that decisions made using the model behave as expected on the system. Second, real-world systems often operate in *a priori* unknown environments. For example, a UAV might not know where obstacles are beforehand or how other agents move in the environment. The existence of these uncertainties make both the decision making as well as the safety analysis of these decisions significantly challenging.

Therefore, a challenging research question is how can we (a) design efficient decision-making schemes for UAVs operating in real-world environments, and (b) analyze the safety of the resulting, potentially learning-enabled, decisions despite the unknown environment and model-mismatches?

II. Technical Approach and Contributions

In our work, we use machine learning to account for model-mismatches by learning unmodeled or hard-to-model dynamics directly based on the data collected on the system. We also design learning-based perception-action loops that can make efficient decisions in unknown, real-world environments by leveraging the robot's prior experience in similar environments, but at the same time, allow for tractable safety analyses. We are partnering with Boeing to implement these perception-action loops on a commercial aircraft for lane tracking on a runway for which this safety analysis is particularly crucial.

We start with a brief overview of our work on using learning to account for model-mismatches, and *safe* navigation in *a priori* unknown environments with learning in the loop. We then present an in-depth look into our work on designing perception-action loops for navigation using learning.

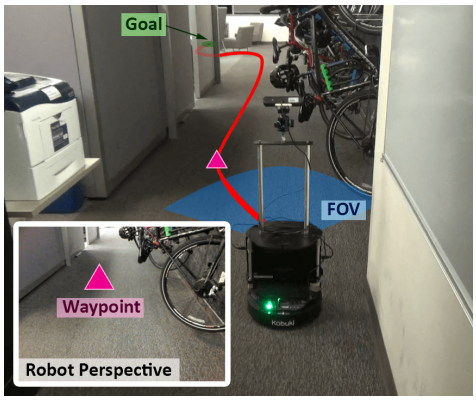

Fig. 1: We consider the problem of navigation from an initial state to a goal state in an *a priori* unknown environment. Our approach consists of a learning-based perception module and a dynamics model-based planning module. The perception module predicts a waypoint based on the current first-person RGB image observation. This waypoint is used by the model-based planning module to design a controller that smoothly regulates the system to this waypoint. This process is repeated for the next image until the robot reaches the goal. Here we demonstrate our approach on a TurtleBot navigating through a cluttered indoor environment.

- ***Using learning to capture unmodeled dynamics or model mismatch:*** the real-world UAVs undergo advanced aerodynamics effects that can affect their flight, such as blade flapping and ground effect [1]. These effects, however, are hard to model and hence difficult to take into account while designing a controller. We use a learning-based approach to model these effects directly using the data from a real system. In particular, we learn a Neural Network (NN)-based dynamics model to capture the unmodeled dynamics for a Crazyflie [2]. This dynamics model is used to track aggressive trajectories that require simultaneous rotational and translational motions. The inclusion of the learned dynamics model in the control loop significantly reduces the tracking error (Fig. 2).
- ***Using learning for closed-loop system identification:*** when the goal is to design an efficient controller to minimize a *given* cost function, a closed-loop identification might lead to a more robust control performance on the actual system compared to an open-loop identification [3, 4].

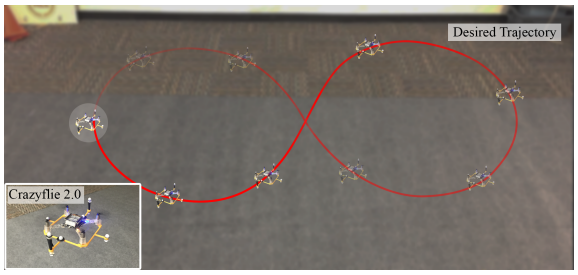

Fig. 2: We use model-based learning to model complex aerodynamic effects acting on a Crazyflie using open-loop and closed-loop identification techniques. The resultant model significantly improves the tracking performance in both cases.

We propose aDOBO, a Bayesian optimization-based active learning framework for optimizing the dynamics model for a given cost function, directly based on the observed cost values on the actual system [5]. Hence, unlike traditional system identification approaches, aDOBO does not necessarily correspond to finding the most accurate dynamics model, but rather the model yielding the best controller performance when provided to the optimal control method used. We also demonstrate how aDOBO can result in an improved control performance on an actual Crazyflie 2.0 compared to a state-of-the-art controller obtained using the 12D nonlinear dynamics model of the quadrotor.

Both the open-loop and closed-loop identification methods above focus on learning a good dynamics model, assuming that the system is operating in a known environment. We now present an approach to account for the unknown environment using learning in the perception-action loop.

- ***Using learning for decision making in unknown environments:*** other than unmodeled dynamics, one of the biggest challenges for real-world autonomous systems is that they operate in unknown environments, where they must leverage on-board perception to make decisions. To overcome this challenge, we propose a novel framework to couple model-based control with learning-based perception [6]. Our framework uses learning to tackle *unknown* environments and leverages optimal control using a *known* system dynamics to produce smooth locomotion (see Figure 1).

  More specifically, the learning-based perception module uses a Convolutional Neural Network (CNN) to produce a series of *waypoints* based on a monocular RGB observation that guide the robot to the goal via a collision-free path. These waypoints are used by a model-based planner to generate a smooth and dynamically feasible trajectory that is executed on the physical system using feedback control. Our experiments in simulated real-world cluttered environments and on an actual ground vehicle (TurtleBot) demonstrate that the proposed approach can reach goal locations more reliably and efficiently in novel environments as compared to a purely end-to-end learning-based alternative. Our approach is successfully able to exhibit goal-driven behavior without relying on explicit 3D maps of the environment, works well with low frame rates, and generalizes well from simulation to the real world.

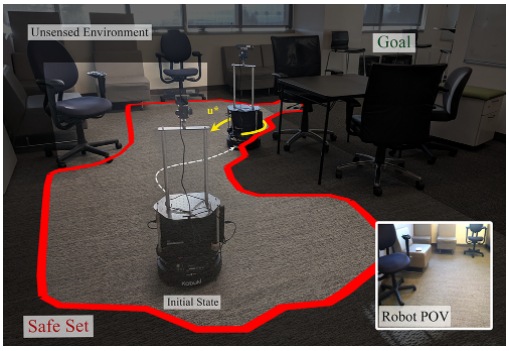

Fig. 3: We now consider the problem of *safe* navigation from an initial state to a goal state in an *a priori* unknown environment in Figure 1. Our approach treats the unsensed environment as an obstacle, and uses a HJ reachability-based framework to compute a safe controller for the vehicle, which is updated in real-time as the vehicle explores the environment. We show an application of our approach on a TurtleBot using the vision-based planner in Figure 1. When the robot is at risk of colliding, the safe controller (u^*) keeps the system safe.

- ***Ensuring safety in unknown environments:*** incorporating learning in the perception-action loop allows us to leverage the robot's prior experience to make efficient decisions in novel similar environments. However, these actions can still lead to unsafe behaviors when the training data priors do not hold in the novel environment. Therefore, it is important to perform a safety analysis of this perception-action loop.

  We recently proposed a Hamilton-Jacobi (HJ) reachability [7]-based safety framework for ensuring safe navigation for autonomous vehicles operating in *a priori* unknown static environments under the assumption that the sensors work perfectly within their ranges (see Figure 3). In particular, we treat the unknown environment at any given time as an obstacle and use HJ reachability to compute the *backward reachable set (BRS)*, i.e. the set of states from which the vehicle can enter the unknown and potentially unsafe part of the environment, despite the best control effort. The complement of the BRS therefore represents the safe set for the vehicle. With this computation, we also obtain the corresponding least restrictive safety controller, which does not interfere with the planner unless the safety of the vehicle is at risk. Because of the planner and sensor agnostic nature of our framework, we use it with the learning-based planner discussed above to ensure the safe navigation of a TurtleBot in an unknown cluttered indoor environment.

Due to the space constraints, in this document we will primarily focus on our work on designing perception-action loops for decision making in unknown environments. We share some initial results, how our model-based approach compares with an end-to-end learning approach, and discuss the key challenges we experienced while integrating perception and learning in a control loop.

III. Decision Making In Unknown Environments

There are several types of uncertainties present in unknown environments that must be taken into account for efficient nav-

igation, such as obstacles and other moving agents. As a first step towards this problem, we focus on single-agent, static, but *a priori* unknown environments. Classical robotics factorizes this problem into sub-problems of mapping and localization [8–10], path planning [11–13] and trajectory tracking [14–16]. Typically, the intermediate representations between these modules are purely geometric that either have a limited notion of navigational affordances (e.g., to go out of a room, I should look for a door, etc.), or use hand tuned heuristics that incorporate semantics in planning [17–21]. This makes it challenging to perform a guided and efficient exploration in novel environments.

In contrast, end-to-end learning approaches have also been used for autonomous navigation, where on-board sensor readings are directly mapped to motor torques [22–26]. Such approaches can incorporate semantics and common-sense reasoning for navigation, which allow for a guided decision making in unseen environments. However, they often require millions of samples to learn desired behavior [27] and are extremely specialized to the system they were trained on.

A. A factorized approach to the perception-action loop

We take a factorized approach to robot navigation that uses learning to tackle *unknown* environments and leverages optimal control using *known* system dynamics to produce smooth locomotion. More specifically, we train a CNN-based high-level policies, that incrementally use the RGB image observations to produce a sequence of intermediate *waypoints*. These waypoints are used as targets for a model-based optimal controller to generate smooth, dynamically feasible control sequences to be executed on the robot.

This approach benefits from the advantages of classical control and learning-based approaches in a way that addresses their individual limitations. Learning allows for generalization to unknown environments by leveraging statistical regularities to make predictions about the environment given only partial observations of the environment. Knowledge of the underlying dynamics for control allows for generalization to systems with different physical properties. At the same time, use of feedback-based control leads to smooth, continuous, and efficient trajectories that are naturally robust to noise in actuation. Furthermore, this factorization also takes the burden off the learning system. Learning now does not need to spend interaction samples to learn about the dynamics of the underlying system, and can exclusively focus on dealing with generalization to unknown environments.

Our approach, which we refer to as *WayPtNav* (WayPoint-based Navigation) here on, is summarized in Fig. 4. The CNN takes as input a 224×224 pixel RGB image, I_t, captured from the onboard camera, the target position, p_t^*, specified in the vehicle's current coordinate frame, and vehicle's current linear and angular speed, u_t, and outputs a waypoint $\hat{w}_t := (\hat{x}_t, \hat{y}_t, \hat{\theta}_t)$. Here $\hat{x}_t, \hat{y}_t$ is the desired next position, $\hat{\theta}_t$ is the desired heading, and $\hat{w}_t$ is the desired next state. Given $\hat{w}_t$ and the current linear and angular speed u_t, the planning module uses the system dynamics model to design a smooth

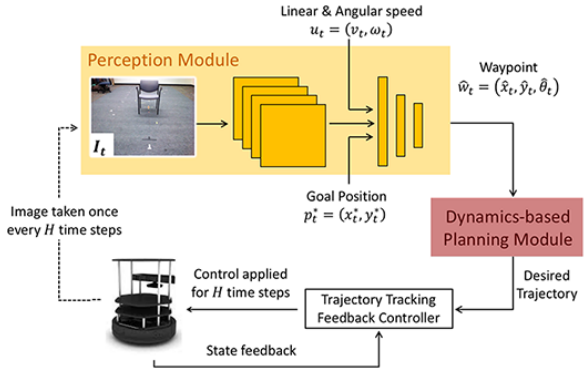

Fig. 4: **Proposed Framework**: Our approach to navigation consists of a learning-based perception module and a dynamics model-based planning module. The perception module consists of a CNN that outputs a desired next state or waypoint. The model-based planning module uses this waypoint to design a controller to smoothly regulate the system to the waypoint.

trajectory from the current vehicle state to the waypoint, $\{z^*, u^*\}_{t:t+H}$, where H is the planning horizon. To track the generated trajectory $\{z^*, u^*\}$, we design a LQR [28] based linear feedback controller. The control commands generated by the controller are executed on the system over a time horizon of H seconds, and a new image is obtained. This image is used by the perception module to predict a new waypoint, that is subsequently used by the planner to generate a new plan. This process is repeated until the robot reaches the goal position.

B. Training details

We train the perception module with supervised learning, using automatically generated globally optimal trajectories as a source of supervision. To generate these trajectories, we assume that the location of all obstacles is known during training time. However, during test time, no such assumption is made and the robot relies only on the image data. We use the Stanford large-scale 3D Indoor Spaces dataset to generate our training and test data [29], which consists of 3D scans (in the form of textured meshes) collected in 6 large-scale indoor areas. Scans from 3 areas were used for training and the robot was tested on the remaining areas. For more details on the training procedure, we refer the interested readers to [6].

C. Comparison with End-to-End (E2E) learning

We compare the proposed approach (WayPtNav) with the E2E learning approach. For E2E learning, we train the CNN in Fig. 4 to directly predict the control commands over the time horizon H. The results across 100 randomly selected, previously unseen navigational tasks are shown in Fig. 5. ***Compared to an end-to-end learning approach, the proposed approach is better (26% more successful at reaching the goals), more efficient (takes 35% less time in reaching the goal), and results in smoother trajectories (56% less jerk).***

We looked further into the failure cases of the E2E learning approach, and noticed that it particularly struggles when the robot needs to make aggressive maneuvers such as making

Fig. 5: We compare the proposed model-based planning approach (WayPtNav) to the end-to-end (E2E) learning approach across various metrics for the test navigation tasks. (a) Success rate of each approach in reaching the target (higher is better). (b) Average time and standard deviation to reach the target (lower is better). WayPtNav is 26% more successful and takes 2/3rd as much time as the E2E learning approach to reach the target. (c, d) Average acceleration and jerk along the robot trajectory (lower is better). Trajectories produced by the WayPtNav are considerably less jerky than ones produced by E2E learning approach.

tight turns around obstacles or going through narrow hallways. In addition, WayPtNav also guides the robot to the target more quickly than the E2E learning approach. Also, even though smooth control profiles generated by the expert policy were used as supervision for E2E learning, the learned control profile is discontinuous and jerky. On the other hand, since the CNN only acts as a high-level planner in WayPtNav, the control profile is still generated by a model-based planner and is smooth. This is also evident from the significantly lower average acceleration and jerk for WayPtNav compared to E2E learning. This has a significant implication for actual robots since the consumed power is directly proportional to the average acceleration. To facilitate the smoothness of the control profile, we also try adding a smoothing penalty in the loss function for E2E learning; however, we see a significant decline in the success rate as a result.

D. Hardware experiments

We next test WayPtNav on a TurtleBot 2 hardware testbed[1] (Fig. 1). We use the network trained in simulation and deploy it directly on the TurtleBot without any finetuning or additional training. The experiments demonstrate that WayPtNav can handle dynamics mismatches between a real robot and the simplified model used in simulation, and generalizes well from simulation to real world, partially due to the presence of the model-based feedback controller in the perception-action loop. We also compare WayPtNav with E2E learning across different metrics on hardware; the results are consistent with our simulation-based results in Fig. 5 (see [6]).

[1]Experiment videos can be found at https://vtolani95.github.io/WayPtNav/

E. Key learnings and challenges

We now share some key practical insights that were instrumental in implementing the proposed perception-action loop.

- *Data representation is important*: We notice that training on egocentric images and goal positions leads to a much better performance compared to the ones in global frame.
- *Optimal control can be too optimal*: We notice that using the waypoints (or control) that result in the system trajectory being too close to an obstacle as supervision for training the CNN leads to a worse performance. Such trajectories are not uncommon if optimal control schemes are used to generate supervision, but such a behavior is hard for the CNN to replicate. Thus, we pad our obstacles by a small margin while generating supervision.
- *Image distortion is instrumental to generalization*: We notice that applying random image distortions such as adding blur, changing brightness, during the training significantly improves the generalization. Furthermore, applying random perspective distortions during training significantly improves the performance on the real robot, which might have different camera parameters than what the network is trained on.
- *Finetuning is better than training from scratch*: We notice that fine tuning a pre-trained ResNet is much better than training the CNN from scratch. This is also in line with the observations made by the vision community [30].

IV. Conclusion and Next Steps

We propose a factorized model-based approach to robot navigation in *a priori* unknown environments that uses learning in the perception-action loop. We compare the proposed approach with an end-to-end learning approach across several metrics, and the quality of control obtained by the two approaches. The proposed approach is better and more efficient at reaching goals, and results in smoother trajectories. Use of a model-based feedback controller, allows the proposed approach to successfully generalize from simulation to physical robots.

Even though we compare the proposed model-based and the E2E learning approach within the context of a ground vehicle, we expect the model-based approaches to be only more advantageous for UAVs given their significantly complex dynamics compared to ground vehicles, which demand even more sophisticated control strategies. In such cases, additional burden of learning these control strategies on the learning system can significantly degrade its performance. Regardless, we will next like to formally compare the two approaches on the Crazyflie testbed. We are also excited to deploy the proposed approach on a Boeing commercial aircraft. Exploring the role of factorized decision making approaches for navigation in dynamic, multi-agent environments is another very interesting research direction.

References

[1] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007, p. 6461.

[2] S. Bansal, A. K. Akametalu, F. J. Jiang, F. Laine, and C. J. Tomlin, "Learning quadrotor dynamics using neural network for flight control," in *CDC*, 2016.

[3] M. Gevers, "Identification for control: From the early achievements to the revival of experiment design," *European journal of control*, vol. 11, no. 4-5, 2005.

[4] H. Hjalmarsson, M. Gevers, and F. De Bruyne, "For model-based control design, closed-loop identification gives better performance," *Automatica*, vol. 32, no. 12, pp. 1659–1673, 1996.

[5] S. Bansal, R. Calandra, T. Xiao, S. Levine, and C. J. Tomlin, "Goal-driven dynamics learning via Bayesian optimization," *CDC*, 2017.

[6] S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin, "Combining optimal control and learning for visual navigation in novel environments," *arXiv preprint*, 2019.

[7] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, vol. 50, no. 7, pp. 947–957, 2005.

[8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[9] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[10] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial Intelligence Review*, 2015.

[11] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[12] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *RA*, 1996.

[13] J. Canny, *The complexity of robot motion planning*. MIT press, 1988.

[14] T. J. Koo and S. Sastry, "Differential flatness based full authority helicopter control design," in *CDC*, vol. 2. IEEE, 1999, pp. 1982–1987.

[15] R. Walambe, N. Agarwal, S. Kale, and V. Joshi, "Optimal trajectory generation for car-type mobile robot using spline interpolation," *IFAC*, vol. 49, no. 1, pp. 601 – 606, 2016.

[16] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *ICRA*. IEEE, 2011, pp. 2520–2525.

[17] C. Galindo, J.-A. Fernández-Madrigal, J. González, and A. Saffiotti, "Robot task planning using semantic maps," *Robotics and autonomous systems*, vol. 56, no. 11, pp. 955–966, 2008.

[18] C. Becker, J. Salas, K. Tokusei, and J.-C. Latombe, "Reliable navigation using landmarks," in *ICRA*, vol. 1. IEEE, 1995, pp. 401–406.

[19] C. Nieto-Granda, J. G. Rogers, A. J. Trevor, and H. I. Christensen, "Semantic map partitioning in indoor environments using regional analysis," in *IROS*. IEEE, 2010, pp. 1451–1456.

[20] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *ICRA*. IEEE, 2017.

[21] J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardós, and J. Montiel, "Towards semantic slam using a monocular camera," in *ICRA*. IEEE, 2011, pp. 1277–1284.

[22] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *ICRA*, 2017.

[23] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *CVPR*, 2017.

[24] A. Khan, C. Zhang, N. Atanasov, K. Karydis, V. Kumar, and D. D. Lee, "Memory augmented control networks," in *ICLR*, 2018.

[25] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *JMLR*, 2016.

[26] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. Theodorou, and B. Boots, "Agile off-road autonomous driving using end-to-end deep imitation learning," *arXiv preprint arXiv:1709.07174*, 2017.

[27] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annual Review of Control, Robotics, and Autonomous Systems*, 2018.

[28] D. J. Bender and A. J. Laub, "The linear-quadratic optimal regulator for descriptor systems: discrete-time case," *Automatica*, vol. 23, no. 1, pp. 71–85, 1987.

[29] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese, "Joint 2D-3D-Semantic Data for Indoor Scene Understanding," *ArXiv e-prints*, Feb. 2017.

[30] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3712–3722.