Interpretable, Data-Efficient and Verifiable Autonomy with High-Level Knowledge

Zhe Xu [1]

[1]The University of Texas at Austin

October 30, 2023

Abstract

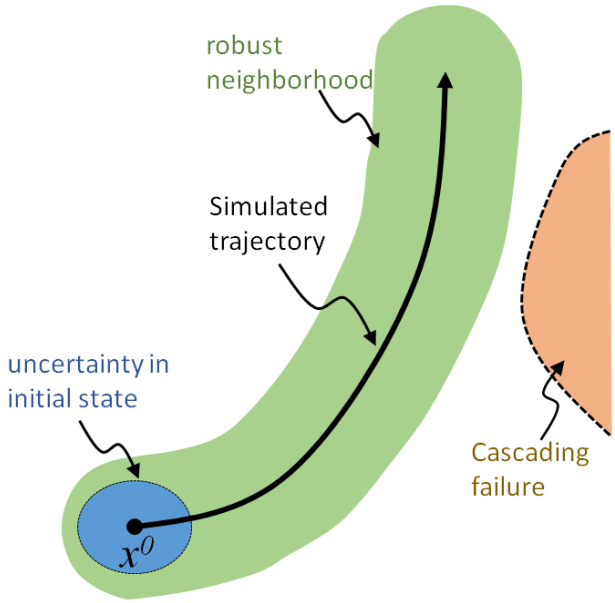
Despite the fact that artificial intelligence boosted with data-driven methods (e.g., deep neural networks) has surpassed human-level performance in various tasks, its application to autonomous

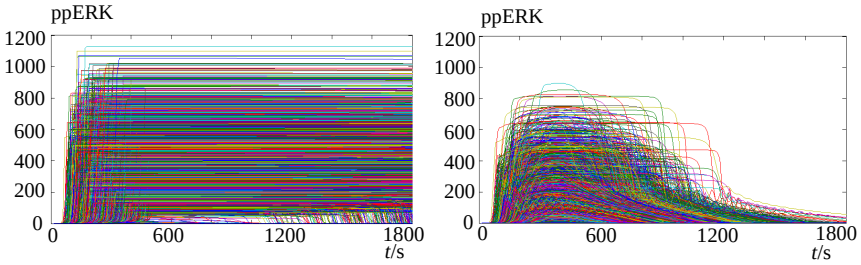
systems still faces fundamental challenges such as lack of interpretability, intensive need for data and lack of verifiability. In this overview paper, I overview some attempts to address these fundamental challenges by explaining, guiding and verifying autonomous systems, taking into account limited availability of simulated and real data, the expressivity of high-level

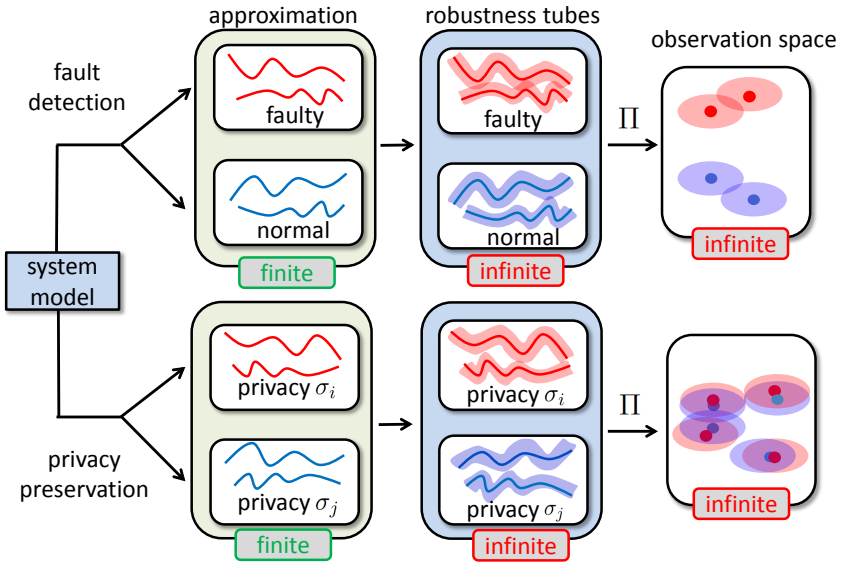
knowledge representations and the uncertainties of the underlying model. Specifically, this paper covers learning high-level knowledge from data for interpretable autonomous systems,

guiding autonomous systems with high-level knowledge, and

verifying and controlling autonomous systems against high-level specifications.







Hosted file

`Autonomy_arXiv.tex` available at https://authorea.com/users/662892/articles/675885-interpretable-data-efficient-and-verifiable-autonomy-with-high-level-knowledge

Hosted file

`Autonomy_arXiv.aux` available at https://authorea.com/users/662892/articles/675885-interpretable-data-efficient-and-verifiable-autonomy-with-high-level-knowledge

Hosted file

`ref_RS.bib` available at https://authorea.com/users/662892/articles/675885-interpretable-data-efficient-and-verifiable-autonomy-with-high-level-knowledge

Hosted file

`Autonomy_arXiv.bbl` available at https://authorea.com/users/662892/articles/675885-interpretable-data-efficient-and-verifiable-autonomy-with-high-level-knowledge

Hosted file

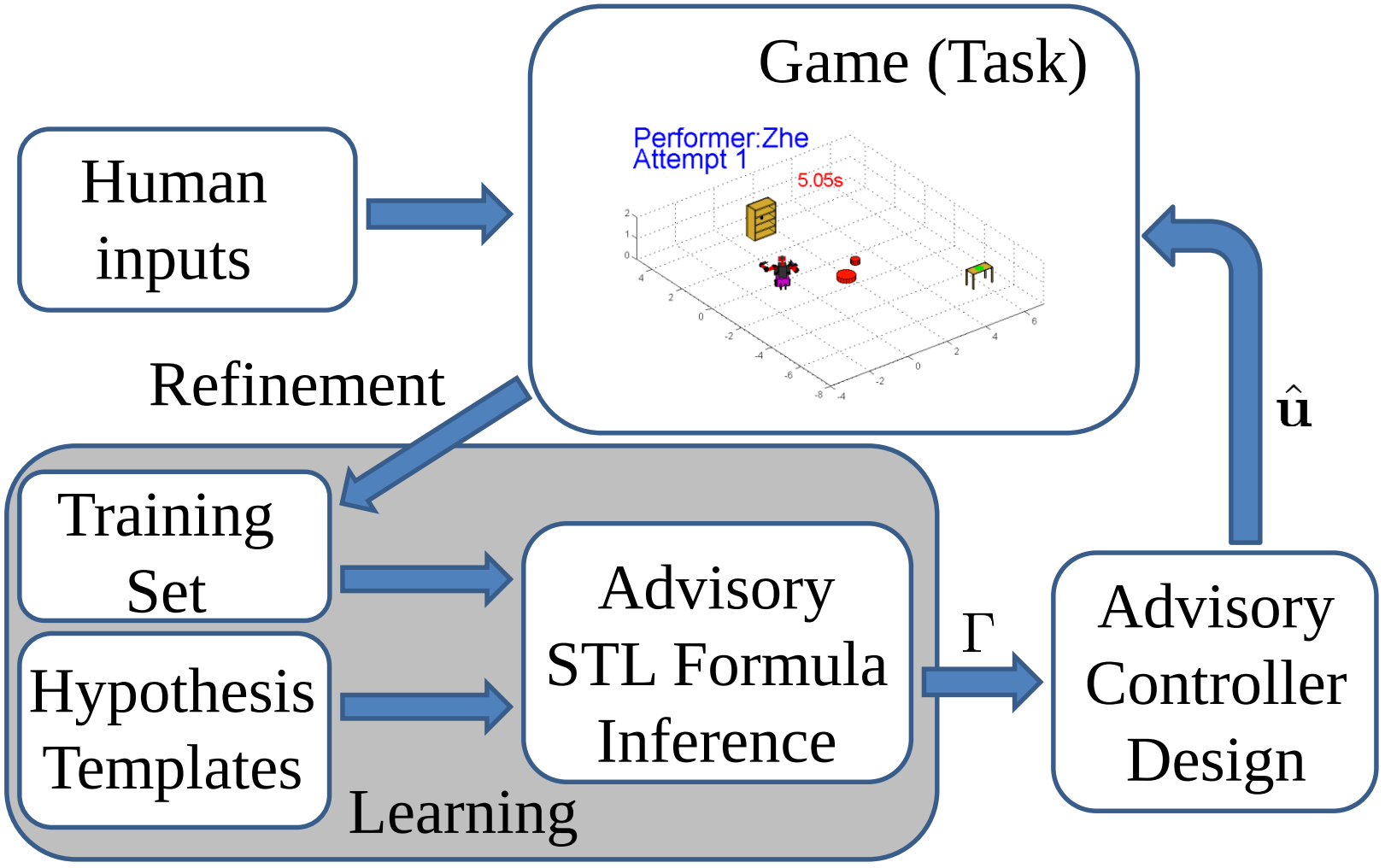
`Autonomy_arXiv.blg` available at https://authorea.com/users/662892/articles/675885-interpretable-data-efficient-and-verifiable-autonomy-with-high-level-knowledge

Hosted file

Autonomy_arXiv.log available at https://authorea.com/users/662892/articles/675885-interpretable-data-efficient-and-verifiable-autonomy-with-high-level-knowledge

Hosted file

Autonomy_arXiv.synctex.gz available at https://authorea.com/users/662892/articles/675885-interpretable-data-efficient-and-verifiable-autonomy-with-high-level-knowledge


Game (Task)
Performer:Zhe
Attempt 1
5.05s
Human inputs
Refinement
Training Set
Hypothesis Templates
Advisory STL Formula Inference
Learning
Γ
Advisory Controller Design
û

Interpretable, Data-Efficient and Verifiable Autonomy with High-Level Knowledge

Zhe Xu

1 introduction

Despite the fact that artificial intelligence boosted with data-driven methods (e.g., deep neural networks) has surpassed human-level performance in various tasks [1], its application to autonomous systems still faces fundamental challenges. Several recent high-profile traffic incidents involving autonomous vehicles have revealed the serious consequences of current applications and the imperative need to address these challenges [2].

- **Lack of Interpretability**: Few of the autonomous systems using data-driven methods can explain their behaviors and reason over the decision-making process in a way that humans can understand. In order to work seamlessly with humans, these systems need to communicate and explain their motivations, strategies and competence in performing various tasks to humans. Interpretability is especially needed for safety-critical tasks such as autonomous driving.
- **Intensive Need for Data**: Most data-driven methods in autonomous systems are data-intensive and lack *commonsense knowledge and reasoning* that are natural to humans. For example, *reinforcement learning* tasks often require extensive exploration of the environment to achieve satisfactory performance. On the other hand, the data available for performing such tasks are often limited or costly to obtain.
- **Lack of Verifiability**: The autonomous systems using data-driven methods tend to lack the deterministic features of traditional software, making the application of standard verification approaches substantially less effective. The lack of verifiability causes safety and security concerns, hence it is imperative to build cost-effective tools to verify such systems.

I address these fundamental challenges by explaining, guiding and verifying autonomous systems, taking into account limited availability of simulated and real data, the expressivity of high-level knowledge representations and the uncertainties of the underlying model. To that end, my approaches weave together the theories and techniques in *machine learning*, *formal methods* and *control theory*.

- **Learning High-Level Knowledge from Data for Interpretable Autonomous Systems**: I develop computationally efficient methods to learn high-level knowledge representations from data generated from autonomous systems with embedded data-driven modules. Such high-level knowledge representations need to be both understandable and informative to

humans, and amenable to automated reasoning. For example, I have used various variants of *temporal logics* to represent such high-level knowledge. In comparison with precise system identification and coarse sub-goal identification, inference of temporal logic formulas offers a balance between expressivity and human understanding in characterizing the task specifications. Traditional algorithms for inferring temporal logic representations do not scale to complex concepts, and the informativeness of the inferred formulas over prior knowledge is rarely considered. I develop an algorithm to learn informative temporal logic formulas with polynomial time complexity with respect to the size of the formula. I have also provided the first set of methods to learn temporal logic formulas to analyze multi-agent group behaviors, discover spatial-temporal patterns, and detect fault in a cyber-physical system in a privacy-preserving manner.

- **Guiding Autonomous Systems with High-Level Knowledge**: High-level knowledge can provide contextual information for guiding the autonomous systems towards better task performance. I conceptualize and develop a framework that enables a reinforcement learning agent to reason over its exploration process and distill high-level knowledge for effectively guiding its future explorations. Such knowledge can also be transferred from an original well-studied task to a new task, if these two tasks share some *logical similarities*. The resulting performance shows that the high-level knowledge can improve the sampling efficiency of the learning agent by up to two orders of magnitude.
- **Verifying and Controlling Autonomous Systems Against High-Level Specifications**: I provide safety and correctness guarantees for autonomous systems through *formal verification* and *provably correct synthesis*. While there has been research on verifying physical systems with neural networks in the decision and control loop, such verification has been limited to properties without a temporal evolution. I have developed verification and provably correct synthesis methods with high-level temporal logic specifications for non-linear hybrid systems, multi-agent systems with intermittent communication, etc. The developed methods have wide applications in robotic systems, power systems, smart buildings and biological systems.

2 Learning (inferring) temporal logic formulas from data

With the increasing development of artificial intelligence and machine learning, there has been a growing interest in learning (inferring) dense time temporal logic formulas from system trajectories. The process of extracting knowledge from data is crucial in reverse engineering. In comparison with precise system identification and coarse high-level specification inference, temporal logic inference may provide the right amount of precision in characterizing the system behaviors. Recently, there has been a growing interest in learning (inferring) dense-time temporal logic formulas from system trajectories [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14].

Given two sets of trajectories, I am interested in finding a temporal logic formula that can classify these two sets of trajectories. If only one set of trajectories are available, I am interested in identifying a temporal logic formula that best fits the trajectories with respect to certain fitness measures. Both the classification and identification of temporal logic belong to temporal logic

inference from data.

How to utilize a priori information in performing temporal logic inference?

I applied the temporal logic inference algorithm in the classification and identification of robot arm movements of Phantom Omni haptic devices [15]. The motivation for our work is to enable robots to learn from human demonstrations and generate temporal logic specifications from the demonstration data. I inferred STL formulas that can classify different robot arm movements, predict sequential robot arm movements and identify different goals and obstacles during different time intervals. For these purposes, it is natural to assume that **a priori information** about (some of) atomic predicates involved in the formulas is available. For example, we can naturally assign a set of the state-space to the predicate "*the arm is stretched upright*". By including such a prior information in the inference process, we can ensure that the STL formulas are composed of atomic predicates with a priori assigned meaning.

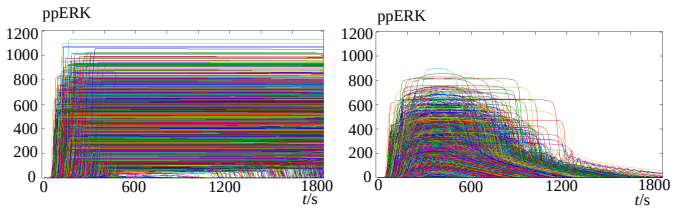


Figure 1: Trajectories generated from Model 1 (left) and Model 2 (right) with EGF dose of 0.1nM using 20000 trajectories generated from Model 1 and 20000 trajectories from Model 2.

How to perform temporal logic inference for model discrimination?

Mathematical models can be used to generate hypotheses that can guide experiments of biological systems. How to select better models and check which model is more representative of the real biological systems has always been a challenge. I designed a method for discriminating among competing models for biological systems by learning temporal logic formulas from data obtained by simulating the models. I applied the temporal logic inference method in discriminating between two competing mathematical models of *extracellular signal-regulated kinase* (ERK) responses to *epidermal growth factor* (EGF) stimulation [4] (see Fig. 1).

How to perform temporal logic inference for classification with spatial and temporal uncertainties?

Besides classifying finitely many trajectories in different sets, I can also classify infinitely many trajectories with initial state variations and disturbances, which accounts for the spatial uncertainties; I also consider the time variations when the switching events occur in a switched system, which accounts for the temporal uncertainties. I designed observation maps in the form of temporal logic formulas for **fault detection** and **privacy preservation** of **cyber-physical Systems** with temporal and spatial uncertainties in a provably correct fashion (see Fig. 2). I implemented the method on the simulation model of a smart building testbed for detecting the open window fault while preserving multiple privacy conditions of the room occupancy [16, 17]. The approach can also be applied to distinguish the infinitely many trajectories in different locations of a hybrid

system to improve the observability of the system [18].

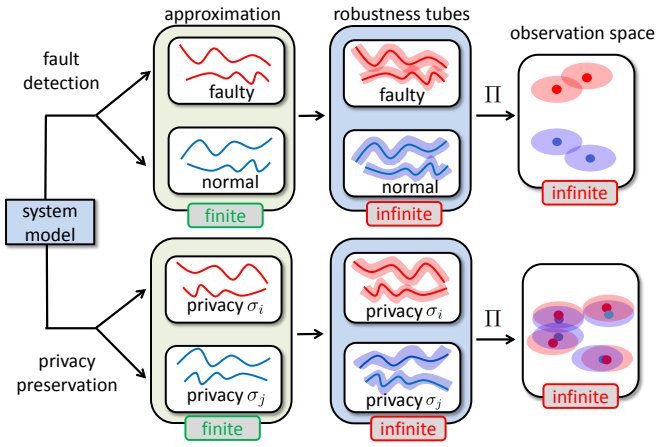


Figure 2: A compact set of infinitely many trajectories can be approximated with a finite set and the observation map Π can project the system behaviors into an observation space where the images of the normal behaviors and the faulty behaviors are separate while for privacy preservation the images of the behaviors with privacy conditions σ_i and σ_j have to be close.

How to extend the inference approach to multi-agent systems?

I defined a new type of signal temporal logic specifically for multi-agent systems: census signal temporal logic (CensusSTL) [19]. The CensusSTL consists of an "*inner logic*" formula that characterizes a consistent, frequent and specific task and an "*outer logic*" formula that characterizes the pattern of the number of agents in certain subgroups whose behaviors satisfy the "*inner logic*" formula. I proposed a new inference algorithm that can infer the CensusSTL formula directly from individual agent trajectories and applied the inference algorithm to analyzing the strategies of a soccer game with the data of body sensors equipped on each player.

3 Reinforcement learning and transfer learning utilizing high-level knowledge

The sampling efficiency and performance of reinforcement learning can be improved if some high-level knowledge (e.g., temporal logic formulas, finite-state machines) can be incorporated in the learning process [20, 21].

How to utilize high-level knowledge for improving reinforcement learning?

I conceptualized and developed a framework to achieve *joint inference of high-level knowledge and policies* for reinforcement learning [20]. In this framework, the high-level knowledge (e.g., temporal logic formulas, finite-state machines) is inferred simultaneously while the reinforcement learning proceeds, and the inferred high-level knowledge can effectively guide the future explorations of the learning agent. The experiments show that learning high-level knowledge can lead to fast convergence to optimal policies, while standard reinforcement learning methods fail to converge to optimal policies after a substantial number of training steps in many tasks.

How to utilize high-level knowledge for transfer learning?

The inferred temporal logic formulas can be also transferred from a *source task* to a *target task* if

these tasks are *logically similar*. While the transfer of logical knowledge *without a temporal context* has been studied in reinforcement learning, i.e., learning for sequential decision-making, transfer between tasks in *temporal context* has attracted significantly less attention. I concretized similarity between temporal tasks through a notion of *logical transferability*, and developed a transfer learning approach between different yet similar temporal tasks. If logical transferability is identified through this inference, I construct an automaton for each subformula of the inferred temporal logic formulas from both source and target tasks, and perform reinforcement learning on the extended state which includes the states of the automata representing the subformulas for the source task. I then establish mappings between the corresponding components of the automata from the two tasks, and transfer the extended reward and policy information based on these established mappings. Finally, I perform reinforcement learning on the extended state for the target task, starting with the transferred extended reward and policy information. The experimental results show, depending on how similar the source task and the target task are, that the sampling efficiency for the target task can be *improved by up to one order of magnitude* by performing reinforcement learning in the extended state space, and further *improved by up to another order of magnitude* using the transferred extended reward and policy information [8].

4 System verification with respect to temporal logic specifications

Given a temporal logic specification, I am interested in analyzing whether all the trajectories within certain robust neighbourhood around a nominal trajectory simulated by a system (e.g. hybrid system, non-linear system) can satisfy the given temporal logic specification within certain time horizon. This problem is referred to as robust testing or verification problem with temporal logic specifications [22].

How to perform robust testing of temporal logic specifications in nonlinear hybrid systems?

Complex systems such as power systems can be very vulnerable to disturbances or intended attacks. How can we make sure that these systems are stable or behave as they are designed to behave? If the desired behavior is specified in the form of temporal logic formulas such as "*line currents should never exceed the threshold value for more than 0.1 seconds*", then I can perform robust testing of temporal logic specifications for the nonlinear hybrid system model of power systems. I first proposed the algorithm of computing the *bounded disturbance local discrepancy function* for a general nonlinear system and then extended the method to hybrid systems [22]. I applied the robust testing algorithm in power systems cascading failures mitigations in two different scenarios (see Fig. 3): *robust testing of various generator mechanical power dispatch schedules; robust testing of post-fault remedial actions based on quick-start storage.* I performed robust testing on a three-machine power system model for the Italian blackout in 2003 and the IEEE 39-bus benchmark system.

5 Controller synthesis with respect to temporal logic specifications

With a temporal logic specification, I am also interested in designing a controller such that the trajectories of the controlled system satisfy the given temporal logic specification. This problem is referred to as the controller synthesis problem with temporal logic specifications [23, 24, 25, 26, 27, 28].

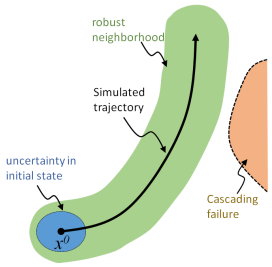


Figure 3: A simulated trajectory from initial state x^0 can be equipped with a robust neighborhood such that variations in the initial state or bounded disturbance will not deviate the trajectory beyond the robust neighborhood, thus safety is guaranteed (cascading failures are proven to be avoided).

How to synthesize controller with temporal logic specifications for nonlinear DAE (differential-algebraic equations) systems?

I proposed a controller synthesis method to regulate grid frequencies with energy storage systems, so that the system trajectories satisfy the MTL specifications about the grid frequency deviations and the wind turbine generator rotor speed deviations [29]. I formulated the metric temporal logic (MTL) specification as a constraint and applied the functional gradient descent method to both satisfy the MTL constraint and minimize an objective function as a performance metric of the controller. The gradients of both the objective and the constraint functions are calculated specifically for DAE systems.

I simulated finitely many post-fault trajectories (after the fault is cleared) with different fault clearing time such that the initial robust neighborhoods of these simulated trajectories can cover all the post-fault initial states (all the possible states when the fault is cleared) with given uncertainties in the fault clearing time. In this way, all the post-fault trajectories that start from the set of post-fault initial states (including the uncertainties in the fault clearing time) are guaranteed to stay in the robust neighborhoods around the nominal (simulated) trajectories and satisfy the MTL specifications.

Is it possible to learn a feedback control law from the state and input data of the feedforward controller?

I learned a piecewise linear control law from the data of the optimal input signals and the states of the simulated trajectories. I used robust linear programming to find the classification functions for the subclasses and construct piecewise linear classifiers in partitioning the state space so that the state space is totally covered. I have proven and tested with simulations that any trajectory starting from the initial set with the feedback controller are guaranteed to satisfy the metric temporal logic (MTL) specification [29]. Besides, simulations show that even when unexpected disturbances occur, trajectories generated with the feedback controller can still satisfy the MTL specification in certain cases and have better performance in comparison with the trajectories generated with the feedforward controller.

How to synthesize controller with temporal logic specifications in stochastic environment?

I proposed the *stochastic control bisimulation function*, which bounds the divergence of the trajectories of the stochastic control system and the diffusionless deterministic control system in a probabilistic fashion [30]. I designed a feedforward controller by solving an optimization problem for the nominal trajectory of the deterministic control system with robustness against initial state variations and stochastic uncertainties. Then I learned a feedback control law from the state and input data of the feedforward controller. As an implementation, I applied the proposed approach in controlling a wind farm and an energy storage system for frequency regulation with provable probabilistic safety guarantees in the stochastic environment of wind power generation.

Is it possible to combine the temporal logic inference and controller synthesis processes?

Few of the existing works utilize the inferred temporal logic formulas as features of the desired set for iteratively improving the performance of future generated trajectories. As an example, in uncertain or adversarial environment, robots with all the necessary sensors may still fail to complete tasks within specified time if wrong strategies are employed. To improve the performance of the robots in these situations, we can utilize the trajectories of the robots generated in successful or failed attempts to provide valuable information or knowledge for advising or guiding the future operations.

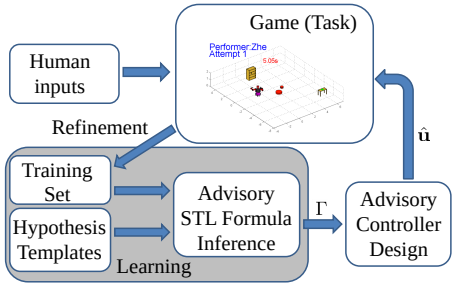


Figure 4: Block Diagram of the advisory STL subformula inference (learning), controller design and refinement process.

I proposed a method to iteratively learn (infer) and refine a set of advices from the trajectories generated in the successful and failed attempts in a task, with each advice in the form of advisory signal temporal logic (STL) formulas [25]. Each advice consists of an advisory motion STL formula that characterizes the spatial-temporal pattern of the motion as a feature of success and an advisory selection STL formula as a criterion for the environment to select the advice. The advisory controller can advise or guide the human operators or the robots for better performance with the shared autonomy between the human operator and the controller (see Fig. 4).

I implemented the approach in two case studies to test the effectiveness of the advisory controller, one with a Baxter-On-Wheels simulator using the keyboard control and the other with two quadrotors in an experimental testbed using the joystick control in iteratively improving the success rates of completing the tasks with the help of the designed advisory controller.

References

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[2] F. M. Favarò, N. Nader, S. O. Eurich, M. Tripp, and N. Varadaraju, "Examining accident reports involving autonomous vehicles in california," *PLOS ONE*, vol. 12, no. 9, pp. 1–20, 09 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0184952

[3] Z. Kong, A. Jones, and C. Belta, "Temporal logics for learning and detection of anomalous behavior," *IEEE Trans. Automatic Control*, vol. 62, no. 3, pp. 1210–1222, March 2017.

[4] Z. Xu, M. Birtwistle, C. Belta, and A. Julius, "A temporal logic inference approach for model discrimination," *IEEE Life Sciences Letters*, vol. 2, no. 3, pp. 19–22, Sept 2016.

[5] Z. Xu and A. A. Julius, "Census signal temporal logic inference for multiagent group behavior analysis," *IEEE Trans. Autom. Sci. and Eng.*, 2016, in press. [Online]. Available: http://ieeexplore.ieee.org/document/7587357/

[6] G. Bombara, C.-I. Vasile, F. Penedo, H. Yasuoka, and C. Belta, "A decision tree approach to data classification using signal temporal logic," in *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '16. New York, NY, USA: ACM, 2016, pp. 1–10. [Online]. Available: http://doi.acm.org/10.1145/2883817.2883843

[7] Z. Xu, C. Belta, and A. Julius, "Temporal logic inference with prior information: An application to robot arm movements," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 141 – 146, 2015.

[8] Z. Xu and U. Topcu, "Transfer of temporal logic formulas in reinforcement learning," in *Proc. IJCAI'2019*, 7 2019, pp. 4010–4018. [Online]. Available: https://doi.org/10.24963/ijcai.2019/557

[9] E. Asarin, A. Donzé, O. Maler, and D. Nickovic, "Parametric identification of temporal properties," in *Proc. Second Int. Conf. Runtime Verification*, Berlin, Heidelberg, 2012, pp. 147–160.

[10] R. Yan, Z. Xu, and A. Julius, "Swarm signal temporal logic inference for swarm behavior analysis," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 3021–3028, 2019.

[11] Z. Xu, M. Ornik, A. A. Julius, and U. Topcu, "Information-guided temporal logic inference with prior knowledge," in *2019 American Control Conference (ACC)*, July 2019, pp. 1891–1897.

[12] B. Hoxha, A. Dokhanchi, and G. Fainekos, "Mining parametric temporal logic properties in model-based design for cyber-physical systems," *International Journal on Software Tools for Technology Transfer*, Feb 2017. [Online]. Available: http://dx.doi.org/10.1007/s10009-017-0447-4

[13] Z. Xu, A. J. Nettekoven, A. Agung Julius, and U. Topcu, "Graph temporal logic inference for classification and identification," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, Dec 2019, pp. 4761–4768.

[14] X. Jin, A. Donze, J. V. Deshmukh, and S. A. Seshia, "Mining requirements from closed-loop control models," in *Proc. Int. Conf. Hybrid Systems: Computation and Control*, 2013, pp. 43–52.

[15] Z. Xu, C. Belta, and A. Julius, "Temporal logic inference with prior information: An application to robot arm movements," *IFAC Conference on Analysis and Design of Hybrid Systems (ADHS)*, pp. 141 – 146, 2015.

[16] Z. Xu and A. A. Julius, "Robust temporal logic inference for provably correct fault detection and privacy preservation of switched systems," *IEEE Systems Journal*, vol. 13, no. 3, pp. 3010–3021, 2019.

[17] Z. Xu, S. Saha, and A. Julius, "Provably correct design of observations for fault detection with privacy preservation," in *IEEE Conference on Decision and Control (CDC), Melbourne, Australia, 2017*.

[18] Z. Xu, Y. Deng, and A. Julius, "Robust temporal logic inference for hybrid system observation– an application on occupancy detection of smart buildings," in *2018 Annual American Control Conference (ACC)*, June 2018, pp. 610–615.

[19] Z. Xu and A. A. Julius, "Census signal temporal logic inference for multiagent group behavior analysis," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 264–277, Jan. 2018.

[20] Z. Xu, I. Gavran, Y. Ahmad, R. Majumdar, D. Neider, U. Topcu, and B. Wu, "Joint inference of reward machines and policies for reinforcement learning," in *Proc. International Conference on Automated Planning and Scheduling (ICAPS), Special Track on Planning and Learning*, 2020.

[21] F. Memarian, Z. Xu, B. Wu, M. Wen, and U. Topcu, "Active task-inference-guided deep inverse reinforcement learning," in *Submitted to IEEE Conference on Decision and Control (CDC)*, 2020.

[22] Z. Xu, A. A. Julius, and J. H. Chow, "Robust testing of cascading failure mitigations based on power dispatch and quick-start storage," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–12, 2017.

[23] Z. Xu, F. M. Zegers, B. Wu, W. Dixon, and U. Topcu, "Controller synthesis for multi-agent systems with intermittent communication. a metric temporal logic approach," in *Allerton'19*, pp. 1015–1022.

[24] Z. Xu, K. Yazdani, M. T. Hale, and U. Topcu, "Differentially private controller synthesis with metric temporal logic specifications," in *Proc. American Control Conference (ACC)*, 2020.

[25] Z. Xu, S. Saha, B. Hu, S. Mishra, and A. A. Julius, "Advisory temporal logic inference and controller design for semiautonomous robots," *IEEE Trans. Autom. Sci. Eng.*, pp. 1–19, 2018.

[26] M. Hibbard, Y. Savas, Z. Xu, A. A. Julius, and U. Topcu, "Minimizing the information leakage of high-level task specifications," in *21st IFAC World Congress*, 2020.

[27] M. Cubuktepe, Z. Xu, and U. Topcu, "Policy synthesis for factored mdps with graph temporal logic specifications," in *Proc. International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2020.

[28] F. Djeumou, Z. Xu, and U. Topcu, "Probabilistic swarm guidance with graph temporal logic specifications," in *Submitted to Robotics: Science and Systems (RSS)*, 2020.

[29] Z. Xu, A. Julius, and J. H. Chow, "Energy storage controller synthesis for power systems with temporal logic specifications," *IEEE Systems Journal*, Early access on IEEE Xplore.

[30] Z. Xu, A. A. Julius, and J. H. Chow, "Coordinated control of wind turbine generator and energy storage system for frequency regulation under temporal logic specifications," in *Proc. Amer. Control Conf.*, 2018, pp. 1580–1585.