Concurrent Policy Blending and System Identification for Generalized Assisitve Control

Luke Bhan[1], Marcos Quinones-Grueiro[1], and and Gautam Biswas[1]

***Abstract*— In this work, we defined a simoultaneous policy blending and system identification approach to create generalized policies that are robust to system parameter changes. To do this, we employ a blending policy whose state space relies soley on estimated parameters from any system indentification technique. This blending policy then only must learn how to utilize it's subpolicies to handle various parameter changes instead of learning a complex task for a generalized parameter set simoultaneously. We demonstrate our schemes ability on a collaborative robot and human itching task in which the human contains potential impairments. We then showcase our methods efficiency with a variety of system identification techniques resulting in consistent outperformance of standard domain randomization. The code is available at Luke Bhan's Github.**

I. INTRODUCTION

Over the last few years, there has been significant interest in developing models that are trained in simulation and then transferred to the real world [1] [2] [3]. However, despite the progress in learning techniques, simulated policies still suffer from long simulation times as they require large amounts of experience to handle the unknown environements present in the 'real world'. Additionally, these policies struggle to generalize to complex situations as they can become unpredictable when faced with new challenges. As such, researchers have approached this training these policies in two distinct ways. The first approach relies on system identification of parameters as a set of information that can inform the policy on how to respond in different environments [4]. However, these policies struggle to generalize and often require retuning for different situations [4]. The second approach involves randomizing a series of system parameters during training such that the policy learns to handle a wide variery of situations. However, this approach - domain randomization [5] - requires that the actual parameters be in the set that is randomized and as such, the robustness of the policy is directly correlated to the range of randomized parameters. For complex tasks with many parameters, creating a large range takes significant training time before a robust policy can be utilized [6].

However, both these approaches require a single policy that can 1) solve the task at hand for a single set of parameters and 2) can then generalize the solution to a wide set of parameters of which some may create very different challenges than others. As such these policies are in essence attempting to solve an overload of challenges and can fail for large randomization problems. Given this, we attempt to decouple the process of learning to generalize to system parameters and learning policies that can solve the task at hand efficiently by utilizing a blending technique. With this, we train single policies that are efficient for a certain set of distinct parameters and then utilize a blending policy to identify how best to weight these distinct policies based on the current parameters of the system. As such, we can then more efficiently train generalized policies comlpetely in simulation.

For this paper, our main contributions are:

- Formalizing the idea to decouple the process of learning a single task and generalizing to a large set of system parameters using a blending policy technique
- Designing an architecture that integrates a blending policy which accurately handles the generalization of its sub-models to system parameters
- Implementing our scheme in a collaborative human and robotic locomotion task to demonstrate its effectiveness across different system identification methods

II. RELATED WORK

There have been many approaches to learning policies based on estmiation of simulation parameters; however, to the author's knowledge, none have yet to combine system identification with a blending approach. For example, [7] demonstrates the use of simoulatenous learning where they explore a series of predictive error methods to minimzie the differences between the observed parameters and estimated paramters for model predictive control (MPC). Additionally, domain randomizaiton has been used for a robotic control similar to ours [8]; however this task does not consider a collaborative environment nor does it handle multiple faults introduced by the interacting agents. Furthermore, [9] solves a challenging Rubik's cube control task by automatic domain randomization that slowly increase the difficulty of the task, but can take siginficant time as it does not consider the integration of any real-world sampling. Lastly [10] considers an adapative domain randomization strategy where they attempt to identify domains that can create challenging enviornments for the policy. This approach is similar to ours, except that their approach is purely data driven based on the result of their policy whichrequires significant training due to potential sample inefficiency while we can utilize prior domain knowledge to identify environments that have a high potential of being challenging for the policy.

In addition to the large amount of research tackling policies designed for efficient sim2real transfer, there have been a series of recent work that demonstrate the effectiveness of policy blending. [11] demonstrate the use of policy blending

[1] All authors are currently members of Vanderbilt University, Nashville, TN. Correspondence to luke.bhan@vanderbilt.edu

for simple tasks such as opening a cap, flipping a breaker, and turning a dial. However, their policy learns directly from sensor measurements and does not consider impairments in the environment. Furthermore, [12] has shown a policy blending technique between a human and robot policy for robot-assisted control to accurately assist the human with various tasks such as fetching a water bottle. However, this work does not consider training models using modern DRL techniques. Given these approaches, it seems worthwhile to combine robust policy blending with modern system identification as a new approach to generalized assistive modelling.

III. Background

A. Formalizing RL Problems

Consider the problem defined by a Partially Observable Markov Decision Process (POMDP) $\mathscr{M} = (S, O, A, R, \gamma)$ where S is the state space, O is the agent's observation space, A is the agent's action space, $R : SxA \implies \mathbb{R}$ defines a reward function mapping an action in some state to a number in $\mathbb{R}$ and γ as a discount factor on the reward [13]. In practice, it is difficult to know exactly how the state space S behaves without consistent exploration and thus simulation is required given that real-world exploration is generally either expensive, inefficient, or physically challenging. To efficiently solve this problem, the goal is to create a policy π^* such that we maximize the expected reward function [14] $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$ where r_t is the immediate reward function at time t

B. Introduction to Domain Randomization

To effectively identify π^* in simulation, a set of parameters must be defined to create the environment. Domain randomization attempts to sample a set of some N parameters which we will denote ξ for which a reasonable range of potential values is constructed - usually from domain specific knowledge [2]. In this paper, we will consider domain randomization of the uniform type such that the parameters are uniformly sampled within a feasible range for that particular parameter. For example, the weaknes s of a certain human joint can be sampled uniformly between 0 and 1 where 0 invokes no mobility while 1 is a joint that is at full strength.

C. System Identification Via Parameter Estimation

System identification through parameter estimation is a well studied subject in which a estimator can consistently recieve samples from a real world environment and generalize these samples into an estimated true value. In this work, we utilize the Unscented Kalman Filter (UKF) for our estimator [15] and make the assumption that our real-world parameters can be measured with some confidence, but may be cluttered with noise.

D. Autotuned Search Parameter Model (SPM)

Given that our assumption that an environment's parameters can be measured does not always hold true, we utilize a new technique that can estimate the parameters by interacting with the environment as an agent over measuring them directly. Recently, Du Et. Al. have formulated a new approach to system identification where they define a data driven model that learns a map from $(o_{1:T}, a_{1:T}, \xi_{guess}) \implies (0,1)^N$ such that the current parameters ξ_{guess} are greater than, less than, or equal to the true parameters [16]. This binary classifier is than iteratively trained concurrently to the policy such that it slowly converges to the real world parameters by attempting to sample the real world parameters through its own policies interaction trajectories. By utilizing this iterative search, we can then perform a level of system identification that does not comlpetely rely on domain knowledge for our experiments.

IV. Approach

A. A Blending Model

To create a decoupled policy in which we can solve individual tasks while maintaining robustness to a variety of system parameters, we introudce a blending policy in which we consider soley the N system parameters ξ as its state space. This policy then only needs to output the weights $\mathbf{W}$ of its sub-policies at each time step to generate the action for the environment. The sub-policies of this model are trained on a single set of constant system parameters in which a unique environment is identified through previous domain knowledge. We then define the action value of this system as $a = \frac{1}{N}\sum_{i=0}^{N} w_i \pi_i(s_t)$ where $w_i \in \mathbb{R}$, N is the number of sub policies and $\pi_i(s_t)$ represents the action taken by the sub policy given a state at time t.

B. Concurrent System Identification

We then combine this blending model with a concurrent system identification scheme to train a generalized policy that is robust to different enviormental challenges. To do this, we let the state space of the blending policy consist of only the estimated parameters and as such must learn to associate certain parameters with its sub-policies. In practice, after a certain number of training steps defined by the researcher, we utilize our system identification method to reupdate the state space with a more accurate set of estimate system parameters and continue training. We emphasize that our approach is independent of the system identification method chosen and thus can tailored based on the available domain knowledge of the environment. This approach can be visualized in **??**.

V. Experiments

A. Training of Sub-Policies

For demonstrating our model, we attempt to solve a collaborative itching task using assistive gym [17] where a robot is assisting an impaired human in itching. We consider 3 impairments similar to [18] for the human:

(a) Involuntary Movement: The first impairment is involuntary movement which is handled by adding noise normally distributed to the joint actions of the human. For this policy, we sample the noise according to a normal distribution where each joint in the arm has a

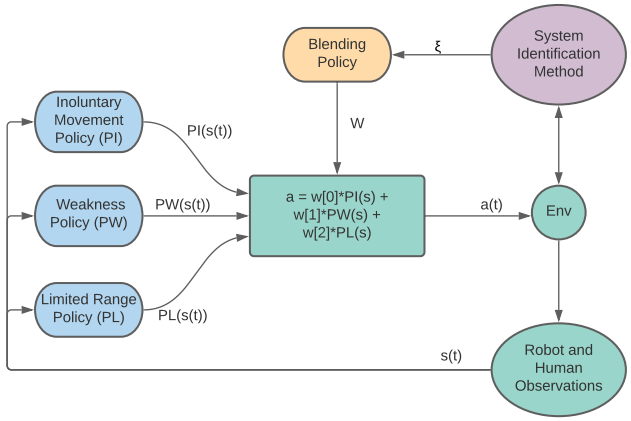

Fig. 1. Our Proposed model architecture

mean of 0 noise and a standard deviation of 5 degrees of noise.

(b) Weakened Strength: The second impairment involves weakness in the ability for the human to move thier arms which is introduced by lowering the strength factor in the pid controller of the joints. This value is also sampled normally with a mean of 0.66 and standard deviation of 0.2 with 1 representing full strenght and 0 representing immobility.

(c) Limited Range: Lastly, we consider a limitation in the range of movement for each joint in the arm of the human. Like above, full joint movement is represented by 1 and immobile joints are represented by 0. As such, we sample the limited movement from a normal distribution with mean 0.75 and standard deviation of 0.1.

Initially, we begin by training a single policy for each individual impairment on 2 million timesteps (5000 episodes) using PPO [19] where each network consists of 2 layers of 64 nodes. For all sinlge impairment policies, we define a state space of 64 joints between the robot and human along with an action space of 17 joint targets. All policies use the same reward function as defined in [18]. This reward function considers a weighted combination of the distance of the robot arm to the target itch position, the human's preferences, and the contact induced with the itch target.

B. Training of Blending Policy

Similar to the sub-policies, we consider the same reward and utilize a PPO model with 2 layers of 64 nodes each for training the blending policy. However, for the blending policy we train for 400k timesteps and the state space only consists of the system parameters. Unlike the subpolicies, our blending policy is trained on a human with all three impairments and as such must consider many more cases of how the robot needs to act. By training the policy on a general all three impairments, we allow our blending policy to become more robust to parameter identification and improve on the notion that training a single policy to handle all three impairments is comlpex and time-consuming due to sample inefficiency.

TABLE I

TRAINED POLICIES AND THEIR RESPECTIVE OBSERVATION AND ACTION SPACES

Policy	Observation Space	Action Space
Involuntary Movement	34 human joint values 30 robot joint values	10 human joint values 7 robot joint values
Weakness	34 human joint values 30 robot joint values	10 human joint values 7 robot joint values
Limit Range of Motion	34 human joint values 30 robot joint values	10 human joint values 7 robot joint values
Blending Policy	Only System Parameters: 1 for Estimate Weakness 1 for Estimated Range Limit 10 for Estimated Involuntary Movement Joints	3 weighted values for blending the policies

TABLE II

TRAINED POLICIES AND THEIR RESPECTIVE OBSERVATION AND ACTION SPACES

Method	Policy Blending	State Space
Domain Randomization	No	Trained from Human and Robot Observation Space
UKF	Yes	12 Parameters Estimated By UKF Sampling Real World
Autotuned SPM	Yes	12 Parameters Estimated by Mapping Function of Interaction Between Policy and Real World
Perfect Parameters	Yes	Parameters are Passed as the State Space at the Start of Each Epsiode

C. Training of Domain Randomization

To train the domain randomization model, we train on a human invoking exhibiting all three impairments. Similar to above, we use PPO with layers of 64 nodes each. However, these impairments are now sampled uniformly as such:

(a) Involuntary Movement: The noise for each joints angle is between $[-10, 10]$ degrees.

(b) Weakened Strength: We consider a weakness coefficient between $[0.25, 1]$.

(c) Limited Range: We consider range limitations between $[.5, 1]$ times the original motion.

VI. Discussion and Results

For our intial sub-policies, we can see that the weakness and limit based policies can achieve a higher reward consistenly over the tremor policy in Fig. 3. For our blending policy, we consider the best performing sub-policies and only train on humans with a variation of all three impairments. As such, in Fig. 5 can see that the rewards are much lower than those of the individual policies. Additionally, we notice that there is a significant advantage to using a clustering based policy with system identification over general domain randomization. Furthermore, we can see that the ability to sample the real-world parameters enhances the policies overal convergence as the auto tuned policies struggle to achieve the same success as the UKF based or the baseline

Fig. 2. Example of our robot completing the itching task even when the human is disfunctionally moving thier arm upward

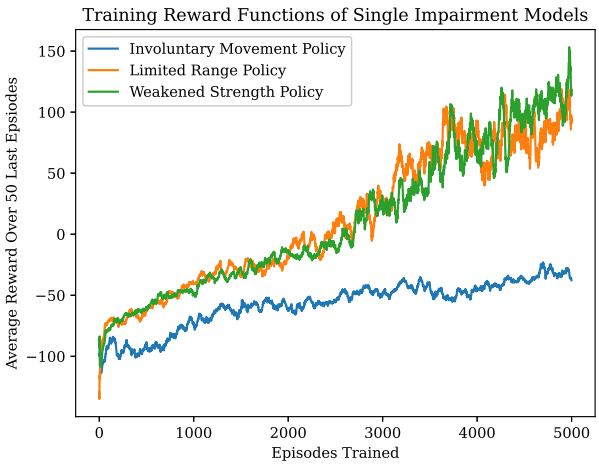

Fig. 3. Training reward average over 50 episodes for single impairment policies. The rewards here are averaged over the training of 3 seeds.

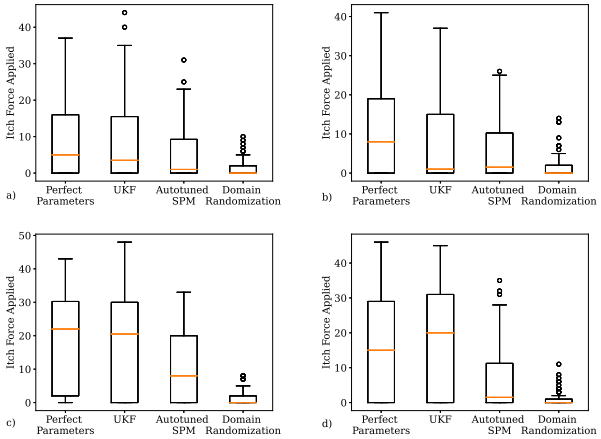

Fig. 4. Application of the trained policy to a real environment for 100 separate epsiodes. We use the highest reward policy for each situation. a) Shows the itch force applied when the human has a variation of all three impairments. b), c), and d) show the force applied when the human has a single impairment in the form of limited range, weakness, or involuntary motion respectively.

(system parameters are pefectly fed everytime). To further evaluate our policies, we define a testing experiment in which we undergo 100 episodes of our human exhibiting all three impairments in which the impairment values are sampled as above. We still utilize the given system indentification method for estimating the state space of the blending policy and Fig. 4 demonstrates a boxplot of each in a. Furthermore, we consider experiments in which the human only inacts a single impairment and the results are shown in b, c, and d of Fig. 4 and Table III respectively.

From the boxplots, it is not obvious to see that the variations between each group are statstically significant and thus, from the plots, we cannot fully conclude one policy is significantly better than another. As such, we perform a Wilcoxon Signed Rank Test [20] between each pair of mdoels in the case of the generalized (3 impairment human) experiment. The results are shown in Table IV where those p-values that are statistically significant are in green. The only models that are not significantly different from each are the UKF-based model and the baseline model fed with the system parameters. As such, we can determine two important things about our approach. First, the policy blending approach has a significant improvement over general domain randomization in terms of both sample efficiency and performance. Second, our design can successfully employ various types of system identification; however, those identification methods may significantly affect the overall performance of the policy and should be bsaed on the max amount of domain knowledge available.

TABLE III

MEAN AND STDEV OF EACH METHOD GIVEN A SEPECIFIC IMAPIRMENT

	Combined Impairments		Involuntary Movement Impairment		Limited Range of Motion		Weakness in Joints	
Method	Mean	STDEV	Mean	STDEV	Mean	STDEV	Mean	STDEV
Domain Randomization	1.33	2.18	0.96	1.94	1.56	2.77	1.07	1.90
UKF	8.68	**10.58**	**18.14**	**14.62**	8.05	10.09	18.13	**14.49**
Autotuned SPM	5.26	7.35	6.34	8.67	5.71	7.35	10.42	10.32
Perfect Parameters	**9.03**	10.23	15.02	14.40	**11.2**	**11.71**	**19.0**	13.81

Given this, we must note limitation of our scheme is that we need to develop the sub-policies; however, these theoretically provide us stability and robustness when faced with unknown environments. Additionally, given that these sub-policies can be reused as they are now decoupled from the main blending policy, different approaches can quickly be tested and tuned - a problem limiting current domain randomization methods.

TABLE IV

WILCOXON PAIRED TEST P-VALUE SCORES

Wilcoxon Test P-Values	Domain Randomization	UKF	Autotuned SPM	Perfect Parameters
Domain Randomization	-	1.02E-8	6.52E-6	1.05E-9
UKF	1.02E-8	-	2.67E-4	0.53
Autotuned SPM	6.52E-6	2.67E-4	-	3.02E-4
Perfect Parameters	1.05E-9	0.53	3.02E-4	-

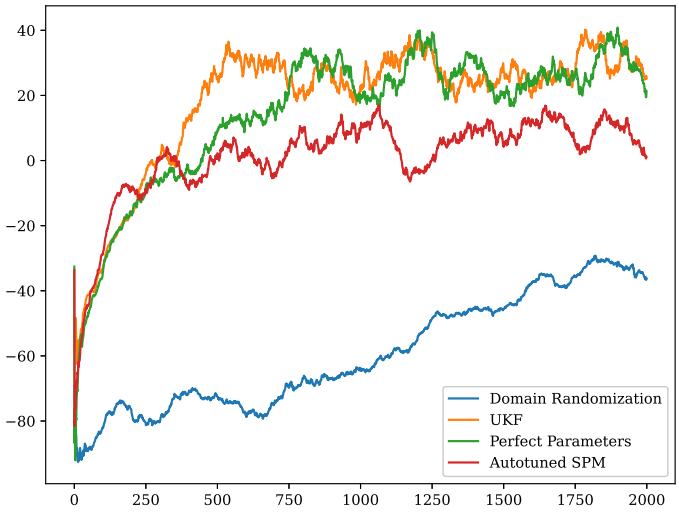

Fig. 5. Training reward averaged over 50 episodes for our policy exploration methods. The rewards here are averaged over the training of 3 seeds.

VII. Conclusion and Future Work

In this work, we present a concurrent policy blending and system identification scheme for learning generalized models with respect to system parameters. With this scheme, we demonstrate the ability to solve a collaborative human and robot task in which the human is impaired with multiple separate, but impactful conditions. Additionally, we demonstrate that our policy outperforms the sample inefficient domain randomization as we can utilize state-of-the-art system identification methods to significantly outtrain a single general policy. As such, in this work we provide a framework for efficiently training generalized policies that are robust to a ever changing system parameters.

References

[1] M. Kaspar, J. D. M. Osorio, and J. Bock, "Sim2real transfer for reinforcement learning without dynamics randomization," *CoRR*, vol. abs/2002.11635, 2020. [Online]. Available: https://arxiv.org/abs/2002.11635

[2] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," 2017.

[3] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," 2018.

[4] L. Ljung, *System Identification: Theory for the User*. USA: Prentice-Hall, Inc., 1986.

[5] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018. [Online]. Available: http://dx.doi.org/10.1109/ICRA.2018.8460528

[6] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," *CoRR*, vol. abs/1806.07851, 2018. [Online]. Available: http://arxiv.org/abs/1806.07851

[7] A. B. Martinsen, A. M. Lekkas, and S. Gros, "Combining system identification with reinforcement learning-based mpc," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 8130–8135, 2020, 21st IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896320329542

[8] X. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," 05 2018, pp. 1–8.

[9] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, "Solving rubik's cube with a robot hand," 2019.

[10] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, "Active domain randomization," 2019.

[11] T. Narita and O. Kroemer, "Policy blending and recombination for multimodal contact-rich tasks," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2721–2728, 2021.

[12] A. Dragan and S. Srinivasa, "A policy blending formalism for shared control," *International Journal of Robotics Research*, vol. 32, no. 7, pp. 790 – 805, June 2013.

[13] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[14] K. Åström, "Optimal control of markov processes with incomplete state information," *Journal of Mathematical Analysis and Applications*, vol. 10, no. 1, pp. 174–205, 1965. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022247X6590154X

[15] E. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.

[16] Y. Du, O. Watkins, T. Darrell, P. Abbeel, and D. Pathak, "Auto-tuned sim-to-real transfer," 2021.

[17] Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, and C. C. Kemp, "Assistive gym: A physics simulation framework for assistive robotics," 2019.

[18] A. Clegg, Z. Erickson, P. Grady, G. Turk, C. C. Kemp, and C. K. Liu, "Learning to collaborate from simulation for robot-assisted dressing," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2746–2753, 2020.

[19] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[20] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945. [Online]. Available: http://www.jstor.org/stable/3001968