Preprints, 4th IFAC Conference on
Advances in Proportional-Integral-Derivative Control
Almería, Spain | June 12-14, 2024

Gain Scheduled PI controller design using Multi-Objective Reinforcement Learning

Kranthi Kumar P * **Ketan P Detroja** **

* *Department of Artificial Intelligence, Indian Institute of Technology, Hyderabad, India (e-mail: ai20resch14002@iith.ac.in).*
** *Department of Artificial Intelligence, Indian Institute of Technology, Hyderabad, India (e-mail: ketan@ee.iith.ac.in).*

Abstract: Gain scheduling, a widely adopted control design approach, performs tasks by decomposing into sub-problems. It has found successful applications across diverse fields, including aerospace and industrial process control (Leith and Leithead (2000)). Reinforcement Learning (RL) has become increasingly significant in process control applications, thanks to the advancement of sophisticated algorithms and the rise in computational speeds. An attempt has been made to enhance the performance of a Proportional-Integral (PI) controller by integrating the benefits of both gain scheduling and RL. The manuscript focuses on designing a Gain Scheduled PI controller using Multi-Objective RL (GS-MORL). Proposing the training of multiple RL-based PI controllers with distinct objectives with customized reward functions to derive tuning parameters for each goal. The switching between these parameter sets is facilitated by a gain schedule variable. The controller parameters are adjusted based on the error between actual and desired responses. The aim is to enhance overall performance across various performance indices, including the integral of absolute error and time response specifications. The performance improvement from the proposed work has been demonstrated using two simulation case studies, one involving an underdamped second-order system and an integrating type steam turbine system. Comparative analysis against existing literature revealed superior performance across all indices.

Keywords: Gain Scheduling; Multi-Objective RL; PI controller; PPO algorithm; Steam Turbine System; Step response.

1. INTRODUCTION

Evolution in Reinforcement Learning (RL) has surged and fueled by advanced algorithms, enhanced computational speeds, and efficient hardware resource utilization. This progress has enabled the application of RL techniques in control systems, overcoming certain limitations of traditional control methods.

A recent study (Nian et al. (2020)) suggests that employing reinforcement learning (RL) for PID (Proportional-Integral-Derivative) controller tuning is preferable over replacing the controller entirely with an RL network in industrial process control applications. Control systems are crucial in industrial and experimental applications, ensuring smooth and efficient operation. Various control schemes exist for industrial applications, with PID being the most commonly used (Ang et al. (2005)) and effective controller. The conventional PID controllers may prove ineffective for higher-order systems, time-delay systems, time-varying systems, parameter-varying systems, nonlinear systems, and systems with uncertainties due to their linear structure. Dealing with uncertainties in a system stands as a fundamental concern in control theory. One of the prominent techniques to deal with these systems is gain scheduling. Gain-scheduling control of linear parameter varying systems (Rugh and Shamma (2000)) has garnered substantial focus in both theory and practical applications. The authors emphasized linearization-focused scheduling and approaches based on linear parameter variations, providing examples of their applications in flight control and automotive engine control. Recent work aimed to achieve an efficient and non-conservative gain-scheduled control system (Elkhatem and Engin (2023)), ensuring high performance and stability across various flight conditions and transitions for the B747 aircraft using evolutionary algorithms. Various gain scheduling techniques (da Silva Campos et al. (2021), Man et al. (2022), Coutinho et al. (2022), Romero et al. (2022)) have been proposed for different systems. Singh and Kaur (2016) took a similar approach. The authors employed Fuzzy rule-based gain scheduling to adjust parameters based on error and error change, emphasizing the necessity of understanding model dynamics. The performance comparison of a fixed PI controller with the gain-scheduled PI controller for the control of the boiler-turbine system showed that the gain-scheduled PI controller performs better than the other PI and PID controllers (Kashyap et al. (2022)). In this work, it is proposed to employ gain scheduling techniques to enhance the performance of linear systems across various performance metrics, including Integral of Absolute Error (IAE) and time domain specifications.

Tuning controllers for improved performance is a delicate balance. Achieving better rise time requires aggressive tuning while minimizing overshoot demands a relatively sluggish approach. Therefore, optimizing all indices with a single set of controller parameters may be contradictory to each other. To address these challenges, a multi-objective RL framework has been proposed in this work. Where multiple RL agents will be trained to achieve different objectives with customized reward functions. The efficacy of reinforcement learning hinges on crafting a proficient, dense, and precise reward function tailored to the specific task (Chakraborty et al. (2023)). In process control applications, the reward is designed to achieve desired objectives in terms of overall performance or any particular performance index like IAE, time domain specifications, etc. In earlier works (Kumar and Detroja (2022), Kumar and Detroja (2023)), control relevant reward functions were designed to achieve better set-point tracking and disturbance rejection. In the proposed approach, multiple controllers, each having a specific objective, are designed using RL. The controller tuning parameters obtained with multiple agents will be applied to a single controller using the gain scheduling technique. The scheduling variable is the error between the actual and desired output of the plant. Based on the error magnitude, the controller switches between different sets of tuning parameters to improve overall performance. The proposed framework is termed as 'GS-MORL' (Gain Scheduled PI controller design using Multi-Objective RL) and is further explained in Section 3.

For the first time, multi-objective reinforcement learning has been used for a gain scheduling based controller for linear systems for performance optimization. Recently proposed Proximal Policy Optimization (PPO) algorithm is computationally efficient, can handle continuous variables, and provides better performance (MacHalek et al. (2020)), therefore PPO algorithm has been selected for training the RL agent in the present study. Two realistic simulation case studies are presented in the manuscript to highlight the benefits of the proposed GS-MORL framework. These case studies include a Steam Turbine system (Dulau and Bica (2014)), which is a second order system with the integrator. The changes in operating zones due to load changes will be taken care of by the boiler operation. The results show that the proposed method achieved better performance and smoother control compared to other methods.

The contributions of this study are as follows: 1) Formulating control relevant reward functions for RL based controller design; 2) Multi-objective RL for the training of different agents to achieve desired objectives; 3) Switching of the controller tuning parameters using the gain scheduling technique to achieve improvement across all the performance indices; 4) Design of PI controller for second order system, 5) Design of PI controller for steam turbine system, 6) Comparing the performance of the proposed work with the other benchmark methods.

The manuscript is organized as follows. In Section 2, the basic RL philosophy and PPO algorithms are briefly explained. The proposed Gain scheduled PI controller design using multi-objective RL is explained in Section 4. Subsequently, the performance of the proposed work in process applications is compared followed by detailed analysis and conclusions.

2. MATERIALS AND METHODS

2.1 Reinforcement Learning

Reinforcement learning (Sutton and Barto (2018)), a data-driven AI algorithm rooted in trial-and-error learning, involves training an agent aiming to expand its capabilities. The RL agent learns a policy, mapping states to actions, through a trial-and-error search guided based on a reward signal function. In discrete time steps, the agent interacts with an environment, receiving state s_t, taking action a_t, and obtaining a reward r_t, creating a sequential trajectory throughout an episode. The policy π guides the mapping of states to actions, and the learning process is formalized using a Markov Decision Process (MDP).

2.2 PPO Algorithm

Proximal Policy Optimization (PPO) (Schulman et al. (2017)) is an on-policy, model-free policy-gradient algorithm, suitable for environments with discrete or continuous action spaces. Unlike its predecessor TRPO (Trust Region Policy Optimization) (Schulman et al. (2015)), PPO is simpler to implement yet performs comparably well. It employs first-order methods and clipping techniques to learn policies iteratively, ensuring stability by constraining deviations from the existing policy. PPO's learned policy is inherently stochastic, facilitating exploration through action sampling. As training progresses, the exploration randomness diminishes, and the agent increasingly exploits learned policies.

3. PROPOSED GAIN SCHEDULED PI CONTROLLER DESIGN USING MULTI-OBJECTIVE RL

The manuscript introduces a gain-scheduled PI controller design employing multi-objective reinforcement learning, referred to as GS-MORL for discussion. In this approach, the control task is divided into three distinct segments. Each segment undergoes training with an individual RL-based controller, targeting specific objectives. Subsequently, the parameters of these controllers are employed to switch between them, determined by the gain scheduled parameter. Diverging from traditional approaches, which involve creating a single controller to manage the entire control task with a specific goal. The PI controller's tuning parameters are switched by updating them according to the error band, which is the gain scheduling variable in this case.

The schematic of the proposed GS-MORL architecture is depicted in Fig. 1. The PPO Network in Fig. 1 represents the RL agent/policy, which is a parameterized neural network. The continuous actions of the PPO network are the tuning parameters of the PI controller. These are given as

$$a_j = [Kp_j \; Ti_j] \tag{1}$$

$$Kp_j^{min} \leq Kp_j \leq Kp_j^{max}, \; Ti_j^{min} \leq Ti_j \leq Ti_j^{max} \tag{2}$$

Where Kp_j is the proportional gain, Ti_j is the integral time, and $j = 1,2,3$. j represents each training objective. The action space is constrained to mitigate potential disruptions from the policy network's stochastic nature, ensuring a controlled and safe operation of the control loop. PPO network makes decisions based on state vector x_j and reward r_j from the environment. The output of the PI controller is given as

$$u_j = Kp_j(e_j + \frac{ie_j}{Ti_j}) \tag{3}$$

Where e_j is the error and ie_j is the integral error.

3.1 Objective-1 : Quick initial response

The reward functions are selected to attain the specified objectives for each segment, as elaborated further. The first objective of training focuses on improving rise time, making the controller aggressive in the transient stage. The corresponding reward considered to achieve this goal is given as

$$r_1 = -t_r - e_{ss} \tag{4}$$

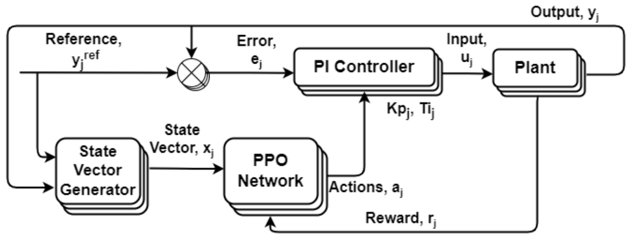

Fig. 1. Schematic of the proposed GS-MORL agent training Architecture

Where t_r represents the rise time (the time required to reach 90% of its steady value) of the response and e_{ss} is the steady-state error. The reward function is formulated in such a way that the controller should be optimized for minimum rise time, ultimately ensuring its convergence to the desired set point. The presence of negative signs signifies that a longer rise time incurs more penalties and diminishes the reward, which applies similarly to other components.

3.2 Objective-2 : Minimizing overshoot

The controller's second objective centers around minimizing maximum overshoot. RL agents strive to achieve this by minimizing the maximum overshoot, and the following reward is considered to attain this goal.

$$r_2 = -M_p - IAE \tag{5}$$

Where M_p represents the maximum overshoot and IAE is the integral of absolute error. The reward function is formulated to tune the controller for a critical damped response i.e. minimizing the overshoot while avoiding over-damped response by incorporating IAE.

3.3 Objective-3 : Fast settling and minimizing steady state error

The third controller objective focuses on minimizing settling time. RL agents aim to expedite the settling process in pursuit of this goal. To attain it, the following reward is taken into account.

$$r_3 = -t_s - e_{ss} \tag{6}$$

Where t_s represents the settling time of the response. The reward formulation requires tuning the controller to minimize settling time.

The plant's output is denoted as y_j, with the reference signal being y_j^{ref}. This consistent reference signal can be applied across all training cases.

3.4 Gain Scheduling

Following the discussed training process, three sets of tuning parameters will be obtained. These parameters will be utilized to create a single PI controller, managing the plant by switching between them through the gain scheduling technique. The schematic representing the same is shown in Fig. 2. The Gain Scheduler block, depicted in Fig. 2, switches between various parameter sets to generate tuning parameters for the controller. The corresponding equations are provided below.

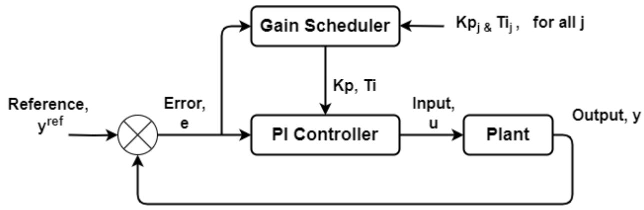

Fig. 2. Gain Scheduled Control Architecture

$$Kp = \begin{cases} Kp_1 & \text{if } |e| \geq \beta_1 \\ Kp_2 & \text{if } \beta_2 \leq |e| < \beta_1 \\ Kp_3 & \text{otherwise} \end{cases} \tag{7}$$

$$Ti = \begin{cases} Ti_1 & \text{if } |e| \geq \beta_1 \\ Ti_2 & \text{if } \beta_2 \leq |e| < \beta_1 \\ Ti_3 & \text{otherwise} \end{cases} \tag{8}$$

Where,

$$\text{error, } e = y^{ref} - y \tag{9}$$

$$\text{constants, } \beta_1, \beta_2 = (0,1), \beta_1 > \beta_2 \tag{10}$$

The selection of constants β_1 and β_2 can be done empirically to get better performance. Equations (7) and (8) indicate that the controller prioritizes aggressive performance, enhancing transient response when the error exceeds β_1. If the error falls within β_1 and β_2, the controller shifts to the second tuning parameters, aiming to minimize maximum overshoot. Finally, when the third set of controller parameters to expedite system settling. Hence with the proposed approach, the performance of the controller will be optimized in terms of all performance indices viz. Integral of Absolute Error, Maximum Overshoot, Rise Time, and Setting Time. Section 4 illustrates this through case studies.

4. RESULTS AND ANALYSIS

4.1 Design of PI Controller for Second Order System

To demonstrate the performance of the proposed framework, an underdamped second-order system has been considered. Typically tuning PI controllers for such systems is difficult as this system has an infinite gain margin and phase margin. The transfer function of the model is given as

$$G(s) = \frac{K}{\tau^2 s^2 + 2\zeta\tau s + 1} \tag{11}$$

Where K is the process gain, τ is the time constant and ζ is the damping factor. A system with $K = 1.5$, $\tau = 3.16$, and $\zeta = 0.95$ has been considered in this case.

The gain scheduled PI controller design presented in Section 3 is employed to train the agents in this system. The plant shown in Fig. 1 is replaced by this second-order system (11) and trained for each of the objectives (4), (5) and (6) separately. The hyperparameter settings for all training objectives are uniform and can be found in Table 1. The rewards obtained during the training processes are shown in Fig. 3. The red (dotted) plot represents the reward during the training process concerning the first objective (Eq. (4)). Blue (dash-dot) and green (dashed) plots represent rewards for Eq. (5) and Eq. (6), respectively. Table 2 presents tuning parameters for each objective alongside their performance indices. Additionally, Fig. 4 displays the step responses corresponding to these tuning parameter sets.

Table 1. Hyper-Parameter Settings

Parameter	Value
Learning Rate	5e-05
Kullback-Leibler(KL) Coefficient	0.2
Kullback-Leibler(KL) target	0.01
Clipping ratio	0.3
Training batch size	5
SGD mini-batch size	5
No of SGD Iterations	1
No of time steps	500
No of episodes	1000

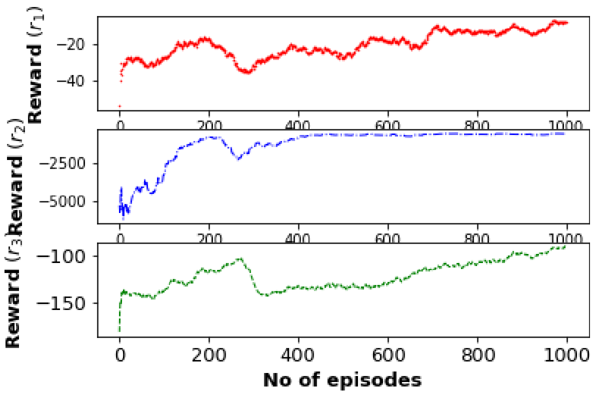

Fig. 3. Reward during training of RL controller with different objectives

Table 2. Static Performance Indices (Training with different objectives)

Objective	Kp	Ti	IAE	$M_p(\%)$	$t_r(\mathrm{s})$	$t_s(\mathrm{s})$
Objective-1	4.67	15.71	34.24	36.3	**2.4**	23.7
Objective-2	1.63	8.2	22	**9.9**	4.6	17.5
Objective-3	2.19	8.64	19.7	18.3	3.7	**14.8**

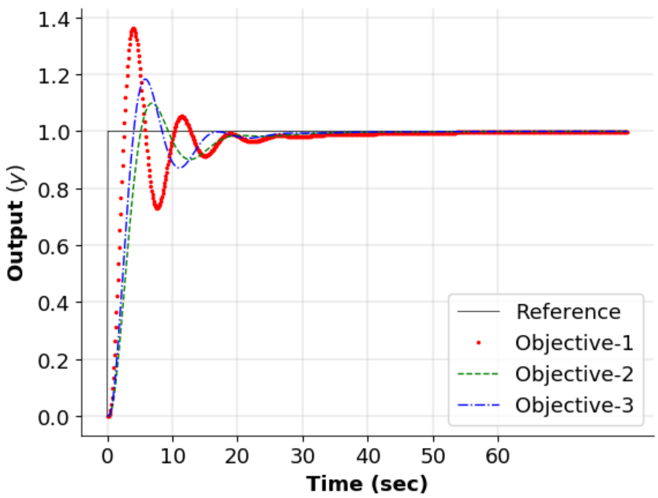

Fig. 4. Step Response of the controllers with different objectives

Observations from Fig. 3, Fig. 4, and Table 2 indicate successful training of RL controllers based on their respective objectives. The first method excels in achieving the best rise time, the second method minimizes overshoot, and the third method

Table 3. Static Performance Indices (For Second Order System)

Method	Kp	Ti	IAE	$M_p(\%)$	$t_r(\mathrm{s})$	$t_s(\mathrm{s})$
MATLAB	1.2	6.1	16.7	8.9	5.4	18.7
Shinskey	3	3.83	39.82	52.7	**2.8**	25.2
Skogestad	1.44	6.6	16.51	12.3	4.8	17.5
GS-MORL	-	-	**16.4**	**8.5**	3.7	**14.8**

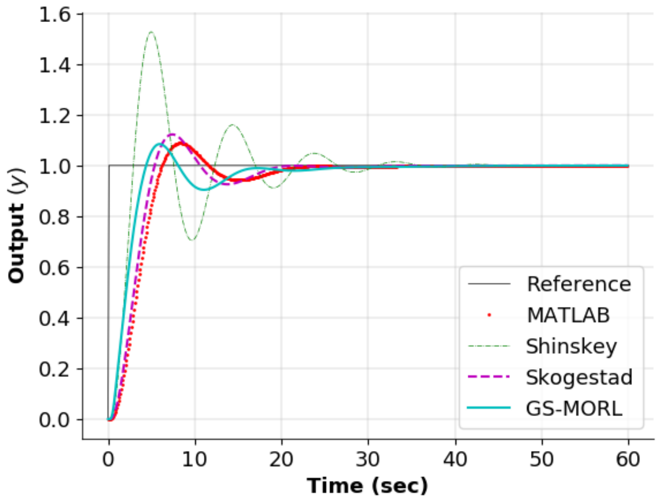

Fig. 5. Comparison of step response among various methods

ensures a quicker settling of the system. However, in all cases, trade-offs exist with other parameters, sacrificing them for the sake of achieving the specified objective indices.

To enhance overall performance metrics, the gain scheduling technique is employed. This technique dynamically adjusts parameters according to (7) and (8) to achieve our performance goals. Experimentation determined the constant parameters β_1 and β_2 to be 0.95 and 0.1, respectively. The label 'GS-MORL' is assigned to the response of this gain-scheduled multi-objective controller for comparison.

Performance benchmarking of the gain scheduled controller involves comparing it with various methods such as Shinskey (1994) as discussed in O'Dwyer (2009) and Skogestad (2003). Additionally, the MATLAB PID tuner, guided by human expertise, is employed for controller tuning for comparative analysis. The tuning parameters and corresponding performance indices are detailed in Table 3. The step responses of these methods are illustrated in Fig. 5. Referring to Table 3 and Fig. 5, it is evident that the gain scheduled controller, formulated according to the proposed framework, exhibits superior overall performance compared to alternative controllers across all performance indices. While there may be instances of better rise time in certain cases, it comes at the expense of compromising other parameters heavily.

4.2 *Design of PI Controller for Steam Turbine System*

The methodology has been applied to a physical process involving a steam turbine unit within a thermal power plant system. In this context, a gain scheduled PI controller is designed using multi-objective RL. Refer to Fig. 6 for the schematic representation of the steam turbine system. The steam turbine transforms high-pressure and high-temperature steam's stored

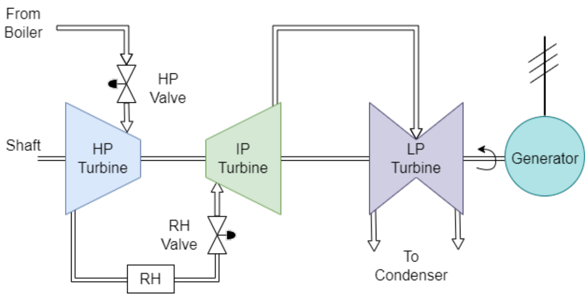

Fig. 6. Steam Turbine system schematic

energy into rotary energy, subsequently converted into electricity by the generator. Each turbine section includes moving blades connected to the rotor and stationary vanes that accelerate steam to high velocity. The units may feature high-pressure (HP), intermediate-pressure (IP), and low-pressure (LP) turbine sections, depending on the configuration. The steam turbine features high-pressure valves (HP Valve) and re-heater valves (RH Valve). Steam enters the HP section via the HP Valve and inlet piping. HP exhaust steam passes through the re-heater (RH). Reheated steam enters the IP turbine section through the RH Valve and inlet piping. The IP turbine's exhaust flows to the LP inlet. The conversion of steam energy into rotational energy by the shaft occurs as the steam expands through three stages. To generate power and maintain grid frequency, it's crucial to keep the shaft speed at a desired value, regardless of the load. Disturbances in shaft speed occur due to load changes, necessitating the regulation of steam flow through the operation of the HP valve position to maintain the required speed. HP valve regulation primarily addresses minor load adjustments, necessitating boiler collaboration for significant load variations. Consequently, this control mechanism operates effectively within a specific operational range, resembling a linear system ideal for testing the proposed work. The mathematical model of the above process based on Dulau and Bica (2014) for the desired speed to HP valve position is given as

$$G(s) = \frac{0.0833(2.25s+1)}{s(7.5s+1)(0.25s+1)} \tag{12}$$

The behavior of the system is of integrating type having 2 poles and 1 zero.

The method proposed in Section .3 is employed to train the RL agent for various objectives, consistent with the earlier case. The responses of the individual training objectives are shown in Fig. 7 and the tuning parameters corresponding to each objective are

$$Kp_1 = 8.31,\ Ti_1 = 296.75 \tag{13}$$
$$Kp_2 = 0.84,\ Ti_2 = 2177.7 \tag{14}$$
$$Kp_3 = 7.38,\ Ti_3 = 305.4 \tag{15}$$

With these sets of parameters, a gain scheduled controller has been envisaged as shown in Fig. 2 using (7) and (8). The performance has been compared with (Bialkowski (1996)) as discussed in (O'Dwyer (2009)) and (Skogestad (2003)) along with MATLAB PID tuner. The bench-marking methods in this scenario differ from previous ones, involving the integration type second-order systems compared to non-integrating second-order systems. The step response of these methods is shown in Fig. 8 and the corresponding performance indices are tabulated in Table 4. The gain scheduled controller exhibits superior overall response compared to others, as evident from Fig. 8 and Table 4. While Bialkowski shows better IAE and rise time, it comes at the cost of significantly high overshoot and

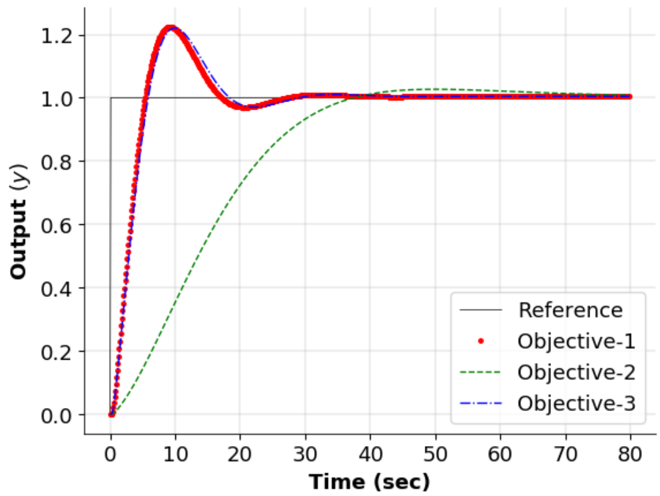

Fig. 7. Step response of the controller with different objectives for Steam Turbine model

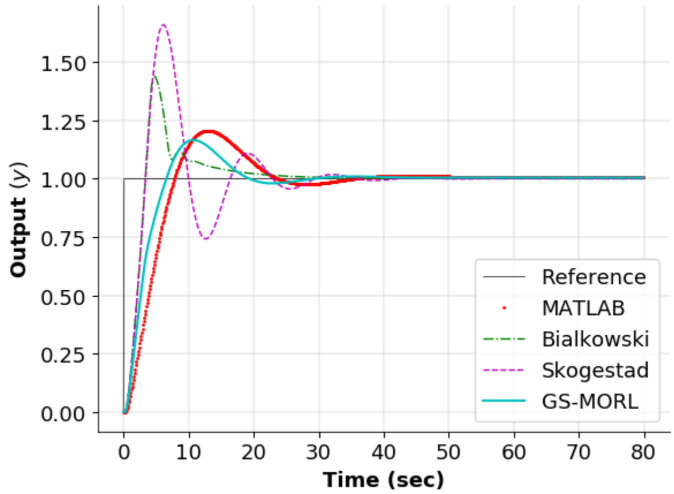

Fig. 8. Comparison of step response of various methods for Steam Turbine model

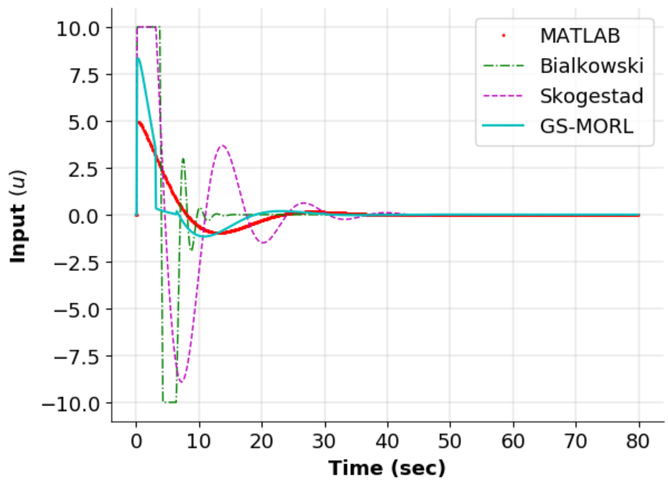

Fig. 9. Controller outputs/Plant Inputs for step change for Steam Turbine model

Table 4. Static Performance Indices
(For Steam Turbine System)

Method	Kp	Ti	IAE	$M_p(\%)$	t_r(s)	t_s(s)
MATLAB	4.932	250	75.02	20.6	6.8	21.8
Bialkowski	90	7.5	**21.33**	44	**3.2**	**17.5**
Skogestad	16.13	4	50.9	65.9	**3.2**	27.4
GS-MORL	-	-	46.4	**16.6**	5.5	**17.5**

prolonged controller output saturation, as illustrated in Fig. 9. A similar case is observed with Skogestad also. This extended saturation, resulting from a high gain, is undesirable for real systems.

5. CONCLUSION

The manuscript introduces a gain scheduled PI controller design through multi-objective RL. The proposed approach is validated on a second-order system and an integrating type realistic steam turbine model. Three distinct controllers are trained using RL for different objectives. A gain scheduled controller is devised, utilizing the three sets of tuning parameters to enhance overall performance. Comparative analysis with various methods reveals significant performance improvements across all indices for the proposed approach. In future research, the authors aim to investigate the development of a gain scheduled RL controller for more complex systems and multi-variable systems.

REFERENCES

Ang, K.H., Chong, G., and Li, Y. (2005). Pid control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, 13, 559–576. doi: 10.1109/TCST.2005.847331.

Bialkowski, W. (1996). The control handbook. 1219–1242.

Chakraborty, S., Bhaskar, A., Singh, A., Tokekar, P., Manocha, D., and Bedi, A.S. (2023). Rebel: A regularization-based solution for reward overoptimization in reinforcement learning from human feedback. URL `https://raaslab.org/projects/REBEL`.

Coutinho, P.H., Peixoto, M.L., Bessa, I., and Palhares, R.M. (2022). Dynamic event-triggered gain-scheduling control of discrete-time quasi-lpv systems. *Automatica*, 141. doi: 10.1016/J.AUTOMATICA.2022.110292.

da Silva Campos, V.C., Nguyen, A.T., and Palhares, R.M. (2021). Adaptive gain-scheduling control for continuous-time systems with polytopic uncertainties: An lmi-based approach. *Automatica*, 133. doi: 10.1016/J.AUTOMATICA.2021.109856.

Dulau, M. and Bica, D. (2014). Mathematical modelling and simulation of the behaviour of the steam turbine. *Procedia Technology*, 12, 723–729. doi:10.1016/j.protcy.2013.12.555.

Elkhatem, A.S. and Engin, S.N. (2023). Enhancing performance and stability of gain-scheduling control system using evolutionary algorithms: A case study on transport aircraft. *Expert Systems with Applications*, 213. doi: 10.1016/J.ESWA.2022.118859.

Kashyap, G.S., Sant, A.V., and Yadav, A. (2022). Gain scheduled proportional integral control of a model based boiler turbine system. *Materials Today: Proceedings*, 62, 7028–7034. doi:10.1016/J.MATPR.2022.01.056.

Kumar, P.K. and Detroja, K.P. (2022). Parameterized adaptive controller design using reinforcement learning and deep neural networks. 121–126. doi: 10.1109/ICC56513.2022.10093404.

Kumar, P.K. and Detroja, K.P. (2023). Design of reinforcement learning based pi controller for nonlinear multivariable system. 1–6.

Leith, D.J. and Leithead, W.E. (2000). Survey of gain-scheduling analysis and design. *International Journal of Control*, 73, 1001–1025. doi:10.1080/002071700411304.

MacHalek, D., Quah, T., and Powell, K.M. (2020). Dynamic Economic Optimization of a Continuously Stirred Tank Reactor Using Reinforcement Learning. *Proceedings of the American Control Conference*, 2020-July, 2955–2960. doi: 10.23919/ACC45564.2020.9147706.

Man, J., Song, X., Song, S., and Lu, J. (2022). Finite-time synchronization of reaction–diffusion memristive neural networks: A gain-scheduled integral sliding mode control scheme. *ISA Transactions*. doi: 10.1016/J.ISATRA.2022.08.011.

Nian, R., Liu, J., and Huang, B. (2020). A review on reinforcement learning: Introduction and applications in industrial process control. *Computers and Chemical Engineering*, 139. doi:10.1016/j.compchemeng.2020.106886.

O'Dwyer, A. (2009). *HANDBOOK OF PI AND PID CONTROLLER TUNING RULES*. Imperial College Press, 3 edition.

Romero, E.E., Bérenguer, C., and Martinez, J.J. (2022). Gain-scheduling wind-turbine control to mitigate the effects of weather conditions on the drive-train degradation. *IFAC-PapersOnLine*, 55, 508–513. doi: 10.1016/J.IFACOL.2022.07.179.

Rugh, W.J. and Shamma, J.S. (2000). Research on gain scheduling. *Automatica*, 36, 1401–1425. doi: 10.1016/S0005-1098(00)00058-3.

Schulman, J., Levine, S., Moritz, P., Jordan, M.I., and Abbeel, P. (2015). Trust Region Policy Optimization. *International conference on machine learning (pp. 1889-1897), PMLR*. URL `http://arxiv.org/abs/1502.05477`.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Openai, O.K. (2017). Proximal Policy Optimization Algorithms. *arXiv:1707.06347*.

Shinskey, F. (1994). Feedback controllers for the process industries. *McGraw-Hill Professional*.

Singh, S. and Kaur, M. (2016). Gain scheduling of pid controller based on fuzzy systems. doi:10.1051/conf/2016.

Skogestad, S. (2003). Simple analytic rules for model reduction and pid controller tuning. *Journal of process control*, 13, 291–309. URL `www.elsevier.com/locate/jprocont`.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement Learning: An Introduction Second edition, in progress*. The MIT Press Cambridge, Massachusetts London, England.