

Transformer for the Design of Automatic Control Systems with Prefilters

Isaac Hughes, John F. O'Brien

Department of Electrical Engineering and Computer Science, University of Wyoming, Laramie, USA

Email: ihughes@uwyo.edu

How to cite this paper: Hughes, I. and O'Brien, J.F. (2024) Transformer for the Design of Automatic Control Systems with Prefilters. *World Journal of Engineering and Technology*, 12, 575-586.

<https://doi.org/10.4236/wjet.2024.123037>

Received: May 22, 2024

Accepted: July 26, 2024

Published: July 29, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

A machine learning model, using the transformer architecture, is used to design a feedback compensator and prefilter for various simulated plants. The output of the transformer is a sequence of compensator and prefilter parameters. The compensator and prefilter are linear models, preserving the ability to analyze the system with linear control theory. The input to the network is a window of recent reference and output samples. The goal of the transformer is to minimize tracking error at each time step. The plants under consideration range from simple to challenging. The more difficult plants contain closely spaced, lightly damped, complex conjugate pairs of poles and zeros. Results are compared to PID controllers tuned for a similar crossover frequency and optimal phase margin. For simple plants, the transformer converges to solutions which overly rely on the prefilter, neglecting the maximization of negative feedback. For more complex plants, the transformer designs a compensator and prefilter with more desirable qualities. In all cases, the transformer can start with random model parameters and modify them to minimize tracking error on the step reference.

Keywords

Feedback, Machine Learning, Compensator Design

1. Introduction

1.1. Motivations

Automatic control systems, employing negative feedback, continue to be powerful tools across the domain of applied technologies. Mass-scale power systems, all forms of powered transportation and small electronics rely on the application of negative feedback. Despite the broad use of this technology, most modern control designs have remained simple and reliant on technologies like the proportional-integral-derivative (PID) controller. Furthermore, designers simplify

to proportional-integral control or proportional control. The original tuning method by Ziegler and Nichols, provided a simple method of achieving zero steady-state error on the step reference [1]. PID tuning does not require expert training in control theory to apply, accounting for its popularity. The trade-off for this ease of application is performance. An expert in control theory may extract more performance out of a system through the development of a feedback compensator which maximizes negative feedback over the functional bandwidth [2]. After the design of a feedback compensator, a prefilter is designed to improve transient response and reduce steady state error. A machine learning (ML) method is proposed, based on modern architecture, to fill in the gap between the potential performance one could extract from a system, and the expertise required to do so.

1.2. Model Goals

The machine learning model is used to minimize tracking error. The model takes a narrow window of input and output samples and generates parameters for a compensator and prefilter. Tracking error is the difference between the input and output signals. Using this error, the model will update itself to provide a compensator and prefilter which reduces the tracking error. Results of this process are presented where a transformer model successfully minimizes tracking error to the step reference. This is shown to be possible with both simple and challenging plants. Prior work on the utilization of machine learning to improve tracking error relies on inserting neural networks inside control loops, a process which complicates the analysis of the system. The proposed method preserves the ability to analyze the system with linear control theory.

2. Literature Review

Once control engineers began to face plants whose parameters varied over the course of operation, adaptive control methods were needed. Model reference adaptive control (MRAC) was an early milestone in adaptive control systems which used gradient descent to minimize a cost function. This cost function represented the difference between a system's actual output y and the desired model output y_m [3]. If plant parameters change, MRAC detects a difference between actual and desired output and adjusts to maintain a desired response. However, this method still requires skilled control engineers to design the desired model. Virtual reference feedback tuning (VRFT) comes much later in 2000 and provides a direct technique for compensator design. This method utilizes sample input/output data from live testing of a plant to automatically design a compensator which achieves a desired model [4]. VRFT achieves this by minimizing the model reference criterion:

$$J_{MR}(\theta) = \left\| \left(\frac{P(z)C(z;\theta)}{1+P(z)C(z;\theta)} - M(z) \right) M(z) \right\|^2 \quad (1)$$

As such, VRFT requires the designer to construct a desired model. Other

methods like Iterated feedback tuning (IFT) utilize gradient descent to improve existing designs [5]. IFT is useful in improving the performance of PID controllers tuned with existing algorithms. The method examined in this paper does not utilize a reference model or start from any known design.

Beyond compensator design, machine learning has directly intersected with control theory by including neural networks (NN) inside control loops. Including a neural network in the feedback loop has the potential to improve performance of existing designs and compensate for unknown nonlinearities and plant parameter variation [6]. The risk in this approach is that NN-based compensators are black boxes, whose exact details cannot be fully known. Stability of such controllers can be determined by utilizing Lyapunov stability theory, but the breadth of linear control theory cannot be applied. The method used in this paper differs by using machine learning outside the loop, to design linear controllers that stabilize and deliver high performance.

3. Methodology

3.1. Feedback

Of primary importance is the rational function, where s is the Laplace variable.

$$T(s) = C(s)P(s) = \frac{y(s)}{r(s)} \quad (2)$$

$T(s)$ is known as the loop transmission (or the return ratio) of the feedback loop. It is further necessary to define two types of bandwidths, the *control bandwidth*, and the *functional bandwidth*. The control bandwidth, ω_b , is where the magnitude of the loop transmission is equal to 1. This frequency is also known as the *0 dB crossover frequency*. The functional bandwidth is the frequency, ω_f , where the magnitude of the return ratio is approximately constant or, $|T(j\omega)| \cong A_0, \forall \omega \leq \omega_f$, for some constant A_0 . Feedback as a quantity is defined as $|F(s)| = |1 + T(s)|$. Broadly, feedback is defined as being *negative*, *positive*, *negligible*, and *large* by the conditions, $|F(s)| > 1$, $|F(s)| < 1$, $|T(s)| \ll 1$ and $|F(s)| \gg 1$, respectively. These conditions quantify the closed loop system's effect on disturbances. Phase margin is defined as the difference between the phase of the loop transmission function at crossover, and -180° .

3.2. Goals of Feedback

In the task of aggressively controlling a single input, single output (SISO) system, feedback is used to: stabilize an unstable system and maximize available negative feedback. If a plant is open-loop unstable, negative feedback must be applied. The cost for stabilization is an amount of negative feedback that could have been extracted if the plant was stable [2]. This cost is directly proportional to the number of unstable open-loop poles and their frequency. After stabilization, the designer endeavors to maximize negative feedback to improve system response time, its resistance to disturbance, and reduced sensitivity to plant parameter variation. For simple plants, processes like PID can achieve reasonable

results. However, complex plants with challenging dynamics (lightly damped, complex-conjugate pairs of poles) require more sophisticated application of compensation. Methods like PID, while capable of achieving zero steady state error to the step, leave available performance behind. It is imperative then for the designer to maximize negative feedback over the functional bandwidth under the constraints presented by the plant [7]. After designing a feedback compensator for maximum negative feedback, a prefilter is designed to improve transient response and fix steady state error [2]. The complete system is shown in **Figure 1**. This paper proposes a machine learning approach to the automatic generation of both.

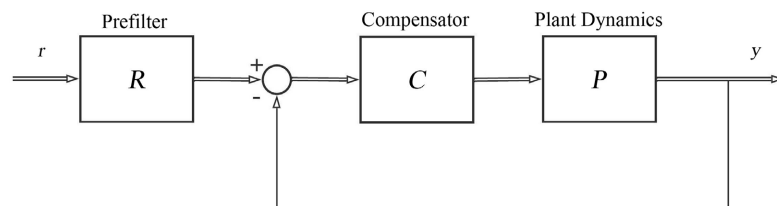


Figure 1. Feedback system with prefilter.

3.3. Transformer Architecture

Progress in complex sequence tasks has been persistent from the development of the long-short-term memory layer to the auto-regressive Transformer. Driven primarily by the challenge of natural language [8], Transformers have been applied to reinforcement learning tasks [9], control tasks [10] and generative models in the natural sciences [11]. The sequence task in this paper requires the Transformer to generate a sequence of parameters for a discrete feedback compensator and prefilter by observing a narrow window of reference and output samples, as in **Figure 2**. The broad behavior of linear dynamic systems requires that the machine learning architecture be capable of responding to many different behaviors over changing time scales. The tendency of the multilayer perceptron network to suffer from catastrophic forgetting requires that the chosen architecture can represent sparse situations like a system's transient response and its lengthy steady state equally. As such, the Transformer is better suited to this task.

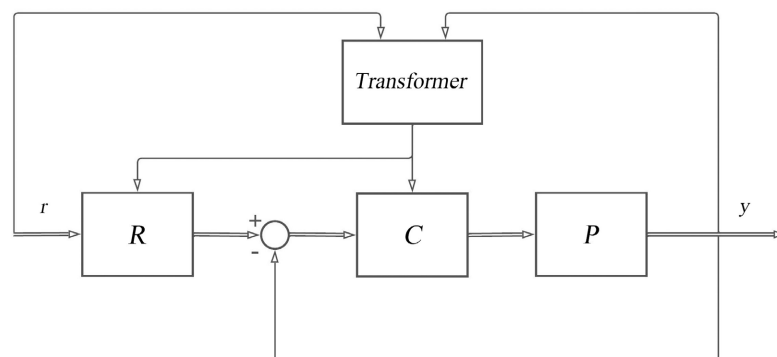


Figure 2. Feedback system with transformer outputting parameters as a function of reference and output signals.

3.4. Model Details

The ML architecture used is a decoder-only transformer. The input to the transformer is a sequence of vectors with recent reference and output samples. The input sequence is as long as the sum of the number of poles in the prefilter, compensator and plant. This sequence length encompasses the samples that will be required to generate the next output. The higher dimensional embedding used as an input to the self-attention mechanism is learned along with the model and consists of a linear layer and a positional encoding layer added together. The output of the transformer is flattened and passed to a final linear layer which outputs the frequencies of the compensator and prefilter parameters. The transformer outputs a vector of model parameters at every timestep. System parameters come in the form of gains, real pole and zero locations, and the real and imaginary parts of complex conjugate pairs. Users select the number of real poles, zeros, complex conjugate pairs of zeros and poles for both the compensator and prefilter. The model used has a single transformer layer, an embedding dimension of 16, a subspace dimension of 64 and eight heads. The sequence of input and output samples is expanded up to the embedding dimension with a densely connected linear layer. This higher dimensional sequence then has the multi-head self-attention equation applied, in parallel with other self-attention heads. The self-attention equation is,

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \quad (3)$$

where Q , K and V are the matrix products between separate weight matrices and the embedded sequence and d_k is the embedding dimension. The output of the attention mechanism is added to the embedded sequence and normalized. The normalized sequence is passed through another linear layer, flattened into a vector and passed through a final linear layer that results in the output dimension. The output vector is a vector of frequencies. Consider an example output where the system has a compensator with one real zero and a complex conjugate pair of poles. The compensator parameters are output as

$$[C_k, Z, I, R]. \quad (4)$$

The parameters are arranged into the compensator:

$$\frac{C_k(z+Z)}{z^2 + 2Rz + (R^2 + I^2)}. \quad (5)$$

This process is repeated for a prefilter, then both systems are convolved with the plant to implement the closed loop transfer function:

$$\frac{RCP}{1+CP} \quad (6)$$

A linear dynamic model computes the output of the system, using the discrete difference equation. Automatic differentiation is used to compute the gradient of the output with respect to the compensator parameters. This limitation is why

accurate plant parameters must be known at training time. Plant parameters do not need to be known after training.

3.5. Training

The model uses the adaptive momentum (Adam) optimizer for model updates [12]. Decay and squared decay parameters for Adam are reduced to 0.75 for both. This reduction is necessary as momentum in parameter updates can easily send model parameters outside the unit circle, resulting in instability in the model and training. Adam is still a necessary optimizer for this application as the output of the feedback system can be close to zero for many time steps, resulting in infinitesimal changes in model parameters with stochastic gradient descent. A learning rate of 10^{-4} was used.

Training data consists of either a single reference, trained on iteratively or a plethora of references generated randomly. Randomly generated references subject the transformer to realistic control inputs and can be selected with specific applications in mind. For this paper, the step reference is used for training.

Loss is computed as the squared tracking error:

$$\frac{1}{2}(r - y)^2 \quad (7)$$

The model is updated at each time step during training. It is possible to train on the mean-squared error of an entire reference and output sequence, but this method tends not to converge.

3.6. Model Environment

All results are trained in a model environment free of disturbances and non-linearities. Any live system includes these challenges. To improve the reliability of a machine learning model in application, the training environment should match the real environment as closely as possible. As well, machine learning models, including the proposed method, pose the risk of sending feedback systems unstable. The proposed method preserves the ability for the system stability to be analyzed using linear control theory.

4. Performance on Known Plants

4.1. Comparison to PID

PID is an industry wide standard and makes a good comparison for the proposed method. The PID tuning was done by MATLAB's PID tuning application [13]. This application lets the user choose a desired bandwidth and phase margin. All bandwidths and phase margins were set for best performance. Performance is measured by the following criteria:

- Rise time: the time to reach 90% of the system's final value.
- Settling time: the time after which the system is bounded within $\pm 5\%$ of its final value.
- Overshoot: the ratio of the peak response value to the final value (reported

as a percent).

- Steady state error: the difference between the step amplitude and the system's final value (reported as a percent).
- Absolute tracking error: the sum of the absolute value of all tracking error samples.

The transformer outputs a new set of parameters at each time step, the frequency domain plots shown are using the most active sample from the step response. Each plant is sampled at 1000 kHz.

4.2. Simple Plants

Consider the plant, P1:

$$\frac{z - 0.1}{z^2 - 0.7z + -0.1}. \quad (8)$$

P1 has a simple lowpass shape and is not difficult for PID to control. The comparison between PID and transformer performance on the step reference is shown in **Figure 3**. The loop transmission of both the transformer output and PID are shown in **Figure 4**. Performance data is compared in **Table 1**.

Table 1. Performance data for P1.

	Transformer	PID
Rise Time (ms)	4	3
Settling Time (ms)	4	3
Overshoot (%)	0	3
Steady State Error (%)	2	0
Absolute Tracking Error	12.2	3.9

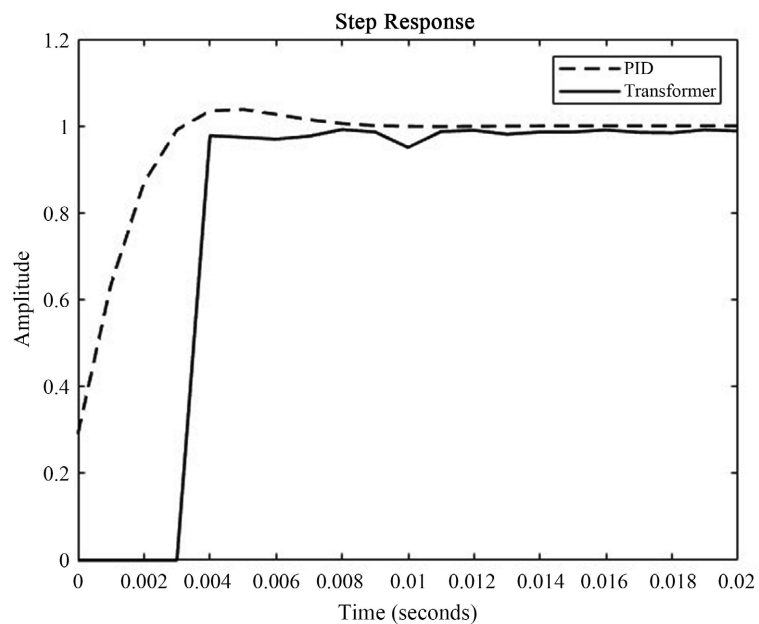


Figure 3. Step response of closed loop systems with plant P1.

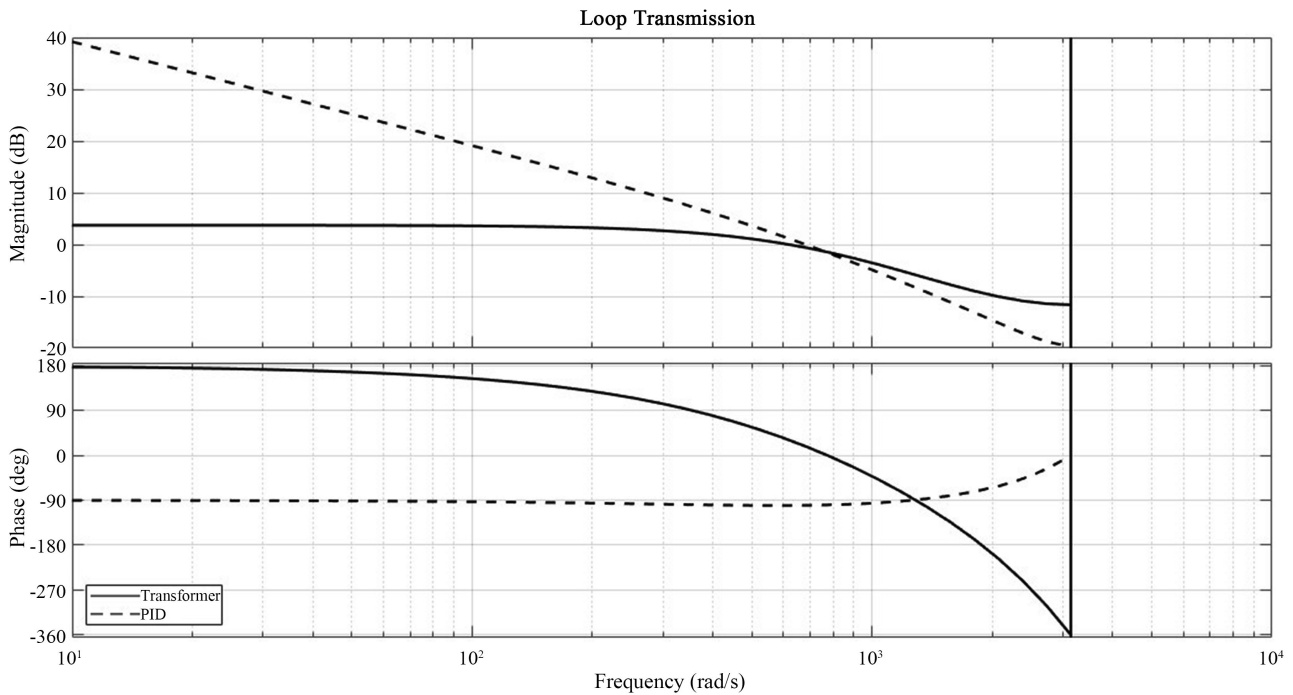


Figure 4. Loop transmission of transformer output and PID with plant P1.

For this plant, the transformer output is a compensator with two real zeros and four real poles, and a prefilter with one real zero and two real poles. The PID was set to the same control bandwidth as the transformer output. Due to the simplicity of this plant, the transformer solves this problem primarily with the prefilter. This results in a design which has less feedback over the functional bandwidth than the PID. Due to the simplicity of the design, PID would be a reasonable choice absent significant bandwidth restrictions.

4.3. Complex Plants

Consider the plant, P2:

$$\frac{z^3 - 1.8z^2 + 1.263z - 0.1885}{z^4 - 2.4z^3 + 2.175z^2 - 0.7495z + 0.0555} \tag{9}$$

This plant contains closely spaced, lightly damped conjugate pairs of poles and zeros, which pose a design problem for an engineer. Similar to P1, the transformer achieves most of its performance with the prefilter and neglects the maximization of negative feedback. Performance on the step reference is shown in Figure 5. The loop transmission function of the transformer output and the PID are shown in Figure 6.

The compensator consisted of two real zeros, five real poles and one complex conjugate pair of zeros. The prefilter contains three real poles and one complex conjugate pair of zeros. Performance data is compared in Table 2.

Consider the plant, P3:

$$\frac{z^2 - 0.8z + 1.006}{z^4 - 2.8z^3 + 3.544z^2 - 2.343z + -0.6298} \tag{10}$$

Table 2. Performance data for P2.

	Transformer	PID
Rise Time (ms)	17	21
Settling Time (ms)	28	23
Overshoot (%)	15	1
Steady State Error (%)	1	0
Absolute Tracking Error	5.2	10.3

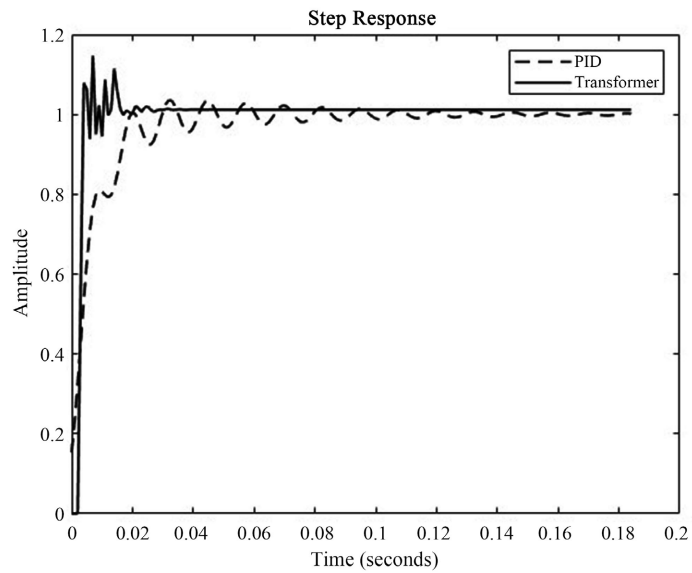


Figure 5. Step response of closed loop system with plant P2.

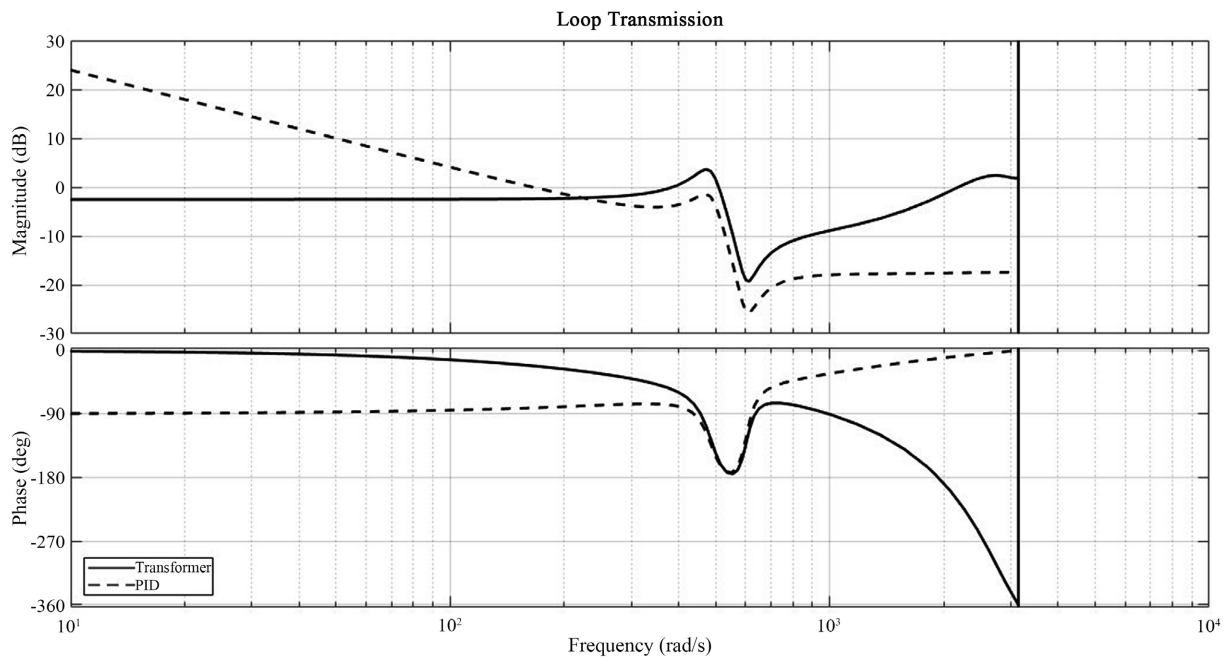


Figure 6. Loop transmission function of transformer and PID with plant P2.

P3 contains close, very lightly damped conjugate pairs of poles and zeros.

These extreme dynamics put significant limitations on control. Selecting a crossover frequency for the PID controller of over 290 radians/s results in an unstable system. The PID for this system was given a crossover frequency of 164 radians/s and a phase margin of 56° . Due to the nature of the plant, the transformer is forced to attempt a more complete design. The transformer achieves a good phase margin of 37° , a quick response and fast settling time. The step response is shown in **Figure 7**. The loop transmission function of closed loop system is shown in **Figure 8**. Performance data is compared in **Table 3**.

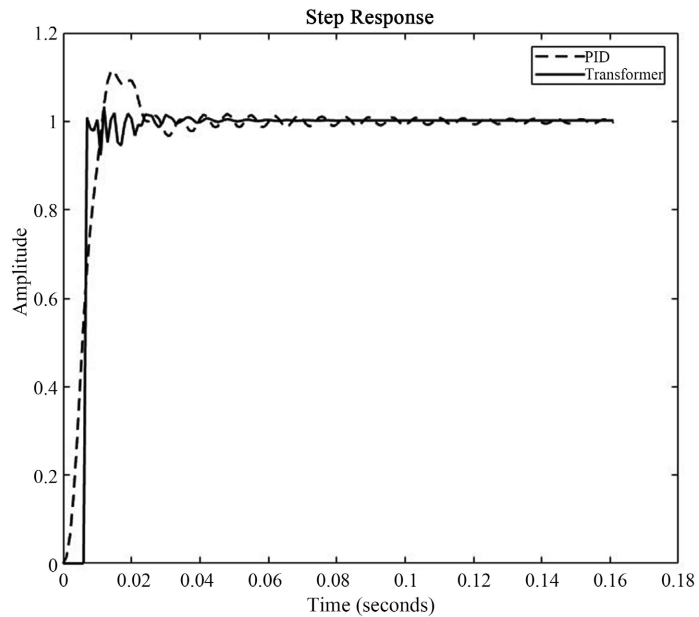


Figure 7. Step response of closed loop system with plant P3.

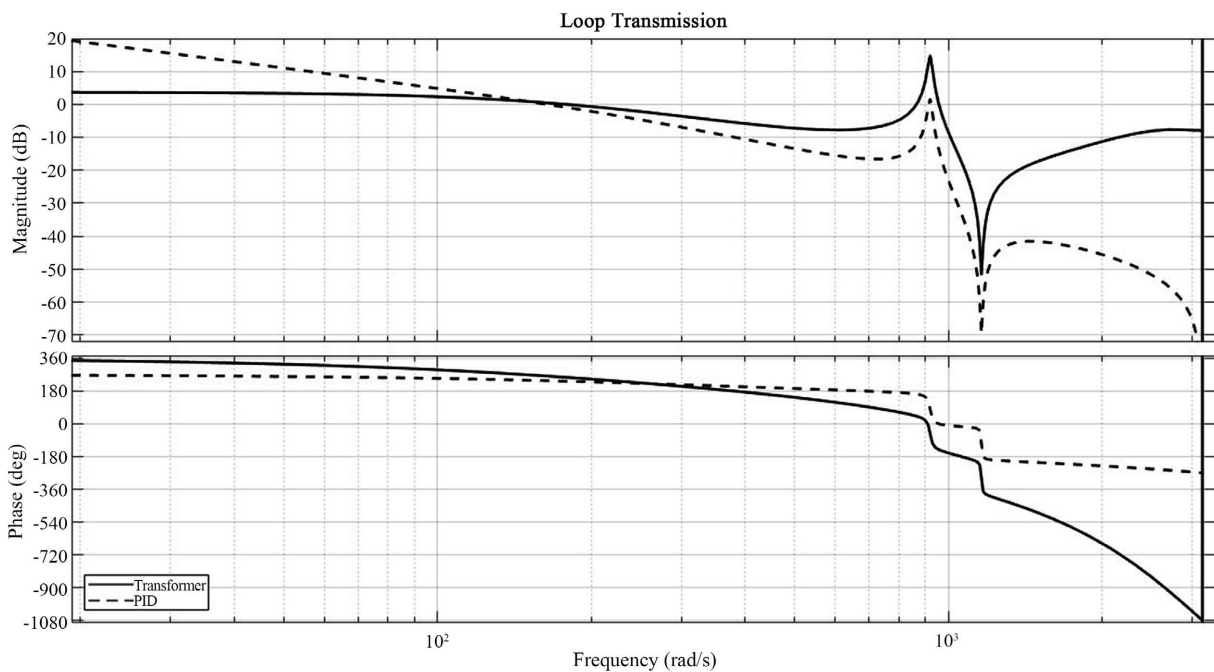


Figure 8. Loop transmission function of transformer and PID with plant P3.

Table 3. Performance data for P3.

	Transformer	PID
Rise Time (ms)	7	10
Settling Time (ms)	7	23
Overshoot (%)	0	14
Steady State Error (%)	1	0
Absolute Tracking Error	7.2	8.4

5. Conclusions

The transformer is an adequate architecture for the generation of compensator and prefilter parameters. The compensation schemes developed by the model do not depend on any prior knowledge and are only refined through experience. However, the training environment can allow for the transformer to make decisions which do not maximize negative feedback, unless the plant is sufficiently challenging.

The maximum amount a designer is required to do is select the type of compensator and prefilter desired and adjust training hyperparameters if necessary. Machine learning continues to develop steadily and has attained greater adoption in the commercial space. As such, ML and control theory may be required to cooperate more closely.

Future Work

Training the network in the presence of disturbances, nonlinearities and plant parameter variation may further leverage the complexity of the transformer. Training in disturbance rich environments may require the transformer to extract more negative feedback. Under plant parameter variation, the transformer could be able to respond to a wide range of plants. If plant parameters were only known in a possible range, training under these conditions could potentially prepare the transformer for reality.

The concurrent training of the prefilter and compensator may not be efficient in the maximization of available feedback. Developing a stable manner of developing the compensator first, then the prefilter may have superior results. This may help mitigate gradient descent's proclivity to search for the simplest solution (gain allocated to the prefilter instead of the compensator).

Acknowledgements

This work was supported and funded by the University of Wyoming Department of Electrical and Computer Engineering.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Ziegler, J.G. and Nichols, N.B. (1942) Optimum Settings for Automatic Controllers. *Journal of Fluids Engineering*, **64**, 759-765. <https://doi.org/10.1115/1.4019264>
- [2] O'Brien, J.F. (2012) Frequency-Domain Control Design for High-Performance Systems. Institution of Engineering and Technology. <https://doi.org/10.1049/PBCE078E>
- [3] Shekhar, A. and Sharma, A. (2018). Review of Model Reference Adaptive Control. 2018 *International Conference on Information, Communication, Engineering and Technology (ICICET)*, Pune, 29-31 August 2018, 1-5. <https://doi.org/10.1109/icicet.2018.8533713>
- [4] Campi, M.C., Lecchini, A. and Savaresi, S.M. (2002) Virtual Reference Feedback Tuning: A Direct Method for the Design of Feedback Controllers. *Automatica*, **38**, 1337-1346. [https://doi.org/10.1016/s0005-1098\(02\)00032-8](https://doi.org/10.1016/s0005-1098(02)00032-8)
- [5] Hjalmarsson, H., Gevers, M., Gunnarsson, S. and Lequin, O. (1998) Iterative Feedback Tuning: Theory and Applications. *IEEE Control Systems Magazine*, **18**, 26-41.
- [6] Rubaai, A., Castro-Sitiriche, M.J., Garuba, M. and Burge, III, L. (2007) Implementation of Artificial Neural Network-Based Tracking Controller for High-Performance Stepper Motor Drives. *IEEE Transactions on Industrial Electronics*, **54**, 218-227. <https://doi.org/10.1109/tie.2006.888785>
- [7] Lurie, B.J. (1986) Feedback Maximization. Artech House.
- [8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Plolosukhin, I. (2017) Attention Is All You Need. arXiv: 1706.03762
- [9] Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Mordatch, I., *et al.* (2021) Decision Transformer: Reinforcement Learning via Sequence Modeling. *Advances in Neural Information Processing Systems*, **34**, 15084-15097.
- [10] Hughes, I. and O'Brien, J.F. (2023). Transformer for Automated Feedback System Design. 2023 *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Seattle, 28-30 June 2023, 1147-1152. <https://doi.org/10.1109/aim46323.2023.10196217>
- [11] Born, J. and Manica, M. (2023) Regression Transformer Enables Concurrent Sequence Regression and Generation for Molecular Language Modelling. *Nature Machine Intelligence*, **5**, 432-444. <https://doi.org/10.1038/s42256-023-00639-z>
- [12] Kingma, D. and Ba, J. (2014) Adam: A Method for Stochastic Optimization. arXiv: 1412.6980
- [13] The MathWorks Inc. (2024) MATLAB Version: 24.1.0.2578822 (R2024a).