2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

# Transformer for Automated Feedback System Design

1st Isaac Hughes
*Department of Electrical Engineering and Computer Science*
*University of Wyoming*
Laramie, United States

2nd John F. O'Brien
*Department of Electrical Engineering and Computer Science*
*University of Wyoming*
Laramie, United States

***Abstract*—Neural-Network (NN) based compensation is a thoroughly investigated topic in automatic control. However, these approaches include the neural network inside the control loop. This paper proposes an alternative approach where the NN is external to the loop, making decisions on the parameters of a linear compensator in cascade with a plant to be controlled with a feedback system. The proposed model utilizes the adept sequence transduction capabilities of the Transformer architecture. This approach is used to design a discrete controller of any order to maximize available performance. This paper applies this method to simple plants without extreme dynamics and a plant with non-minimum phase and very high quality factor modes.**

***Index Terms*—Transformer architecture, disturbance rejection, high-order compensation**

## I. Introduction

The design of a compensator, *C*, for the feedback system in Fig. 1 comes with a panoply of design choices subject to fundamental constraints. Performance requirements are weighed against limitations imposed by the system being controlled. This provides the control designer with an extensive list of targets to meet, dilemmas to solve and scales to balance. This duty is often reduced to the application of proportional-integral (PI) or proportional-integral-derivative (PID) controllers due to these complexities. PID controllers give the designer choices over a gain and two real zeros in the compensator [1]. PID also comes with the inherent benefit of guaranteeing zero steady state error when tracking the step reference. PID controllers have ubiquitous automated tuners and algorithms, making them easy options [1]. This simplicity comes at the cost of performance. High order compensation, as compared to PID, presents the possibility of extracting more performance out of a system. Performance may be quantified as the amount of feedback over the frequencies of interest. Maximizing feedback improves performance, rejects disturbances and reduces sensitivity to plant parameter variation [2]. These desirable features can be difficult to achieve without the proper training.

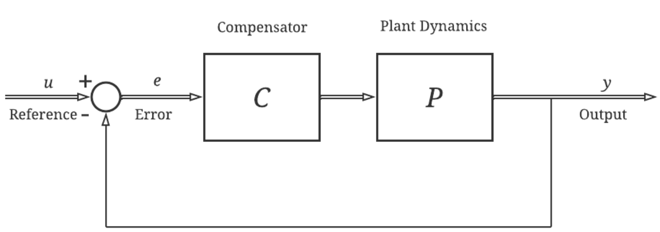


Fig. 1. Block diagram of a control system

In this paper we propose a deep learning approach to compensator design to ease the difficulty of extracting performance out of a system. Digital control systems contain time data, thus, a deep learning model capable of handling sequential information must be chosen. The Transformer architecture is a primary tool for understanding sequential data [3]. Transformers excel at learning long term dependencies by using several Multi-Head Attention stages. Used most prominently in natural language processing, the Transformer can be adapted to the purposes of compensator design by attending to both reference and output signals to produce discrete compensator parameters (pole, zero locations and gain). With the Transformer model arranged in this way, the designer may relegate high order compensator design to the Transformer. The job of the designer is reduced to hyper-parameter choice for the Transformer, compensator order and target performance from the system.

Deep learning has been used frequently in feedback control for the modeling and control of nonlinearities [4]. These approaches share the common approach of putting the NN inside the control loop, in cascade with a standard compensator [5]. The approach proposed in this paper sets the NN outside the loop as in Fig. 2, which makes decisions on the compensator parameters. Advantages of this approach, compared to previous work, are:

1) Ease of determining stability of the closed loop control system.
2) Ease of understanding the current compensation scheme.
3) The closed loop transfer function may be explicitly stated.

### A. Terminology and Background Theory

For the feedback loop in Fig. 1, rational function $T(s) = C(s)P(s) = \frac{u_2(s)}{u_1(s)}$ of the Laplace variable $s$ is the loop transmission (alternatively return ratio) of a feedback loop. For a low pass loop transmission, $\omega_b$, where $|T(j\omega_b)| = 1$ is the *control bandwidth* (alternatively *0 dB crossover frequency*),

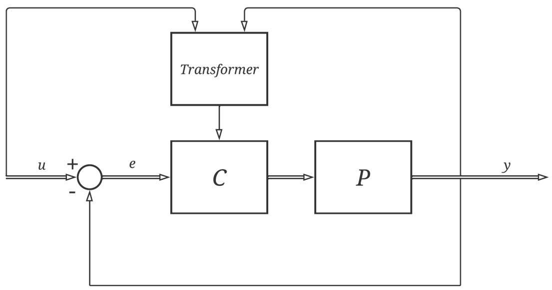


Fig. 2. Block diagram of control system with Transformer

and $\omega_f$ where $|T(j\omega)| \simeq A_0$, $\forall \omega \leq \omega_f$, $A_0$ a constant is the *functional bandwidth*. $|F(s)| = |1 + T(s)|$ is the feedback. $|F(s)| > 1$, $|F(s)| < 1$ and $|T(s)| \ll 1$ define *negative*, *positive* and *negligible* feedback, respectively. $|F(s)| \gg 1$ defines *large* feedback. These definitions indicate the effect of feedback on the logarithmic response of the closed loop system to disturbances. *Nonminimum phase* is the phase lag not found using the Bode phase/gain relationship [2].

### B. The Goals of Feedback

The following enumerated phrases are not intended to be comprehensive. They apply to the challenges of controlling difficult SISO systems aggressively.

1) To stabilize an unstable system.
2) To maximize available negative feedback.
3) To provide sufficient robustness in the presence of sector-type nonlinearities.

Item 1 is obvious. If the plant is unstable, then only the introduction of feedback will provide stability. Available negative feedback is reduced in this case as Bode proved [2]. The complexity of the design to stabilize is directly proportional to the number of unstable open-loop poles and their distance from the imaginary axis.

Item 2 is nuanced. For low-order plants with long time constants, PID usually suffices. For complicated plants with many lightly damped modes within the bandwidth, the compensator design is a delicate process. For low-pass systems, the designer must endeavor to maximize the feedback over the functional bandwidth with the crossover frequency as a constraint [6]. For vibration suppression, an individual mode must be attacked with the maximum available feedback [7].

Item 3 requires a judicious application of feedback. Nyquist-stable compensation applied to systems with limited or imperfect actuation (i.e. saturation, stiction) invariably results in oscillation. Nonlinear dynamic compensation can be applied for aggressive feedback application for these plants [8], however only linear compensation is considered in this work. Large feedback application (linear) for these types of plants requires fractional-order roll-off with carefully designed lead compensation for sufficient relative stability and sharp attenuation of loop gain when feedback becomes negligible. After maximizing feedback, a prefilter is designed to minimize overshoot and mitigate steady state error. [2].

To address adequately these three items requires an engineer highly skilled in feedback theory and practice. *The authors present a method that automates this process.*

## II. Model Elements

### A. The Difference Equation

Given a discrete time transfer function,

$$H(z^{-1}) = \frac{b_0 + b_1 z^{-1} + ... + b_n z^{-n}}{1 + a_1 z^{-1} + ... + a_m z^{-m}} \quad (1)$$

the output at sample $k$ is given as:

$$\begin{aligned} y[k] &= b_0 u[k] + b_1 u[k-1] + ... + b_n u[k-n] \\ &-a_1 y[k-1] - a_2 y[k-2] - ... - a_m y[k-m] \end{aligned} \quad (2)$$

The input to the system at time $k$ is $u[k]$.
For a closed loop system of relative degree $r$, the closed loop transfer function is:

$$H(z^{-1}) = \frac{C(z)P(z)}{1 + C(z)P(z)} \quad (3)$$

Split into numerator and denominator polynomials,

$$C = \frac{C_B}{C_A},$$

$$P = \frac{P_B}{P_A}$$

to express

$$H(z^{-1}) = \frac{C_B P_B}{C_B P_B + C_A P_A} \quad (4)$$

$$= \frac{b_0 z^{-r} + b_1 z^{-r-1} + ... + b_n z^{-m}}{b_0 z^{-r} + b_1 z^{-r-1} + ... + b_n z^{-m} + 1 + a_1 z^{-1} + ... + a_m z^{-m}}. \quad (5)$$

Let the tracking error be $e[k] = u[k] - y[k]$, and the output may be expressed as:

$$\begin{aligned} y[k] &= b_0 e[k-r] + b_1 e[k-r-1] + ... + b_n e[k-m] \\ &-a_1 y[k-1] - a_2 y[k-2] - ... - a_m y[k-m] \end{aligned} \quad (6)$$

### B. Model Architecture

The Transformer architecture is an encoder-decoder structure similar to most sequence transduction models. The encoder is composed of many multi-head self-attention mechanisms that build parallel representational subspaces that attend to patterns in the data. Self-attention refers to the process of weighting the importance of all elements in a sequence to a particular element of the same sequence. This process is carried out by the formula:

$$\text{Attention(Q,K,V)} = \text{softmax}(\frac{QK^T}{\sqrt{dk}})V \quad (7)$$

Details of the Transformer's architecture are described by Vaswani et al. [3].

The input sequence given to the Transformer encoder are the input samples, $u[k], u[k-1], ..., u[k-m+1]$. The output sequence provided to the decoder are the output samples $y[k-1], y[k-2], ..., y[k-m]$. Thus the approach in this paper is auto-regressive as the output of the closed loop model becomes the new leading output fed to the decoder. The encoder self-attention mechanism has a full query matrix $Q$ for input elements where the decoder has a query vector $q$ to provide an output for a given time step. Raw inputs and outputs are projected to the embedding dimension by a linear layer that is trained along with the model.

The output of the Transformer are the parameters of the compensator. These parameters come in seven types:

1) Compensator gain $C_K$
2) Compensator real zero locations $C_z$
3) The real portion of a complex conjugate pair of zeros $C_{z_r}$
4) The imaginary portion of a complex conjugate pair of zeros $C_{z_i}$
5) Compensator real pole locations $C_p$
6) The real portion of a complex conjugate pair of poles $C_{p_r}$
7) The imaginary portion of a complex conjugate pair of poles $C_{p_i}$

The number of zeros and poles, both real and imaginary, is chosen by the user. The produced compensator is then used to generate $y[k]$ through the closed loop system.

The embedding dimension for the Transformer is 32, subspace dimension is 64 and four heads were used. Both encoder and decoder fully connected networks use two linear layers with 64 neurons each.

### C. Training

Training consists of providing a reference sequence to the encoder, which is masked to maintain causality. Each reference time step generates a vector of compensator parameters which is used to generate the output at that time step. The loss function for the model is the mean-squared tracking error:

$$\frac{1}{2}(y[k]-u[k])^2.$$

This loss function was chosen as the overall problem is similar to a regression problem. The network is trained at every time step. Once a reference sequence is finished, a new reference is generated and trained on. The learning rate is varied by the schedule:

- Linear increase from 10% to 10,000% learning rate at one quarter of the training epochs.
- Linear decrease from 10,000% learning rate to 10% learning rate to end of training.

Base learning rate is $10^{-5}$.

### D. Closed Loop System Backpropagation

The output of the Transformer is separated from the loss function by the closed loop system, thus the gradient of the loss function with respect to the compensator parameters must be calculated.

With $P_K$ representing the plant gain, the output terms associated with the compensator zeros are

$$C_K P_K C_B(z^{-1}) P_B(z^{-1}), \tag{8}$$

which in factored form for $n$ zeros is

$$\begin{aligned} C_K P_K (z^{-1}+C_{B_1})(z^{-1}+C_{B_2})(z^{-1}+C_{B_3})... \\ (z^{-1}+C_{B_n}) P_B(z^{-1}) \end{aligned} \tag{9}$$

Equation (8) is expanded and the inverse Z-transform applied to get

$$C_K P_K (e[k-r]+b_1 e[k-r-1]+...+b_n e[k-m]). \tag{10}$$

The derivative of the output $y[k]$ with respect to the gain $C_K$ is

$$\frac{\partial y[k]}{\partial C_K} = P_K(e[k-r]+b_1 e[k-r-1]+...+b_n e[k-m]). \tag{11}$$

The derivative of the output $y[k]$ with respect to the $j^{th}$ real zero is

$$\begin{aligned} \frac{\partial y[k]}{\partial C_{B_j}} = C_K P_K (e[k-r-1]+ \\ b'_1 e[k-r-2]+...+b'_n e[k-m]), \end{aligned} \tag{12}$$

where $b'_1, b'_2, ..., b'_n$ are the polynomial coefficients generated from expanding

$$\frac{C_K P_K C_B(z^{-1}) P_B(z^{-1})}{(z^{-1}+C_{B_j})}. \tag{13}$$

Finally $C_j$ may be part of a complex conjugate pair of zeros. $(z+C_{B_j})(z+C^*_{B_j})$. As $C_{B_j}$ and $C^*_{B_j}$ share two parameters, two derivatives must be calculated for the entire pair. The derivative with respect to the real portion of $C_{B_j}$ is

$$\begin{aligned} \frac{\partial y[k]}{\partial \mathrm{Re}\{C_{B_j}\}} = 2C_K P_K (e[k-r-1]+ \\ b''_1 e[k-r-2]+...+b''_n e[k-m]), \end{aligned} \tag{14}$$

where $b''_1, b''_2, ..., b''_n$ are the polynomial coefficients generated from expanding

$$C_B(z^{-1}) P_B(z^{-1}) \frac{(z^{-1}+\mathrm{Re}\{C_{B_j}\})}{(z^{-1}+C_{B_j})(z^{-1}+C^*_{B_j})}. \tag{15}$$

The derivative with respect to the imaginary portion of $C_j$ is

$$\begin{aligned} \frac{\partial y[k]}{\partial \mathrm{Im}\{C_{B_j}\}} = 2C_K P_K \mathrm{Im}\{C_{B_j}\}(e[k-r-2]+ \\ b'''_1 e[k-r-3]+...+b'''_n e[k-m]), \end{aligned} \tag{16}$$

where $b'''_1, b'''_2, ..., b'''_n$ are the polynomial coefficients generated from expanding

$$\frac{C_B(z^{-1}) P_B(z^{-1})}{(z^{-1}+C_{B_j})(z^{-1}+C^*_{B_j})}. \tag{17}$$

A similar process is carried out for the $m$ compensator poles. The derivative of the output with respect to the $j^{th}$ real pole is

$$\frac{\partial y[k]}{\partial C_{A_j}} = -y[k-2] - a'_1 y[k-3] - ... - a'_m y[k-m]. \tag{18}$$

If $C_{A_j}$ is part of a complex conjugate pair of zeros, the derivative with respect to the real part of the pair is

$$\frac{\partial y[k]}{\partial \text{Re}\{C_{A_j}\}} = -2(y[k-2] - a_1''y[k-3] - ... - a_m''y[k-m]). \tag{19}$$

The derivative with respect to the imaginary part is

$$\frac{\partial y[k]}{\partial \text{Im}\{C_{A_j}\}} = 2\text{Im}\{C_{A_j}\}(-y[k-3] - a_1'''y[k-4] - ... - a_m'''y[k-m]). \tag{20}$$

The coefficients $a_1', a_2', ..., a_m'$ are generated from expanding the polynomial

$$\frac{C_A(z^{-1})P_A(z^{-1})}{(z^{-1} + C_{A_j})}, \tag{21}$$

$a_1'', a_2'', ..., a_m''$ from expanding

$$C_A(z^{-1})P_A(z^{-1})\frac{(z^{-1} + \text{Re}\{C_{A_j}\})}{(z^{-1} + C_{A_j})(z^{-1} + C_{A_j}^*)}, \tag{22}$$

and $a_1''', a_2''', ..., a_m'''$ from expanding

$$\frac{C_A(z^{-1})P_A(z^{-1}))}{(z^{-1} + C_{A_j})(z^{-1} + C_{A_j}^*)}. \tag{23}$$

## III. Performance on known plants

### A. PID Comparison and Tuning

To compare the performance of the Transformer-designed compensator against a PID controller, the step reference will be used. Time domain performance characteristics to be examined are the rise time, settling time, overshoot, steady state error and absolute tracking error.

- Rise time: the time to reach $90\%$ of the systems final value.
- Settling time: the time after which the system is bounded within $\pm5\%$ 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.

PID tuning was done in MATLAB's PID Tuner application [9]. MATLAB allows the user to select for bandwidth and phase margin. The following plants were tuned for maximum allowable bandwidth and minimum phase margin.

### B. Prefilter

Transformer-designed compensators begins with random parameters. These initial parameters results in high error and no tracking. In training the Transformer to minimize tracking error, major sources of error must be solved first. The Transformer will first achieve tracking at the expense of high overshoot and long settling times. Continued training past this point will remove overshoot at the cost of feedback over the functional bandwidth. From the control designer's perspective, this is not a good trade as maximization of feedback over the functional bandwidth is the goal of the feedback compensator [6]. To remove overshoot while retaining performance, a prefilter has been applied to Transformer outputs. The prefilter is the double notch described by the transfer function,

$$R(s) = (\frac{s^2 + \omega_b s + (0.9\omega_b)^2}{s^2 + 2\omega_b s + (0.9\omega_b)^2})^2, \tag{24}$$

which is converted to a discrete-time filter using MATLAB's zero-order hold function. This prefilter was chosen to remove input power from the limited band of frequencies which are causing overshoot while retaining performance over the functional bandwidth.

### C. Simple Plants

*1) $P_1$:* Consider the plant

$$P_1 = \frac{-0.78z^2 - 0.15z - 0.01}{z^3 - 0.43z^2 - 0.20z - 0.01}, \tag{25}$$

sampled at 1kHz. The Transformer outputs new compensator parameters for a closed loop system at every time step, thus the median compensator from a given epoch is considered. The median compensator of the final epoch is

$$C_1 = \frac{-1.02z^3 + 0.95z^2 + 0.14z - 0.34}{z^5 - 0.59z^4 - 0.44z^3 + 0.004z^2 + 0.02z + 10^{-3}}. \tag{26}$$

The resulting loop transmission function in Fig. 3 displays 63dB of feedback over the functional bandwidth and a phase margin of $41°$. These characteristics result in a significantly more aggressive step response as compared to a PID controller as shown in Fig. 4. $C_1$ is $5^{th}$ order compared to the order-one PID compensator. The extra parameters are spent primarily on shaping the response around crossover. The time domain performance characteristics of the Transformer-generated compensator and a PID controller are compared in table I. The aggressive approach taken by the Transformer results in significant overshoot. Application of a prefilter is required to remove this quality while retaining performance. It is noted that the Transformer being external to the loop allows the usage of linear control theory to determine performance and stability. This is a critical capability.

| Performance Measure | Transformer | PID |
|---|---|---|
| Rise Time (ms) | **15** | 226 |
| Settling Time (ms) | **15** | 294 |
| Overshoot (%) | 2 | **0** |
| Steady State Error (%) | **0** | **0** |
| Absolute Tracking Error | **7.3** | 99.87 |

TABLE I
Closed loop performance with $P_1$

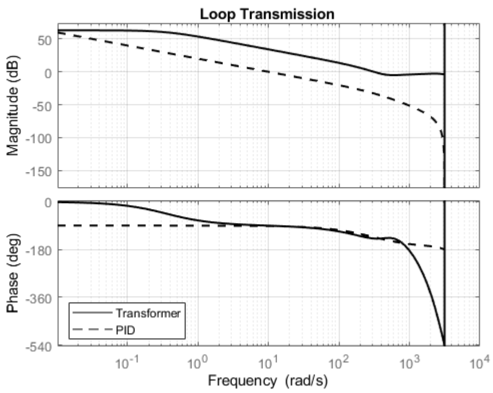


Fig. 3. Discrete loop transmission Bode plot $C_1 P_1$.

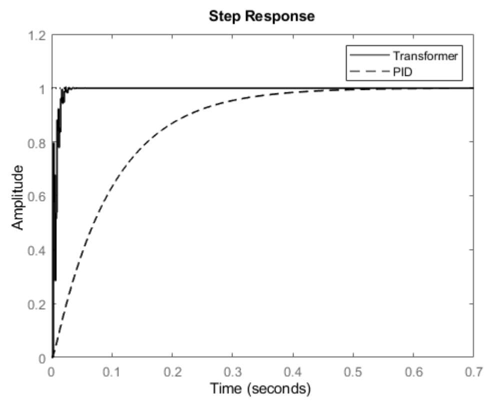


Fig. 4. Closed loop step response using $C_1$.

*2) $P_2$:* For the plant

$$P_2 = \frac{0.5z^5 - 0.22z^4 - 0.47z^3 + 0.1z^2 + 0.17z + 0.03}{z^6 - 2.17z^5 + 1.64z^4 - 0.64z^3 + 0.26z^2 - 0.11z + 0.02}, \quad (27)$$

sampled at 100Hz. The median compensator at the end of training is

$$C_2 = \frac{2.1z^6 + 1.2z^5 + 0.1z^4 + 0.5z^3 + 0.1z^2 - 0.02z + 0.001}{z^7 + 2.1z^6 + 0.8z^5 - 1.1z^4 - z^3 - 0.3z^2 - 0.02z - 0.002}, \quad (28)$$

producing a the loop transmission function in Fig. 5. The step response performance is shown in Fig. 6 and compared in table II. The PID outperforms the Transformer output in feedback at low frequency because of PID integral action. However, the transformer output achieves better rise time and settling time through a more carefully shaped crossover region and a 40° phase margin.

*D. Complex Plants*

Consider a complicated plant with non-minimum phase characteristics and very high quality factor modes:

$$P_3 = \frac{2.16(z + 1.19)(z + 0.02)(z^2 - 1.76z + 1.006)}{z(z^2 - 1.2z + 0.995)(z^2 - 0.91z + 0.67)}, \quad (29)$$

sampled at 200Hz. The Transformer was given control over an $11^{th}$ order compensator, $C_3$, with 8 real zeros, one complex conjugate pair of zeros, 5 real poles and 3 pairs of complex conjugate poles. A potentially problematic element of the Transformer output is the gain-stabilized mode at 200 rad/s seen in Fig. 7. During training, the Transformer attempts to use this element to speed up response to high frequency references, but fails to phase stabilize the section. The time domain performance characteristics and step response are compared in table III and Fig. 8 respectively.

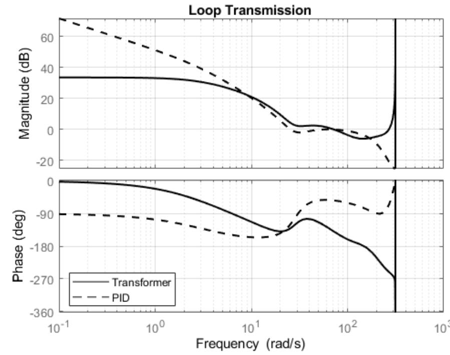


Fig. 5. Discrete loop transmission Bode plot $C_2 P_2$.

| Performance Measure | Transformer | PID |
|---|---|---|
| Rise Time (ms) | **20** | 50 |
| Settling Time (ms) | **170** | 330 |
| Overshoot (%) | **15** | 23 |
| Steady State Error (%) | **0** | **0** |
| Absolute Tracking Error | **4.3** | 5.79 |

TABLE II
CLOSED LOOP PERFORMANCE WITH $P_2$

| Performance Measure | Transformer | PID |
|---|---|---|
| Rise Time (ms) | **150** | 1380 |
| Settling Time (ms) | **165** | 1830 |
| Overshoot (%) | 1 | **0** |
| Steady State Error (%) | **0** | **0** |
| Absolute Tracking Error | **18.2** | 85.6 |

TABLE III
CLOSED LOOP PERFORMANCE WITH $P_3$

## IV. CONCLUSION

Automatic control and frequency domain designs still underpin much of our mechanical world. Low order techniques like PID function and are easy to implement, but do not extract maximum available performance. A deep learning method has been presented which successfully develops high order compensators which outperform conventional tools and which requires limited knowledge of control theory. This method provides superior performance compared to standard control designs for both simple and complex plants as evidenced by the presented examples.

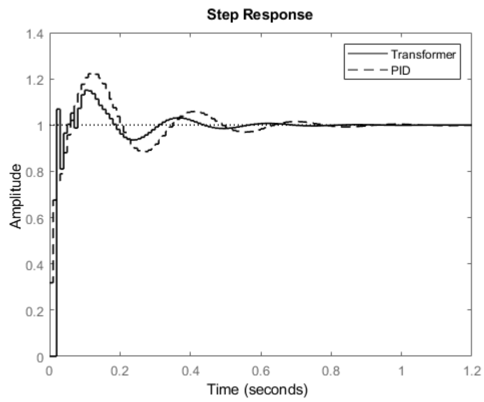


Fig. 6. Closed loop step response using $C_2$.

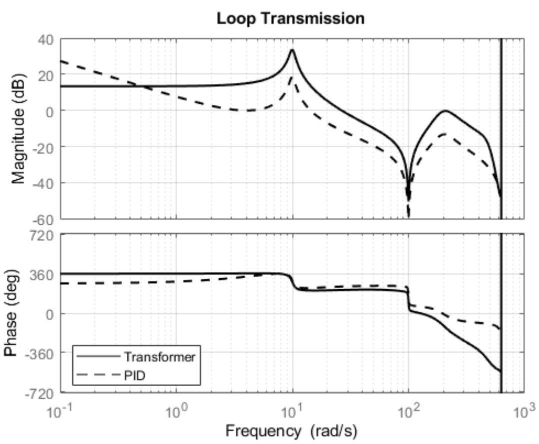


Fig. 7. Discrete loop transmission function $C_3P_3$.

## V. Future Work

### A. Prefilter

The process of error minimization taken by the Transformer can begin to conflict with feedback compensator design goals. The Transformer achieves tracking of the training references and then continues to minimize error beyond this point. Minimization of tracking error can come at the expense of feedback. In this case, a prefilter is used to correct transient response while retaining performance. Extension of the current method to include a learn-able prefilter would complete the automatic design process.

### B. Non-linearity

All control problems come with some degree of non-linearity. Previous work in this field has used deep learning to effectively model and compensate for this issue. Either merging prior work with the proposed approach or expanding the Transformer's capability to also model these non-linearities would greatly improve the potential for deep learning models to be used in control system design (section I, subsection B, item 3).

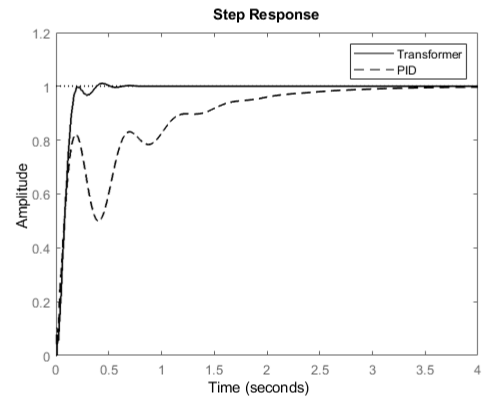


Fig. 8. Closed loop performance using $C_3$.

## VI. Acknowledgments

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

## References

[1] Ziegler, J.G., and Nichols, N.B., "Optimum settings for automatic controllers." trans. ASME 64.11 (1942).
[2] O'Brien, J.F. Frequency-Domain Control Design for High-Performance Systems, Institution of Engineering and Technology. Stevenage, UK, 2012.
[3] Vaswani, A., Shazeer, N., Parmar N., Uszkoreit, J., Jones L., Gomez, A.N., Kaiser, L., Plolosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems 2017*
[4] Fernandes, J. M. D. M., Tanaka, M. C., and Bessa, W. M. (2022). Sliding mode control with a neural network compensation scheme for electro-hydraulic systems.
[5] Tang, J., Huang, Z., Zhu, Y. et al. Neural network compensation control of magnetic levitation ball position based on fuzzy inference. Sci Rep 12, 1795 (2022). https://doi.org/10.1038/s41598-022-05900-w
[6] Lurie, B.J., Feedback Maximization, Artech House, US, 1986
[7] Chu, C-C and O'Brien, J.F., "A Layered Approach to Structural Control System Design for the JPL Phase B Testbed," *1993 AIAA Guidance and Control Conference, Monterey*, CA, August 1993.
[8] O'Brien Y.V. and O'Brien, J.F., "Parallel Loop Recovery with Quiescent Compensation for High Performance Feedback Control of Systems with Imperfect Actuators," *Control Theory and Technology*, August 2019
[9] The MathWorks, Inc. (2023). MATLAB version: 9.11.0 (R2021b).