Vol. 4(2), pp. 1-22, December 2025
eISSN: 2976-4009

ASEAN Journal of Process Control

Research Article

MIMO Control of an Interacting Two-Tank System: A Comparative Study of PID and MPC via Chirp-Based Frequency-Domain System Identification

Ooi Yun Sheng[1], Musab Abdul Razak[1*]

[1] Sustainable Process Engineering Research Centre (SPERC), Department of Chemical and Environmental Engineering, Faculty of Engineering, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia.

[*]Corresponding Author: musab@upm.edu.my

Abstract: Effective control of interacting multiple-input-multiple-output (MIMO) systems is critical in many industrial applications since stable flow and level regulation affect process safety and efficiency. A commonly encountered challenge in such systems is the coupling between inputs and outputs, which complicates controller design and tuning. This study addresses the control of an interacting two-tank system by comparing the performance of conventional Proportional-Integral-Derivative (PID) controllers with Model Predictive Control (MPC). The main objective of the study is to evaluate the suitability and effectiveness of MPC in managing coupled dynamics, actuator constraints, and setpoint tracking, in contrast to classical PID controllers. The system was identified using frequency-domain techniques, where chirp signals were injected into each input, and the resulting output responses were processed using the Discrete Fourier Transform (DFT) to estimate transfer function models. PID controllers were designed using the root locus method, while MPC was implemented based on a discrete-time state-space model. Simulation results showed that while PID controllers performed acceptably in single-loop configurations, they suffered from excessive overshoot (up to 187%) and a prolonged settling time of 53 seconds when applied to the full MIMO system, despite exhibiting zero steady-state error. More importantly, the actuator produced an impractically high flowrate of 2.7×10^4 L/min. In contrast, MPC achieved faster settling times (under 23 seconds) and eliminated overshoot entirely (0% steady-state error). Also, it limited its actuator output to a feasible flowrate of 25 L/min. These findings suggest that MPC provides a more robust and practical solution for controlling interacting MIMO systems.

Keywords: Multiple-Input-Multiple-Output (MIMO) Systems, Chirp Signals, PID Controllers, Root Locus Method, Model Predictive Control (MPC)

1. Introduction

The dynamics of an interacting two-tank system are representative of many real-world industrial processes where multiple inputs and outputs are interlinked, such as in the nuclear power industry, the paper industry, the coating and water treatment industries [1].

In an interacting two-tank system, liquid is transferred from one tank to another. The dynamics must be precisely controlled to ensure safe and efficient operation. Parameters such as temperature, pressure and flow rate must be maintained within strict limits [2]. Among these, liquid level control in interacting two-tank systems is particularly crucial [3]. Such systems are characterized by large time delays, strong coupling, great inertia, nonlinearity and multivariable constraints [4]. This makes the control and analysis of the system significantly more complex than linear systems. Besides, abnormal

liquid levels in chemical processes can lead to hazardous accidents, which underscores the importance of effective control [5].

Before a suitable control strategy can be applied, system identification is necessary to model the process dynamics. Among the various identification methods, the step-response test is widely used in industry due to its ease of implementation and low cost [6]. In this method, the system is first brought to a steady state before a step change is introduced, and the output is recorded. However, the step-response approach is often time consuming and sensitive to disturbances and noise, particularly in interacting or nonlinear systems [7]. To address these limitations, frequency-domain identification methods have gained attention for their robustness and modelling accuracy in complex processes [8]. These methods estimate a system model based on its frequency response and offer the advantage of focusing on specific frequency ranges, thereby reducing computational burden and enabling the estimation of a large number of nonlinear parameters. Periodic inputs are particularly useful because, once the transient response has subsided, any variation in the output across periods can be attributed to measurement noise [9]. This makes it easier to estimate the noise level and assess the quality of the collected data.

Among the various excitation signals used in frequency-domain identification, the chirp signal (also known as a swept-sine) is particularly useful due to its flexibility and efficiency. A chirp signal varies its frequency continuously over time, either linearly or exponentially, to allow the user to target specific frequency ranges relevant to the system under study. One of the key advantages of chirp excitation is its ability to concentrate energy across a wide range of frequencies within a single experiment and reduces the total test duration compared to using multiple single-tone inputs [10]. Furthermore, chirp signals offer joint time–frequency localization, which enables the data to be filtered and analyzed around any chosen frequency after collection [11]. This property improves the effective signal-to-noise ratio, making chirps more robust to measurement disturbances than pseudo-random signals and other broadband excitations.

Given the multivariable and nonlinear nature of chemical processes, selecting an appropriate control strategy remains a significant challenge. Model Predictive Control (MPC) has emerged as a powerful technique for handling such complexities. MPC effectively addresses constrained multivariable control problems by predicting future system behavior and optimizing control moves accordingly [12]. Unlike traditional Proportional-Integral-Derivative (PID) controllers, MPC can anticipate disturbances and handle interactions between variables more effectively [13]. This makes it particularly efficient for processes involving interacting fluid levels, constraints on actuator performance, and dynamic coupling.

This study investigates the application of both PID and MPC control strategies to an experimental interacting two-tank system, with the aim of improving control accuracy and dynamic response. The insights obtained from this work may support the optimization of control strategies in processes that involve such systems and contribute to improvements in safety, efficiency, and sustainability in chemical process industries.

2. Materials and Methods

2.1. Prototype Design and Apparatus Selection

A prototype of the interacting two-tank system was constructed. The process flow diagram of the system is as follows:

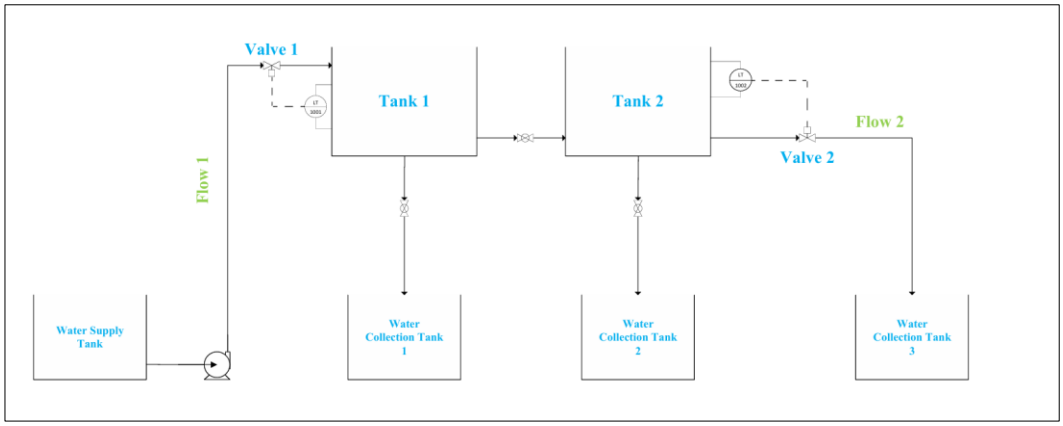

Figure 1: Process flow diagram of the interacting two-tank system.

The process was continuous and involved two identical tanks of 3 gallons (11.4 L). The pipes and fittings were steel and their diameters were ½ inch. The pump capacity was 26 L/min. The level sensor used was Milone eTape PN-12110215TC-12. The control valves (Valve 1 and Valve 2) used were inherited from earlier work conducted in the laboratory. They were functionally tested and deemed suitable for use without further modification. Each control valve consists of a ball valve, a high-torque 25 kg RC servo, and a 3D-printed valve holder. By connecting the servo of the valves to Arduino Mega 2560, they were able to fully open, fully close, or partially open to vary Flow 1 and Flow 2. Figure 2 shows the complete setup of the prototype:

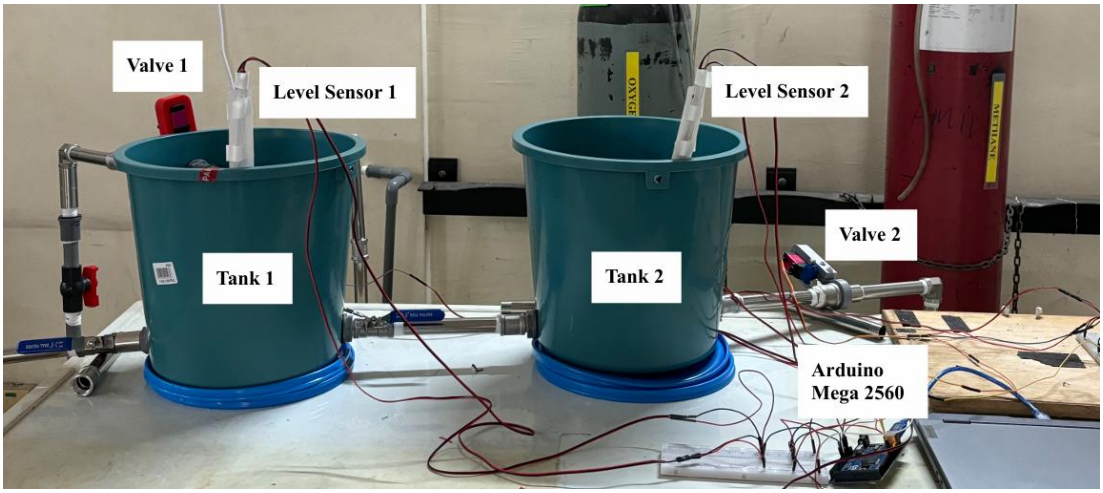

Figure 2: Complete setup of the interacting two-tank system.

As shown in Figure 2, the setup corresponded to the process flow diagram provided in Figure 1. The two tanks were placed on a table. The control valves and level sensors were connected to Arduino Mega 2560.

2.2. Data Collection

Chirp excitation was applied sequentially to each inlet flow valve over a 300 s interval, while the opposing valve remained fixed at full opening. Tank levels were recorded at 10 Hz using analogue water-level sensors connected to Arduino Mega 2560. To improve measurement reliability, each chirp test was repeated three times and the corresponding water level readings were averaged before plotting.

Prior to signal injection, the system was allowed to reach steady-state conditions. Then, a linear chirp signal which swept frequencies from 0.01 Hz to 0.1 Hz over a duration of 300 seconds was employed. As the system had two inputs, namely Valve 1 (Flow 1) and Valve 2 (Flow 2), chirp signals were introduced individually, while the opposite valve remained fixed at its full opening.

Taking into account the mechanical constraints of both servos and valves (for Valve 1, servo angle of 15° corresponded to 0% valve opening, 105° to 100% valve opening; for Valve 2, servo angle of 160° corresponded to 0% valve opening and 70° to 100% valve opening) the angle commands for Valve 1, $\theta_1(t)$ and Valve 2, $\theta_2(t)$ were mathematically expressed as follows:

$$\theta_1(t) = 65° + 40° \cos(\Omega_1(t)t) \quad (1)$$

$$\theta_2(t) = 110° + 40° \cos(\Omega_2(t)t) \quad (2)$$

where

$$\Omega_1(t) = 2\pi f_1(t) \quad (3)$$

$$\Omega_2(t) = 2\pi f_2(t) \quad (4)$$

$$f_1(t) = f_{0,1}(t) + \Delta f_1(t) \quad (5)$$

$$f_2(t) = f_{0,2}(t) + \Delta f_2(t) \quad (6)$$

$$\Delta f_1(t) = \frac{f_{1,1} - f_{0,1}}{T} \quad (7)$$

$$\Delta f_2(t) = \frac{f_{1,2} - f_{0,2}}{T} \quad (8)$$

$f_{o,1}$ and $f_{o,2}$ were the start frequencies of Valve 1 and Valve 2 respectively and were equal to 0.01 Hz. $f_{1,1}$ and $f_{1,2}$ were the end frequencies of Valve 1 and Valve 2 respectively and were equal to 0.1 Hz.

The selected frequency range of 0.01 Hz to 0.1 Hz was deliberately chosen based on the dynamics of the two-tank system, which typically exhibits relatively slow response [14]. The lower limit (0.01 Hz) ensured the capture of slow dynamic responses, while the upper limit (0.1 Hz) avoids mechanical and hydraulic limitations by avoiding unwanted actuator saturation and sensor lag effects.

Considering that the maximum frequency of the chirp signal used was 0.1 Hz (equivalent to approximately 0.628 rad/s), the Nyquist–Shannon criterion required a minimum sampling frequency of at least 0.2 Hz (approximately 1.256 rad/s). To ensure accurate data capture and to provide adequate resistance against noise and higher-order dynamics, a significantly higher sampling rate of 10 Hz was employed in the experiment. This rate corresponded to a sampling interval of 0.1 seconds.

Following these considerations, chirp signals were sequentially injected into the system, and the corresponding data from the sensors were recorded. To ensure accuracy, each sampling process were repeated three times, and the average measurement was used.

2.3. System Identification

System identification was performed by slicing the chirp signal inputs and their corresponding output signals (Tank 1 and Tank 2 levels) into frequency bands at 0.01 Hz intervals from 0.01 Hz to 0.1 Hz. Each slice was zero-padded until its length reached the nearest power of two, and then windowed using a Hamming window generated via the MATLAB command *w = hamming(L)*, where *L* is the signal length.

The Discrete Fourier Transform (DFT) of each windowed signal was computed using the *fft(X)* command. For each input frequency, the DFT amplitude and phase were extracted for both input and output signals. The magnitude response at each frequency was determined by dividing the output amplitude by the input amplitude, while the phase response was obtained by subtracting the input phase from the output phase.

This process was repeated for all frequencies to generate Bode diagrams for the following input-output pairs:

- Flow 1 → Tank 1 level
- Flow 1 → Tank 2 level
- Flow 2 → Tank 1 level
- Flow 2 → Tank 2 level

Once the Bode plots were available, transfer functions were derived by following the guidelines below [15]:

1) Look for any peaks or dips in the magnitude response plot. A peak indicates the presence of an underdamped second-order pole, while a dip may suggest an underdamped zero. The frequency at which a peak occurs can be taken as the natural frequency, ω_n of a second-order system, represented as:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (9)$$

Here, ζ is the damping ratio, which can be estimated from the peak magnitude, M_p using the equation:

$$M_p = \frac{1}{2\zeta\sqrt{1-\zeta^2}} \quad (10)$$

Solving Eq. 10 for ζ results in a quartic equation with four possible roots However, a resonance peak only occurs when $\zeta \leq 0.707$, so only the root(s) satisfying this condition should be accepted [15].

2) A break frequency *a* is the frequency at which the slope of the magnitude plot changes, typically by ±20 dB/decade. If the slope is +20 dB/decade, then the transfer function component is:

$$G(s) = \frac{(s+a)}{a} \tag{11}$$

If the slope is -20 dB/decade, the transfer function component is one with a pole:

$$G(s) = \frac{a}{(s+a)} \tag{12}$$

3) If a horizontal segment with nonzero magnitude is observed (slope = 0 dB/decade), it represents a gain in the system. The gain K can be related to the magnitude M in decibels using:

$$M = 20\ log_{10}\ K \tag{13}$$

4) This procedure is repeated until the magnitude plot flattens to 0 dB, which indicates the end of dynamic frequency contributions.
5) The final estimated transfer function is the product of all the individual components (poles, zeros, and gains) identified from the Bode plot.

Note that GP1(s) was assigned as the transfer function of input Flow 1 and output Tank 1 water level, GP2(s) as that of input Flow 1 and output Tank 2 water level, GP3(s) as that of input Flow 2 and output Tank 1 water level, and GP4(s) as that of input Flow 2 and output Tank 2 water level.

2.4. RGA Computation

The Relative Gain Array (RGA) was computed to determine optimal input-output pairings for decentralized control. In this experiment, the interacting two-tank system comprises two inputs (Flow 1 and Flow 2) and two outputs (Tank 1 and Tank 2 water levels). Accordingly, the RGA at any given frequency is expressed as a 2×2 matrix, where each element reflects the relative coupling strength of a specific input-output pair.

To assess how coupling characteristics vary across frequencies, the dynamic RGA was computed. This involved evaluating the system's frequency response over a defined frequency range (0–100 rad/s). At each frequency ω_k, the complex frequency response $G(j\omega_k)$ of the 2×2 MIMO system was determined, forming the matrix:

$$G(j\omega_k) = \begin{bmatrix} GP1(j\omega_k) & GP2(j\omega_k) \\ GP3(j\omega_k) & GP4(j\omega_k) \end{bmatrix} \tag{14}$$

Next, the RGA matrix $\Lambda(j\omega_k)$ at each frequency was computed using the definition:

$$\Lambda(j\omega_k) = G(j\omega_k) \circ G(j\omega_k)^{-T} \tag{15}$$

where "∘" denotes the element-wise (Hadamard) multiplication, and "$^{-T}$" denotes the inverse transpose of the matrix.

To ensure numerical accuracy and stability, an iterative approach was employed to compute the RGA matrices until convergence. The convergence criterion was based on how closely the RGA at each frequency approached the identity matrix.

Iteration continued until the error at all frequencies was below a predefined tolerance, 10^{-6}, or until a maximum allowable number of iterations was reached (50 iterations). Upon convergence, the magnitude of each RGA element $\lambda_{ij}(j\omega)$ was extracted and plotted on semilogarithmic axes to illustrate clearly how interactions varied across frequency:

1) λ_{11} : magnitude of the coupling from input 1 to output 1.
2) λ_{12} : magnitude of the coupling from input 2 to output 1.
3) λ_{21} : magnitude of the coupling from input 1 to output 2.
4) λ_{22} : magnitude of the coupling from input 2 to output 2.

The resulting plots allowed direct interpretation of optimal pairings [16]:

1) RGA values close to 1 on diagonal terms (λ_{11}, λ_{22}) and close to zero on off-diagonal terms (λ_{12}, λ_{21}) indicate minimal coupling between loops, hence supporting the diagonal pairing strategy (input 1 to output 1 and input 2 to output 2).

2) Conversely, large values or values significantly different from unity in diagonal elements suggest significant interaction or coupling, hence making such pairings undesirable.

2.5. PID Design

Next, PID controllers were designed for the two optimal pair obtained through RGA analysis. For example, if according to RGA analysis, Flow 1 and Tank 1 level formed an optimum input-output pair, the PID controller parameters were designed through root locus method on GP1(s) only. Target transient specifications were approximately 20% overshoot and a settling time of 1.2 s. These criteria were translated into desired closed-loop pole locations, from which controller design proceeded. For each loop:

1) Root locus of the uncompensated plant was plotted using *rlocus(G)* in MATLAB.
2) PD compensation was introduced by placing a zero to reshape the locus, enabling placement of dominant poles at the desired locations. The appropriate gain was determined using *rlocfind*.
3) Integral control was added to eliminate steady-state error. A zero at negative z_i and a pole at the origin were added, thus forming a complete PID compensator. The root locus of the compensated system was replotted, and the final gain was tuned.
4) Final controller parameters (K_p, K_i, K_d) were extracted from the resulting PID structure and verified through closed-loop simulation using MATLAB's *step* and *feedback* functions.

All controller designs were iteratively refined based on simulation results in MATLAB Simulink to better meet performance requirements and improve robustness.

2.6. MPC Design

Model Predictive Control was designed and tuned based on a discrete-time state-space model derived from identified transfer functions, following the steps below [17]:

1) Initially, the identified continuous-time transfer-function matrix obtained from frequency-domain system identification was discretized at an appropriate sampling period T_s. In this experiment, T_s was selected as 0.05 s, which was small enough to satisfy the Nyquist-Shannon Sampling Theorem. The discrete-time state-space representation of the system was obtained using MATLAB's *c2d* command. A general discrete-time state-space model is as follows:

$$x_m[n+1] = A_m x_m[n] + B_m u[n] \tag{16}$$

$$y[n] = C_m x_m[n] \tag{17}$$

Where u[n] is the manipulated variable or input variable; y[n] is the process output; and x_m[n] is the state variable vector.

2) Then, assuming the plant had m inputs, q outputs and n state variables, the augmented model was formed as follows:

$$x[n+1] = Ax[n] + B\Delta u[n] \tag{18}$$

$$y[n] = Cx[n] \tag{19}$$

$$x[n] = [\Delta x_m[n]^T \; y[k]]^T \tag{20}$$

$$\Delta x_m[n] = x_m[n] - x_m[n-1] \tag{21}$$

$$x[n+1] = [\Delta x_m[n+1]^T \; y[n+1]]^T \tag{22}$$

$$\Delta x_m[n+1] = x_m[n+1] - x_m[n] \tag{23}$$

$$\Delta u[n] = u[n] - u[n-1] \tag{24}$$

$$A = \begin{bmatrix} A_m & {o_m}^T \\ C_m A_m & I_{q\times q} \end{bmatrix} \tag{25}$$

$$B = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \tag{26}$$

$$C = [o_m \quad I_{q\times q}] \tag{27}$$

Where $I_{q\times q}$ is the identity matrix with dimensions q×q, and 0_m is a q×n1 zero matrix.

3) After formulation of the augmented model, the next step was to calculate the predicted plant output with the future control signal as the adjustable variables. This prediction was defined within an optimization window, whose length is N_p. Assuming the current time as n_i, the following was defined:

$$\Delta u[n_i], \Delta u[n_i + 1], \dots, \Delta u[n_i + N_c - 1] \tag{28}$$

Where N_c is the control horizon (chosen as less than or equal to N_p) which dictates the number of parameters used to capture the future control trajectory. With given $x[n_i]$, through repeated substitution of Eq. 18 with $x[n_i]$ and Eq. 28, the future state variables were calculated and defined as:

$$x[n_i + 1|\, n_i], x[n_i + 2|\, n_i], \dots, x[n_i + m|\, n_i], \dots, x\left[n_i + N_p|\, n_i\right] \tag{29}$$

Where $x[n_i + m|\, n_i]$ is the predicted state variable at $n_i + m$ with given current plant information $x[n_i]$. Then, through repeated substitution of Eq. 19 with $x[n_i]$ and Eq. 28, the following can be calculated:

$$Y = Fx[n_i] + \Phi\Delta U \tag{30}$$

Where

$$Y = \left[y[n_i + 1|\, n_i] \dots \; y[n_i + m|\, n_i] \dots \; y\left[n_i + N_p|\, n_i\right]\right]^T \tag{31}$$

$$\Delta U = \left[\Delta u[n_i] \quad \Delta u[n_i + 1] \quad \Delta u[n_i + 2] \dots \Delta u[n_i + N_c - 1]\right]^T \tag{32}$$

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_P} \end{bmatrix} \tag{33}$$

$$\Phi = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_P-1}B & CA^{N_P-2}B & CA^{N_P-3}B & \cdots & CA^{N_P-N_C}B \end{bmatrix} \tag{34}$$

Where $y[n_i + m|\, n_i]$ is the predicted output at $n_i + m$ with given current plant information $x[n_i]$.

4) Next, to compute Y, given a constant set-point signal $r[n_i]$ at sample time n_i, the following equation was solved for to obtain the best ΔU that will bring the predicted output as close as possible to $r[n_i]$.

$$\Delta U = (\Phi^T\Phi + \bar{R})^{-1}\Phi^T(R_s - Fx[n_i]) \tag{35}$$

$$\bar{R} = r_w I_{N_C\times N_C} \tag{36}$$

$$R_s^T = \overbrace{[1\ 1\ \cdots 1]}^{N_p} r[n_i] \tag{37}$$

Where r_w is the tuning parameter for the desired close-loop performance. Eq. 35 computes the optimal ΔU that will minimize the cost function, J, defined as follows:

$$J = (R_s - Y)^T(R_s - Y) + \Delta U^T\bar{R}\Delta U \tag{38}$$

5) By adjusting the three MPC tuning parameters, N_p, N_c and r_w, the above equations were solved for using MATLAB. After computing ΔU, however, only the first control increment, $\Delta u[n_i]$ was applied at each instant, thereby following the receding-horizon principle. In essence, the closed-loop MPC controller was computed and simulated iteratively using MATLAB. At each simulation step:

a) ΔU was computed by Eq. 35.
b) The first input increment, $\Delta u[n_i]$ was applied to update the inputs in Eq. 18 and hence Eq. 19.
c) The augmented model was advanced using the updated inputs.
d) System outputs and inputs were logged for analysis.

6) Throughout the process, the three MPC tuning parameters, N_p, N_c and r_w were varied and adjusted iteratively to get the best possible response based on metrics such as overshoot, settling time and actuator smoothness and feasibility.

The final MPC parameters, N_p, N_c and r_w were selected based on an optimal trade-off of system responsiveness, minimal overshoot, rapid settling, and reasonable actuator actions. MPC controller performance was illustrated graphically by plotting output responses and control actions over a simulation duration of 100 points.

The MPC tuning parameters, N_p, N_c and r_w were iteratively adjusted to achieve optimal performance, defined by minimal overshoot, short settling time, and control actions within the physical flow rate limit of 26 L/min.

2.7. Comparison of PID and MPC

After designing and independently tuning both PID and MPC controllers, a comprehensive comparative study was conducted to evaluate their relative effectiveness and performance on the interacting two-tank MIMO system. The comparison focused primarily on four critical performance metrics:

1) Transient response, including settling time and percent overshoot.
2) Steady-state accuracy, which evaluated steady-state errors or offsets.
3) Robustness to disturbances.
4) Control effort such as actuator movements.

3. Results and Discussion

3.1. System Identification

3.1.1. Frequency Responses of the System

Chirp excitation was applied sequentially to each inlet flow valve over a 300 s interval, while the opposing valve remained fixed at its steady-state opening. Tank levels were recorded at 10 Hz using analogue water-level sensors connected to Arduino Mega 2560. To improve measurement reliability, each chirp test was repeated three times and the corresponding water level readings were averaged before plotting.

Frequency responses were computed by isolating each chirp segment around a target frequency, zero-padding to enhance frequency resolution, and applying a Hamming window to reduce spectral leakage. The Discrete Fourier Transform (DFT) of each windowed segment was then calculated. For a given frequency, the system gain magnitude was obtained as the ratio of output and input DFT magnitudes, and the phase shift as the difference between output and input DFT phases. The transfer function models with their respective inputs and outputs are summarized in Table 1 for clarity.

Table 1. Transfer function models with their respective inputs and outputs.

Transfer Function	Input	Output
$GP_1(s)$	Flow 1	Tank 1 water level
$GP_2(s)$	Flow 1	Tank 2 water level
$GP_3(s)$	Flow 2	Tank 1 water level
$GP_4(s)$	Flow 2	Tank 2 water level

Bode plots were plotted separately according to the frequency responses of GP1(s), GP2(s), GP3(s) and GP4(s), as shown in Figures 3 to 6:

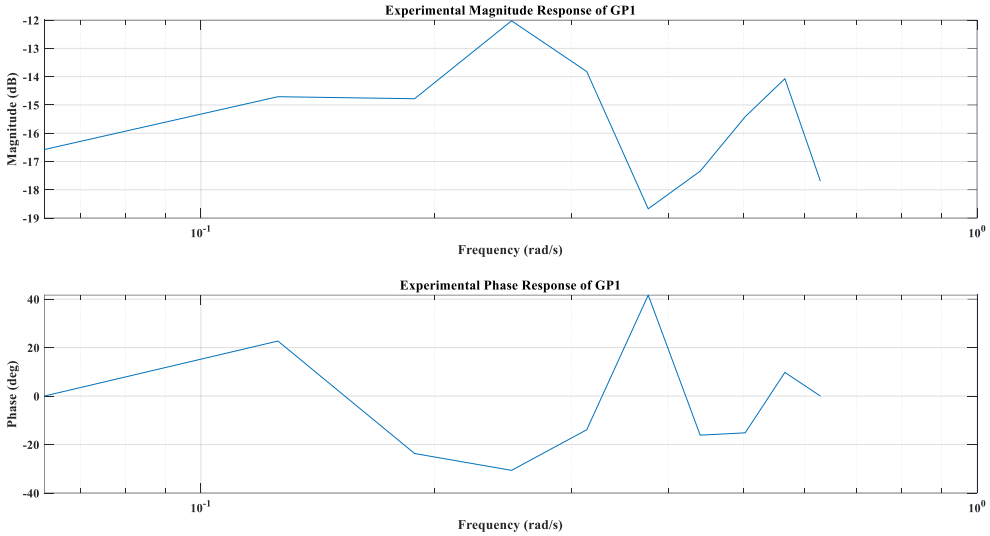

Figure 3: Experimental Bode plot of GP1(s).

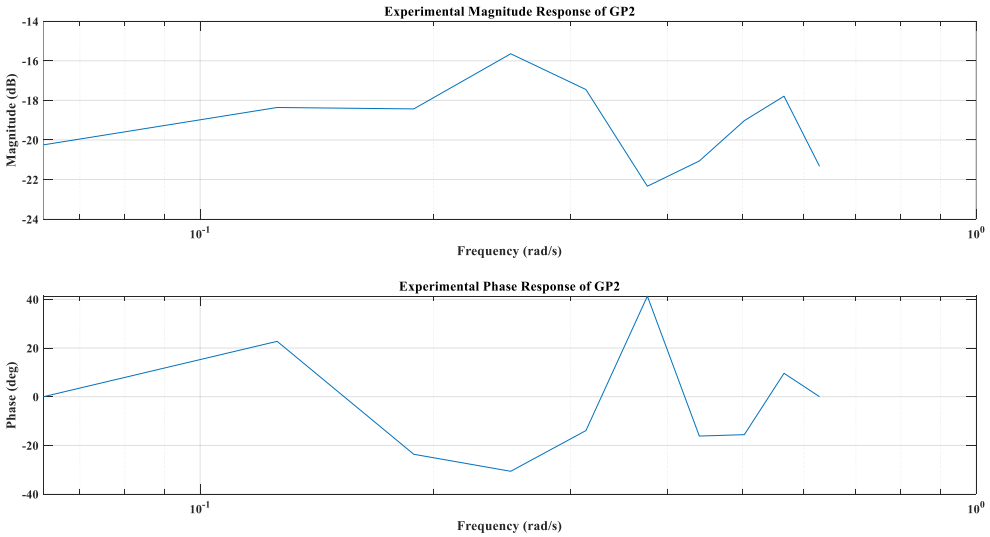

Figure 4: Experimental Bode plot of GP2(s).

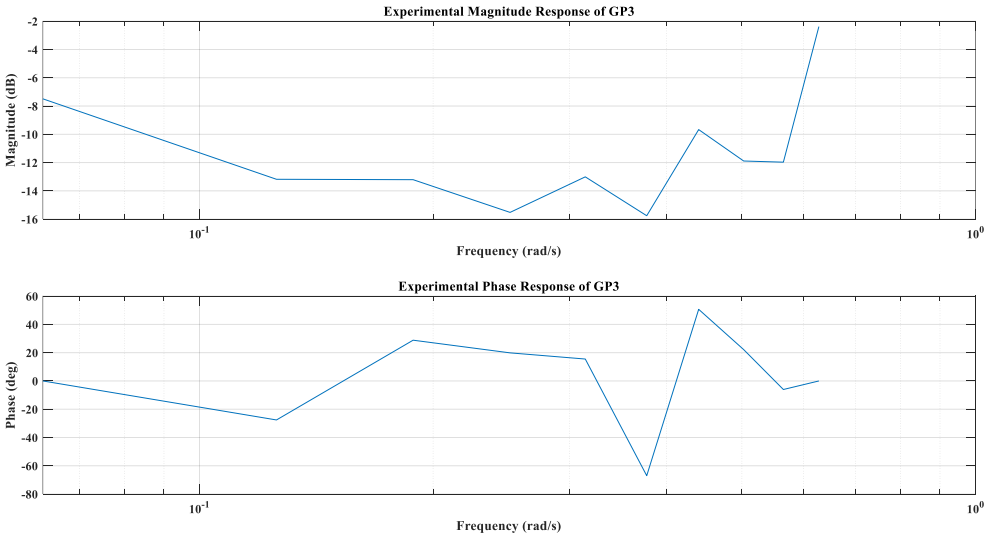

Figure 5: Experimental Bode plot of GP3(s).

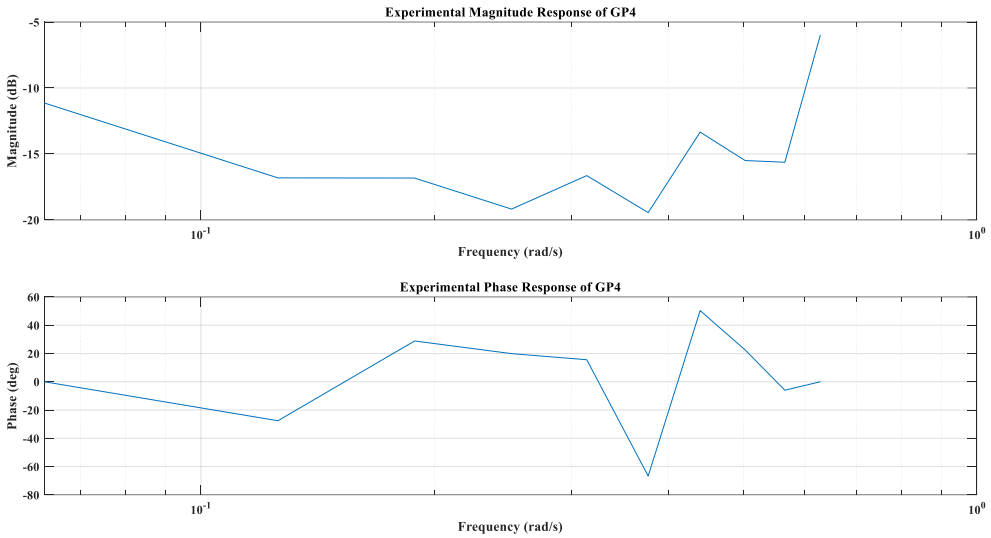

Figure 6: Experimental Bode plot of GP4(s).

The experimental Bode plots for the systems shown from Figures 3 to 6 revealed several characteristic features and non-typical anomalies. In Figures 3 and 4, the magnitude response initially increased up to approximately 0.13 rad/s, which suggests the presence of a zero at the origin for the underlying transfer function. Both figures exhibited dominant magnitude peaks around 0.25 rad/s (-12 dB in Figure 3 and -16 dB in Figure 4) and secondary peaks around 0.55 to 0.56 rad/s (-14 dB in Figure 3 and -18 dB in Figure 4). This suggests the presence of multiple pairs of second-order poles. In contrast, Figures 5 and 6 displayed an initial decrease in magnitude, indicating the presence of a pole at the origin. Figure 5 showed multiple peaks of -13 dB, -10 dB and -2 dB at 0.32 rad/s, 0.45 rad/s and 0.62 rad/s respectively. Each peak suggests the natural frequency of a corresponding second-order pole. Across all figures, the overall negative magnitude implies that the system gain is less than one. Importantly, the phase responses in Figures 3 to 6 exhibited sharp, non-typical peaks (for instance, 40° and 10° in Figures 3 and 4; 50° in Figure 5). This is nontypical, as the phase response of linear time-invariant (LTI) systems does not normally exhibit sharp peaks. Instead, phase typically changes smoothly, with slopes of approximately ±45°/decade for first-order poles or zeros, and ±90°/decade for second-order poles or zeros. These anomalous peaks in the phase plot may be attributed to measurement noise, unmodeled dynamics, or nonlinearities within the system.

3.1.2. Transfer Function Modelling

Transfer functions were derived directly from the experimental Bode plots using a component-by-component estimation approach. The transfer functions were listed along with their corresponding Bode plots as follows:

$$GP1(s) = \frac{0.0070313\ (s + 0.251)^2}{(s + 0.125)^2(s^2 + 0.02093\ s + \ 0.063)} \tag{39}$$

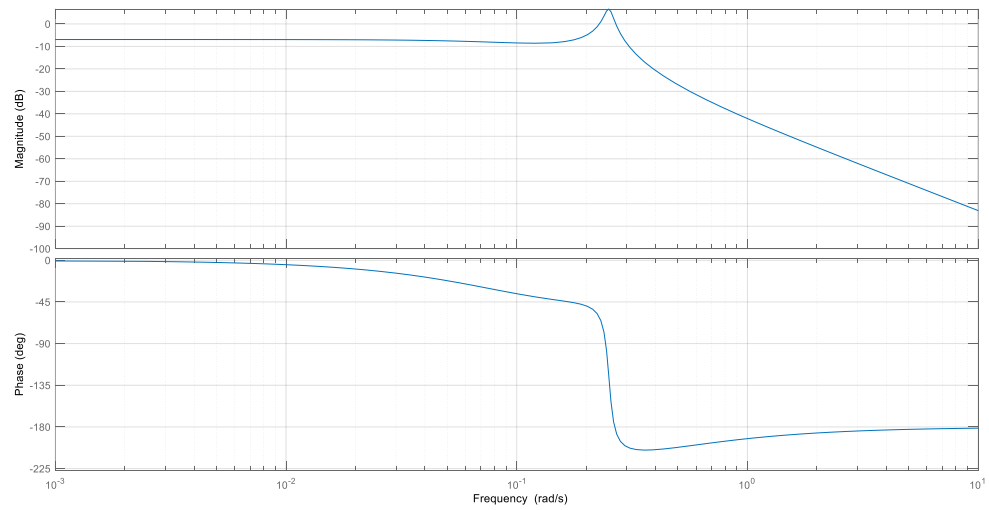

Figure 7: Bode plot of estimated GP1(s).

$$GP2(s) = \frac{0.00019531\ (s + 0.251)^2}{(s + 0.125)^3(s^2 + 0.01605s\ +\ 0.063)} \tag{40}$$

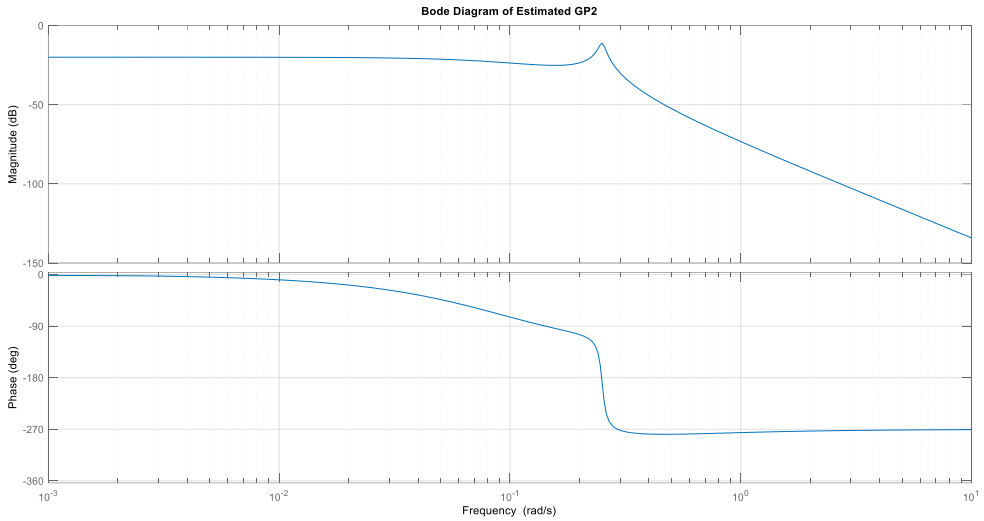

Figure 8: Bode plot of estimated GP2(s).

$$GP3(s) = \frac{0.017209\ (s + 0.251)^2}{s(s^2 + 0.01517s\ +\ 0.1936)} \tag{41}$$

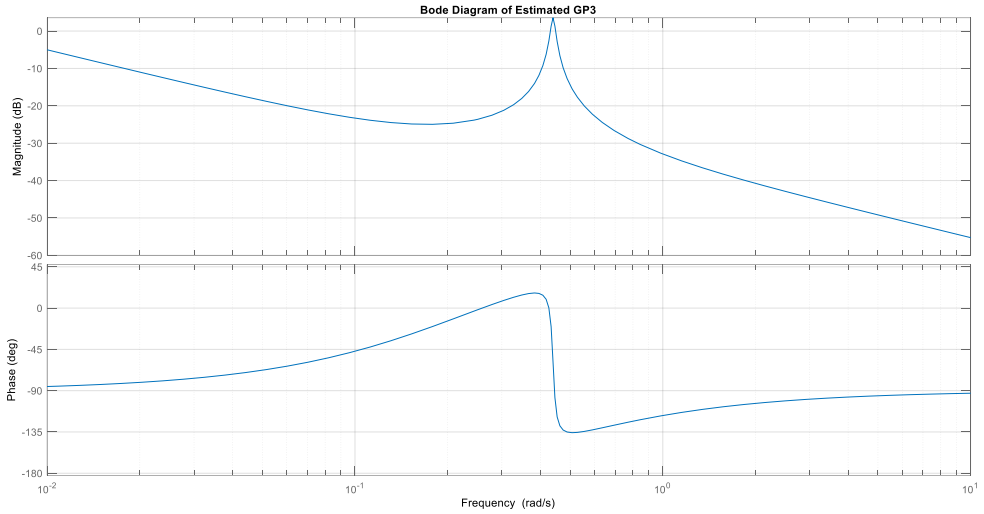

Figure 9: Bode plot of estimated GP3(s).

$$GP4(s) = \frac{0.0062834\ (s + 0.314)^2}{s(s^2 + 0.01467s\ +\ 0.1936)} \tag{42}$$

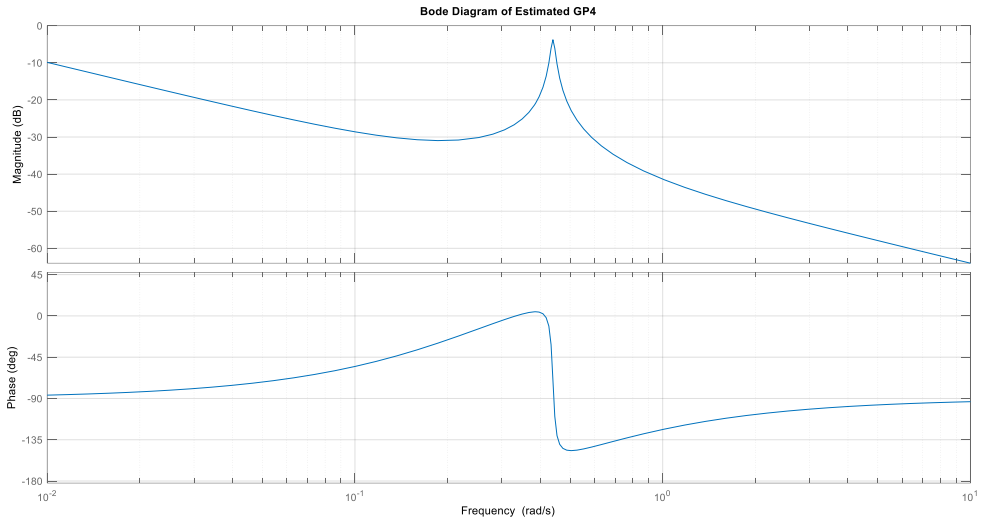

Figure 10: Bode plot of estimated GP4(s).

The magnitude response for the transfer functions GP1(s) and GP2(s) both exhibited a dominant second-order pole pair with a natural frequency of 0.25 rad/s. Specifically, GP1(s) showed a 6.32 dB peak (Figure 7) with a damping ratio of 0.0417, while GP2(s) displayed an -11 dB peak (Figure 8) with a lower damping ratio of 0.032. This lower damping ratio suggests GP2(s) would be more oscillatory than GP1(s). Similarly, transfer functions GP3(s) (Figure 9) and GP4(s) (Figure 10) both showed a dominant second-order pole pair at a natural frequency of 0.44 rad/s. GP3(s) exhibited a 3.38 dB peak with a damping ratio of 0.0172, and GP4(s) showed a -3.79 dB peak with an even lower damping ratio of 0.0167, which implies greater overshoot for GP4(s). Across all four cases, the phase responses of the modelled transfer functions were smooth and lacked the sharp peaks observed in the corresponding experimental Bode plots in Figures 3, 4, 5 and 6. Furthermore, the GP1(s) magnitude response failed to capture the -14 dB peak at 0.55 rad/s present in the experimental data (Figure 3). This suggests that these peaks may be attributed to system noise and nonlinearity. Furthermore, despite the application of DFT, zero-padding, and the Hamming window during signal processing, the noise could not be completely eliminated. This might be due to the noise frequency coinciding with the system's input frequency.

3.2. PID Tuning

3.2.1. RGA Calculation

The dynamic Relative Gain Array (RGA) was computed over the frequency 10^{-5} to 10^{2} rad/s, using 61 logarithmically spaced points. For clarity, Table 2 relates the transfer functions to their corresponding frequency responses and dynamic RGA component, λ_{ij} in the RGA matrix, $\Lambda(j\omega)$.

Table 2. Transfer function models with their respective inputs and outputs.

Transfer Function	Frequency Response	RGA Component
GP1(s)	GP1(jω)	λ_{11}
GP2(s)	GP2(jω)	λ_{12}
GP3(s)	GP3(jω)	λ_{21}
GP4(s)	GP4(jω)	λ_{22}

The MATLAB code aimed to achieve convergence within a tolerance of 10^{-6} in a maximum of 50 iterations. However, convergence within this tolerance was achieved in a single iteration, which indicates the numerical stability of the calculation. Figures 4.17 and 4.18 plot the four RGA element magnitudes $|\lambda_{11}|$, $|\lambda_{12}|$, $|\lambda_{21}|$, $|\lambda_{22}|$ against frequency.

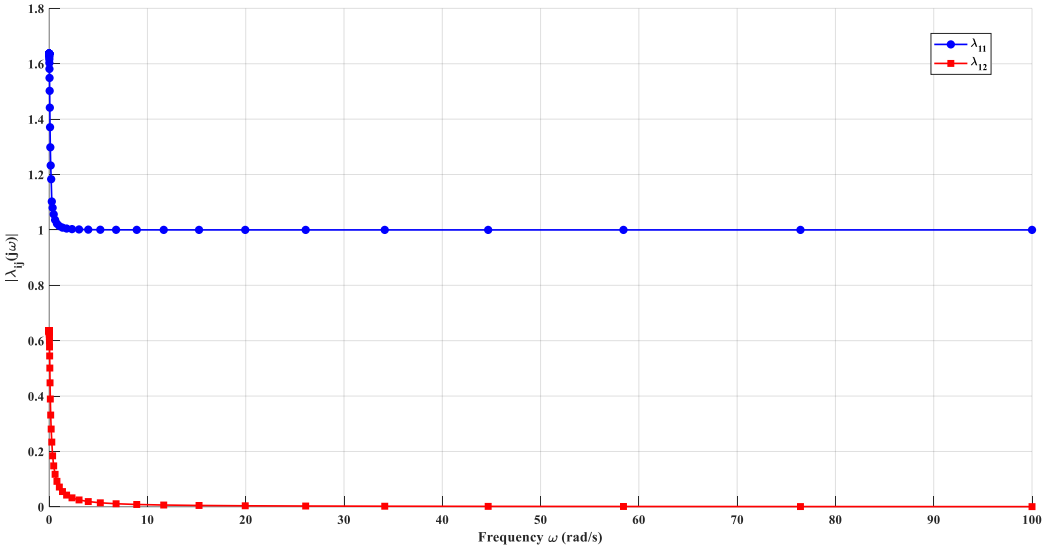

Figure 11: RGA element magnitudes versus frequency for $|\lambda_{11}|$ and $|\lambda_{12}|$.

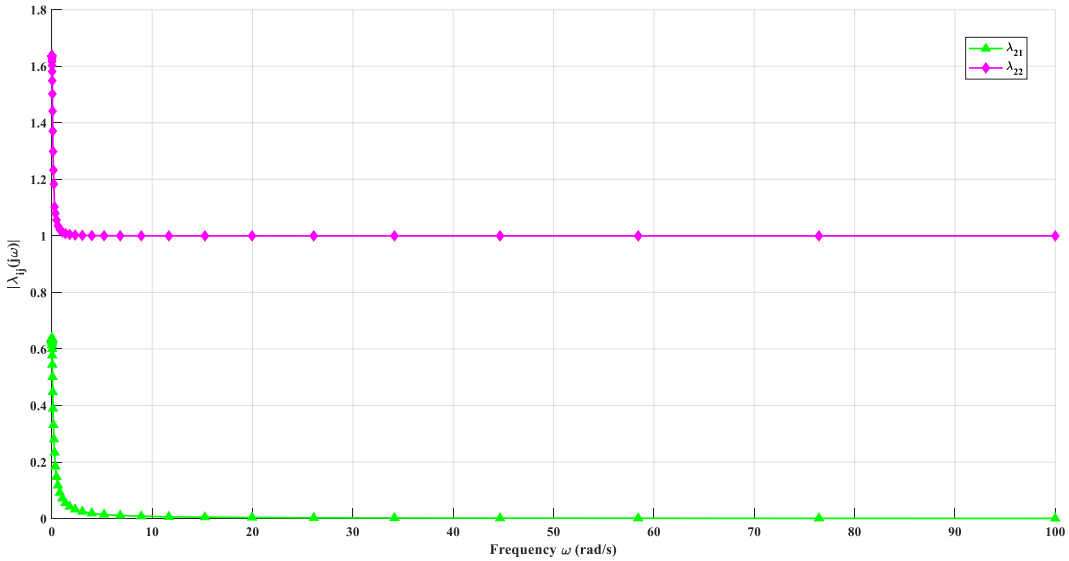

Figure 12: RGA element magnitudes versus frequency for $|\lambda_{21}|$ and $|\lambda_{22}|$.

It is observed as frequency increased, the diagonal elements λ_{11} and λ_{22} remained close to unity, while the off-diagonal elements λ_{12} and λ_{21} stayed below 0.2 and approached 0 eventually. The RGA curves demonstrated clear diagonal dominance. Indeed, strong self-gain and weak cross-gain indicate that each input predominantly influences its matched output, with minimal interaction to the other outputs. Such behaviour confirms that the decentralized pairing of Flow 1 to Tank 1 and Flow 2 to Tank 2 would yield the least loop interaction once controllers are closed.

By selecting the pairing with the smallest off-diagonal RGA values, subsequent PID loops can be tuned as nearly independent SISO systems. Therefore, the design process was greatly simplified since PID controllers were tuned separately for $GP_1(s)$ and $GP_4(s)$. However, RGA analysis is inherently a frequency-domain, steady-state measure, which means it does not account for time-domain transient interactions or nonlinear effects [16]. Moreover, residual coupling could still generate interaction in the real system under large disturbances or aggressive tuning. Nevertheless, the RGA-based pairing provided a reliable starting point for PID tuning, which will be presented in the next section.

3.2.2. PID Tuning on $GP_1(s)$ and $GP_4(s)$

To ensure precise control of the pair Flow 1 – Tank 1, a Proportional–Integral–Derivative (PID) controller was designed and tuned to meet the target transient specifications, namely, 20% overshoot and

a 1.2 s settling time while eliminating steady-state error. The root locus was plotted to design the PID controllers. First, the PID controller for GP1(s), $G_{PID,1}(s)$ was determined to be:

$$G_{PID,1}(s) = 9074.9 + \frac{4261.8}{s} + 1102.8s \tag{43}$$

Next, a PID controller was sought for the pair Flow 2 – Tank 2. However, the PID controller tuning of GP4(s) exhibited anomalies that could be well explained through its root-locus design procedures. Hence, for the sake of clarity, the design procedures were outlined explicitly here:

Adhering to the transient requirements of 20% overshoot and 1.2 s settling time, the root locus of the uncompensated plant, GP4(s) was first plotted in Figure 13:

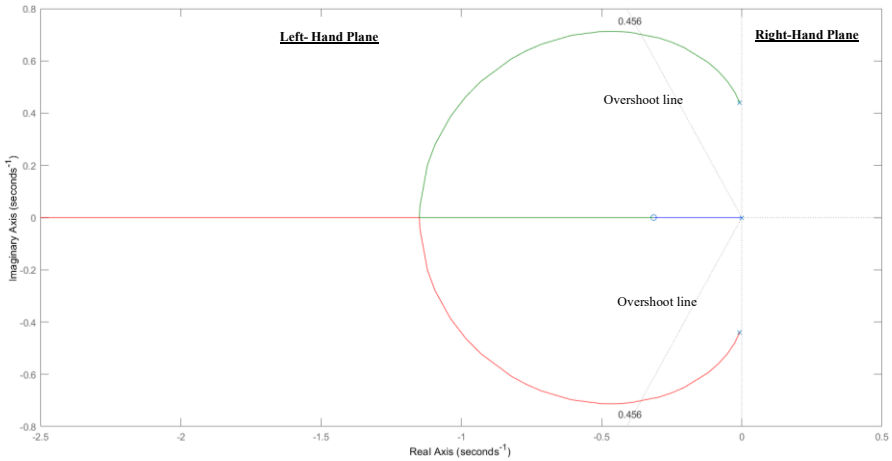

Figure 13: Root locus of GP4(s) with 20% overshoot line.

The root locus of GP4(s) intercepted the 20% overshoot line (dotted). This means that the transient response of 20% overshoot can be achieved by proportional gain adjustment alone. Hence, the point of intersection between the root locus and the overshoot line gave a gain, K of 132.6, and resulted in dominant closed-loop poles at -0.358 ± *j* 0.703. However, the corresponding settling time was calculated to be 11.17 s (almost ten times the 1.2 s target). This demonstrates that proportional control alone could not meet all transient specifications.

Then, a PD zero was introduced to shift the root locus towards higher damping and faster response. However, the root locus of the PD-compensated system is as follows:

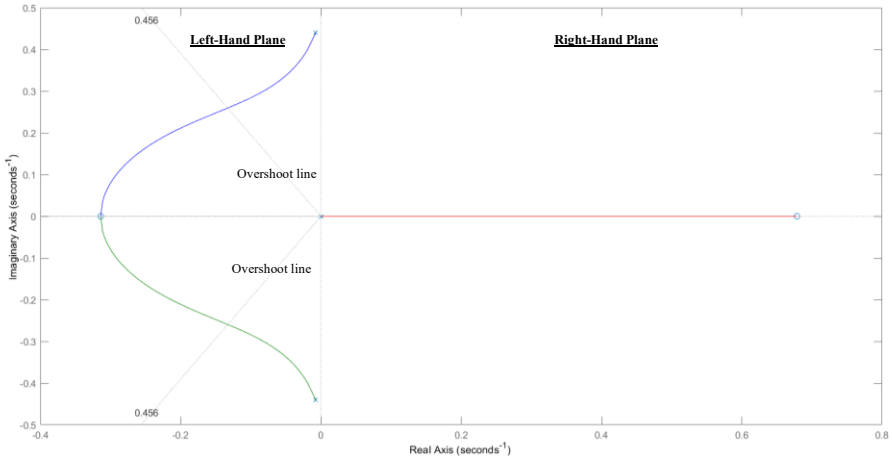

Figure 14: Root locus of PD-compensated GP4(s) with 20% overshoot line.

This presented a nonideal case, since a whole branch of the PD-compensated system's root locus extended into the right-hand plane, which will result in real positive poles (unstable plant) regardless of the gain. The same unstable root loci resulted despite increasing the settling time to 4 s and 10 s, as presented below:

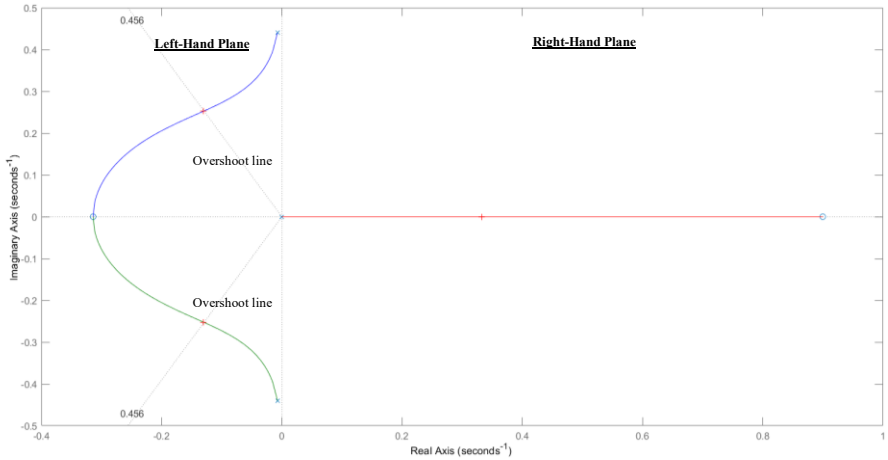

Figure 15: Root locus of PD-compensated GP4(s) with 20% overshoot line and settling time of 4 s.

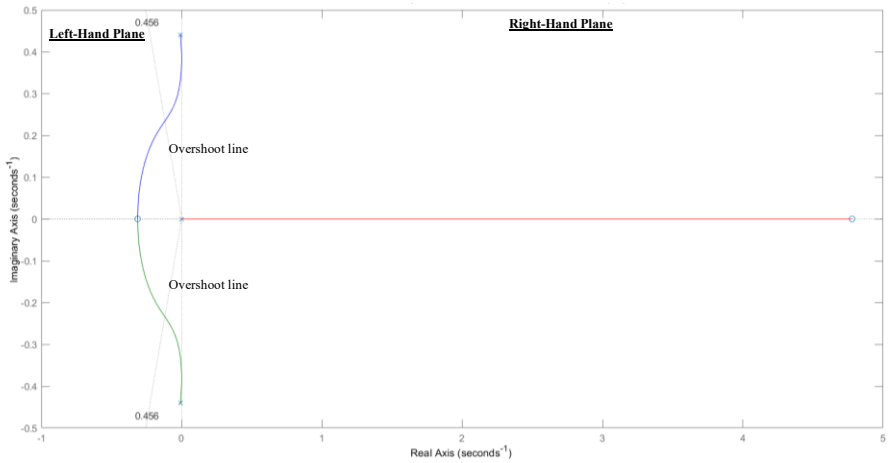

Figure 16: Root locus of PD-compensated GP4(s) with 20% overshoot line and settling time of 10 s.

Far from stabilizing the system, the system grew more prone to instability, as observed from the shrinking of other root locus branches (blue and green) towards the right-hand plane. This outcome indicates that GP4(s) with a pole at the origin (type-1 behaviour) combined with the lightly damped second-order dynamics cannot be tamed by adding only a zero. Additionally, since GP4(s) is a system with a pole at the origin (type-1 system), it will not exhibit offset error when subject to a step input [15].

Because neither proportional nor PD control could simultaneously achieve adequate speed and stability, a standard PID (adding an integrator) was not pursued, since the integrator would introduce yet more low-frequency phase lag and likely more instability. Therefore, the Flow 2 – Tank 2 loop was left with the best achievable proportional gain tuning (K = 132.6), while the Flow 1 – Tank 1 loop employed the PID controller as shown in Equation 43.

3.2.3. Simulation of PID Controller on MIMO System

To determine the efficiency and robustness of both controllers designed via the root locus method, the system, along with the controllers, was simulated in Simulink. Figure 17 illustrates the multiloop configuration of the system in Simulink. Note that in the figure, PID1 corresponds to the PID controller in Loop 1 (Flow 1 – Tank 1), while PID2 corresponds to the PID controller in Loop 2 (Flow 2 – Tank 2). For clarity, the root-locus-tuned PID parameters are restated as follows:

- PID1: K_p = 9074.9, K_i = 4261.8, K_d = 1102.8
- PID2: K_p = 132.6, no integral or derivative action

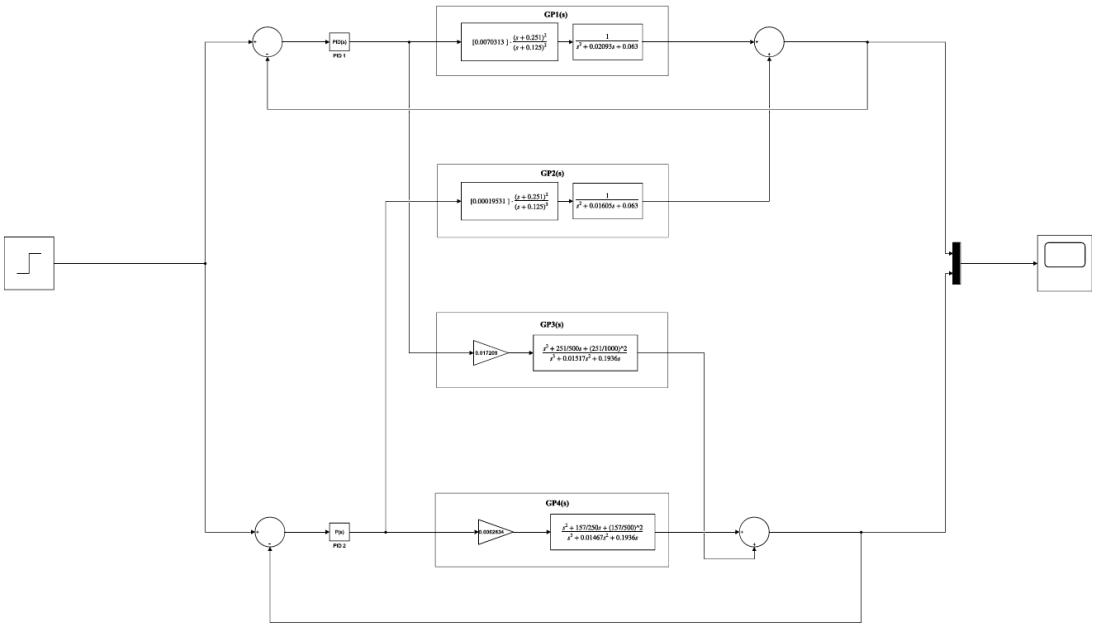

Figure 17: Multiloop configuration of the system in Simulink.

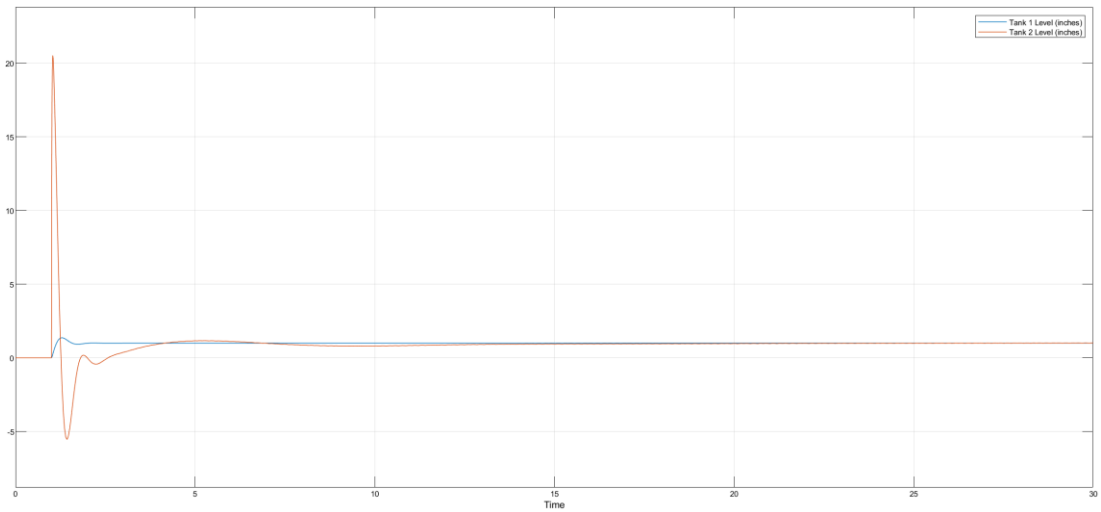

Figure 18: Closed-loop step responses with root-locus PIDs.

Figure 18 shows the closed-loop step responses with root-locus-tuned PID controllers. In the closed-loop simulation using root-locus tuned PIDs as shown in Figure 18, Tank 1 reached its set-point in 1.38 s but exhibited a 36.6% overshoot, whereas Tank 2 required 24.1 s to settle and exhibits a peak magnitude of 20.53, which corresponds to 1953% overshoot. Therefore, when run simultaneously on the coupled plant, the root-locus controllers lost much of their SISO performance. In particular, the Tank 2 loop underwent enormous overshoot and a 24 s settling time. Large cross-coupling disturbances from the Flow 1 loop drove Tank 2 far beyond its set-point.

To improve the performance, the PID Tuner app in MATLAB/Simulink was used. The tuner linearizes the nonlinear plant model and designs a SISO PID controller for each loop independently. Each loop was tuned assuming the other loop to be inactive (open-loop), and no decoupling was applied during tuning. The tuning objective was based on the default PID Tuner trade-off between response speed and robustness. The parameters of the Simulink-tuned PID are as follows:

- Loop 1 (GP1(s)): K_p = 48.22, K_i = 3.37, K_d = 170.61
- Loop 2 (GP4(s)): K_p = 112.29, no integral or derivative action

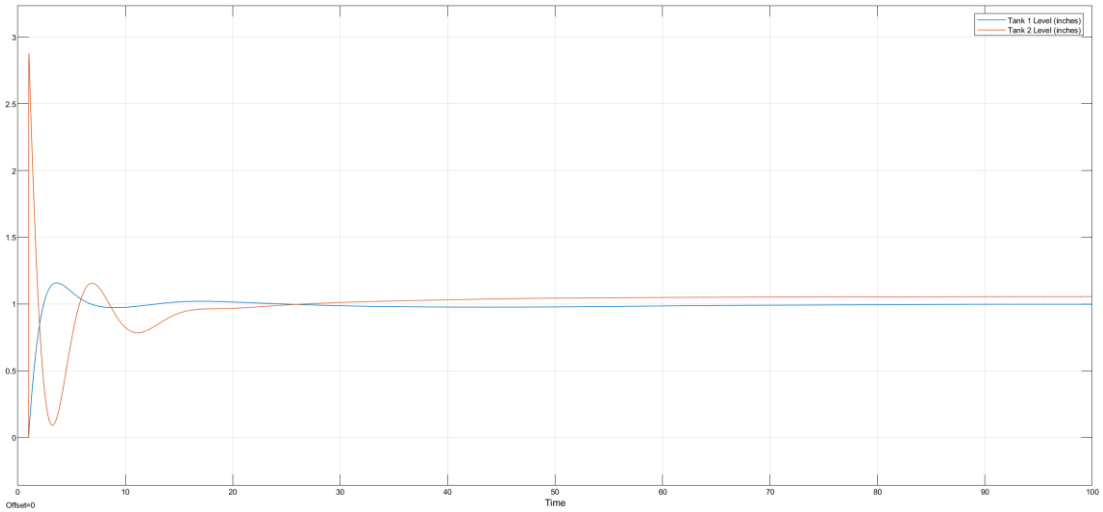

Figure 19: Closed-loop step responses with Simulink-tuned PIDs

Figure 19 shows the closed-loop step responses with Simulink-tuned PID controllers. When the controllers were instead tuned via Simulink's PID Tuner as in Figure 19, Tank 1's response slowed considerably by settling in 53.3 s with a reduced 15.8% overshoot. However, Tank 2 still failed to stabilize within the simulation window and overshot by 187.6%.

Therefore, using the Simulink PID Tuner improved individual loop robustness. In this case, Tank 1 overshoot dropped to 16%, but the Tank 2 loop still could not recover, and settling time grew excessively long. This implies that the single proportional controller simply cannot reject the persistent disturbance introduced by the other loop. Since Simulink-tuned PIDs gave better performance than root-locus PIDs, their control actions are examined in Figure 20:

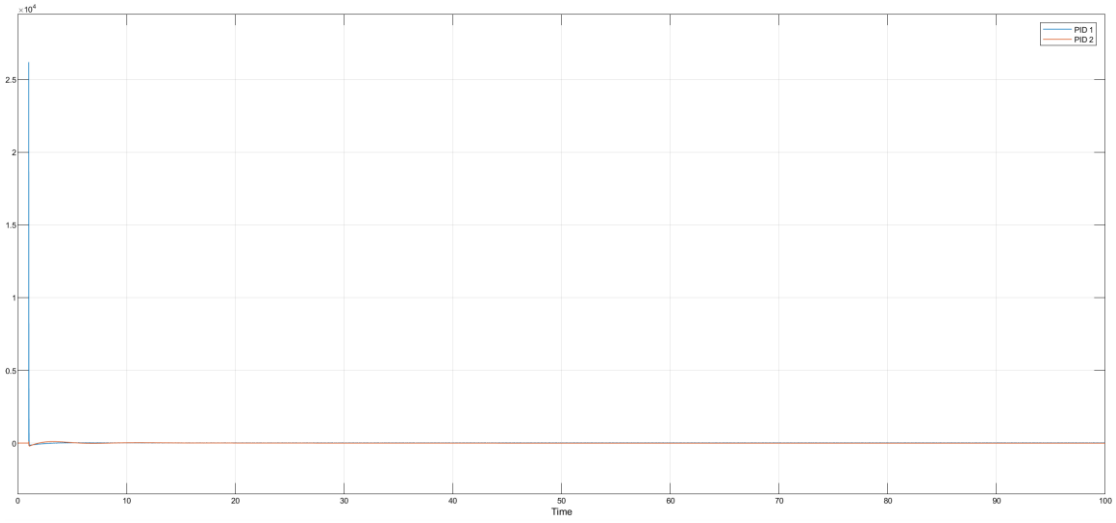

Figure 20: Control action of Simulink-tuned PIDs.

As shown in Figure 20, the control action of the Simulink-tuned PID controllers, though producing better response than root-locus PID controllers, exceeded the mechanical constraints of both Flow 1 and Flow 2 (26 L/min) by a very large amount (peaking at about 2.7×10^4 L/min). This made the control action impractical.

These results demonstrate that when the two loops operate in parallel on the MIMO plant, without any decoupling strategy, the residual cross-coupling overwhelms standard PID action. Even an optimized PID for each SISO loop fails to deliver acceptable multivariable performance and exceeded the mechanical constraints to an impractical extent.

3.3. MPC Tuning

To explore the trade-offs inherent in receding-horizon control, the MIMO MPC was tuned by systematically varying the tuning parameter r_w while holding the prediction horizon N_p and control horizon N_c constant. Using the discrete state-space model with N_p = 100 and N_c = 40, as suggested by Wang [17], seven values of r_w (0, 0.01, 0.1, 0.3, 0.4, 0.5 and 1) were tested. At each setting, closed-loop step responses for both tank levels and the corresponding valve commands were simulated over 500 time-

steps, which corresponded to a duration of 25 s, since the sampling period is 0.05 s. Key metrics (settling time, percent overshoot and maximum control action) were then extracted to identify the set of parameters that best balanced fast transient responses with smooth and realistic actuator motions. The key performances metrics are summarized in Table 3 below:

Table 3. Key performance metrics of MPC controller.

r_w	**Settling Time (s) (Tank 1, Tank 2)**	**% Overshoot (Tank 1, Tank 2)**	**Maximum Flow (L/min) (Flow 1, Flow 2)**
0	0.25, 4.10	2.14, 70.52	2.20×10^5, 6.06×10^5
0.01	10.39, 5.70	8.64, 14.97	56.27, 49.36
0.10	19.87, 7.56	10.12, 7.73	26.51, 33.43
0.30	23.10, 18.58	15.00, 3.20	19.68, 32.70
0.40	23.76, 20.92	17.83, 3.78	18.20, 32.85
0.50	24.13, 22.29	20.28, 4.13	17.12, 32.96
1.00	24.61, 24.59	23.95, 9.63	14.18, 32.71

Based on Table 3, When r_w = 0, the controller achieved exceptionally fast convergence, that is, both tank levels reached the set-point in under 4.10 s. However, the controller drove Flow 1 and Flow 2 instantly to their extremes by delivering extremely rapid settling but impossible valve commands of 2.20×10^5 L/min and 6.06×10^5 L/min. These flowrates far exceed the physical limit of 26 L/min. Increasing r_w to 0.01 resulted in longer settling times of 10.39 s, 5.70 s. However, its valve commands of 56.27 L/min and 49.36 L/min were drastically lower than those of r_w = 0, though they still exceeded the physical limit. By trying on different r_w, it was found that r_w = 0.3 resulted in the best response, with overshoots of 15.00% and 3.20% and realistic valve commands of 19.68 L/min and 32.70 L/min (though they still exceeded the physical limits). However, its settling times of 23.10 s and 18.58 s were relatively slow. In addition, other r_w values of 0.1, 0.4, 0.5 and 1.0 gave no significant improvements. Therefore, selecting r_w = 0.3, the outcomes of additional tuning parameters sweeping the prediction horizon (N_p = 100, 200, 250, 300, 350, 400) and control horizon (N_c = 40, 60, 80) were run and presented in Table 4.

Table 4. Key performance metrics of varying N_p and N_c and constant r_w (0.3).

N_p	N_c	**Settling Time (s) (Tank 1, Tank 2)**	**% Overshoot (Tank 1, Tank 2)**	**Maximum Flow (L/min) (Flow 1, Flow 2)**
100	40	23.10, 18.58	15.00, 3.20	19.68, 32.70
100	60	23.23, 19.32	16.32, 3.48	19.40, 33.25
100	80	23.23, 19.42	16.56, 3.67	19.38, 33.39
200	40	15.74, 18.52	9.74, 7.28	17.56, 29.00
250	40	Do not settle	N/A	9.23×10^{24}, 2.23×10^{25}
300	40	22.65, 10.50	0, 0	16.4, 24.98
350	40	8.72, 14.32	0.21, 210.17	1.92×10^5, 5.29×10^5
400	40	18.42, 22.04	8.73, 0	15.14, 19.54

According to Table 4, varying N_c (first three rows) and keeping N_p constant at 100 give no significant improvements to the outcomes, since the largest settling time difference across different N_c is just around 3.9%, while percent overshoots differ in around 8%. Nonetheless, the maximum flows of around 33 L/min for N_c = 40, 60 and 80 still exceed the physical flow limit of 26 L/min. However, changing N_p varied the transient responses and maximum valve commands significantly, with N_p = 300 being the best. The input and output responses of Np =300 is plotted in Figure 21 below:

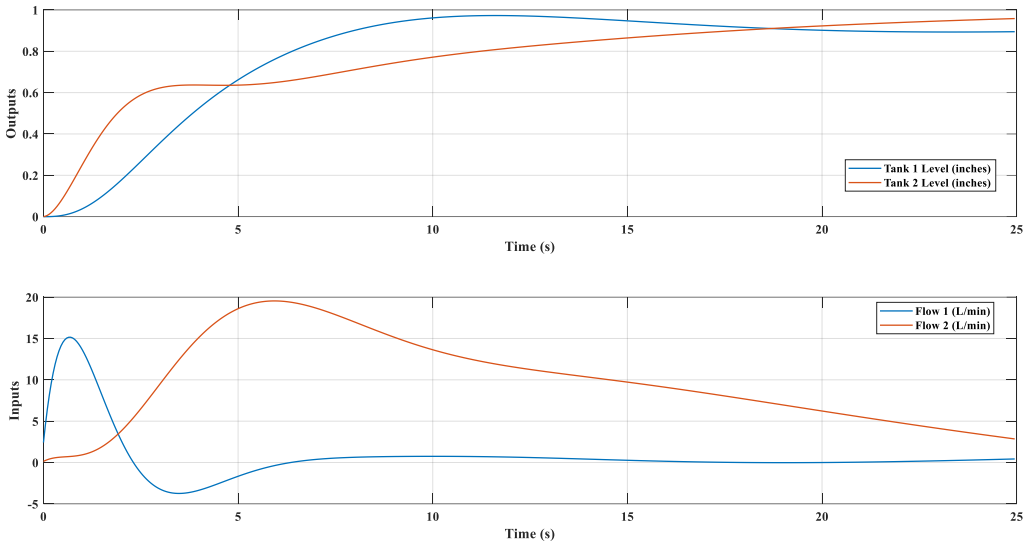

Figure 21: Inputs and outputs when r_w = 0.3, N_p = 300 and N_c = 40.

According to Figure 21, the outputs exhibited 0% overshoots and delivered valve commands (maximum flow of 24.98 L/min) that are well within the physical limits (26 L/min). However, notable cases occur when N_p = 250 and 350, whose responses are plotted in Figure 22 and Figure 23, respectively.

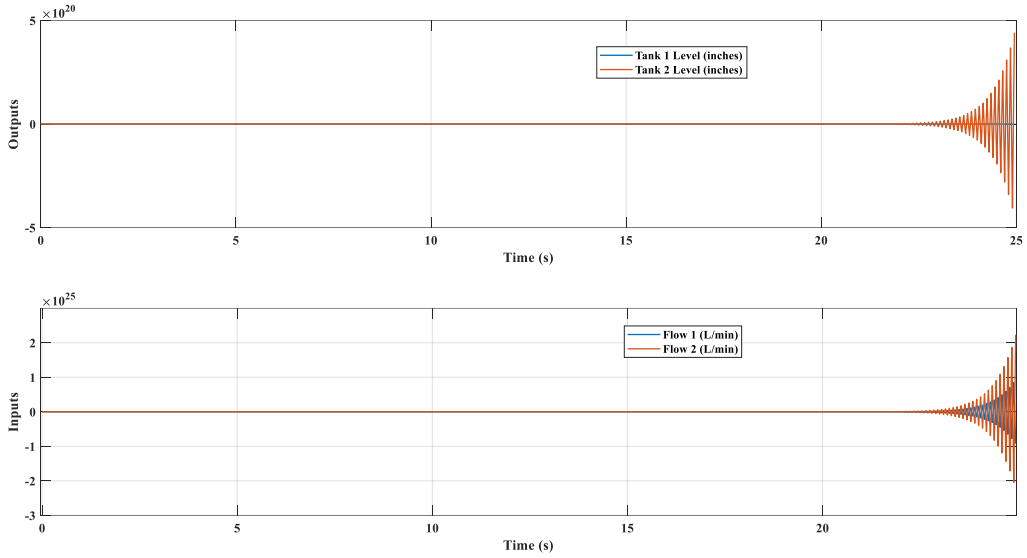

Figure 22: Inputs and outputs when r_w = 0.3, N_p = 250 and N_c = 40.

Based on Figure 22, when N_p equals 250, Tank 1 and Tank 2 water levels do not settle. Instead, they become unstable that their magnitudes grow unboundedly large. The MPC also demands impossible flow rates of 9.23×10^{24} L/min and 2.23×10^{25} L/min to control Tank 1 and Tank 2 water levels, respectively.

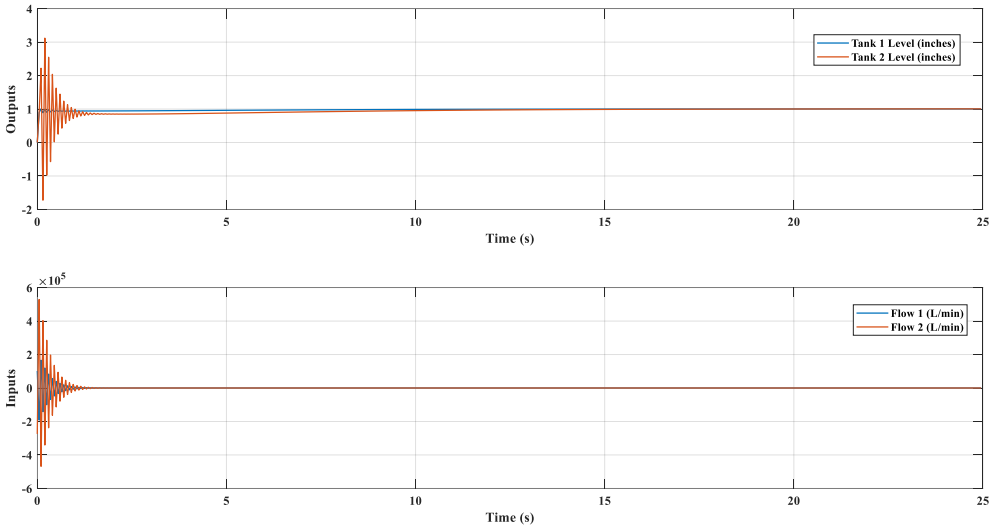

Figure 2:. Inputs and outputs when r_w = 0.3, N_p = 350 and N_c = 40,

On the other hand, N_p = 350 gives a rapid output response that settles within 8.72 s (Tank 1 water level) and 14.32 s (Tank 2 water level). Also, the overshoot of Tank 1 water level is 0.21%, which is remarkably small. However, the overshoot of Tank 2 water level (210.17%) is large. Other than that, the MPC also demands impossible flow rates of 1.92×10^5 L/min and 5.29×10^5 L/min to control Tank 1 and Tank 2 water levels, respectively.

Although N_p = 400 lowered valves' flowrates, it exhibited an overshoot of 8.73% (higher than that of N_p = 300) and settling time of 22.04 s. To conclude, while N_c has little influence on the MPC's outcomes, increasing r_w and N_p do not necessarily give better results. More importantly, alter the parameters without care will result in unstable responses, as when N_p = 250. In other words, the optimal MPC parameters do not follow a linearly increasing pattern and must be determined through trial and error. Consequently, the final MPC parameters were chosen to be N_p = 300, N_c = 40 and r_w = 0.3. A side-by-side comparison with PID controllers was detailed in the next section.

3.4. Comparative Evaluation of PID versus MPC

To assess the practical benefits of multivariable prediction versus decentralized PID loops, the two best-case PID configurations (Simulink-tuned PIDs) were compared side-by-side with the final MPC setup (N_p = 300, N_c = 40 and r_w = 0.3). Table 5 summarized the key performance metrics for both approaches.

Table 5. Performance metrics for PID vs. MPC.

Controller	Settling time (s) (Tank 1, Tank 2)	Percent Overshoot (Tank 1, Tank 2)	Peak Control Action (L/min)
PID (Simulink-Tuned)	53.3 s, did not settle within the simulation period.	15.8%, 187.6%	2.7×10^4
MPC (N_p = 300, N_c = 40 and r_w = 0.3)	22.56 s, 10.50 s	0%, 0%	24.98

Under the Simulink-tuned PID scheme, Tank 1 eventually tracked the set-point but required 53 s to settle, with a moderate 16% overshoot. Tank 2 failed to converge entirely within the simulated timeframe and exhibited an excessive 187% overshoot. Moreover, the PID action drove the valve command for Flow 1 up to an impractical 2.7×10^4 L/min (orders of magnitude above the 26 L/min physical limit). This underscores the inability of decentralized PID to reject cross-coupling disturbances.

In contrast, the MPC controller achieved shorter settling times than PID controllers did for both tanks (22.56 s and 10.50 s) and remarkable overshoots of 0%. More importantly, the MPC-controlled valve flows never exceeded 24.98 L/min, which fits comfortably within hardware constraints. By optimizing both inputs simultaneously, MPC inherently managed inter-loop interactions and enforced realistic actuator limits.

This direct side-by-side comparison demonstrated that, for tightly coupled processes such as the two-tank system, MPC delivers far superior multivariable performance. While decentralized PID struggles with cross-disturbances and produces unimplementable control actions, MPC meets transient objectives, bounds overshoot, and stays within physical flow limits.

4. Conclusions

This study aimed to enhance the control of an interacting two-tank system by applying both classical and advanced control strategies. Transfer function models were derived by injecting chirp signals into the valves and processing the recorded level responses through Discrete Fourier Transform (DFT) techniques. These models captured the essential dynamics of the coupled tanks and enabled the design of both PID and MPC controllers. PID controllers were tuned via root locus techniques and further refined in simulation, whereas MPC was developed using a discrete-time state-space model and optimized over multiple horizon and tuning parameters. Simulation results demonstrated that while PID provided satisfactory performance in single-loop simulations, it struggled with multivariable coupling effects. MPC, on the other hand, offered improved stability, control feasibility, and better handling of interactions. Therefore, MPC demonstrates better suitability for complex MIMO systems.

Declaration Of Competing Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this manuscript.

Acknowledgments: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

1. Manna, S.; Singh, D. K.; Ghadi, Y. Y.; Yousef, A.; Kotb, H.; AboRas, K. M. Probabilistic bi-level assessment and Adaptive Control Mechanism for two-tank interacting system. *IEEE Access* **2023**, *11*, 118268–118280. https://doi.org/10.1109/access.2023.3326727
2. Mohayeji, M.; Farsi, M.; Rahimpour; Shariati, A. Modeling and operability analysis of water separation from crude oil in an industrial gravitational coalescer. *Journal of the Taiwan Institute of Chemical Engineers* **2015**, *60*, 76–82. https://doi.org/10.1016/j.jtice.2015.10.025
3. Balaram Naik, R. B.; Kanagalakshmi, S. Mathematical modelling and controller design for interacting hybrid two tank system (IHTTS). *2020 Fourth International Conference on Inventive Systems and Control (ICISC)* **2020**, 297–303. https://doi.org/10.1109/icisc47916.2020.9171218
4. Xu, T.; Yu, H.; Yu, J.; Meng, X. Adaptive disturbance attenuation control of two tank liquid level system with uncertain parameters based on Port-controlled hamiltonian. *IEEE Access* **2020**, *8*, 47384–47392. https://doi.org/10.1109/access.2020.2979352
5. Huo, M.; Luo, H.; Wang, X.; Yang, Z.; Kaynak, O. Real-time implementation of plug-and-play process monitoring and control on an experimental three-tank system. *IEEE Transactions on Industrial Informatics* **2021**, *17*, 6448–6456. https://doi.org/10.1109/tii.2020.3030812
6. Liu, T.; Wang, Q.-G.; Huang, H.-P. A tutorial review on process identification from step or relay feedback test. *Journal of Process Control* **2013**, *23*, 1597–1623. https://doi.org/10.1016/j.jprocont.2013.08.003
7. Zhu, Y. System identification for process control: Recent experience and outlook. *IFAC Proceedings Volumes* **2006**, *39*, 20–32. https://doi.org/10.3182/20060329-3-au-2901.00003
8. Pang, Z.-Y.; Ma, Z.-S.; Ding, Q.; Yang, T. An improved approach for frequency-domain nonlinear identification through feedback of the outputs by using separation strategy. *Nonlinear Dynamics* **2021**, *105*, 457–474. https://doi.org/10.1007/s11071-021-06595-w
9. Ljung, L. Frequency domain versus time domain methods in system identification – Revisited. In *Control of uncertain systems: Modelling, approximation, and design*; Francis, B. A., Smith, M. C., Willems, J. C., Eds.; Springer, Berlin, Heidelberg, 2006; Volume 329, pp. 116–133. https://doi.org/10.1007/11664550_15
10. Vuojolainen, J.; Nevaranta, N.; Jastrzebski, R.; PyrhÃ¶nen, O. Comparison of excitation signals in active magnetic bearing system identification. *Modeling, Identification and Control: A Norwegian Research Bulletin* **2017**, *38*, 123–133. https://doi.org/10.4173/mic.2017.3.2
11. Xia, X. System identification using chirp signals and time-variant filters in the joint time-frequency domain. *IEEE Transactions on Signal Processing* **1997**, *45*, 2072–2084. https://doi.org/10.1109/78.611210
12. Hu, J.; Shan, Y.; Guerrero, J. M.; Ioinovici, A.; Chan, K. W.; Rodriguez, J. Model predictive control of microgrids – an overview. *Renewable and Sustainable Energy Reviews* **2021**, *136*, 110422. https://doi.org/10.1016/j.rser.2020.110422
13. Rodriguez, J.; Garcia, C.; Mora, A.; Flores-Bahamonde, F.; Acuna, P.; Novak, M.; Zhang, Y.; Tarisciotti, L.; Davari, S. A.; Zhang, Z.; Wang, F.; Norambuena, M.; Dragicevic, T.; Blaabjerg, F.; Geyer, T.; Kennel, R.; Khaburi, D. A.; Abdelrahem, M.; Zhang, Z.; ... Aguilera, R. P. Latest advances of model predictive control in

electrical drives—part I: Basic concepts and advanced strategies. *IEEE Transactions on Power Electronics* **2022**, *37*, 3927–3942. https://doi.org/10.1109/tpel.2021.3121532

14. Ramos, J. A.; Lopes dos Santos, P. Mathematical Modeling, system identification, and controller design of a two-tank system. *2007 46th IEEE Conference on Decision and Control* **2007**, 2838–2843. https://doi.org/10.1109/cdc.2007.4434949
15. Nise, N. S. *Control Systems Engineering*, 7th ed.; John Wiley & Sons, Inc.: 2014; pp. 381-629.
16. Albertos P.; Sala, A. *Multivariable control systems—an engineering approach*, 1st ed.; Springer: London, The United Kingdom. 2003; pp. 125-162.
17. Wang, L.P. Model predictive control: Design and implementation using MATLAB (T-3). *2009 American Control Conference* **2009**, 25–26. https://doi.org/10.1109/acc.2009.5159781