A Nonlinear 2D Ball Balancing System with Robotic Arm Actuation and PID-Based Numerical Feedback Linearization

Ege Özbülbül

June 7, 2026

Abstract

This paper presents the modeling, simulation, and PID control of a nonlinear coupled two-dimensional ball balancing system actuated by two robotic planar arms. The objective of the system is to regulate and track the position of a ball on a movable plate by controlling the plate orientation through servo-driven mechanisms. Due to the nonlinear relationship between ball motion, plate inclination, and actuator angles, the control problem is formulated using a virtual acceleration-based approach. In this framework, the PID controller generates desired ball accelerations, which are then converted into physically realizable actuator commands through a numerical feedback linearization procedure. Simulation results are used to evaluate tracking accuracy, transient response, actuator effort, saturation behavior, and empirical frequency-domain characteristics. The study demonstrates how a classical PID controller can be applied to a nonlinear coupled mechatronic system when combined with an appropriate numerical mapping layer, robotic arm kinematics, and actuator constraint modeling.

1 Introduction

The system studied in this project is a two-dimensional ball balancing platform where a rolling ball is stabilized on a movable plate via two orthogonal 2R planar robotic arms. The primary objective is to regulate or track the spatial trajectory of the ball by dynamically manipulating the plate's orientation. Unlike conventional setups that tilt the plate directly via a gimbal, this system utilizes an indirect actuation mechanism. Consequently, the ball motion, plate orientation, and joint angles are governed by a highly coupled, nonlinear mechanical network.

1.1 System Interdependencies and Challenges

The core complexity of the platform stems from three interdependent layers:

1. **Underactuated Ball-on-Plate Dynamics:** The ball moves freely under gravity, driven solely by the plate's inclination. It possesses no direct actuation, making the system inherently open-loop unstable.
2. **Kinematic Cross-Axis Coupling:** Each 2R planar arm is connected to a perpendicular edge of the same rigid plate. Because the plate is a single rigid body, a motion initiated by one arm to control a specific axis inevitably induces structural forces and tilt variations on the orthogonal axis. This cross-coupling prevents treating the plant as two independent 1D problems.
3. **Physical Actuator Constraints:** The system cannot instantaneously track ideal control commands. The driving servo motors are subject to strict velocity rate limits (±60°/s), introducing a heavy saturation boundary between the computed mathematical commands and real-world execution.

1.2 Proposed Methodology and Document Structure

To overcome these challenges, a sequential model-based PID control framework is implemented. Instead of directly commanding actuator angles from position error, the controller first generates a virtual acceleration command. This command is then processed through a sequence of mathematical transformations:

- A real-time numerical optimization layer (`fsolve`) that numerically inverts the nonlinear plant dynamics and maps desired ball accelerations into target plate angles.
- An Inverse Kinematics (IK) block that translates target plate angles into specific servo joint angles.
- An Actuator Dynamics filter that enforces the physical velocity saturation constraints.

- A Forward Kinematics (FK) block using Homogeneous Transformation Matrices (HTM) to close the physical state loop.

This report focuses on the modeling, simulation, and validation of a PID-controlled robotic-arm-actuated ball balancing system under actuator velocity constraints. The subsequent sections detail the physical derivation, kinematic mapping, control architecture, time-domain tracking performance, actuator saturation behavior, and empirical frequency-domain characterization of the closed-loop system.

2 System Modeling & Kinematics

In this section, the comprehensive mathematical foundation of the 2D ball-and-plate system is established. Following the sequential execution of the simulation loop, we first derive the nonlinear Equations of Motion (EoM) governing the ball's physics, followed by the geometric transformations of the robotic mechanism.

2.1 Ball on Plate Dynamics (Equations of Motion)

The plant skeleton consists of a solid spherical ball rolling without slipping on a flat, square plate with side length $2a$. To establish a rigorous mathematical framework, the coordinate system is defined with its origin $(0,0)$ placed precisely at the geometric center of the plate, implying a bounded workspace of $x, y \in [-a, a]$.

The orientation of the plate is actively manipulated in two rotational degrees of freedom. The primary actuation is provided by two independent 2R planar robotic arms connected to the midpoints of the orthogonal edges:

- The end-effector of the first 2R planar arm is connected directly to the right edge of the plate at $(x = a, y = 0)$ to actuate the pitch angle ϕ_x, which is defined as positive in the counter-clockwise (CCW) direction. A positive increase in ϕ_x tilts the plate upward at this boundary, causing the gravitational component to accelerate the ball along the negative x-axis.
- The end-effector of the second 2R planar arm is connected to the adjacent orthogonal edge at $(x = 0, y = a)$ to control the roll angle ϕ_y, adhering to a symmetric kinematic and physical mapping along the y-axis frame.

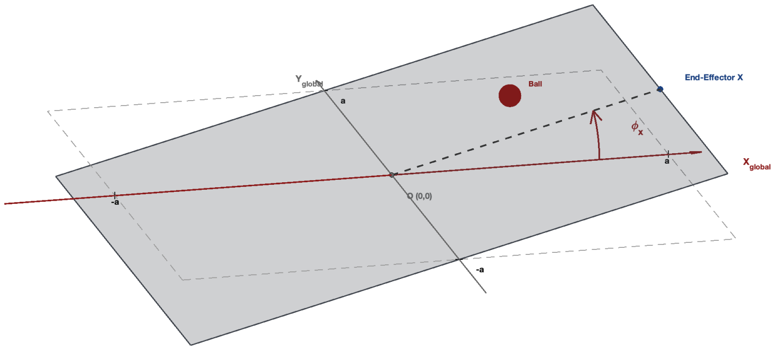
The nonlinear ball-on-plate dynamics used in this study are based on the Lagrangian formulation reported by Hadoune et al. [1], with the notation adapted to the present robotic-arm-actuated plate geometry. Substituting the solid sphere's moment of inertia $I_b = \frac{2}{5}mr_b^2$ into the effective inertial mass term yields the standard geometric scaling factor:

$$m + \frac{I_b}{r_b^2} = m + \frac{\frac{2}{5}mr_b^2}{r_b^2} = \frac{7}{5}m. \tag{1}$$

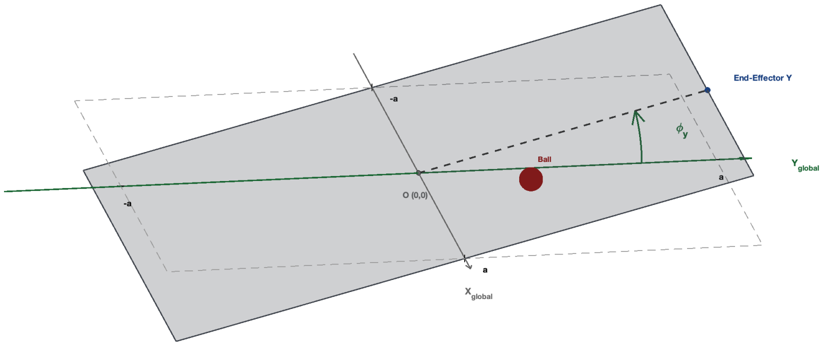
By isolating the linear accelerations $\ddot{x}$ and $\ddot{y}$ based on the exact kinematic implementation of the plant, we obtain the explicit nonlinear state equations deployed in the simulation:

$$\ddot{x} = \frac{5}{7}\left(x\dot{\phi}_x^2 + y\dot{\phi}_x\dot{\phi}_y - g\sin(\phi_x)\right) \tag{2}$$

$$\ddot{y} = \frac{5}{7}\left(y\dot{\phi}_y^2 + x\dot{\phi}_x\dot{\phi}_y - g\sin(\phi_y)\right) \tag{3}$$



(a) Tilt orientation along the x-axis (ϕ_x), showcasing the positive counter-clockwise rotation and the displacement of **End-Effector X**.



(b) Tilt orientation along the y-axis (ϕ_y), showcasing the corresponding orthogonal rotation and the displacement of **End-Effector Y**.

Figure 1: Geometric schematic of the ball balancing platform illustrating the square plate workspace boundaries ($2a \times 2a$), the central global coordinate origin $O(0,0)$ located at the pivot, the designated attachment locations for the 2R planar arms (**End-Effector X** and **End-Effector Y**), and the positive counter-clockwise (CCW) directional conventions for the plate rotations (ϕ_x, ϕ_y).

2.1.1 Analysis of Nonlinear Coupling Effects

The explicit equations reveal two critical control challenges that disqualify decoupled linear PID controllers from achieving optimal high-speed tracking:

1. **Direct Gravitational Non-linearity:** The primary control authority is dictated by the nonlinear trigonometric terms $\sin(\phi_x)$ and $\sin(\phi_y)$. Because the control input enters through a transcendental function, the system gain heavily scales with the operating tilt angle.
2. **Centrifugal and Coriolis Coupling:** The terms $x\dot{\phi}_x^2$, $y\dot{\phi}_y^2$, and $x\dot{\phi}_x\dot{\phi}_y$ represent the coupled cross-axis inertial forces. When a planar arm rotates rapidly to stabilize its primary axis, it inherently induces a severe, state-dependent disturbing acceleration on the orthogonal axis. This multi-input multi-output (MIMO) coupling requires a control strategy capable of dynamic decoupling.

2.2 Inverse Kinematics of Planar Robotic Arms

To command the plate to a desired orientation (ϕ_x, ϕ_y), the target plate angles must be mapped into the joint configurations (θ_1, θ_2) of the driving actuators. This inverse mapping is performed independently for both orthogonal 2R planar robotic arms: the pitch arm along the x-axis and the roll arm along the y-axis.

2.2.1 Stage 1: Angle to Base-Frame Constraint Space

The global origin $(0, 0)$ is anchored at the central pivot of the square plate $(2a \times 2a)$, so the radial distance from the pivot to the edge midpoint connected to the arm end-effector must remain equal to a. This imposes a rigid **geometric constraint**: the end-effector must trace a circular arc of radius a. Deviating from this path would generate severe **tension** or **compression** forces, potentially causing structural binding or joint failure.

For a positive counter-clockwise (CCW) plate angle ϕ, the target spatial coordinates in the global frame are:

$$x_{2,global} = a\cos(\phi) \tag{4}$$

$$y_{2,global} = a\sin(\phi) \tag{5}$$

The actuator base frame is mounted a vertical distance L below the global origin and a horizontal offset d from the actuation axis. Transforming the global target into the actuator base frame gives:

$$x_{2,base} = x_{2,global} - d = a\cos(\phi) - d \tag{6}$$

$$y_{2,base} = y_{2,global} + L = L + a\sin(\phi) \tag{7}$$

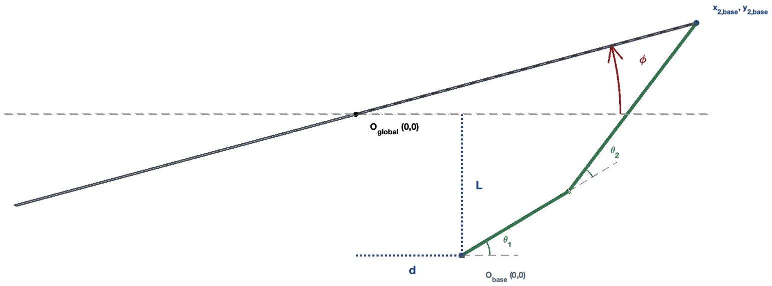


Figure 2: Geometric schematic of Stage 1 inverse mapping, illustrating the spatial relationship between the global origin $(0, 0)$ at the central pivot, the actuator base frame offset (d, L), and the target end-effector coordinates $(x_{2,base}, y_{2,base})$ as a function of the plate tilt angle ϕ.

2.2.2 Stage 2: Planar 2R Joint Extraction

Using the local spatial coordinates $(x_{2,base}, y_{2,base})$ isolated in Stage 1, the explicit link configurations (l_1, l_2) of the 2R mechanism are resolved. To facilitate the trigonometric decomposition, the radial distance r from the actuator base origin to the end-effector and the associated base orientation angle α are defined as:

$$r = \sqrt{x_{2,base}^2 + y_{2,base}^2} \tag{8}$$

$$\alpha = \operatorname{atan2}(y_{2,base}, x_{2,base}) \tag{9}$$

Next, applying the Law of Cosines to the structural triangle formed by r, l_1, and l_2 yields the interior elbow joint angle θ_2:

$$D = \frac{x_{2,base}^2 + y_{2,base}^2 - l_1^2 - l_2^2}{2l_1 l_2}, \quad D \in [-1, 1] \tag{10}$$

$$\theta_2 = \arccos(D) \tag{11}$$

The primary shoulder joint angle θ_1, which dictates the angular position of the first link relative to the horizontal axis of the actuator frame, is subsequently extracted. By utilizing the pre-calculated base target angle α and accounting for the geometric offset induced by the elbow flexion, θ_1 is rigorously computed as:

$$\theta_1 = \operatorname{atan2}(y_{2,base}, x_{2,base}) - \operatorname{atan2}\left(l_2 \sin(\theta_2),\ l_1 + l_2 \cos(\theta_2)\right) \tag{12}$$

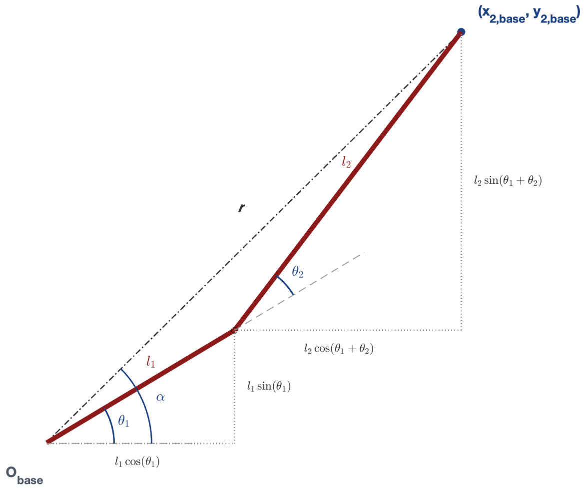


Figure 3: Kinematic diagram of the 2R planar robotic arm viewed from the orthogonal axis, defining the link lengths (l_1, l_2), the baseline primary joint angle θ_1, and the interior elbow angle θ_2 resolved via the Law of Cosines using local base coordinates.

2.2.3 Home Configuration and Relative Actuator Commands

In the physical system, the servo motors operate on relative command profiles wrapped around a flat nominal equilibrium state ($\phi = 0$). To optimize the structural efficiency of the platform, a specific mechanical constraint is enforced: when the plate is perfectly flat, the primary link of each arm must rest perfectly horizontal ($\theta_{1,home} = 0$).

To systematically guarantee this equilibrium condition, the second link length l_2 is selected freely based on physical space constraints, while the first link length l_1 is computed automatically using the nominal layout geometry:

$$l_1 = (a - d) - l_2 \cos\left(\arcsin\left(\frac{L}{l_2}\right)\right) \tag{13}$$

Evaluating the inverse kinematics pipeline at $\phi = 0$ with this structured link design yields the absolute baseline values $\theta_{1,home}$ and $\theta_{2,home}$. The final relative command signals $(\theta_{1,cmd}, \theta_{2,cmd})$ dispatched to the physical actuator hardware are normalized via:

$$\theta_{1,cmd} = \theta_{1,abs} - \theta_{1,home} \tag{14}$$

$$\theta_{2,cmd} = \theta_{2,abs} - \theta_{2,home} \tag{15}$$

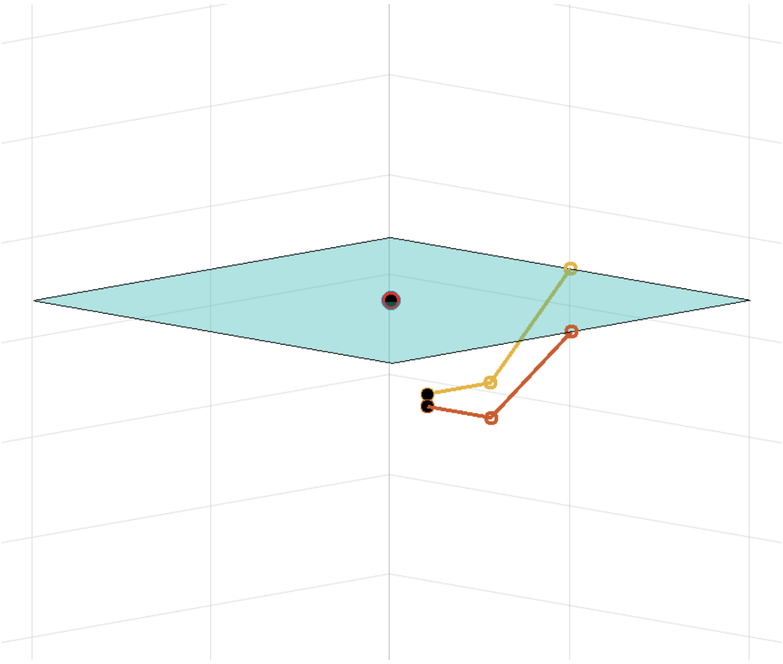


Figure 4: The ball balancing system in its absolute home configuration where plate tilt angles are $\phi_x = \phi_y = 0°$, and the first link joint angles are $\theta_1, x = \theta_1, y = 0°$, resulting in the ball resting at the central origin.

2.3 Forward Kinematics of the Planar Arms

The forward kinematics loop executes the inverse mapping of the kinematic pipeline: given the actual absolute joint angles of the planar robotic arms, it reconstructs the physical tilt angles (ϕ_x, ϕ_y) of the plate. This mathematical synthesis is evaluated independently for the pitch arm (utilizing joint states θ_{1x}, θ_{2x} to find ϕ_x) and the roll arm (utilizing joint states θ_{1y}, θ_{2y} to find ϕ_y). The loop is structured into a two-stage forward formulation.

To establish a systematic geometric foundation for the forward kinematics, the Denavit-Hartenberg (DH) convention is utilized. Choosing the local Z-axes as the axes of rotation for both joints, the geometric parameters defining the link lengths (l_i), link twists (α_i), link offsets (d_i), and joint angles (θ_i) are summarized in Table 1.

Table 1: Denavit-Hartenberg (DH) Parameters for the 2R Planar Robotic Arm

Link (i)	θ_i	α_i	l_i	d_i
1	θ_1	0	l_1	0
2	θ_2	0	l_2	0

2.3.1 Stage 1: Homogeneous Transformation Matrices (HTM) Derivation

To analytically isolate the end-effector position relative to the local actuator base, rigid reference frames are attached to each manipulator link. Frame 0 defines the stationary robot base frame, located at a horizontal offset d and a vertical offset $-L$ from the global central pivot. Frame 1 is locked to the elbow joint, and Frame 2 is embedded at the end-effector contact interface.

Assuming pure planar rotations about the local Z-axis, the rigid transformation matrix from the base frame (Frame 0) to the elbow joint (Frame 1) after a primary rotation θ_1 and a link translation l_1 is given by:

$$T_0^1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & l_1\cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & l_1\sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

Similarly, the transformation from the elbow joint to the end-effector contact point (Frame 2) via a relative rotation θ_2 and link translation l_2 is defined as:

$$T_1^2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & l_2\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & l_2\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

The total forward kinematic transformation matrix T_0^2 mapping the local actuator base directly to the plate contact interface is evaluated via post-multiplication of the independent link matrices:

$$T_0^2 = T_0^1 \cdot T_1^2 = \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 & l_1\cos(\theta_1)+l_2\cos(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & l_1\sin(\theta_1)+l_2\sin(\theta_1+\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

Extracting the translation elements from the fourth column of T_0^2 yields the explicit coordinates of the end-effector $(x_{2,base}, y_{2,base})$ mapped strictly within the local actuator's base frame:

$$x_{2,base} = l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2) \tag{19}$$

$$y_{2,base} = l_1\sin(\theta_1) + l_2\sin(\theta_1+\theta_2) \tag{20}$$

2.3.2 Stage 2: Spatial Re-Projection and Plate Angle Extraction

To close the physical feedback loop, the local coordinates resolved in Stage 1 must be projected back into the global coordinate frame centered at the fixed plate pivot. Accounting for the physical mount configuration where the actuator base sits at a horizontal offset d and a vertical drop of L relative to the central pivot, the global spatial position $(x_{2,global}, y_{2,global})$ of the edge contact point is mapped via linear translation:

$$x_{2,global} = d + x_{2,base} = d + l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \tag{21}$$

$$y_{2,global} = y_{2,base} = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \tag{22}$$

Finally, evaluating the geometric vector from the fixed central pivot $(0, L)$ to the calculated global contact point $(x_{2,global}, y_{2,global})$, the actual instantaneous plate tilt angle ϕ is extracted using the robust four-quadrant inverse tangent function:

$$\phi = \text{atan2}(y_{2,global} - L,\ x_{2,global}) \tag{23}$$

This calculation is performed simultaneously and separately across both control channels, yielding ϕ_x using $(\theta_{1x}, \theta_{2x})$ and ϕ_y using $(\theta_{1y}, \theta_{2y})$. These computed angles provide the real-time geometric states required to evaluate the plant's underlying equations of motion.

2.3.3 Actuator Dynamics and Rate Limiting

In a realistic physical deployment, the commercial servo motors driving the 2R planar arms do not exhibit instantaneous step responses. Due to internal rotor inertia, viscous friction, and low-level control loop latencies, the real joint positions (θ_1, θ_2) lag behind the analytical inverse kinematics commands $(\theta_{1,cmd}, \theta_{2,cmd})$.

To model this physical limitation and ensure smooth control trajectories, the actuator behavior is approximated as a continuous first-order low-pass dynamical system governed by a characteristic servo time constant τ_{servo}:

$$\dot{\theta}_{1,des} = \frac{\theta_{1,cmd}(t) - \theta_1(t)}{\tau_{servo}} \tag{24}$$

$$\dot{\theta}_{2,des} = \frac{\theta_{2,cmd}(t) - \theta_2(t)}{\tau_{servo}} \tag{25}$$

Furthermore, physical actuators are strictly bounded by maximum angular velocity constraints imposed by their internal DC motor windings and gear ratio ratings. To prevent unfeasible high-speed demands that induce severe wind-up or numerical instability during aggressive trajectory tracking, a strict saturation wrapper (rate limiter) is implemented:

$$\dot{\theta}_1(t) = \max\left(-\Omega_{max},\ \min\left(\dot{\theta}_{1,des},\ \Omega_{max}\right)\right) \tag{26}$$

$$\dot{\theta}_2(t) = \max\left(-\Omega_{max},\ \min\left(\dot{\theta}_{2,des},\ \Omega_{max}\right)\right) \tag{27}$$

where Ω_{max} denotes the maximum allowable angular velocity of the servo hardware. The actual absolute joint angles at the next discrete iteration are subsequently resolved via numerical integration of the saturated rates: $\theta(t + \Delta t) = \theta(t) + \dot{\theta}(t)\Delta t$.

3 Control System Design

The primary objective of the control system is to stabilize the open-loop unstable ball-and-plate plant and ensure high-fidelity tracking of complex trajectories under strict actuator velocity constraints. To overcome the highly coupled, nonlinear equations of motion (EoM) without losing physical intuition, a hybrid PID-based control architecture is developed. This framework separates outer-loop trajectory tracking from inner-loop mechanical constraints through a numerical feedback linearization bridge.

3.1 Control Architecture and Sequential Signal Cascade

The simulation and control loop is executed sequentially in discrete time with a fundamental plant integration step of $dt = 0.001\,\mathrm{s}$, while the outer-loop PID trajectory controller operates at a downsampled control sampling period of $T_{s,control} = 0.03\,\mathrm{s}$ to mimic realistic industrial controller frequencies.

Rather than treating the control law, optimization routine, and kinematics as isolated blocks, the entire architecture operates as a continuous, deterministic signal cascade. Each computational stage ingests the outputs of the preceding block and synthesizes the exact inputs required by the next. The operational flow of a single control update cycle follows a strict mathematical pipeline.

The chronological execution of this signal cascade is structured as follows:

1. **State Feedback and Error Evaluation:** At the boundary of each $T_{s,control}$ interval, the actual ball states $(x, y, \dot{x}, \dot{y})$ are sampled from the plant. These are compared directly against the reference trajectory states to evaluate the tracking error vectors $(e, \dot{e})$ and accumulate the numerical integrals (I_x, I_y).

2. **Virtual Control Synthesis:** The error states are processed by the outer-loop PID controller. The controller synthesizes a pair of unconstrained virtual desired accelerations $(\ddot{x}_{des}, \ddot{y}_{des})$, which represent the ideal linear acceleration commands required to correct the ball's trajectory.

3. **Numerical Feedback Linearization Layer:** Because the physical inputs to the mechanism are plate angles rather than direct linear forces, the virtual accelerations $(\ddot{x}_{des}, \ddot{y}_{des})$ are routed into a nonlinear root-finding routine. Utilizing MATLAB's `fsolve`, the algorithm numerically inverts the coupled plant physics to solve for the target commanded plate angles $(\phi_{cmd,x}, \phi_{cmd,y})$ that drive the localized dynamic residuals to zero.

4. **Inverse Kinematics Mapping:** The resolved plate angle commands $(\phi_{cmd,x}, \phi_{cmd,y})$ are passed into the multi-stage inverse kinematics pipeline. The algorithm maps these orientations into relative joint space targets, yielding the absolute desired servo commands $(\theta_{1,cmd}, \theta_{2,cmd})$ for both orthogonal driving arms.

5. **Actuator Dynamics and Rate Limiting:** The ideal joint commands ($\theta_{1,cmd}, \theta_{2,cmd}$) are dispatched to the physical servo hardware model. Here, they pass through a first-order low-pass lag filter governed by the time constant τ_{servo}, and their derivative rates are strictly bounded by the maximum physical motor velocity limit (Ω_{max}). This stage outputs the actual, physically achievable joint positions ($\theta_{1,actual}, \theta_{2,actual}$).

6. **Forward Kinematics Re-Projection:** The bounded joint positions are mapped through the Homogeneous Transformation Matrices (HTM) of the forward kinematics loop. This step calculates the true, physically realized tilt angles ($\phi_{actual,x}, \phi_{actual,y}$) and velocities ($\dot{\phi}_{actual}$) of the plate platform.

7. **Nonlinear Plant Integration:** The realized plate states and ball positions are fed directly into the core nonlinear equations of motion. The system evaluates the true physical ball accelerations ($\ddot{x}_{actual}, \ddot{y}_{actual}$) and integrates them forward across the discrete dt sub-steps, yielding the updated plant state vector $X(k+1)$ for the next feedback cycle.

3.2 Numerical Feedback Linearization via Nonlinear Root-Finding

The main difficulty in controlling the ball-on-plate system is that the PID controller naturally generates desired ball accelerations, whereas the physical commands of the mechanism are plate angles and actuator joint angles. Moreover, the plant dynamics include centripetal terms, cross-axis coupling terms, and trigonometric nonlinearities. Instead of deriving an analytical feedback linearization law through Lie derivatives, this study uses a numerical inverse mapping layer based on MATLAB's `fsolve`.

At every control update interval $T_{s,control}$, the PID controller computes the virtual acceleration commands

$$\ddot{x}_{des}, \qquad \ddot{y}_{des}. \tag{28}$$

The objective of the numerical inverse mapping layer is to find the commanded plate angles

$$z = \begin{bmatrix} \phi_{cmd,x} \\ \phi_{cmd,y} \end{bmatrix} \tag{29}$$

such that the nonlinear acceleration model reproduces these desired accelerations.

Inside the root-finding routine, the commanded plate angular velocities are approximated using the current solution guess and the previously realized plate angles:

$$\dot{\phi}_{cmd,x} = \frac{\phi_{cmd,x} - \phi_{x,prev}}{T_{s,control}}, \qquad \dot{\phi}_{cmd,y} = \frac{\phi_{cmd,y} - \phi_{y,prev}}{T_{s,control}}. \tag{30}$$

Using these quantities, the model-predicted accelerations associated with the current command guess are computed as

$$\ddot{x}_{model} = \frac{5}{7}\left(x(k-1)\dot{\phi}_{cmd,x}^2 + y(k-1)\dot{\phi}_{cmd,x}\dot{\phi}_{cmd,y} - g\sin(\phi_{cmd,x})\right), \tag{31}$$

$$\ddot{y}_{model} = \frac{5}{7}\left(y(k-1)\dot{\phi}_{cmd,y}^2 + x(k-1)\dot{\phi}_{cmd,x}\dot{\phi}_{cmd,y} - g\sin(\phi_{cmd,y})\right). \tag{32}$$

The nonlinear residual vector is then defined as

$$F(z) = \begin{bmatrix} \ddot{x}_{model} - \ddot{x}_{des} \\ \ddot{y}_{model} - \ddot{y}_{des} \end{bmatrix}. \tag{33}$$

The numerical solver searches for the root

$$z^* = \begin{bmatrix} \phi_{cmd,x}^* \\ \phi_{cmd,y}^* \end{bmatrix} \tag{34}$$

that minimizes the residual norm:

$$\min_z \|F(z)\|^2 \to 0. \tag{35}$$

When the solver converges, the resulting commanded plate angles locally satisfy

$$\ddot{x}_{model} \approx \ddot{x}_{des}, \qquad \ddot{y}_{model} \approx \ddot{y}_{des}. \tag{36}$$

Therefore, from the perspective of the outer PID controller, the nonlinear coupled plant is approximately transformed into two decoupled double-integrator-like channels. This allows the PID controller to operate in the intuitive acceleration-command domain, while the numerical inverse mapping layer handles the nonlinear conversion from virtual acceleration commands to plate angle commands.

To validate this numerical inverse mapping step, the residual norm $\|F(z^*)\|$ was recorded at every control update. Figure 5 shows that the residual remains close to numerical precision throughout the trajectory, indicating that `fsolve` successfully finds plate angle commands that reproduce the desired virtual accelerations within the nonlinear model.

It should be emphasized that this validation verifies only the numerical feedback linearization step. It does not imply that the final actuator-limited plant acceleration is always equal to the desired acceleration, since the commanded plate angles are subsequently passed through inverse kinematics, servo dynamics, rate limiting, and forward kinematics.

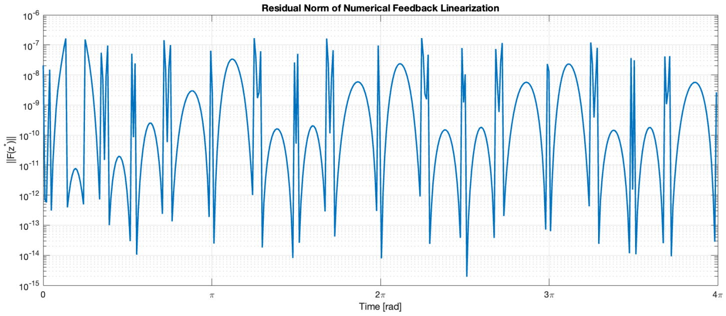


Figure 5: Residual norm of the numerical feedback linearization layer. The small values of $\|F(z^*)\|$ indicate that the commanded plate angles obtained from `fsolve` reproduce the desired virtual acceleration commands before actuator dynamics and rate limiting are applied.

3.3 PID Controller

Benefiting from the decoupled double-integrator plant model established by the numerical feedback linearization layer, independent Parallel PID controllers are designed for the x and y trajectory channels. The controllers utilize full-state tracking error feedback, where the position error $e(k)$ and velocity error $\dot{e}(k)$ are evaluated at each control step against the reference trajectory inputs:

$$e_x(k) = x(k-1) - x_{ref}(k), \quad e_y(k) = y(k-1) - y_{ref}(k) \tag{37}$$

$$\dot{e}_x(k) = \dot{x}(k-1) - \dot{x}_{ref}(k), \quad \dot{e}_y(k) = \dot{y}(k-1) - \dot{y}_{ref}(k) \tag{38}$$

To eliminate steady-state tracking errors induced by unmodeled structural friction or slight geometric asymmetries, discrete numerical integration is performed on the position errors using the control sampling period $T_{s,control}$:

$$I_x(k) = I_x(k-1) + e_x(k) \cdot T_{s,control} \tag{39}$$

$$I_y(k) = I_y(k-1) + e_y(k) \cdot T_{s,control} \tag{40}$$

3.3.1 Actuator-Aware Integral Anti-Windup Clamping

A notorious vulnerability of standard PID controllers in systems with physical speed and position boundaries is *Integral Windup*. When the robotic arm actuators hit their velocity rate limits (Ω_{max}), the physical plant can no longer accelerate fast enough to track the aggressive control command. Consequently, the tracking error remains non-zero, causing the mathematical integral term (I) to accumulate unbounded values. This accumulation delays the controller's recovery, inducing massive overshoot and potential system destabilization.

To guarantee physical robustness, a rigid conditional integration clamping mechanism (Anti-windup) is hardcoded into the discrete simulation loop.

The accumulated error integrals are strictly constrained by a saturation ceiling ($I_{max} = 0.5$):

$$I_x(k) = \max\left(-I_{max},\, \min\left(I_x(k),\, I_{max}\right)\right) \tag{41}$$

$$I_y(k) = \max\left(-I_{max},\, \min\left(I_y(k),\, I_{max}\right)\right) \tag{42}$$

3.3.2 Synthesized PID Control Law

Finally, the virtual desired accelerations ($\ddot{x}_{des}, \ddot{y}_{des}$) fed directly into the `fsolve` block are synthesized by combining the feedforward reference accelerations with the proportional, derivative, and clamped integral feedback terms:

$$\ddot{x}_{des}(k) = \ddot{x}_{ref}(k) - K_{d,x}\dot{e}_x(k) - K_{p,x}e_x(k) - K_{i,x}I_x(k) \tag{43}$$

$$\ddot{y}_{des}(k) = \ddot{y}_{ref}(k) - K_{d,y}\dot{e}_y(k) - K_{p,y}e_y(k) - K_{i,y}I_y(k) \tag{44}$$

Where the controller parameters are empirically tuned to maintain stability under nominal conditions.

4 Methodology and Test Scenario

To evaluate the PID-controlled system under realistic physical constraints, a single time-varying trajectory tracking scenario was established. The reference path incorporates non-uniform frequencies across the axes: the x-axis follows a sinusoidal trajectory at a fundamental frequency of ω, while the y-axis operates at 2ω. This structure increases the dynamic complexity of the motion by forcing the controller to handle different inertial demands along the two axes within the same test case.

The PID controller was tuned to achieve sufficiently accurate trajectory tracking while respecting the actuator velocity limits and maintaining stable closed-loop behavior under the nonlinear plant dynamics. Rather than pursuing an unrealistically aggressive tuning, the selected gains were chosen to provide a practical balance between tracking accuracy, actuator activity, and saturation avoidance.

The resulting closed-loop system was evaluated from two complementary perspectives:

- **Time-domain performance:** The transient and steady-state tracking behavior, position errors, actuator commands, joint velocity effort, and saturation characteristics are analyzed directly from simulation results.
- **Frequency-domain characterization:** The closed-loop tracking response, normalized tracking error behavior, and disturbance-to-error sensitivity are evaluated using empirical frequency-response analysis under small-amplitude unsaturated excitation.

It should be noted that the PID gains used throughout the study were obtained from the time-domain tracking scenario with the non-zero initial condition defined for the main trajectory test. The same finalized gains were then used without retuning in the frequency-domain analysis. However, for the frequency-response tests, the initial conditions were set to zero in order to isolate the steady-state sinusoidal response of the closed-loop system and prevent the non-zero initial impulse from contaminating the empirical frequency-response measurements.

By combining time-domain trajectory tracking with empirical frequency-domain characterization, this methodology evaluates not only whether the PID controller can track the desired nonlinear trajectory, but also how the closed-loop system behaves under increasing reference frequency and small external acceleration disturbances.

5 Results and Discussion

In this section, the simulation results of the tuned PID-controlled system are presented for a single time-varying trajectory tracking scenario. The nonlinear coupled plant model, robotic-arm actuation structure, numerical feedback linearization layer, actuator dynamics, and velocity rate limits are all included in the simulation. The actuator velocity rate limit was restricted to a stringent threshold of 60°/s (1.047 rad/s).

The closed-loop system was evaluated from two complementary perspectives. First, the time-domain behavior was analyzed in terms of trajectory tracking, axis-specific errors, actuator effort, and saturation characteristics. Second, the frequency-domain behavior was investigated through empirical frequency-response analysis, focusing on tracking behavior, normalized error dynamics, and disturbance-to-error sensitivity.

5.1 Test Scenario

In the selected tracking scenario, the PID controller was tuned to achieve accurate trajectory tracking under strict physical constraints. Instead of zero initial states, a specific non-zero initial condition (IC) was introduced where the ball starts at the origin with a high initial horizontal velocity:

$$x(0) = 0 \text{ m}, \quad y(0) = 0 \text{ m}, \quad \dot{x}(0) = 0.3 \text{ m/s}, \quad \dot{y}(0) = 0 \text{ m/s}. \tag{45}$$

This initial impulse forces the controller to immediately counteract a high-momentum state while operating under tight actuator rate limits.

The finalized PID tuning parameters are documented in Table 2. The selected gains were chosen to provide a practical balance between tracking accuracy, actuator activity, and saturation avoidance. With this tuning, the system achieved a total trajectory tracking error of Total RMS = 0.0308 m.

Table 2: Finalized PID Controller Parameters and Global Tracking Metric

Controller	**X-Axis Parameters**	**Y-Axis Parameters**
PID	$K_{px} = 10.00$, $K_{dx} = 4.00$, $K_{ix} = 0.02$	$K_{py} = 5.50$, $K_{dy} = 1.95$, $K_{iy} = 0.02$

5.2 Time-Domain Analysis

The axis-specific error breakdown, joint-level control efforts, and exact motor saturation durations are detailed in Table 3.

Table 3: Time-Domain Axis-Specific Errors, Control Efforts, and Saturation Metrics

Metric Description	**PID Controller**	**Interpretation**
RMS Tracking Error x	0.001529 m	The lower-frequency x-axis trajectory is tracked with very small RMS error.
RMS Tracking Error y	0.011679 m	The higher-frequency y-axis trajectory produces a larger tracking error due to its more demanding reference motion.
X-Axis Motor 1 (Shoulder) Effort	0.446932	Integrated squared joint velocity effort for the first motor of the x-axis arm.
X-Axis Motor 2 (Elbow) Effort	0.444580	Integrated squared joint velocity effort for the second motor of the x-axis arm.
Y-Axis Motor 1 (Shoulder) Effort	4.569023	Higher effort is required on the y-axis because the reference trajectory operates at twice the fundamental frequency.
Y-Axis Motor 2 (Elbow) Effort	5.123409	The second y-axis motor exhibits the largest velocity effort due to the more aggressive high-frequency tracking demand.
Motor Saturation (% of Time)		
X-Axis Motor 1 Saturation	0.92% (115 steps)	The x-axis shoulder motor rarely reaches the imposed rate limit.
X-Axis Motor 2 Saturation	0.87% (109 steps)	The x-axis elbow motor also remains mostly within the feasible velocity range.
Y-Axis Motor 1 Saturation	7.06% (887 steps)	The higher-frequency y-axis command causes more frequent rate saturation.
Y-Axis Motor 2 Saturation	11.70% (1470 steps)	The second y-axis motor is the most constrained actuator in the tested scenario.

5.2.1 Time-Domain Performance Evaluation

As documented in Table 3, the PID-controlled system successfully tracks the nonlinear infinity-shaped trajectory while respecting the imposed actuator velocity constraints. The x-axis achieves a very small RMS tracking error of $\text{RMS}_x = 0.001529$ m, while the y-axis exhibits a larger error of $\text{RMS}_y = 0.011679$ m. This difference is expected because the reference trajectory was intentionally designed with unequal axis frequencies: the x-axis follows a sinusoid at ω, while the y-axis follows a sinusoid at 2ω. Therefore, the y-axis must respond to faster position, velocity, and acceleration variations.

The actuator effort metrics also reflect this asymmetric reference demand. The x-axis motors require relatively low joint velocity effort, with values of 0.446932 and 0.444580. In contrast, the y-axis motors require substantially higher effort, with values of 4.569023 and 5.123409. This indicates that the high-frequency component of the trajectory places a significantly greater burden on the corresponding robotic arm.

The saturation metrics further confirm this trend. The x-axis motors remain mostly within the feasible rate range, saturating for less than 1% of the simulation time. However, the y-axis motors enter saturation more frequently, especially the second motor, which reaches the rate limit for 11.70% of the simulation time. Despite this, the closed-loop system remains stable and continues to track the reference trajectory with acceptable accuracy. This demonstrates that the numerical feedback linearization layer, inverse kinematics mapping, and PID controller can operate effectively even when the actuator dynamics introduce realistic velocity constraints.

5.2.2 Time-Domain Visual Simulation Results

The spatial trajectory, internal state timelines, and actuator behaviors analyzed in the time-domain performance evaluation are graphically illustrated below.

First, the macro-level geometric tracking profile on the plate surface is shown in Figure 6. Next, the time-domain state responses and the physical platform tilt angles are detailed in Figure 7. Finally, the servo-level actuator dynamics, mapping the reference commands directly against velocity rate saturation thresholds, are documented in Figure 8.

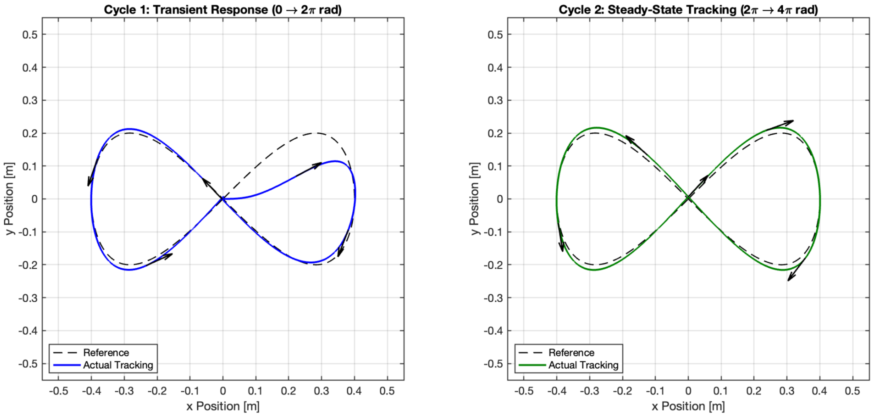


Figure 6: PID: 2D spatial ball path tracking for the infinity trajectory over $0 \rightarrow 4\pi$ cycles.

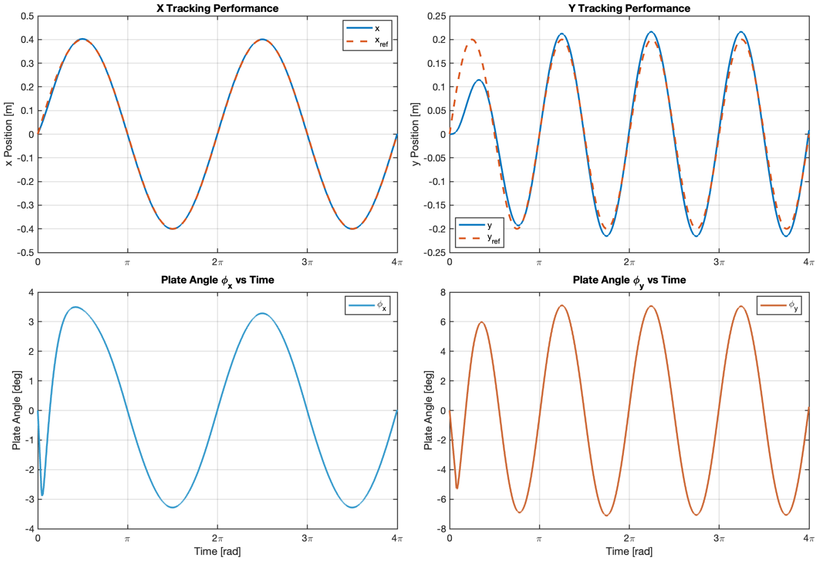


Figure 7: PID: Time-domain X/Y ball tracking profiles and corresponding plate tilt angles (ϕ_x, ϕ_y).

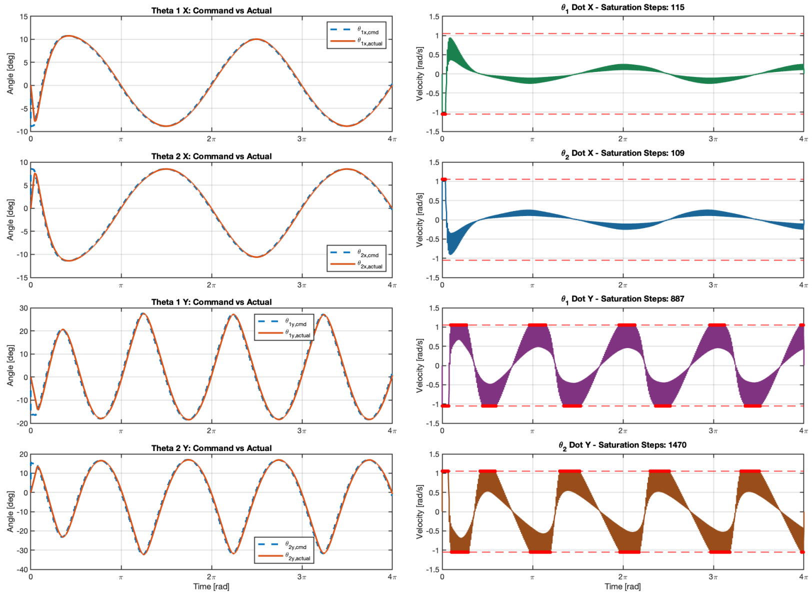


Figure 8: PID: Comprehensive servo actuator analysis mapping angular positions (command vs. actual) directly against velocity saturation profiles.

5.3 Frequency-Domain Analysis

In addition to the time-domain trajectory tracking evaluation, the PID-controlled system was analyzed in the frequency domain. The objective of this analysis was to characterize the closed-loop behavior under small-signal sinusoidal excitation while intentionally avoiding actuator saturation. Therefore, the finalized PID gains obtained from the time-domain tracking test were preserved, but the frequency-domain simulations were performed with zero initial conditions:

$$x(0) = 0 \text{ m}, \quad y(0) = 0 \text{ m}, \quad \dot{x}(0) = 0 \text{ m/s}, \quad \dot{y}(0) = 0 \text{ m/s}. \tag{46}$$

This choice prevents the non-zero initial velocity used in the time-domain tracking test from contaminating the empirical frequency-response estimates. Thus, the obtained frequency responses represent the closed-loop response to chirp reference or disturbance excitation rather than the transient recovery from an initial impulse.

Two types of empirical frequency-response tests were conducted. First, closed-loop tracking and tracking-error frequency responses were estimated by applying small-amplitude chirp reference signals independently along the x- and y-axes. Second, disturbance-to-error sensitivity responses were estimated by injecting sinusoidal acceleration disturbances into the plant dynamics while keeping the reference position equal to zero.

5.3.1 Empirical Closed-Loop Tracking and Error Frequency Response

For the closed-loop tracking analysis, the reference input was selected as a small-amplitude chirp signal with amplitude

$$A_{\text{chirp}} = 0.01 \text{ m}, \tag{47}$$

sweeping from 0.02 Hz to 0.8 Hz over the simulation duration. The chirp amplitude was intentionally kept small so that the system remained in the unsaturated operating region. Therefore, the resulting frequency-response estimates represent small-signal closed-loop behavior.

The empirical closed-loop tracking frequency-response functions were defined as

$$T_{cl,x}(j\omega) \approx \frac{X(j\omega)}{X_{\text{ref}}(j\omega)}, \qquad T_{cl,y}(j\omega) \approx \frac{Y(j\omega)}{Y_{\text{ref}}(j\omega)}. \tag{48}$$

The corresponding normalized tracking-error frequency-response functions were defined as

$$E_{x,\text{FRF}}(j\omega) \approx \frac{E_x(j\omega)}{X_{\text{ref}}(j\omega)}, \qquad E_{y,\text{FRF}}(j\omega) \approx \frac{E_y(j\omega)}{Y_{\text{ref}}(j\omega)}. \tag{49}$$

The closed-loop tracking and tracking-error frequency-response metrics are summarized in Table 4. No actuator saturation occurred during these chirp tracking tests, confirming that the empirical frequency-response estimates correspond to the unsaturated small-signal behavior of the closed-loop PID-controlled system.

Table 4: Empirical Closed-Loop Tracking and Tracking Error Frequency Response Metrics

Metric Description	**PID** x	**PID** y
Low-Frequency Tracking Gain	1.0227	1.0275
Maximum Tracking Gain	1.2009	1.1736
Peak Tracking Frequency	0.7042 Hz	0.4558 Hz
Low-Frequency Error Gain	0.0228	0.0276
Maximum Error Gain	0.2402	0.2631
Peak Error Frequency	0.7682 Hz	0.7682 Hz
Actuator Saturation	0%	0%

5.3.2 Empirical Disturbance-to-Error Sensitivity Frequency Response

The second frequency-domain test evaluated the disturbance rejection characteristics of the PID-controlled system. In this analysis, the reference trajectory was kept at zero:

$$x_{\text{ref}}(t) = 0, \qquad y_{\text{ref}}(t) = 0. \tag{50}$$

A chirp acceleration disturbance was then injected directly into the plant acceleration dynamics. The disturbance amplitude was selected as

$$D_{\text{acc}} = 0.02 \text{ m/s}^2, \tag{51}$$

with the same frequency sweep from 0.02 Hz to 0.8 Hz. The disturbance-to-error sensitivity functions were defined as

$$S_{d,x}(j\omega) \approx \frac{E_x(j\omega)}{D_{\text{acc},x}(j\omega)}, \qquad S_{d,y}(j\omega) \approx \frac{E_y(j\omega)}{D_{\text{acc},y}(j\omega)}. \tag{52}$$

Since the input is an acceleration disturbance in m/s^2 and the output is position error in meters, the unit of the disturbance sensitivity magnitude is approximately s^2.

The disturbance sensitivity metrics are summarized in Table 5. No actuator saturation occurred in any of the disturbance sensitivity tests, confirming that the results represent the unsaturated small-signal disturbance behavior of the closed-loop PID-controlled system.

Table 5: Empirical Disturbance-to-Error Sensitivity Frequency Response Metrics

Metric Description	**PID** x	**PID** y
Low-Frequency Sensitivity	0.101581 s^2	0.185601 s^2
Maximum Sensitivity	0.103373 s^2	0.258972 s^2
Peak Sensitivity Frequency	0.2479 Hz	0.3218 Hz
Maximum Position Error	0.00208 m	0.00521 m
Actuator Saturation	0%	0%

5.3.3 Frequency-Domain Performance Evaluation

The empirical closed-loop tracking, tracking-error, and disturbance-sensitivity frequency-response results provide additional insight into the behavior of the PID-controlled nonlinear system beyond the time-domain trajectory plots. The tracking and error FRF metrics are summarized in Table 4. Both axes exhibit near-unity low-frequency tracking gains, indicating that the PID-controlled system accurately follows slow reference variations.

The closed-loop tracking FRF shows that the x- and y-axis responses are similar at low frequency, with tracking gains of 1.0227 and 1.0275, respectively. However, the frequency-domain behavior is not perfectly identical because the two PID loops were tuned with different gains. The x-axis controller uses larger proportional and derivative gains, whereas the y-axis controller is tuned more conservatively to handle the higher-frequency trajectory component without excessive actuator saturation.

The maximum tracking gains are 1.2009 for the x-axis and 1.1736 for the y-axis. These values indicate mild gain amplification within the tested frequency range, but the response remains bounded and actuator saturation remains inactive during the chirp tests. The normalized tracking-error FRF further shows that low-frequency tracking errors remain small, with error gains of 0.0228 for the x-axis and 0.0276 for the y-axis. As the excitation frequency approaches the upper end of the tested range, the maximum error gains increase to 0.2402 and 0.2631, respectively, indicating the expected degradation in tracking accuracy at higher frequencies.

The disturbance-sensitivity results, summarized in Table 5, reveal that the x-axis rejects acceleration disturbances more effectively than the y-axis. The x-axis has a low-frequency disturbance sensitivity of 0.101581 s^2 and a maximum sensitivity of 0.103373 s^2, whereas the y-axis has a low-frequency sensitivity of 0.185601 s^2 and a maximum sensitivity of 0.258972 s^2. This difference is consistent with the axis-specific PID tuning and the more demanding reference dynamics assigned to the y-axis.

Overall, the frequency-domain analysis confirms that the finalized PID controller provides accurate low-frequency tracking, bounded tracking-error amplification, and stable disturbance rejection under small-amplitude unsaturated excitation. The results also show that the closed-loop behavior is axis-dependent, which is expected because the PID gains were tuned separately for the x- and y-axis dynamics.

5.3.4 Frequency-Domain Visual Simulation Results

The following figures provide the visual frequency-domain characterization of the finalized PID-controlled system. Figure 9 shows how accurately the ball follows small-amplitude chirp references, Figure 10 shows how the normalized tracking error grows with frequency, and Figure 11 shows how external acceleration disturbances are transferred into position error.

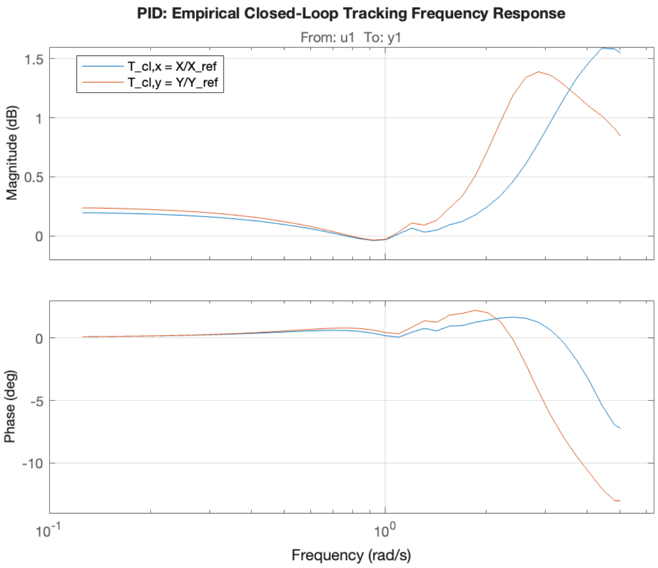


Figure 9: PID empirical closed-loop tracking frequency response for the x- and y-axis chirp reference tests.

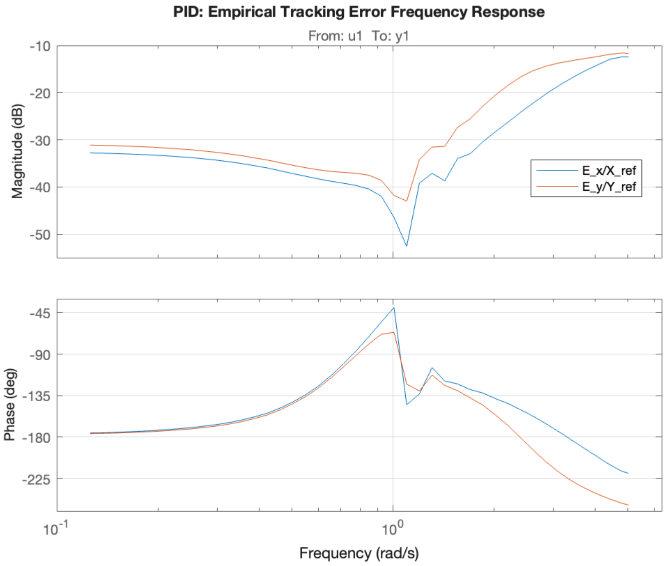


Figure 10: PID empirical tracking error frequency response for the x- and y-axis chirp reference tests.

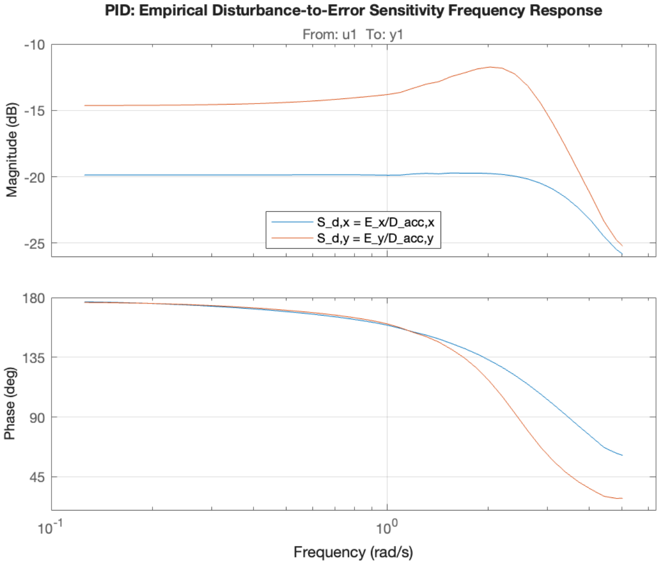


Figure 11: PID empirical disturbance-to-error sensitivity frequency response for x- and y-axis acceleration disturbance inputs.

6 Conclusion

This study presented a nonlinear coupled 2D ball balancing system actuated by two orthogonal 2R planar robotic arms. A PID-based control architecture was implemented in the virtual acceleration domain, and the nonlinear mapping from desired ball accelerations to physically realizable actuator commands was handled through numerical feedback linearization, inverse kinematics, actuator dynamics, and forward kinematics.

The simulation results demonstrated accurate trajectory tracking under strict servo velocity limits, with the x-axis achieving lower RMS error and lower actuator effort than the higher-frequency y-axis. Empirical frequency-domain tests further showed stable low-frequency tracking, bounded tracking-error amplification, and disturbance rejection under small-amplitude unsaturated excitation.

Overall, the project demonstrates that a classical PID controller can be effectively applied to a nonlinear coupled mechatronic system when combined with a suitable numerical inverse mapping layer and realistic actuator constraint modeling.

References

[1] O. Hadoune, M. Benouaret, A. Zeghida, and H. Saker, "Tracking control of a ball on plate system using pid controller and lead/lag compensator with a double loop feedback scheme," *European Journal of Science and Technology*, no. 28, pp. 375–380, 2021.