Digital Twin Co-Simulation: PID/PD Control of a Differential Drive Robot in VSI

Presented to: Dr. Mohamed Abdelsalam and Eng. Mohamed El-Leithy
Presented by: Patrick Ramez

1. Introduction

This project implements and evaluates a Digital Twin framework for a differential drive robot using the VSI platform. Three core components were developed:
- Simulator – models robot kinematics and injects disturbances/noise.
- Controller – implements PID/PD control laws for path following.
- Visualizer – plots trajectories, computes KPIs, and logs results.

The project investigates controller performance under various conditions (gain tuning, path curvature, noise/disturbances, and controller type ablations).

2. System Modeling

The robot follows differential drive kinematics:

$\dot{x} = v \cos(\theta)$, $\dot{y} = v \sin(\theta)$, $\dot{\theta} = \omega$

where:
- v is linear velocity command,
- ω is angular velocity command,
- (x, y, θ) is the robot state.

Discrete update with timestep Δt:
$x_{k+1} = x_k + v \cos(\theta_k) \Delta t$
$y_{k+1} = y_k + v \sin(\theta_k) \Delta t$
$\theta_{k+1} = \theta_k + \omega \Delta t$

3. Controller Design

The controller applies a PID-inspired lateral + heading correction:

Lateral error: $e_y = y_{ref} - y$
Heading error: $e_\theta = \theta_{desired} - \theta$

Control law:
$\omega = Kp_{lat} * e_y + Kd_{lat} * \dot{e}_y + Kp_{head} * e_\theta + f_{ff}$

Feedforward term $f_{ff} = 0.5 * dy/dx$.

Forward velocity is adjusted adaptively:
$v = v_{nom} * \max(0.3, 1 - |e_\theta|)$

4. VSI Gateway Architecture

The VSI platform connects three clients via CAN bus simulation:

- Simulator (Component 0): Publishes state (IDs 12–15), Receives control (IDs 16–17).
- Controller (Component 1): Subscribes to 12–14, Sends control on 16–17.
- Visualizer (Component 2): Subscribes to 12–13, Logs trajectory, plots vs reference, computes KPIs.

This modular design allowed seamless experimentation.

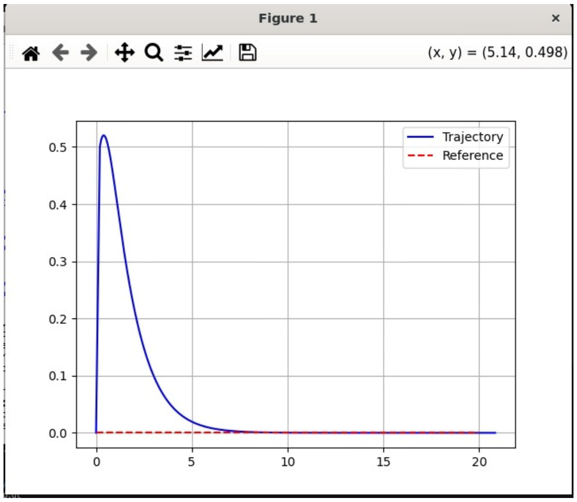
5. Experiments

E1 – PID Gain Sweep

Objective: Study effect of different gain sets on straight path tracking.

Gains tested:
- Set 1: Kp_lat=1.0, Kp_head=2.0, Kd_lat=0.05
- Set 2: Kp_lat=1.5, Kp_head=2.5, Kd_lat=0.1
- Set 3: Kp_lat=2.0, Kp_head=3.0, Kd_lat=0.2

KPIs Recorded: Overshoot, Settling Time, Steady-State Error.



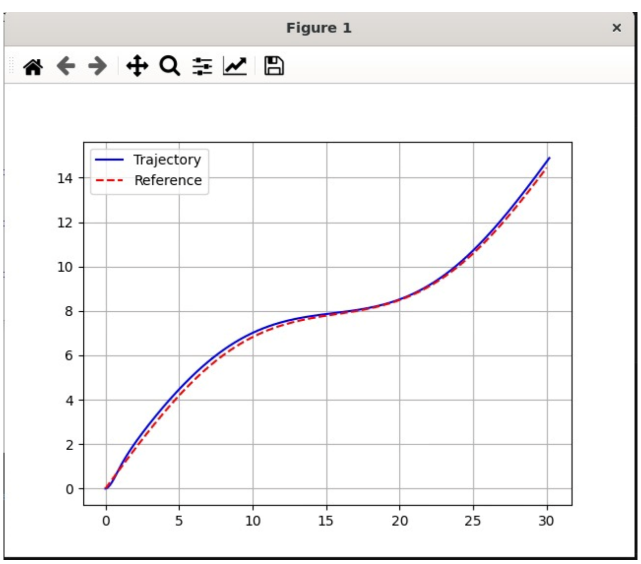
Results: Increasing gains reduced steady-state error but introduced overshoot and oscillations. Best trade-off observed in Set 2.

E2 – Curved Path Robustness

Objective: Evaluate best gain set (from E1) on a curved reference path.

Path: y = 0.5x + 2 sin(0.2x)
Controller: Set 2 gains

KPIs: Similar to E1.



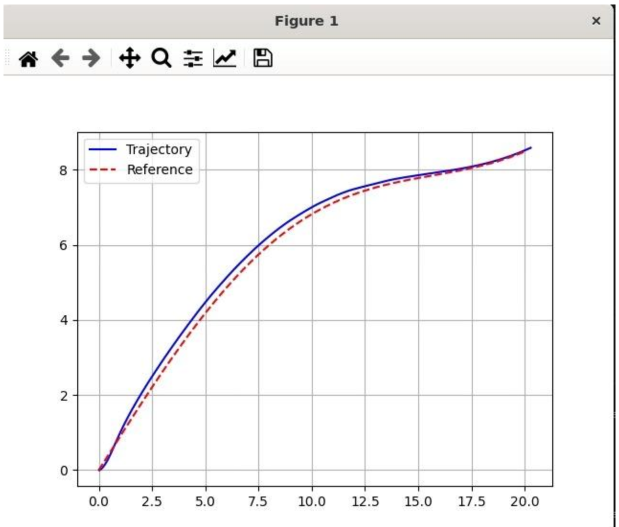
Results: Robot successfully tracked curvature with small lag. Error magnitude increased in high curvature regions.

E3 – Noise and Disturbance Rejection

Objective: Evaluate robustness under Gaussian noise and random disturbances.

Noise: $\sigma = 0.02$ on v, ω
Disturbance: ±0.15 rad/s impulse, 0.1s duration, probability 1% per step

KPIs: Overshoot, Settling Time, Steady-State Error.



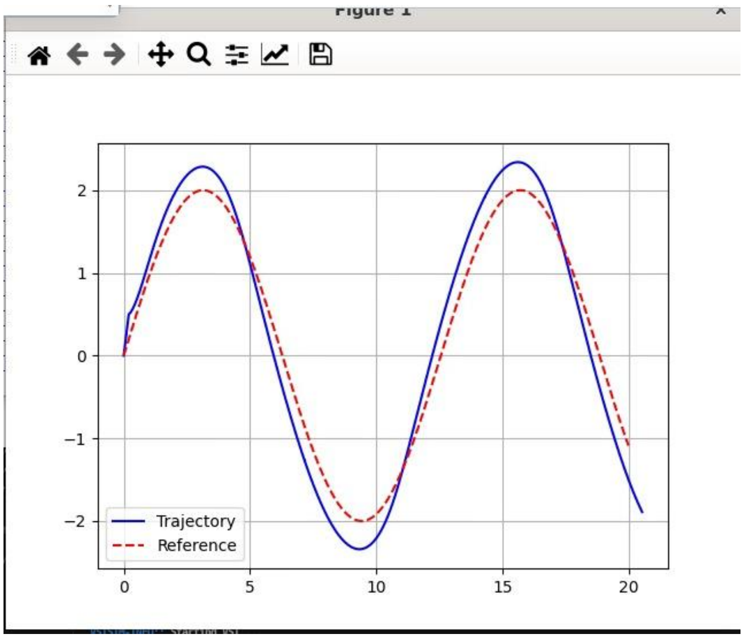
Results: Controller showed robustness; short disturbances caused momentary deviation but system recovered. Slight increase in steady-state error observed.

E4 – PD vs PID Ablation

Objective: Compare performance of PD (Ki=0) vs PID on a curved path with noise.

Conditions: $\sigma = 0.02$, disturbance = 0.15 rad/s
Controllers:
- PD: Ki=0
- PID: Ki=0.05

KPIs: Overshoot, Settling Time, Steady-State Error.



Results:
- PD: faster response, but nonzero steady-state error.
- PID: eliminated bias, improved final accuracy, but settling time slightly increased.

6. Results Summary

Experiment	Key Finding
E1 – Gain Sweep	Moderate gains give best trade-off between stability & accuracy
E2 – Curved Path	Controller tracks curvature, but larger error in sharp turns
E3 – Noise & Disturbance	Robust performance, small steady-state error increase
E4 – PD vs PID	PID eliminates steady-state bias, PD reacts faster but less accurate

7. Conclusion

This project successfully integrated a Digital Twin workflow with simulator, controller, and visualizer using VSI. The experiments demonstrated the trade-offs between gains, robustness under noise/disturbances, and controller variants.

Future work:
- Implement adaptive gain scheduling

- Test with nonlinear disturbances
- Extend to multi-robot coordination

8. Deliverables

- Source code: Simulator, Controller, Visualizer
- Plots: Figures from E1–E4 (insert above each experiment)
- Screencast: 2–3 minute demo of simulation runs and integration
- Report: This document