
KU LEUVEN

FACULTEIT
INGENIEURSWETENSCHAPPEN

**B-KUL-H04X3A: Control Theory**

**Team members:**

Lefebure Tiebert (r0887630)
Campaert Lukas (r0885501)

# Assignment 2: Velocity Control of the Cart

Professor:

Prof. Dr. Ir. Jan Swevers

Academic Year 2025-2026

## *Declaration of Originality*

*We hereby declare that this submitted draft is entirely our own, subject to feedback and support given us by the didactic team, and subject to lawful cooperation which was agreed with the same didactic team. Regarding this draft, we also declare that:*

1. *Note has been taken of the text on academic integrity* `https://eng.kuleuven.be/studeren/masterproef-en-papers/documenten/20161221-academischeintegriteit-okt2016.pdf`.
2. *No plagiarism has been committed as described on* `https://eng.kuleuven.be/studeren/masterproef-en-papers/plagiaat#Definitie:%20wat%20is%20plagiaat?`.
3. *All experiments, tests, measurements, …, have been performed as described in this draft, and no data or measurement results have been manipulated.*
4. *All sources employed in this draft – including internet sources – have been correctly referenced.*

# 1 Design of a velocity controller

## 1.1 Controller choice

The velocity loop must remove steady-state error for a constant command, which we quantify as keeping the residual tracking error below the encoder noise floor once the transient fades. Meeting this requirement forces the closed loop to be at least type 1, while the cart still has to settle in roughly a second with under $\sim$10% overshoot to stay within mechanical limits. Those timing targets map to a phase margin near 55° and a bandwidth that fits inside the 100 Hz sampling budget.

A proportional-integral (PI) compensator satisfies these constraints with minimal design effort. Its transfer function is

$$D(s) = \frac{K}{s}\left(s + \frac{1}{T_I}\right),$$

where the integral term lifts the system type and guarantees infinite gain at DC, while the proportional term fixes the crossover. With only two tunable constants the PI law is easy to calibrate on the embedded controller and avoids the noise-sensitive derivative leg of a full PID.

Its frequency response preserves the needed physics: below the corner $\omega = 1/T_I$ the magnitude behaves like $K/\omega$, so constant disturbances are strongly rejected. The same integral branch introduces negative phase, therefore $1/T_I$ is kept roughly one decade below the crossover to protect the phase margin while still delivering robust steady-state accuracy.

## 1.2 Implementation of the Controller

**Design Formulas:**

**Phase Margin Selection:** Target PM = 55° for $\sim$ 10% overshoot and robustness.

**Crossover Frequency:** Choose $\omega_c$ where plant phase is $\angle G(j\omega_c) \approx -110°$. This accounts for PI lag ($\sim 15°$) to yield total phase of $-125°$ at crossover, achieving PM = 55°.

**Integral Time Constant:** Design PI to contribute 15° lag at $\omega_c$:

$$\angle D(j\omega_c) = -\arctan\left(\frac{1}{\omega_c T_I}\right) = -15° \implies T_I = \frac{1}{\omega_c \tan 15°}$$

Numerical values: $\omega_c =$ `<value>` rad/s $\implies T_I =$ `<value>` s.

**Proportional Gain:** Enforce unity gain at crossover $|D(j\omega_c)G(j\omega_c)| = 1$. Discretize PI using Tustin's method and tune iteratively:

$$K_A \approx \texttt{<value>}, \quad K_B \approx \texttt{<value>}$$

**Design Trade-offs:**

- **Phase Margin:** Higher PM → more damping, less overshoot; Lower PM → faster response, risk of instability
- **Crossover Frequency:** Higher $\omega_c$ → faster bandwidth, tighter tracking; Limited by phase margin and sampling rate
- **Integral Time $T_I$:** Larger $T_I$ → less phase lag, better stability; Slower error correction

**Bode Diagram Verification:**

Figure **??** shows open-loop $L(j\omega) = D(j\omega)G(j\omega)$. Crossover at $\omega_c$ with phase $\approx -125°$ confirms PM = 55°. Gain margin also indicated.

Figure **??** shows closed-loop $T(j\omega) = \frac{L}{1+L}$. Bandwidth $\omega_{BW} \approx \omega_c$. Resonant peak due to finite PM. Gain rolls off beyond control bandwidth.

### 1.3 Bandwidth Limitations

**Theoretical Limitations:** Yes. Bandwidth fundamentally limited by plant dynamics and stability requirements. Increasing $\omega_c$ requires reducing PM (to shift phase condition), leading to reduced damping or instability. Cannot arbitrarily increase bandwidth without violating closed-loop stability.

**Practical Limitations:**

- **Sampling rate:** System sampled at 100 Hz (Nyquist at 50 Hz). Bandwidth must stay well below Nyquist ($\sim 1/10$ to $1/5$ of sampling rate) to avoid aliasing.
- **Actuator saturation:** Arduino driver limited to $\pm 12$ V. High bandwidth demands large control effort; saturation degrades performance.
- **Unmodeled dynamics:** High-frequency effects (actuator delays, sensor noise, friction) not captured in model. Very high bandwidth amplifies these, causing instability.

**Implementation Constraints:** Digital implementation imposes hard limit via sampling rate. Chosen bandwidth $\omega_{BW} \approx$ `<value>` Hz balances responsiveness with robustness and practical constraints.

## 2 Experimental Validation

### 2.1 Step Response Validation

**Experiment:** Step reference $0 \rightarrow$ `<value>` rad/s applied to both motors. Measured velocity compared to simulation.

**Step Response (Fig. ??):** Measured (red) matches simulated (blue) closely. Small initial delay (digital sampling), overshoot $\sim 10\%$ (as designed), settling time similar. **Zero steady-state error achieved** in both cases. Minor differences: measured response shows slightly higher oscillations due to friction/quantization.

**Tracking Error (Fig. ??):** Initial error identical. Transient decay similar. Both converge to zero, validating integrator action.

**Control Signal (Fig. ??):** Large initial spike $\sim$ `<value>` V. **Saturation at 12 V** visible in measurement (flat plateau), not in simulation. Explains slightly slower measured acceleration. Steady-state voltage $\sim$ few volts to overcome friction. Good agreement aside from saturation effect.

**Performance Comparison:** Rise time, overshoot, settling align with design specs (PM $= 55°$, $\omega_c =$ `<value>`). Model accuracy confirmed.

### 2.2 Disturbance Rejection (Constant Force)

**Experiment:** Cart driven on incline; gravity provides constant opposing force. Tests Type-1 disturbance rejection.

**Disturbance Model:** Constant force $f[k]$ produces acceleration $a = f/m$. Discretized: $v[k+1] = v[k] + \frac{T_s}{m} f[k]$. In $z$-domain: persistent integrator effect on velocity. Included in simulation.

**Step Response with Disturbance (Fig. ??):** Measured response (red) shows lower overshoot and additional damping compared to no-disturbance case. Disturbance acts as braking force, reducing oscillations. **Steady-state error remains zero** — PI integrator compensates for constant force.

**Tracking Error (Fig. ??):** Larger transient error (slower response due to opposing force). Oscillations damped. Error decays to zero in steady state, confirming successful disturbance rejection.

**Control Signal (Fig. ??):** Higher steady-state voltage required to counteract gravity ($\sim$ `<value>` V vs $\sim$ `<value>` V). Integrator accumulates to provide necessary torque. Demonstrates Type-1 property: integral action eliminates effect of constant disturbance.

**Theoretical Justification:** PI controller (Type-1) guarantees zero steady-state error for step reference. For step disturbance at plant output, closed-loop is Type-0 w.r.t. disturbance if disturbance enters before integrator. Since force integrates to affect velocity (plant has integrator effect), and controller has integrator, overall loop is Type-1 to disturbance. Experiment confirms: PI successfully rejects constant force with zero steady-state error.

**Summary:** Experimental results validate PI design. Zero steady-state error achieved for step reference (no disturbance and with constant disturbance). Performance characteristics (rise time, overshoot, settling) match design specifications. Saturation and minor unmodeled effects explain small discrepancies. Controller demonstrates robust velocity tracking and disturbance rejection.

# References