KU LEUVEN
FACULTEIT
INGENIEURSWETENSCHAPPEN

B-KUL-H04X3A: Control Theory

Team members:

Lefebure Tiebert (r0887630)
Campaert Lukas (r0885501)

Assignment 3: State Feedback and State Estimation

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 State estimator and state feedback controller design using pole placement

The goal of this assignment is to control the position of the cart along a straight line. The state is the cart position $x(t)$ [m] along the x-axis, with the wall at the origin $x = 0$. The cart is positioned in front of the wall, so $x < 0$. The infrared sensor measures the positive distance to the wall, i.e. it measures $-x$.

The system parameters are:

- Wheel radius: $r = 0.033$ m
- Sampling time: $T_s = 0.01$ s (100 Hz control loop)

The input $u(t) = \omega(t)$ [rad/s] is the common wheel angular velocity setpoint, applied equally to both motors via the inner velocity controllers designed in Assignment 2. The velocity control loop is assumed ideal, meaning the actual wheel velocity tracks the setpoint perfectly.

1.1 Discrete-time state equation

Continuous-time state equation: The velocity control loop from Assignment 2 is assumed ideal, so the actual cart velocity equals the commanded velocity. The continuous-time kinematic relation is:

$$\dot{x}(t) = v(t) = r \cdot \omega(t) = r \cdot u(t), \tag{1}$$

where $v(t)$ [m/s] is the linear cart velocity and $u(t) = \omega(t)$ [rad/s] is the wheel angular velocity input.

In state-space form with scalar state $x(t)$ [m] and input $u(t)$ [rad/s]:

$$\dot{x}(t) = A_c x(t) + B_c u(t), \quad A_c = 0, \quad B_c = r = 0.033 \text{ m}. \tag{2}$$

Forward Euler discretization: The forward Euler method approximates the derivative as:

$$\dot{x}[k] \approx \frac{x[k+1] - x[k]}{T_s}. \tag{3}$$

Substituting into the continuous-time equation:

$$\frac{x[k+1] - x[k]}{T_s} = A_c x[k] + B_c u[k] = 0 \cdot x[k] + r \cdot u[k]. \tag{4}$$

Solving for $x[k+1]$:

$$x[k+1] = x[k] + T_s \cdot r \cdot u[k]. \tag{5}$$

The discrete-time state-space model is:

$$x[k+1] = A_d x[k] + B_d u[k], \tag{6}$$

with the discrete-time system matrices:

$$\boxed{A_d = 1, \qquad B_d = T_s \cdot r = 0.01 \times 0.033 = 3.3 \times 10^{-4} \text{ m/(rad/s)}.} \tag{7}$$

Note: In the state estimator implementation, the measured average wheel speed $u[k] = (\omega_A[k] + \omega_B[k])/2$ is used for the prediction step.

1.2 Measurement equation

The front infrared (IR) sensor measures the distance from the cart to the wall. Since the cart is at position $x < 0$ (in front of the wall at $x = 0$), the measured distance is a positive quantity equal to $|x| = -x$.
The measurement equation (ignoring noise) is:

$$y[k] = -x[k]. \tag{8}$$

In state-space form:

$$y[k] = Cx[k] + Du[k], \tag{9}$$

with the output matrices:

$$\boxed{C = -1, \qquad D = 0.} \tag{10}$$

Here $C = -1$ because the sensor measures the negation of the state, and $D = 0$ since the input does not directly affect the measurement.

1.3 Design of state feedback controller gain K using pole placement

The position controller outputs the desired wheel angular velocity based on the position error. Assuming full state feedback (no estimator), the control law is:

$$u[k] = K \cdot (x_{\text{ref}}[k] - x[k]), \tag{11}$$

where $x_{\text{ref}}[k]$ [m] is the reference position, $x[k]$ [m] the measured position, K [rad/(s·m)] the state feedback gain, and $u[k]$ [rad/s] the commanded wheel angular velocity.

Derivation of closed-loop pole as function of K: Substituting the control law into the discrete-time state equation:

$$\begin{aligned} x[k+1] &= A_d x[k] + B_d u[k] && (12)\\ &= x[k] + T_s r \cdot K(x_{\text{ref}}[k] - x[k]) && (13)\\ &= (1 - T_s r K)x[k] + T_s r K \cdot x_{\text{ref}}[k]. && (14) \end{aligned}$$

The closed-loop system has the form:

$$x[k+1] = A_{\text{cl}} x[k] + B_{\text{cl}} x_{\text{ref}}[k], \tag{15}$$

where the closed-loop system matrix is:

$$\boxed{A_{\text{cl}} = 1 - T_s r K.} \tag{16}$$

For this first-order system, the closed-loop pole equals the system matrix:

$$\boxed{z_{\text{cl}}(K) = 1 - T_s r K = 1 - 3.3 \times 10^{-4} \cdot K.} \tag{17}$$

Pole behavior as function of K: At $K = 0$, $z_{\text{cl}} = 1$ (marginally stable integrator). As K increases, the pole moves left along the real axis, reaching $z_{\text{cl}} = 0$ (deadbeat) at $K \approx 3030$ rad/(s·m) and the stability boundary $z_{\text{cl}} = -1$ at $K \approx 6060$ rad/(s·m).

Stability analysis: For discrete-time stability, the pole must lie inside the unit circle:

$$|z_{\text{cl}}(K)| = |1 - T_s r K| < 1. \tag{18}$$

This yields the stability condition:

$$\boxed{0 < K < \frac{2}{T_s r} = \frac{2}{0.01 \times 0.033} \approx 6060 \text{ rad/(s·m)}.} \tag{19}$$

Yes, the system can become unstable if $K > 6060$ rad/(s·m), causing the pole to exit the unit circle through $z = -1$.

Pole-zero map: Figure 1 shows the closed-loop pole location for varying K. The pole moves along the real axis from $z = 1$ (at $K = 0$) toward $z = -1$ (at $K = K_{\max}$).

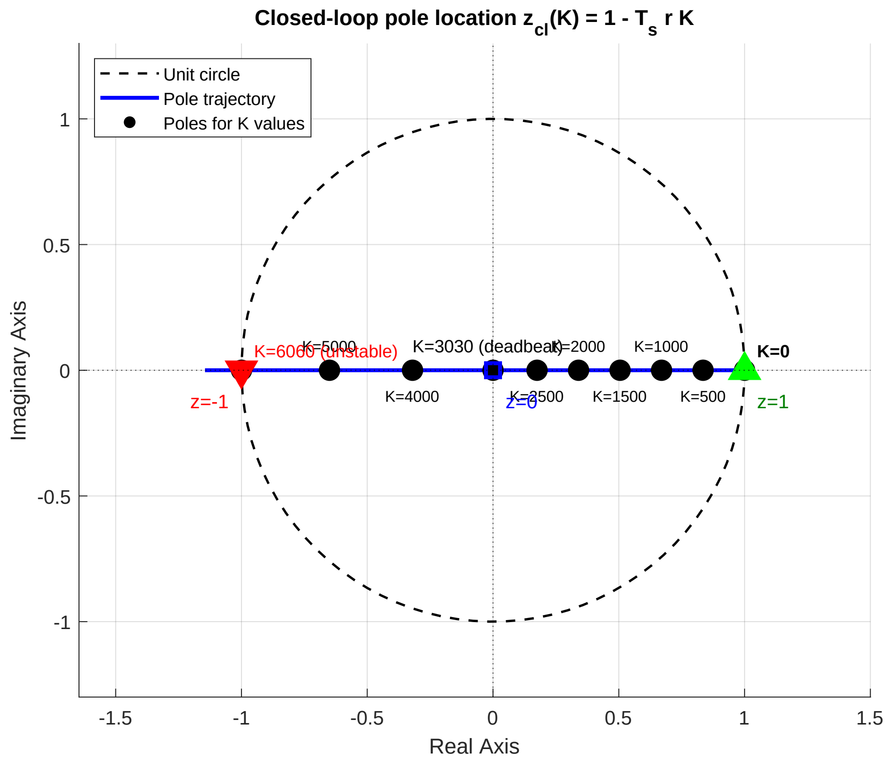

Figure 1: Closed-loop pole location $z_{\text{cl}}(K) = 1 - T_s r K$ for varying K. Unit circle shown for stability reference.

Simulated step responses: The discrete-time step response is simulated for $K \in \{20, 40, 80\}$ rad/(s·m):

$$x[k+1] = (1 - T_s r K)x[k] + T_s r K \cdot x_{\text{ref}}, \quad x[0] = x_0. \tag{20}$$

K [rad/(s·m)]	z_{cl}	Pole location	Expected behavior
20	0.9934	Close to 1	Slow convergence, long settling time
40	0.9868	Moderate	Faster response, good compromise
80	0.9736	Closer to 0	Fast response, higher control effort

Table 1: Closed-loop poles for different K values.

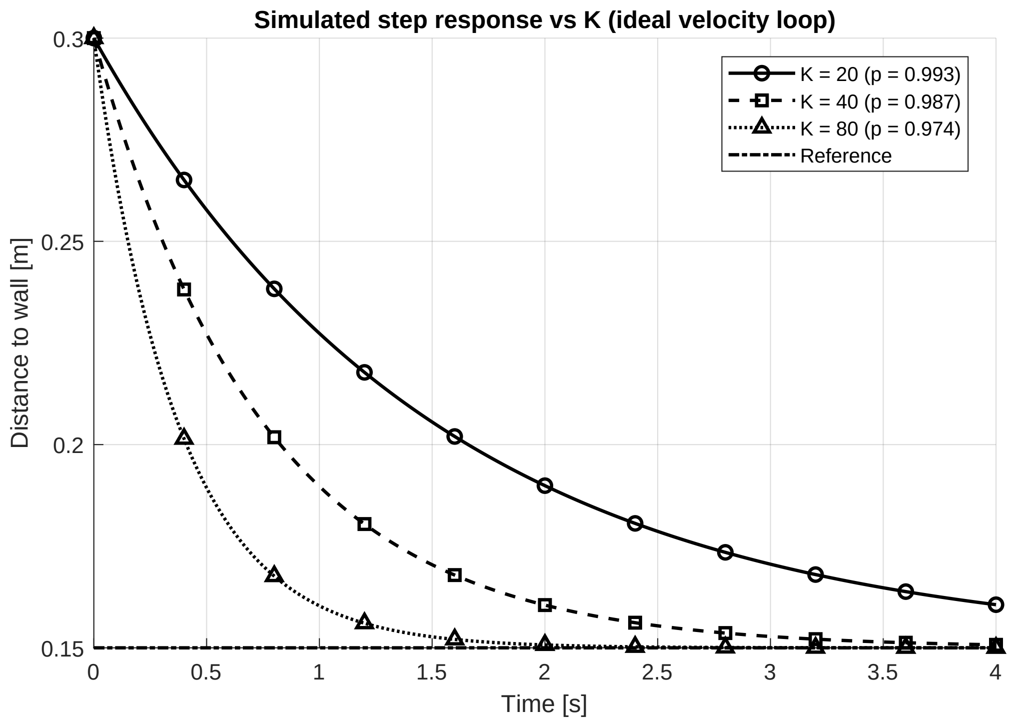

Figure 2: Simulated closed-loop step responses. Larger K yields faster convergence but higher peak velocity commands.

Relationship between pole location, K, and time response:

- **Small** K ($z_{\text{cl}} \approx 1$): Slow exponential decay, long settling time, smooth but sluggish response.
- **Moderate** K ($0 < z_{\text{cl}} < 1$): Faster convergence, monotonic response without oscillation.
- **Large** K ($z_{\text{cl}} < 0$): Sign-alternating (oscillatory) discrete-time response. If $|z_{\text{cl}}| < 1$, still stable but with overshoot.
- $K > K_{\mathbf{max}}$ ($|z_{\text{cl}}| > 1$): Unstable, diverging oscillations.

Choice of K: The theoretical stability limit is $K_{\max} \approx 6060$ rad/(s·m), but practical constraints are more restrictive:

- **Actuator saturation**: The motor voltage is limited to ±11 V. Large K causes large velocity commands for small position errors, potentially saturating the inner velocity loop.
- **Sensor noise**: High K amplifies measurement noise into the control signal.
- **IR sensor range**: The sensor is accurate only for 5–30 cm. Aggressive control may drive the cart outside this range.

Selected value:

$$\boxed{K = 40 \text{ rad/(s}\cdot\text{m)}.} \tag{21}$$

This gives $z_{\text{cl}} = 1 - 0.01 \times 0.033 \times 40 = 0.9868$, providing:

- Stable operation well within the stability margin
- Reasonable settling time (≈ 3–5 s for 2% criterion)
- Control effort within actuator limits for typical position steps (0.1–0.2 m)

1.4 Design of state estimator gain L using pole placement

A discrete-time Luenberger observer is used to estimate the cart position from the IR sensor measurement:

$$\hat{x}[k+1] = A_d\hat{x}[k] + B_d u[k] + L \cdot \nu[k], \tag{22}$$

where $\nu[k] = y[k] - C\hat{x}[k]$ is the **innovation** (measurement prediction error).

With $A_d = 1$, $B_d = T_s r$, and $C = -1$:

$$\hat{x}[k+1] = \hat{x}[k] + T_s r \cdot u[k] + L \cdot (y[k] + \hat{x}[k]). \tag{23}$$

Derivation of estimator pole as function of L: Define the estimation error:

$$e[k] = x[k] - \hat{x}[k]. \tag{24}$$

The true state evolves as $x[k+1] = x[k] + T_s r \cdot u[k]$. Subtracting the observer equation:

$$\begin{aligned}
e[k+1] &= x[k+1] - \hat{x}[k+1] && (25)\\
&= (x[k] + T_s r \cdot u[k]) - (\hat{x}[k] + T_s r \cdot u[k] + L(y[k] - C\hat{x}[k])) && (26)\\
&= (x[k] - \hat{x}[k]) - L(Cx[k] - C\hat{x}[k]) && (27)\\
&= e[k] - L \cdot C \cdot e[k] && (28)\\
&= (1 - LC) \cdot e[k]. && (29)
\end{aligned}$$

Substituting $C = -1$:

$$e[k+1] = (1 - L \cdot (-1)) \cdot e[k] = (1 + L) \cdot e[k]. \tag{30}$$

The estimator error dynamics have the closed-loop pole:

$$\boxed{z_{\text{est}}(L) = 1 + L.} \tag{31}$$

Pole behavior as function of L:

- At $L = 0$: $z_{\text{est}} = 1$ (no correction, estimator ignores measurements)
- As L becomes more negative: pole moves **left along the real axis**
- At $L = -1$: $z_{\text{est}} = 0$ (deadbeat estimator, instant convergence)
- At $L = -2$: $z_{\text{est}} = -1$ (stability boundary)

Stability analysis: For the estimator to be stable:

$$|z_{\text{est}}(L)| = |1 + L| < 1. \tag{32}$$

This yields:

$$\boxed{-2 < L < 0.} \tag{33}$$

Yes, the estimator can become unstable:

- If $L > 0$: pole $z_{\text{est}} > 1$, estimation error grows exponentially.
- If $L < -2$: pole $z_{\text{est}} < -1$, estimation error diverges with oscillations.

Trade-offs in pole placement:

- **Faster estimator** (larger $|L|$, pole closer to 0):
  - + Faster convergence from wrong initial estimate
  - + Quicker correction of model errors
  - − Higher sensitivity to measurement noise (noise is amplified by L)
  - − Risk of oscillatory behavior if $z_{\text{est}} < 0$
- **Slower estimator** (smaller $|L|$, pole closer to 1):
  - + Smoother estimate, better noise rejection
  - − Slow convergence from initialization errors
  - − Poor tracking of rapid state changes

Design guideline: The estimator pole should typically be 2–6 times faster than the controller pole, ensuring the estimation error decays before significantly affecting control performance.

Pole placement calculation: With the chosen controller gain $K = 40$ rad/(s·m), the controller pole is:

$$z_{\text{cl}} = 1 - T_s r K = 1 - 0.01 \times 0.033 \times 40 = 0.9868. \tag{34}$$

To make the estimator 4 times faster (in continuous-time sense), we use:

$$z_{\text{est}} = z_{\text{cl}}^4 = 0.9868^4 \approx 0.948. \tag{35}$$

Solving for L:

$$L = z_{\text{est}} - 1 = 0.948 - 1 = -0.052. \tag{36}$$

Selected value:

$$\boxed{L = -0.05 \text{ (dimensionless).}} \tag{37}$$

This gives $z_{\text{est}} = 0.95$, which:

- Lies safely within the stability region $-2 < L < 0$
- Provides faster convergence than the controller pole
- Maintains reasonable noise rejection

Note: The gain L is dimensionless because both $y[k]$ and $x[k]$ have units of meters. The correction term $L \cdot \nu[k]$ has units [m], matching the state.

Alternative gains for experiments: To study the effect of L on estimator performance, we test:

Gain	z_{est}	Expected behavior
$L = -0.05$ (slow)	0.95	Slow convergence, smooth estimate
$L = -0.18$ (nominal)	0.82	Moderate speed, good compromise
$L = -0.35$ (fast)	0.65	Fast convergence, noisier estimate

Table 2: Estimator gains tested in experiments.

2 Implementation and testing of state estimator and state feedback controller

Now the designed estimator and controller are implemented on the Arduino, following the assignment structure. Data and figures will be populated after running the Arduino experiments and MATLAB post-processing. Each plot placeholder corresponds to a file exported to `images/`.

2.1 Estimator only: wrong initial estimate, different L

The controller is disabled; the estimator starts from a wrong initial position $\hat{x}[0]$ set via `readValue(5)` in the Arduino code.

Configuration:

- Estimator active, controller off. The cart is moved manually, or driven with a known, simple input signal.
- The state estimator is initialized with a wrong position estimate $\hat{x}[0]$ (e.g. $10cm$ closer to the wall than reality).
- Different values of L are tested (e.g. L = -0.02, -0.07, -0.15).

For each L:

- The measured distance $y[k]$ (positive, measured by IR sensor) is logged.
- The estimated state $\hat{x}[k]$ is logged.

Figure 3 overlays the measured distance $y = -x$ and the estimated distance $-\hat{x}$ for $L \in \{-0.05, -0.18, -0.35\}$. Larger $|L|$ yields faster convergence but more noise on $\hat{x}$ and ν.

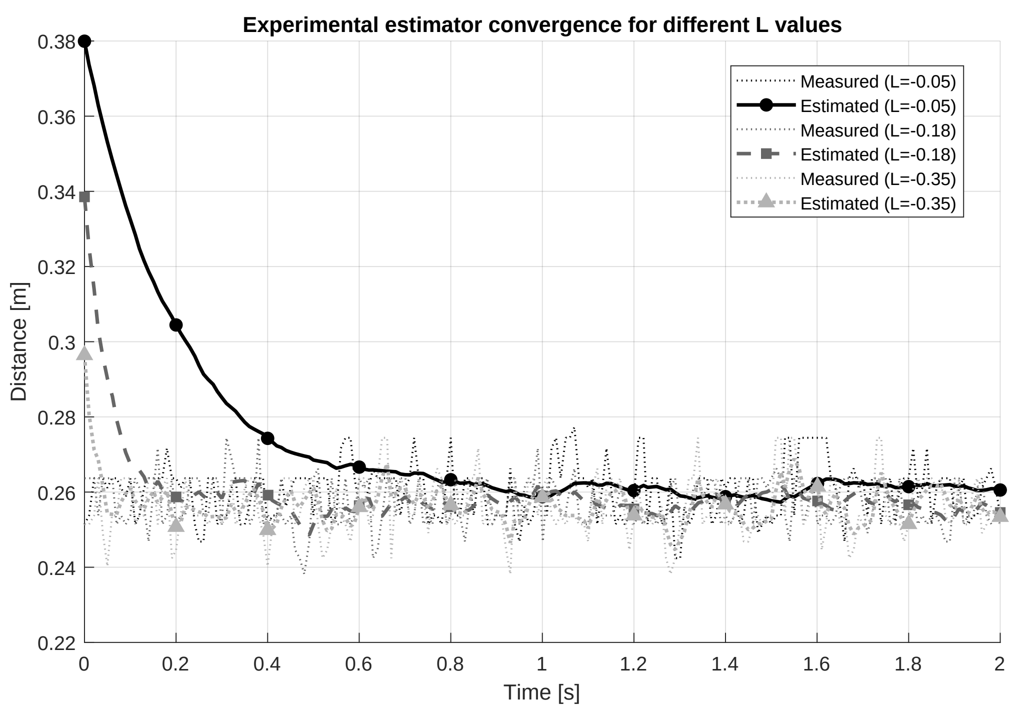

Figure 3: Measured vs. estimated distance for different L (wrong initial estimate).

Interpretation and link to design of state estimator gain L: For small $|L|$ (e.g. $L \approx -0.02$, so $z_{\text{est}} \approx 0.98$):

- The estimator reacts slowly to the discrepancy between measurement and prediction.
- The estimate converges slowly to the measurement, a substantial transient remains for a long time.
- This matches the formula $z_{est} = 1 + L$: pole close to 1 leads to slow dynamics.

For moderate $|L|$ (e.g. the designed $L \approx -0.07$, giving $z_{est} \approx 0.93$):

- The estimate converges significantly faster to the measurement.
- Noise on the measurement is visible but not overly amplified.

For large $|L|$ close to -2 (e.g. $L =$ -1.5, so $z_{\text{est}} =$ -0.5):

- Convergence is very fast, but the estimate closely follows measurement noise, becoming noisy and irregular.
- This shows the trade-off: faster convergence vs. noise sensitivity.

So:

- Convergence speed increases with $|L|$, consistent with $z_{\text{est}} = 1 + L$.
- Noise on $\hat{x}$ and the innovation ν grows with $|L|$, illustrating the trade-off from 1(d).
- The chosen L_{nom} balances settling time ($\approx$ TODO s) and noise (RMS $\approx$ TODO m).

From the plots, one concludes that larger negative L values (closer to -2) give faster convergence of the estimates to the measurements, consistent with the expression $z_{\text{est}} = 1 + L$ and the trade-offs discussed in Section 1.4 (design of state estimator gain L).

2.2 Controller only: proportional feedback for different K

The estimator is disabled, feedback uses the raw distance $y[k] = -x[k]$. A $0.15m$ step reference is applied for $K \in \{K_{\text{slow}}, K_{\text{nom}}, K_{\text{fast}}\}$.

Configuration:

- Controller active, estimator off. The controller directly uses the IR measurement (converted to position) for feedback.
- The control law becomes: $u[k] = K(r[k] - x[k])$, with $x[k]$ obtained from the IR sensor relation $x[k] = -y[k]$.
- A step in position reference is applied. The car starts at, for example, $30cm$ from the wall. The reference is stepped from $30cm$ to $15cm$ (in x: $x = -0.30$ to $x = -0.15$).

Repeat this experiment for several values of K (e.g. $K =$ 10,50,100 $rad/(s \cdot m)$).

Figure 4 shows position responses and Figure 5 the corresponding motor voltages.

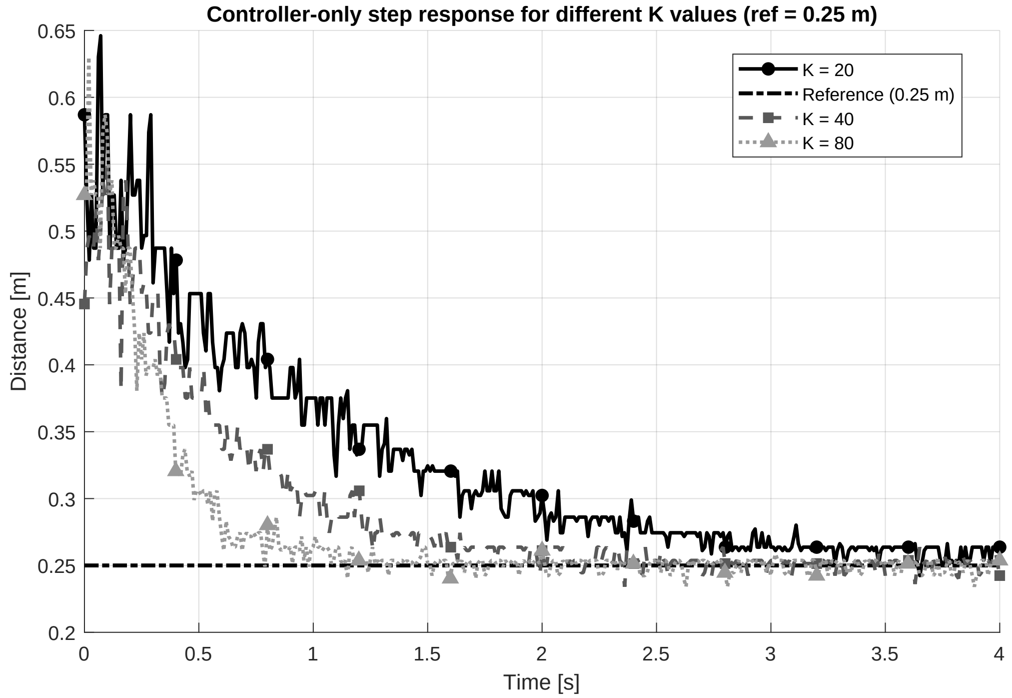

Figure 4: Measured position step responses for different K (estimator disabled).

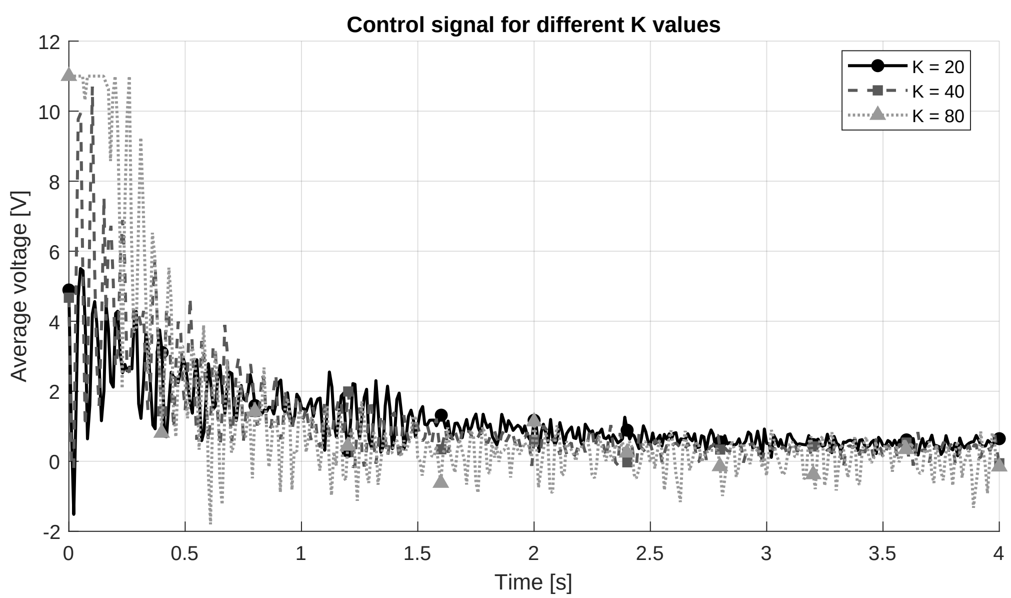

Figure 5: Motor voltage commands for the responses in Fig. 4.

Interpretation and link to design of state feedback controller gain K: As shown analytically, the closed-loop pole is

$$z_{\text{cl}}(K) = 1 - T_s r K. \tag{38}$$

For small K (e.g. $K = 20\ rad/(s \cdot m)$):

- Pole close to 1: slow rise, long settling time, very smooth control signal.
- This matches time-responses with slow approach to the reference.

For moderate K (e.g. the designed $K = 50\ rad/(s \cdot m)$):

- Pole closer to zero: faster response, reduced settling time.
- Acceptable overshoot and control signals well within approximately $12V$.
- Good tracking performance.

For large K close to theoretical maximum $\frac{2}{T_s r}$ (e.g. $K = 100\ rad/(s \cdot m)$):

- Pole approaches -1: discrete-time oscillatory response and possibly overshoot.
- The control signals reach or exceed saturation ($12V$), so the inner velocity loop cannot follow the demanded angular velocity.

So:

- Rise time and steady-state error match the simulated trend from Section 1.3 (design of state feedback controller gain K), higher K speeds up the response but increases peak voltage.
- For $K = K_{\text{fast}}$ the voltage briefly saturates, explaining the mild overshoot; this bounds feasible K in practice.
- The selected K_{nom} avoids saturation while providing $t_s \approx$ TODO s and negligible steady-state error.

These observations correspond to the theoretical dependence $z_{cl}(K) = 1 - T_s r K$: as K increases, the closed-loop pole moves left towards -1, giving faster dynamics until saturation and discrete oscillations become limiting.

Choice of K: There are both theoretical and practical limits for K. Theoretical limit: stability in discrete-time requires

$$0 < K < \frac{2}{T_s r} \approx 6060 \text{ rad/(s·m)}. \tag{39}$$

Practical limits:

- The low-level motor voltage is bounded: too large K causes saturation, making the actual behavior deviate from the linear model.
- Large K amplifies sensor noise and model errors.
- The IR sensor has a limited range (approximately 5-30cm). Overshoot may drive the cart outside this range, resulting in invalid measurements.

In light of these constraints, a value around

$$K = 40 \text{rad/(s·m)} \tag{40}$$

offers a good compromise between tracking performance and actuator/sensor limitations, and is therefore used in the remainder of the assignment.

2.3 Combined estimator and controller: influence of estimator pole choice

In this experiment, both the state feedback controller and the state estimator are active. The controller uses the estimated position $\hat{x}[k]$ instead of the true position:

$$u[k] = K(x_{\text{ref}}[k] - \hat{x}[k]). \tag{41}$$

Per the assignment specification, the estimator is designed to be **10 times slower** than the controller.

Recap of closed-loop poles: From Sections 1(c) and 1(d), the closed-loop poles are:

- **Controller pole**: $z_{\text{cl}}(K) = 1 - T_s r K$
- **Estimator pole**: $z_{\text{est}}(L) = 1 + L$

With $K = 40$ rad/(s·m):

$$z_{\text{cl}} = 1 - 0.01 \times 0.033 \times 40 = 0.9868. \tag{42}$$

Designing the estimator to be 10× slower: A pole closer to 1 corresponds to slower dynamics. To make the estimator 10 times slower in continuous-time:

1. Convert controller pole to continuous-time: $s_{\text{cl}} = \frac{\ln(z_{\text{cl}})}{T_s} = \frac{\ln(0.9868)}{0.01} \approx -1.33$ rad/s
2. Estimator pole 10× slower: $s_{\text{est}} = \frac{s_{\text{cl}}}{10} = -0.133$ rad/s
3. Convert back to discrete-time: $z_{\text{est}} = e^{s_{\text{est}} \cdot T_s} = e^{-0.00133} \approx 0.9987$

Solving for L using $z_{\text{est}} = 1 + L$:

$$L = z_{\text{est}} - 1 = 0.9987 - 1 = -0.0013. \tag{43}$$

Selected slow estimator gain:

$$\boxed{L_{\text{slow}} = -0.00132 \text{ (dimensionless).}} \tag{44}$$

This matches the value used in the Arduino code for Section 2(c).

Full closed-loop system poles: The combined system (controller + estimator) has two poles:

$$\boxed{\lambda_1 = z_{\text{cl}} = 0.9868, \qquad \lambda_2 = z_{\text{est}} = 0.9987.} \tag{45}$$

Key observation (Separation Principle): The controller pole z_{cl} depends only on K, and the estimator pole z_{est} depends only on L. The two dynamics are decoupled—the estimator design does not affect the controller pole, and vice versa.

Experiments: Two experiments are performed with the slow estimator ($L = -0.00132$):

1. **Good initial estimate**: $\hat{x}[0] \approx x[0]$. A step reference is applied.
2. **Wrong initial estimate**: $\hat{x}[0] = x[0] + \Delta x$ (e.g., 10–15 cm offset). A step reference is applied simultaneously.

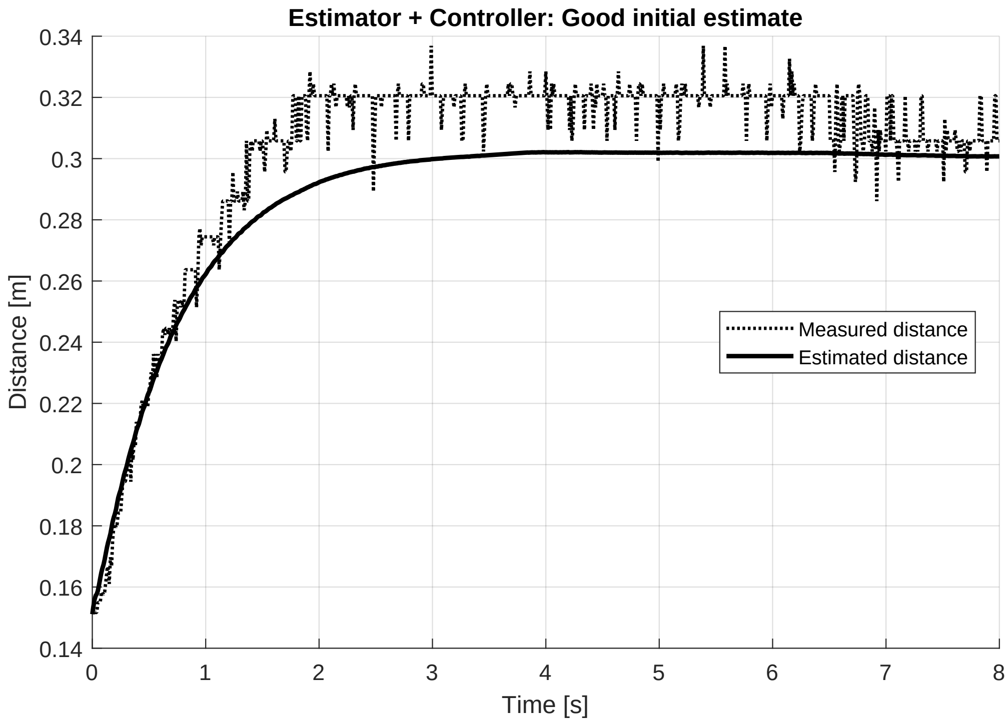

Figure 6: Combined estimator and controller with good initial estimate. The estimator tracks the measurement, and the controller achieves the design settling time.

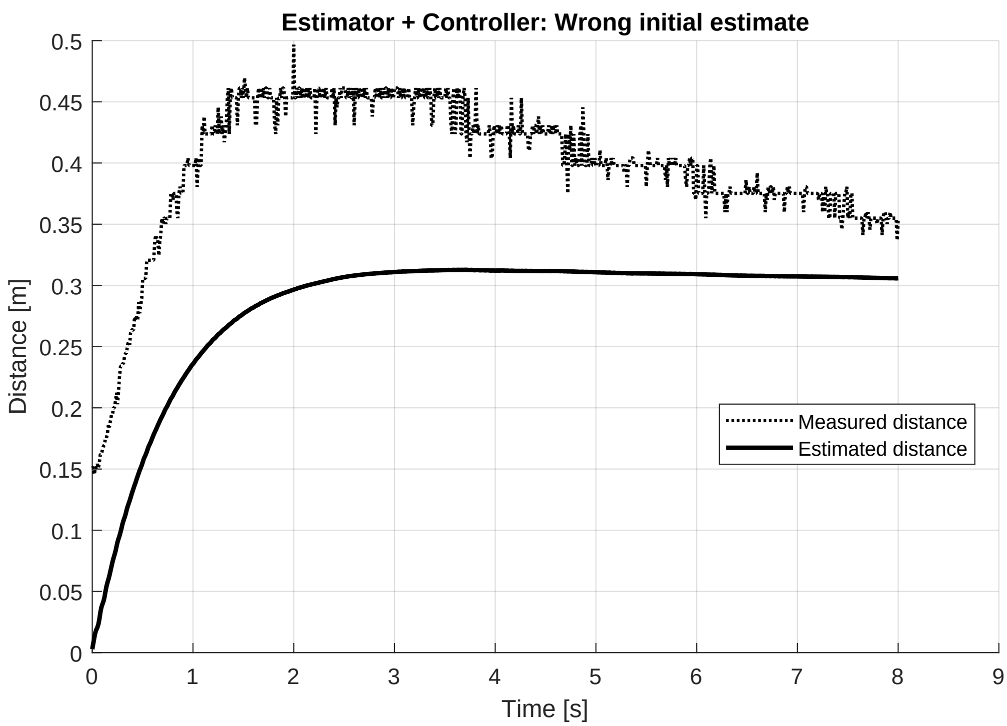

Figure 7: Combined estimator and controller with wrong initial estimate. The slow estimator takes a long time to converge, but the control performance remains acceptable.

Analysis of results: 1. Good initial estimate ($\hat{x}[0] \approx x[0]$):

- The estimation error is small from the start.
- The controller responds to the step reference with dynamics governed by $z_{\mathrm{cl}} = 0.9868$.
- Control performance is satisfactory—the cart reaches the reference position with the expected settling time.

2. Wrong initial estimate ($\hat{x}[0] \neq x[0]$):

- The estimator starts with a large error and converges slowly (pole at 0.9987).
- During convergence, the controller acts on a biased estimate, causing suboptimal transient behavior.
- However, the **steady-state** control performance is unaffected—once the estimator converges, the controller achieves zero steady-state error.

Is control performance satisfactory?

- **Good initialization**: Yes, the controller meets design specifications.
- **Wrong initialization**: Transient performance is degraded due to the slow estimator, but steady-state is correct. For better transient performance, a faster estimator (larger $|L|$) should be used.

Recommended estimator pole placement: For practical applications, the estimator should be **faster** than the controller (2–6 times), not slower. This ensures:

- Quick convergence from initialization errors
- Minimal impact of estimation error on control performance

A value such as $L = -0.05$ to -0.18 (giving $z_{\mathrm{est}} = 0.82$–0.95) would be more appropriate for real applications.

Separation principle verification: The closed-loop transfer function from reference to output can be written as:

$$\frac{X(z)}{X_{\mathrm{ref}}(z)} = \frac{T_s r K}{(z - z_{\mathrm{cl}})(z - z_{\mathrm{est}})} \cdot (z - z_{\mathrm{est}}) = \frac{T_s r K}{z - z_{\mathrm{cl}}}. \tag{46}$$

The estimator pole z_{est} cancels in the transfer function from reference to output, confirming that:

- The **reference tracking** dynamics depend only on z_{cl} (controller pole).
- The estimator pole affects only the **estimation error** dynamics, not the control dynamics.
- This is the **separation principle**: controller and estimator can be designed independently.

References