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 Design of a state feedback controller and estimator (pole placement)

Unless stated otherwise, the sampling time is $T_s = 0.01$ s and the wheel radius is $r = 0.033$ m. The state is the cart position x (m) along the track, measured from the wall at the origin ($x < 0$ when the cart is in front of the wall). The input v is the common wheel velocity command (rad/s).

1.1 Discrete-time state equation (1a)

Continuous-time model. With ideal inner velocity control, $\dot{x}(t) = r\,\omega(t)$, where ω is the average wheel speed. In matrix form with state x and input $u = \omega$:

$$\dot{x}(t) = A_c x(t) + B_c u(t), \quad A_c = 0, \quad B_c = r. \tag{1}$$

Forward Euler discretization. Using $x[k+1] = x[k] + T_s\dot{x}[k]$:

$$x[k+1] = x[k] + T_s r\, u[k] = A_d x[k] + B_d u[k], \tag{2}$$

with

$$A_d = 1, \qquad B_d = T_s r = 3.3 \times 10^{-4} \text{ m/rad}. \tag{3}$$

For the estimator we also use $u[k] \approx (\omega_A[k] + \omega_B[k])/2$ to incorporate measured wheel speeds.

1.2 Measurement equation (1b)

The infrared sensor measures the distance to the wall, i.e. $y[k] = -x[k] + v_d[k]$ with v_d measurement noise. The discrete measurement model is

$$y[k] = Cx[k] + Du[k], \qquad C = -1, \qquad D = 0. \tag{4}$$

The positive distance reading corresponds to $-x$, consistent with Fig. 3.

1.3 Design of state feedback gain K (1c)

The state feedback acts on the position estimate: $u[k] = K\,(x_{\text{ref}} - \hat{x}[k])$, with K in rad/(s·m). In closed loop (no estimator) the pole is

$$p_{\text{cl}}(K) = A_d - B_d K = 1 - T_s r K. \tag{5}$$

The pole moves left on the real axis as K increases; stability requires $|p_{\text{cl}}(K)| < 1 \Rightarrow 0 < K < \frac{2}{T_s r} \approx 6.1 \times 10^3$ rad/(s·m). A faster response is obtained by choosing p_{cl} closer to 0, at the cost of higher velocity commands.

Pole-zero map. Figure 1 shows $p_{\text{cl}}(K)$ for $K \in \{K_{\text{slow}}, K_{\text{nom}}, K_{\text{fast}}\} = \{80, 120, 250\}$ rad/(s·m).

Placeholder for pole-zero map from MATLAB (images/pole_map_K.pdf).

Figure 1: Closed-loop pole location versus K.

Step responses. Figure 2 groups simulated position step responses for the same K sweep. Larger K yields shorter rise time and higher control action; excessive K would saturate the motors and could destabilize the discretized model.

Placeholder for simulated step responses (images/step_K_sweep.pdf).

Figure 2: Simulated closed-loop step responses for varying K.

Gain selection. A nominal choice $K_{\text{nom}} = 120$ rad/(s·m) gives $p_{\text{cl}} \approx 0.96$, keeping the velocity within ± 20 rad/s for 0.2 m position steps while preserving stability margin. This value will be updated after on-cart tests; the MATLAB script exports the active value to the Arduino code.

1.4 Design of state estimator gain L (1d)

The one-state observer update is $\hat{x}[k+1] = A_d \hat{x}[k] + B_d u[k] + L\,(y[k] - C\hat{x}[k])$. The estimation error $e[k] = x[k] - \hat{x}[k]$ evolves as

$$e[k+1] = (A_d - LC)e[k] \quad \Rightarrow \quad p_{\text{est}}(L) = A_d - LC = 1 + L. \tag{6}$$

Because $C = -1$, negative L moves the observer pole inside the unit circle; $L > 0$ makes it unstable. Fast convergence requires $|p_{\text{est}}|$ small, but a large $|L|$ amplifies sensor noise.

Pole placement and trade-offs. We select L such that the estimator is 3–5 times faster than the controller for tasks in Section 2(a)–(b), then set it ten times slower than the controller for Section 2(c) per the specification. The practical limits are set by sensor noise and discretization: $|1 + L| < 1 \Rightarrow -2 < L < 0$.

Chosen gain. A nominal value $L_{\text{nom}} = -0.18$ gives $p_{\text{est}} = 0.82$ (fast but noise-aware). Alternative gains $L \in \{-0.05, -0.18, -0.35\}$ are swept in the experiments below.

2 Implementation and experiments

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 (2a)

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

Placeholder for estimator convergence plots (images/estimator_L_sweep.pdf).

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

Observations:

- Convergence speed increases with $|L|$, consistent with $p_{\text{est}} = 1 + L$.
- 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).

2.2 Controller only (2b)

The estimator is disabled; feedback uses the raw distance $y = -x$. A 0.15 m step reference is applied for $K \in \{K_{\text{slow}}, K_{\text{nom}}, K_{\text{fast}}\}$. Figure 4 shows position responses and Figure 5 the corresponding motor voltages.

Placeholder for measured step responses without estimator (images/controller_K_response.pdf).

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

Placeholder for control signals (images/controller_K_voltage.pdf).

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

Observations:

- Rise time and steady-state error match the simulated trend from 1(c); 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.

2.3 Estimator and controller (2c)

The estimator and controller run together. The estimator pole is placed ten times slower than the controller: $(1 - p_{\text{est}}) = (1 - p_{\text{cl}})/10$, leading to $L_{\text{slow}} \approx -0.004$ for K_{nom}. Two experiments are executed with a 0.15 m step:

1. **Good initial estimate** $\hat{x}[0] \approx x[0]$.
2. **Wrong initial estimate** $\hat{x}[0] = \hat{x}_0^{\text{bad}}$ (set via `readValue(5)`).

Figures 6 and 7 plot measured and estimated distances. The closed-loop poles of the combined system are

$$\lambda_{1,2} = \{p_{\text{cl}}(K_{\text{nom}}),\ p_{\text{est}}(L_{\text{slow}})\}, \tag{7}$$

confirming that the estimator dynamics do not affect the control pole (separation principle).

Placeholder for measured vs. estimated position, good initial estimate (images/est_ctrl_good.pdf).

Figure 6: Measured and estimated distance with good initial estimate.

Placeholder for measured vs. estimated position, wrong initial estimate (images/est_ctrl_bad.pdf).

Figure 7: Measured and estimated distance with wrong initial estimate.

Observations:

- With correct initialization, the estimator tracks the measurement and the controller meets the design settling time.
- With a wrong $\hat{x}[0]$, the estimator converges slowly by design while the control performance remains acceptable, illustrating that the estimator pole does not alter the control pole.
- If faster estimator convergence is required, L can be decreased (more negative) at the cost of more noise, as seen in 2(a).

References