KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

B-KUL-H04X3A: Control Theory

Team members: Lefebure Tiebert (r0887630), Campaert Lukas (r0885501)

Assignment 5: EKF and LQR for the 2WD Swivel Cart

Due: [DD/MM/2025]

1 System modelling

1.1 Inputs → wheel speeds and state equation

Let ω_A, ω_B denote wheel angular velocities (rad/s), r the wheel radius and $b = \frac{\text{WHEELBASE}}{2}$ the half-track. The forward/rotational inputs are

$$v = \frac{r}{2}(\omega_A + \omega_B), \qquad \omega = \frac{r}{2b}(\omega_B - \omega_A). \tag{1}$$

Inverting,

$$\omega_A = \frac{v}{r} - \frac{b}{r}\omega, \qquad \omega_B = \frac{v}{r} + \frac{b}{r}\omega. \tag{2}$$

With $\xi = [x_c \; y_c \; \theta]^T$ the pose in the world XY frame and $u = [v \; \omega]^T$, the nonlinear continuous model is

$$\dot{\xi} = f(\xi, u) = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix}, \qquad \xi(0) = [[-0.30 \;\; -0.20 \; 0]^T \text{ m, rad}]. \tag{3}$$

1.2 Measurement equation (general and Fig. 4)

Let a wall be $\mathcal{W} = \{(x, y) \mid px + qy = r\}$ and $n = \sqrt{p^2 + q^2}$. Front sensor position (body frame $(\alpha, 0)$), side sensor position (body frame (β, γ) with $\gamma > 0$ to the left) expressed in world coordinates are

$$(x_f, y_f) = \big(x_c + \alpha\cos\theta, \; y_c + \alpha\sin\theta\big), \tag{4}$$

$$(x_s, y_s) = \big(x_c + \beta\cos\theta - \gamma\sin\theta, \; y_c + \beta\sin\theta + \gamma\cos\theta\big). \tag{5}$$

Distance to a wall is the signed orthogonal projection

$$h(x, y; p, q, r) = \frac{r - px - qy}{n}. \tag{6}$$

Hence the measurement equation is

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} h(x_f, y_f; p_1, q_1, r_1) \\ h(x_s, y_s; p_2, q_2, r_2) \end{bmatrix} + w, \quad w \sim \mathcal{N}(0, R). \tag{7}$$

For Fig. 4 (walls at $y = 0$ and $x = 0$) take $(p_1, q_1, r_1) = (0, 1, 0)$ and $(p_2, q_2, r_2) = (1, 0, 0)$ so that $z_1 = -y_f$, $z_2 = -x_s$.

2 Discretisation and linearisation (EKF)

2.1 Forward Euler discretisation

With sampling time T_s,

$$\xi_{k+1} = \xi_k + T_s f(\xi_k, u_k), \tag{8}$$

$$z_k = h(\xi_k) + v_k, \qquad v_k \sim \mathcal{N}(0, R). \tag{9}$$

The Jacobian of f is

$$A_k = \frac{\partial f_d}{\partial \xi}\Big|_{\xi_k, u_k} = \begin{bmatrix} 1 & 0 & -T_s v_k \sin\theta_k \\ 0 & 1 & T_s v_k \cos\theta_k \\ 0 & 0 & 1 \end{bmatrix}, \quad B_k = \frac{\partial f_d}{\partial u} = \begin{bmatrix} T_s\cos\theta_k & 0 \\ T_s\sin\theta_k & 0 \\ 0 & T_s \end{bmatrix}.$$

For the measurement model, with wall norms $n_i = \sqrt{p_i^2 + q_i^2}$,

$$C_k = \frac{\partial h}{\partial \xi} = \begin{bmatrix} -\frac{p_1}{n_1} & -\frac{q_1}{n_1} & -\frac{p_1\dot{x}_f + q_1\dot{y}_f}{n_1} \\ -\frac{p_2}{n_2} & -\frac{q_2}{n_2} & -\frac{p_2\dot{x}_s + q_2\dot{y}_s}{n_2} \end{bmatrix}$$

with $\dot{x}_f = -\alpha\sin\theta_k$, $\dot{y}_f = \alpha\cos\theta_k$, $\dot{x}_s = -\beta\sin\theta_k - \gamma\cos\theta_k$, $\dot{y}_s = \beta\cos\theta_k - \gamma\sin\theta_k$.

2.2 Linear vs. extended Kalman filter

A linear KF assumes a fixed (A, B, C) around $\xi^\star$; the EKF recomputes A_k, C_k at each step. When the robot orientation changes appreciably (turn phase) the EKF is required; in straight, low-curvature motion the linearisation around $\xi^\star$ with $\theta^\star \approx \theta_k$ is acceptable. Process/measurement noise statistics do not change between KF and EKF, only the model fidelity does.

3 EKF design, tuning and experiments

3.1 Noise sources

- Process noise Q: unmodelled wheel slip, encoder quantisation, mismatch between commanded and actual v, ω, floor friction changes.
- Measurement noise R: IR sensor quantisation and nonlinearity, wall reflectivity, mounting offsets (α, β, γ).

3.2 Implementation notes (Arduino)

- Files: `arduino_files/ass5_ino/extended_kalman_filter.cpp` (EKF), `robot.cpp` (trajectory, state feedback, telemetry).
- Tunables: Q (`kQx,kQy,kQθ`), R (`kRz1,kRz2`), initial covariance $P_{0|0}$ and $\hat{\xi}_{0|0}$ in `resetKalmanFilter()`.
- Channel map for QRC logging: ch0–19 as listed in the experiment plan.
- Buttons: 0 control on/off, 1 EKF reset/on/off, 2 start/stop trajectory, 3 reset trajectory pointer.

3.3 Q/R sweep (Spec 3b)

Planned combinations (units: Q in $\mathrm{m}^2/\mathrm{rad}^2$, R in m^2):

Run	$Q = \mathrm{diag}(\cdot)$	$R = \mathrm{diag}(\cdot)$	File
[Nominal]	$[[1\times10^{-5}, 1\times10^{-5}, 5\times10^{-6}]]$	$[[1\times10^{-4}, 1\times10^{-4}]]$	ekf_Q1_R1.csv
[Q×5]	[...]	[...]	ekf_Q5_R1.csv
[R×5]	[...]	[...]	ekf_Q1_R5.csv
[Both×5]	[...]	[...]	ekf_Q5_R5.csv

Use the built-in trajectory (2 cm/s straight, turn, straight). Sensors are active only when `hasMeasurements=1`. Figure 1 overlays state estimates for the sweep.

[*Figure: EKF state trajectories for multiple Q/R choices*]

Figure 1: EKF state trajectories for multiple Q/R choices. [Add discussion of convergence vs. noise weighting.]

3.4 Uncertainty before/after the turn (Spec 3c)

Figure 2 shows the 95% confidence bands for the nominal Q/R. When z_1, z_2 are disabled (turn + post-turn), $P_{xx}, P_{yy}, P_{\theta\theta}$ grow monotonically due to prediction-only updates; with sensors active they contract rapidly. If both sensors were hypothetically left on during the turn, the measurement Jacobian would change because the wall normals rotate in the sensor frame; the current $h(\cdot)$ would no longer be valid.

[*Figure: State estimates with 95% CI*]

Figure 2: State estimates with 95% CI. [Describe divergence during dead-reckoning and re-convergence when measurements resume.]

4 LQR state-feedback tracking

4.1 Rotation to cart frame

With $\hat{e}_k = [x_{c,ref} - \hat{x}_c \;\; y_{c,ref} - \hat{y}_c \;\; \theta_{ref} - \hat{\theta}_c]^T$,

$$R(\hat{\theta}_{c,k}) = \begin{bmatrix} \cos\hat{\theta}_{c,k} & \sin\hat{\theta}_{c,k} & 0 \\ -\sin\hat{\theta}_{c,k} & \cos\hat{\theta}_{c,k} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \hat{e}'_k = R(\hat{\theta}_{c,k})\,\hat{e}_k. \tag{10}$$

4.2 Feedback matrix structure

Discrete error model (forward Euler, linearised at $\xi_{ref} = \xi_c$, $\dot{\xi}_{ref} = \dot{\xi}_c$):

$$A_d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -T_s v_{ref} \\ 0 & 0 & 1 \end{bmatrix}, \quad B_d = \begin{bmatrix} -T_s & 0 \\ 0 & 0 \\ 0 & -T_s \end{bmatrix}.$$

Choose $Q = \mathrm{diag}(q_x, q_y, q_\theta)$, $R = \mathrm{diag}(r_v, r_\omega)$; $Q \in \mathbb{R}^{3\times3}$ penalises position/orientation error, $R \in \mathbb{R}^{2\times2}$ penalises v, ω. Using `dlqr(Ad,Bd,Q,R)` in MATLAB gives

$$K = \begin{bmatrix} k_{v,x} & k_{v,y} & k_{v,\theta} \\ k_{\omega,x} & k_{\omega,y} & k_{\omega,\theta} \end{bmatrix} = \text{[fill from MATLAB]}.$$

The first row shapes forward velocity based on longitudinal/lateral error; the second row shapes yaw rate.

4.3 Q/R tuning experiments

Four combinations (see `assignment5_solution.m`) are logged as `lqr_*.csv`. Feedforward is disabled for this subsection to highlight feedback behaviour. Figures 3–4 summarise tracking errors and control signals.

[*Figure: Tracking errors for different Q/R weights*]

Figure 3: Tracking errors for different Q/R weights. [Comment on convergence speed and oscillations.]

[*Figure: Control signals v, ω for the same Q/R set*]

Figure 4: Control signals v, ω for the same Q/R set. [Note any saturation.]

Final chosen weights (units included) and resulting K:

Item	Value	Units	Rationale
Q	[diag(…)]	m^{-2}, rad^{-2}	[fast lateral correction, moderate yaw]
R	[diag(…)]	$(\mathrm{m/s})^{-2}$, $(\mathrm{rad/s})^{-2}$	[limit wheel saturation]
K	[$[k_{v,\cdot}; k_{\omega,\cdot}]$]	–	[copied into robot.cpp]

5 Conclusion

- EKF corrects dead-reckoning drift when IR data are available; covariance grows as expected when sensors are off.

- Increasing Q/R ratio speeds measurement convergence but amplifies noise; higher R slows correction and increases bias during turns.
- LQR gains derived from the linearised error model stabilise the trajectory; tuning Q/R trades convergence speed for actuator effort/saturation.
- [Insert final numeric choices for Q, R, P_0, K and brief justification.]