TTIC 31170: Robot Learning and Estimation (Spring 2025)

Problem Set #2

Due Date: April 24, 2025

1 Multivariate Gaussians [10 pts]

Consider the multivariate Gaussian distribution $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ over a random vector $\boldsymbol{x} \in \mathbb{R}^n$. Recall that the mean and covariance are given by $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{x}]$ and $\Sigma = \mathbb{E}\left[(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])^\top\right]$, respectively, where $\mathbb{E}[\cdot]$ denotes the expectation operation.

(a) **[1pts]** Prove that covariance matrices are symmetric.

To show symmetry, we compute the transpose with the third equality following from the fact that the transpose acts entrywise:

$$\Sigma^T = \left(E([(x-\mu)(x-\mu)^T])\right)^T = E([(x-\mu)(x-\mu)^T])^T = E[(x-\mu)(x-\mu)^T] = \Sigma$$

(b) **[2pts]** A matrix M is positive semi-definite (PSD) (represented as $M \succeq 0$) iff $\boldsymbol{x}^\top M \boldsymbol{x} \geq 0 \; \forall \boldsymbol{x} \in \mathbb{R}^n$. Prove that covariance matrices are positive semi-definite (PSD).

For any $a \in \mathbb{R}^n$,

$$a^T \Sigma a = a^T E[(x-\mu)(x-\mu)^T] a = E[a^T(x-\mu)(x-\mu)^T a] = E[(a^T(x-\mu))^2]$$

Since $(a^T(x-\mu))^2$ has real entries, it is always ≥ 0 and since expectation preserves non-negativity, we have that Σ is PSD here.

(c) **[3pts]** Suppose that we don't know the parameters $\boldsymbol{\mu}$ or Σ, but are given a set of m samples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$ drawn from the distribution. Write the equations for the maximum likelihood estimate of the mean vector based upon these samples.

Given these samples, the joint density is

$$L(\mu, \Sigma) = \prod_i^m \frac{1}{(2\pi)^{n/2}\Sigma^{1/2}} \exp(-\frac{1}{2}(x_i - \mu)^T \Sigma^{-1}(x_i - \mu))$$

The log likelihood that depends on μ is therefore

$$l(\mu) = -\frac{1}{2}\sum_{i=1}^m (x_i - \mu)^T \Sigma^{-1}(x_i - \mu)$$

$$\frac{\partial l}{\partial \mu} = -\frac{1}{2}\sum_{i=1}^{m}[-2\Sigma^{-1}(x_i - \mu)] = 0 \rightarrow m\mu = \sum_{i=1}^{m} x_i$$

Therefore,

$$\hat{\mu}_{MLE} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

Bonus (5pts): Derive the maximum likelihood estimate of the covariance matrix.

The log likelihood that depends on Σ from above would be

$$l(\Sigma) = -\frac{1}{2}m\log(\Sigma) - \frac{1}{2}\sum_{i=1}^{m}(x_i - \mu)^T\Sigma^{-1}(x_i - \mu)$$

We set $\ln\Sigma = -\ln\Lambda$ for ease of use,

$$\frac{\partial l}{\partial \Lambda} = \frac{m}{2}(\Lambda^{-1})^T - \frac{1}{2}(\sum_{i=1}^{m}(x_i - \mu)(x_i - \mu)^T)^T = 0$$

The second term is symmetric, so we can drop the transpose, giving us

$$\Lambda^{-1} = \Sigma = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)(x_i - \mu)^T$$

We can get the joint MLE by using the MLE for μ,

$$\hat{\Sigma}_{MLE} = \frac{1}{m}\sum_{i=1}^{m}(x_i - \hat{\mu}_{MLE})(x_i - \hat{\mu}_{MLE})^T$$

(d) **[2pts]** Describe the process by which you can sample from $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$. Suppose that you have access to a function that draws samples from $\boldsymbol{y} \sim \mathcal{N}(\mathbf{0}, I)$, where I is the identity matrix and y is of the same dimension as x. Hint: Recall that if $\boldsymbol{x}$ is a linear function of $\boldsymbol{y}$ and $\boldsymbol{y}$ is Gaussian, then $\boldsymbol{x}$ is also Gaussian with mean and covariance that are linear transformations of those of $\boldsymbol{y}$.

We can simple from y and then linearly transform our result with

$$x = \mu + Sy$$

where S is the matrix such that $\Sigma = SS^T$ (exists since Σ is symmetric PSD). We then have by the hint that

$$E[x] = \mu + LE[y] = \mu$$
$$Cov[x] = LCov[y]L^T = LIL^T = \Sigma$$

(e) **[2pts]** Show that a Gaussian distribution can be parameterized in terms of the information vector $\boldsymbol{\eta} = \Sigma^{-1}\boldsymbol{\mu}$ and information matrix $\Lambda = \Sigma^{-1}$.

We start with our usual form,

$$p(x) = \frac{1}{(2\pi)^{n/2}\sqrt{\Sigma}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

Consider the exponent,

$$\begin{aligned}-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) &= -\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu - \frac{1}{2}\mu^T\Sigma^{-1}\mu \\ &= -\frac{1}{2}x^T\Lambda x + x^T\eta - \frac{1}{2}\mu^T\eta\end{aligned}$$

Plug back in to see

$$\begin{aligned}p(x) &= \frac{1}{(2\pi)^{n/2}\sqrt{\Sigma}} \exp\left(-\frac{1}{2}x^T\Lambda x + x^T\eta - \frac{1}{2}\mu^T\eta\right) \\ &= \frac{1}{(2\pi)^{n/2}\sqrt{\Lambda^{-1}}} \exp\left(-\frac{1}{2}x^T\Lambda x + x^T\eta - \frac{1}{2}\eta^T\Lambda^{-1}\eta\right)\end{aligned}$$

2 Measurement Update [15 pts]

Let $\boldsymbol{x} \sim \mathcal{N}(\hat{\boldsymbol{x}}^-, \Sigma^-)$ be a Gaussian multivariate random variable with mean $\hat{\boldsymbol{x}}^-$ and covariance Σ^-. Let $\boldsymbol{Z} = H\boldsymbol{x} + \boldsymbol{v}$ be a linear measurement of $\boldsymbol{x}$ corrupted by additive Gaussian noise $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, R)$ that is independent of $\boldsymbol{x}$.

(a) **[3pts]** Derive an expression for the mean and covariance of $\boldsymbol{z}$ using their definitions in terms of expectation.

$$\begin{aligned}E[z] &= HE[x] + E[v] = H\hat{x}^- + 0 = H\hat{x}^- \\ Cov[z] &= E[(Z - E[z])(Z - E[z])^T] \\ &= E[(Hx + v - H\hat{x}^-)(Hx + v - H\hat{x}^-)^T] \\ &= E[H(x-\hat{x}^-)(x-\hat{x}^-)^T H^T] + E[vv^T] + E[H(x-\hat{x}^-)v^T] + E[v(x-\hat{x}^-)^T H^T] \\ &= H\Sigma^- H^T + R + 0 + 0 = H\Sigma^- H^T + R\end{aligned}$$

(b) **[4pts]** Derive an expression for the exponential component of $p(\boldsymbol{x} \mid \boldsymbol{z}) \propto \exp\{\cdot\}$ in terms of the parameters of the prior over $\boldsymbol{x}$ and those of the measurement.

We expect the exponential component to be of the form

$$p(x) \propto -\frac{1}{2}(x-\hat{x}^-)^T(\Sigma^-)^{-1}(x-\hat{x}^-)$$

$$p(z|x) \propto -\frac{1}{2}(z-Hx)^T R^{-1}(z-Hx)$$

Which gives us

$$p(x|z) \propto -\frac{1}{2}(x - \hat{x}^-)^T(\Sigma^-)^{-1}(x - \hat{x}^-) + (z - Hx)^T R^{-1}(z - Hx)$$

And if we simplify primarily to terms that depend on x,

$$= -\frac{1}{2}[x^T((\Sigma^-)^{-1} + H^T R^{-1} H)x - 2x^T((\Sigma^-)^{-1}\hat{x}^- + H^T R^{-1} z)] + \text{constants}$$

(c) **[6pts]** Simplify the result of part (b) to obtain an expression of the form

$$p(\boldsymbol{x} \mid \boldsymbol{z}) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \hat{\boldsymbol{x}}^+)^\top \Sigma^{+^{-1}}(\boldsymbol{x} - \hat{\boldsymbol{x}}^+)\right\} \tag{1}$$

where $\hat{\boldsymbol{x}}^+$ and Σ^+ are expressed in terms of the parameters of the prior over $\boldsymbol{x}$ and those of the measurement.

We define the information vector naturally as

$$\Lambda^+ = (\Sigma^-)^{-1} + H^T R^{-1} H$$

and

$$\eta^+ = (\Sigma^-)^{-1}\hat{x}^- + HTR^{-1}z$$

We can then see that, by essentially completing the square

$$x^T\Lambda^+ x - 2x^T\eta^+ = (x - (\Lambda^+)^{-1}\eta^+)^T\Lambda + (x - (\Lambda^+)^{-1}\eta^+) - \eta^{+^T}(\Lambda^+)^{-1}\eta^+ s$$

Which gives us, considering the x-dependent terms that

$$p(x|z) \propto \exp(-\frac{1}{2}(x - \hat{x}^+)^T(\Sigma^+)^{-1}(x - \hat{x}^+))$$

with

$$\Sigma^+ = (\Lambda^+)^{-1}$$
$$\hat{x}^+ = (\Lambda^+)^{-1}\eta^+$$

(d) **[2pts]** Can you relate the above expression to anything that you have seen in class?

We essentially just derived the measurement-update step for a Kalman filter! An equivalent form would be to consider the kalman gain as we described in class,

$$K = \bar{\Sigma}C^T(C\bar{\Sigma}C^T + Q)^{-1}$$

with $\bar{\Sigma}$ denoting Σ^-, C denoting H and Q denoting R here.

3 Kalman Filter [15 pts]

The Kalman Filter maintains a Gaussian representation of the belief, parametrized by a mean vector and covariance matrix, or a mean value and variance in the case of a univariate distribution. In class, we derived analytical expressions that describe how the mean and covariance evolve according to the prediction and measurement update steps. This problem asks you take an alternative approach to deriving these steps.

Before we begin, it is useful to consider the key operations of marginalization and conditioning as they are performed for a Gaussian distribution. Suppose that we have a Gaussian distribution over two random vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$:

$$p(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_\alpha \\ \boldsymbol{\mu}_\beta\end{bmatrix}, \begin{bmatrix}\Sigma_{\alpha\alpha} & \Sigma_{\alpha\beta} \\ \Sigma_{\beta\alpha} & \Sigma_{\beta\beta}\end{bmatrix}\right) \tag{2}$$

It is easy to show (after some refactoring of the distributions) that marginalization is easy when the distribution is parametrized in terms of the mean and covariance

$$\begin{aligned} p(\boldsymbol{\alpha}) = \int p(\boldsymbol{\alpha},\boldsymbol{\beta})\, d\boldsymbol{\beta} &= \mathcal{N}(\boldsymbol{\mu}, \Sigma) \\ \boldsymbol{\mu} &= \boldsymbol{\mu}_\alpha \\ \Sigma &= \Sigma_{\alpha\alpha} \end{aligned} \tag{3}$$

Similarly, one can show that conditioning one random vector on the other preserves Gaussianity and is equivalent to

$$\begin{aligned} p(\boldsymbol{\alpha} \mid \boldsymbol{\beta}) = \frac{p(\boldsymbol{\alpha}, \boldsymbol{\beta})}{p(\boldsymbol{\beta})} &= \mathcal{N}(\boldsymbol{\mu}, \Sigma) \\ \boldsymbol{\mu} &= \boldsymbol{\mu}_\alpha + \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_\beta) \\ \Sigma &= \Sigma_{\alpha\alpha} - \Sigma_{\alpha\beta}\Sigma_{\beta\beta}^{-1}\Sigma_{\beta\alpha} \end{aligned} \tag{4}$$

For this problem, we will assume that the process and measurement models are linear with additive, zero mean Gaussian noise:

$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + B\boldsymbol{u}_t + \boldsymbol{v}_{t-1} \qquad \boldsymbol{v}_t \sim \mathcal{N}(\mathbf{0}, R_t) \tag{5a}$$

$$\boldsymbol{z}_t = H\boldsymbol{x}_t + \boldsymbol{w}_t \qquad \boldsymbol{w}_t \sim \mathcal{N}(\mathbf{0}, Q_t) \tag{5b}$$

where $\boldsymbol{x}_t \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times k}$, and $H \in \mathbb{R}^{m\times n}$.

(a) **[4pts]** Let's assume that the state and data are one-dimensional (i.e., $n = k = m = 1$, $A = a$, $B = b$, $R_t = \sigma_R^2$, and $Q_t = \sigma_Q^2$), and that the belief at time $t-1$ is given by $p(x_{t-1} \mid z^{t-1}, u^{t-1}) = \mathcal{N}(\mu_{t-1}, \sigma_{t-1}^2)$. Derive the closed-form expression for the prediction step by (i) augmenting the state to include x_t along with x_{t-1}, i.e., $p(x_t, x_{t-1} \mid z^{t-1}, u^t)$ and then (ii) marginalizing out the previous state, i.e., $p(x_t \mid z^{t-1}, u_t) = \int p(x_t, x_{t-1} \mid z^{t-1}, u^t)\, dx_{t-1}$. Note: the augmentation step will yield a random vector of the form in Equation 2 where each of the two terms in the mean vector and four terms in the covariance matrix are scalars (e.g., $\Sigma_{t-1,t} = \rho\sigma_{t-1}\sigma_t$).

$$\begin{aligned} p(x_t, x_{t-1}|z^{t-1}, u^t) &= \mathcal{N}\left(\begin{bmatrix}\mu_t \\ \mu_{t-1}\end{bmatrix}, \begin{bmatrix}\Sigma_{x_t,x_t} & \Sigma_{x_t,x_{t-1}} \\ \Sigma_{x_{t-1},x_t} & \Sigma_{x_{t-1},x_{t-1}}\end{bmatrix}\right) \\ &= \mathcal{N}\left(\begin{bmatrix}a\mu_{t-1} + bu_t \\ \mu_{t-1}\end{bmatrix}, \begin{bmatrix}a^2\sigma_{t-1}^2 + \sigma_R^2 & a\sigma_{t-1}^2 \\ a\sigma_{t-1}^2 & \sigma_{t-1}^2\end{bmatrix}\right) \end{aligned}$$

$$p(x_t|z^{t-1}, u_t) = \mathcal{N}\left(a\mu_{t-1} + bu_t, a^2\sigma_{t-1}^2 + \sigma_R^2\right)$$

(b) **[4pts]** Now that we have the mean and variance for the predictive posterior $\overline{\text{bel}}(x_t)$, derive the closed-form expression for the mean and variance of the belief $\text{bel}(x_t) = \mathcal{N}(\mu_t, \sigma_t)$ as a result of the measurement update by (i) augmenting the distribution to include the measurement z_t and then (ii) conditioning x_t on z_t.

$$p(x_t, z_t|\mu_t, \sigma_t) = \mathcal{N}\left(\begin{bmatrix}\mu_t \\ H\mu_t\end{bmatrix}, \begin{bmatrix}\sigma_t^2 & H\sigma_t^2 \\ H\sigma_t^2 & H^2\sigma_t^2 + \sigma_Q^2\end{bmatrix}\right)$$

$$p(x_t|z_t) = \mathcal{N}\left(\mu_t + H\sigma_t^2(H^2\sigma_t^2 + \sigma_q^2)^{-1}(z_t - H\mu_t), \sigma_t^2 - H\sigma_t^2(H^2\sigma_t^2 + \sigma_Q^2)^{-1}H\sigma_t^2\right)$$

(c) **[0.5pts]** How, if at all, would the mean μ_t of the belief $\text{bel}(x_t) = \mathcal{N}(\mu_t, \sigma_t)$ differ if the variances of the process model noise and measurement model noise were both multiplied by some positive constant $\lambda > 0$, i.e., $\sigma_R^2 \to \lambda\sigma_R^2$ and $\sigma_Q^2 \to \lambda\sigma_Q^2$?

If we apply the scaling, the predictive variance becomes

$$(\sigma_t^-)^2 = a^2\sigma_{t-1}^2 + \lambda\sigma_R^2$$

and our gain becomes

$$K_t = \frac{H(a^2\sigma_{t-1}^2 + \lambda\sigma_R^2)}{H^2(a^2\sigma_{t-1}^2 + \lambda\sigma_R^2) + \lambda\sigma_Q^2} = \frac{H(\frac{a^2\sigma_{t-1}^2}{\lambda} + \sigma_R^2)}{H^2(\frac{a^2\sigma_{t-1}^2}{\lambda} + \sigma_R^2) + \sigma_Q^2}$$

We see though that since σ_{t-1}^2 is not affected by λ here, if $\lambda \to \infty$, the inner fraction divided by λ goes to 0, but the numerator and denominator will grow at the same rate. In the case that $\lambda \to 0$, we see that $K_t \to 1/H$ and the measurement dominates, which overall means that the posterior mean does not change if both the variances are scaled by the same λ.

(d) **[2pts]** Repeat the derivations from (a) and (b) for the case that the distribution is multivariate and $\boldsymbol{u}_t$ and $\boldsymbol{z}_t$ are vectors.

$$p(x_t, x_{t-1}|z^{t-1}, u^t) = \mathcal{N}\left(\begin{bmatrix}A\mu_{t-1} + Bu_t \\ \mu_{t-1}\end{bmatrix}, \begin{bmatrix}A\Sigma_{t-1}A^T + R & A\Sigma_{t-1} \\ \Sigma_{t-1}A^T & \Sigma_{t-1}\end{bmatrix}\right)$$

$$p(x_t|z_{1:t-1}, u_t) = \mathcal{N}\left(A\mu_{t-1} + Bu_t, A\Sigma_{t-1}A^T + R\right)$$

$$p(x_t, z_t|\mu_t, \sigma_t) = \mathcal{N}\left(\begin{bmatrix}\mu_t^- \\ H\mu_t^-\end{bmatrix}, \begin{bmatrix}\Sigma_t^- & \Sigma_t^- H^T \\ H\Sigma_t^- & H\Sigma_t^- H^T + Q\end{bmatrix}\right)$$

$$p(x_t|z_t) = \mathcal{N}\left(\mu_t^- + \Sigma_t^- H^T(H\Sigma_t^- H^T + Q)^{-1}(z_t - H\mu_t^-), \Sigma_t^- - \Sigma_t^- H^T(H\Sigma_t^- H^T + Q)^{-1}H\Sigma_t^-\right)$$

(e) **[4pts]** Consider a time-invariant stochastic system governed by the following linear process and measurement models subject to additive Gaussian noise

$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + B\boldsymbol{u}_t + \boldsymbol{v}_t \tag{6a}$$

$$\boldsymbol{z}_t = H\boldsymbol{x}_t + \boldsymbol{w}_t \tag{6b}$$

where $\boldsymbol{v}_t \sim \mathcal{N}(\mathbf{0}, R)$ and $\boldsymbol{w}_t \sim \mathcal{N}(\mathbf{0}, Q)$ denote zero-mean Gaussian noise.

We can estimate the posterior distribution $p(\boldsymbol{x}_t \mid \boldsymbol{z}^t)$ using a Kalman filter, which is the optimal Bayesian estimator, since the problem is linear-Gaussian. Derive a recursive expression for the posterior mean and covariance at time step t in terms of the posterior mean and covariance at time $t-1$. What is the steady state formulation?

We have

$$x_t^- = Ax_{t-1} + Bu_t$$
$$P_t^- = AP_{t-1}A^T + R$$
$$K_t = P_t^- H^T(HP_t^- H^T + Q)^{-1}$$
$$x_t = x_t^- + K_t(z_t - Hx_t^-)$$
$$P_t = (I - K_t H)P_t^-$$

For a steady state, we expect that $P_t^- \to P^-, K_t \to K, P_t \to P$, which gives us that

$$P^- = APA^T + R$$

$$P = (I - P^- H^T(HP^- H^T + Q)^{-1}H)P^-$$

Which we can combine to yield

$$\begin{aligned} P &= APA^T + R - (APA^T + R)H^T(H(APA^T + R)H^T + Q)^{-1}H(APA^T + R) \\ &= APA^T + R - APH^T(HPH^T + Q)^{-1}HPA^T \end{aligned}$$

This impliess a steady-state Kalman gain of

$$K = PH^T(HPH^T + Q)^{-1}$$

and thus

$$x_t = (Ax_{t-1} + Bu_t) + K(z_t - H(Ax_{t-1} + Bu_t))$$

(f) **[0.5pts]** Does the resulting expression for the recursive covariance calculation exhibit any interesting properties in the linear case?

It's interesting that P is completely deterministic once parameters of the model are set (z_t doesn't appear in it anywhere), which means that you could precompute the steady-state gain and run the mean update in real time.

4 Extended Kalman Filter [25 pts]

You are developing a new vacuum cleaning robot that will estimate its position and orientation in a room in order to be more efficient and precise at vacuuming. The robot's state $\boldsymbol{x}_t = [x_t\ y_t\ \theta_t]^\top$ consists of its position (x_t, y_t) and orientation θ_t. The robot is nonholonomic and its motion is described by the following nonlinear motion model:

$$\begin{aligned} x_t &= x_{t-1} + (d_t + v_{d,t})\cos(\theta_{t-1}) \\ y_t &= y_{t-1} + (d_t + v_{d,t})\sin(\theta_{t-1}) \\ \theta_t &= \theta_{t-1} + \delta\theta_t + v_{\theta,t}, \end{aligned} \tag{7}$$

where the control data $\boldsymbol{u}_t = [d_t \quad \delta\theta_t]^\top$ consists of the body-relative forward distance that the robot moved d_t and its change in orientation $\delta\theta_t$, both measured using wheel encoders. The term $\boldsymbol{v}_t = [v_{d,t} \;\; v_{\theta,t}]^\top \sim \mathcal{N}(\mathbf{0}, R)$ is zero-mean Gaussian noise that reflects errors in the measured linear and angular motion (e.g., as a result of wheel slippage on carpet and hardwood floors).

Along with the robot, you plan on shipping a sensor that will be placed on the floor of the room that the robot is vacuuming. This sensor provides a measurement $\boldsymbol{z}_t = [z_r \quad z_\theta]^\top$ of the (squared) distance z_r and bearing z_θ to the robot. The measurement model then takes the form

$$\begin{aligned} z_{r,t} &= x_t^2 + y_t^2 + w_{r,t} \\ z_{\theta,t} &= \arctan\left(\frac{y_t}{x_t}\right) + w_{\theta,t} \end{aligned} \tag{8}$$

where, $\boldsymbol{w}_t = [w_{r,t} \quad w_{\theta,t}]^\top \sim \mathcal{N}(\mathbf{0}, Q)$ is additive zero-mean Gaussian noice and, without loss of generality, we assume that the sensor is at the origin, $(x, y) = (0, 0)$.

Owing to the uncertainty in the motion and measurements, you would like to deliver the robot with software that maintains the belief over its current pose based upon an uncertain motion model and noisy observations from the ceiling-mounted sensor. The system is nonlinear and you'd like to use an Extended Kalman Filter (EKF) to maintain the belief $bel(\boldsymbol{x}_t)$ as a Gaussian distribution, which involves linearizing the process and measurement models about the current mean:

$$\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{\mu}_{t-1}, \boldsymbol{u}_t) + F(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_t) + G\boldsymbol{v}_t \qquad \boldsymbol{v}_t \sim \mathcal{N}(\mathbf{0}, R_t) \tag{9a}$$

$$\boldsymbol{z}_t = \boldsymbol{h}(\boldsymbol{\mu}_t) + H(\boldsymbol{x}_t - \boldsymbol{\mu}_t) + \boldsymbol{w}_t \qquad \boldsymbol{w}_t \sim \mathcal{N}(\mathbf{0}, Q_t) \tag{9b}$$

where the Jacobians F and H are evaluated at mean state estimate:

$$F = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x}_{t-1}=\boldsymbol{\mu}_{t-1}, \boldsymbol{u}_t)} \qquad G = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}}\right|_{(\boldsymbol{x}_{t-1}=\boldsymbol{\mu}_{t-1}, \boldsymbol{u}_t)} \qquad H = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x}_t=\boldsymbol{\mu}_t)} \tag{10}$$

Recall from class that the EKF prediction step follows as:

$$\overline{\boldsymbol{\mu}}_t = f(\boldsymbol{\mu}_{t-1}, \boldsymbol{u}_t) \tag{11a}$$

$$\overline{\Sigma}_t = F_t \Sigma_{t-1} F_t^\top + G_t R_t G_t^\top \tag{11b}$$

The subsequent measurement update step proceeds as

$$K_t = \overline{\Sigma}_t H_t^\top (H_t \overline{\Sigma}_t H_t^\top + Q_t)^{-1} \tag{12a}$$

$$\boldsymbol{\mu}_t = \overline{\boldsymbol{\mu}}_t + K_t(\boldsymbol{z}_t - \boldsymbol{h}_t(\overline{\boldsymbol{\mu}}_t)) \tag{12b}$$

$$\Sigma_t = (I - K_t H_t)\overline{\Sigma}_t \tag{12c}$$

This problem asks you to implement an EKF to estimate the robot's trajectory (pose history) given a stream of control and measurement data. For this purpose, assume that the process and measurement noise covariances, R_t and Q_t, respectively, are as follows:

$$R_t = 1.0\,E^{-1} \cdot \begin{bmatrix} 0.1 & 0.0 \\ 0.0 & 4\pi/180 \end{bmatrix} \qquad Q_t = 1.0\,E^{-1} \cdot \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & \pi/180 \end{bmatrix}$$

(a) **[3pts]** What is the closed-form expression for the three Jacobian matrices (10) evaluated about the current mean $\boldsymbol{\mu}_t = [\mu_{x,t} \; \mu_{y,t} \; \mu_{\theta,t}]^\top$?

$$F = \frac{\partial f}{\partial x} = \begin{bmatrix} 1 & 0 & -d_t \sin\hat{\theta}_{t-1} \\ 0 & 1 & d_t \cos\hat{\theta}_{t-1} \\ 0 & 0 & 1 \end{bmatrix}, G = \frac{\partial f}{\partial v} = \begin{bmatrix} \cos\hat{\theta}_{t-1} & 0 \\ \sin\hat{\theta}_{t-1} & 0 \\ 0 & 1 \end{bmatrix}$$

$$H = \frac{\partial h}{\partial x} = \begin{bmatrix} 2\hat{x}_t & 2\hat{y}_t & 0 \\ -\frac{\hat{y}_t}{\hat{x}_t^2+\hat{y}_t^2} & \frac{\hat{x}_t}{\hat{x}_t^2+\hat{y}_t^2} & 0 \end{bmatrix}$$

(b) **[16pts]** Having determined the Jacobians, we are now ready to implement the EKF to recursively estimate the robot's pose, given an initial pose estimate and a sequence of control and measurement data.

The robot vacuums the floor by following a "mow the lawn" trajectory to ensure full coverage of the room. As it operates, wheel encoders measure the forward distance traveled and its change in orientation between successive time steps, $\boldsymbol{u}^T = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_T\}$. Meanwhile, the sensor in the middle of the room provides observations of the (squared) distance and bearing to the robot at each time step, $\boldsymbol{z}^T = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_T\}$.

What do do: Update the `EKF` class defined in `EKF.py` to implement the EKF prediction and update steps to maintain the posterior distribution over the robot's pose given the history of motion and measurement data. In particular, add your code to the `prediction` and `update` functions defined in the class. These functions are called for each time step by the `run` function.

Hint: When calculating the innovation in the measurement update step (i.e., the difference between the actual and predicted observation), make sure that the error in the bearing is between $-\pi$ and π. The `EKF` class includes an `angleWrap` function that you can use for this purpose. Also, you may want to use `atan2` (link) (or `arctan2` (link) if you are using NumPy) for the measurement model (Eqn. 8) since it reasons over the quadrant (whereas `atan` and `arctan`) do not.

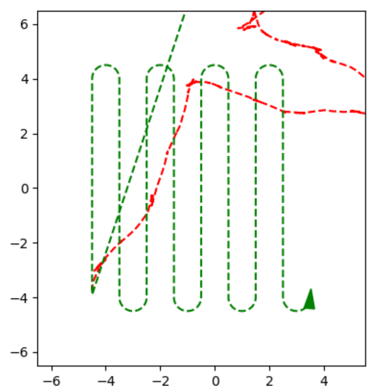
What is included: The problem set includes two text files `U.txt` and `Z.txt` that contain the control and measurement data, respectively, and an `XYT.txt` text file that includes the robot's ground-truth pose. Each row corresponds to a different time instant with the first row being $t = 1$ (i.e., the first step will be a prediction from $t = 0$ to $t = 1$, since there is no measurement at $t = 0$). The problem set includes a `Renderer.py` file that provides routines for visualizing the estimated robot pose and the associated uncertainty in the form of a level-set ellipse (both in red), the ground-truth pose (in green), as well as the final errors along with the three-sigma error bars. The included `RunEKF.py` file, which reads the data files and runs the EKF, can be called as

```
$ python RunEKF.py U.txt Z.txt XYT.txt
```

Things to try: Once you have implemented the EKF and estimated the robot's pose trajectory based upon the data, you can experiment with different settings for the initial uncertainty and the settings for the covariance matrices Q and R (though be sure to preserve symmetry and positive semi-definiteness). In particular, an important aspect of KFs and EKFs is their sensitivity to the noise parameters. If you over-estimate the accuracy of the process and measurement models/data, the filter will become *overconfident*. This occurs when the variance of the estimator (i.e., the uncertainty in the state estimates), as expressed by the covariance, is too low. This can be seen when the errors fall outside the intervals indicated by the standard deviation more often than should occur.

What to hand in: Your `EKF.py` file along with all other Python files necessary to run your code via the call to `RunEKF.py` as described above.

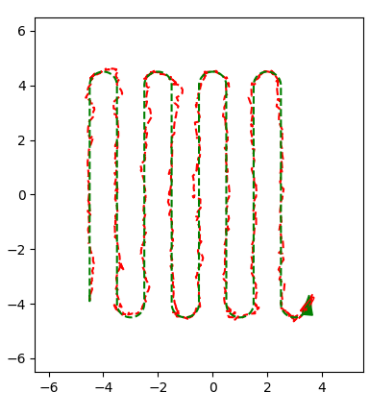
(c) **[2pts]** Run your code using only the prediction step and provide a plot that compares the estimated trajectory of the robot to the ground-truth trajectory.

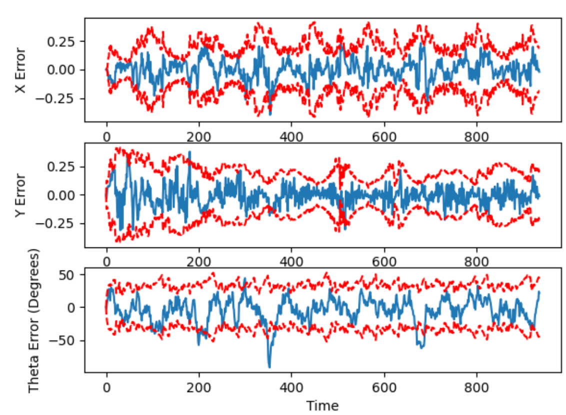


It gets wacky pretty quick

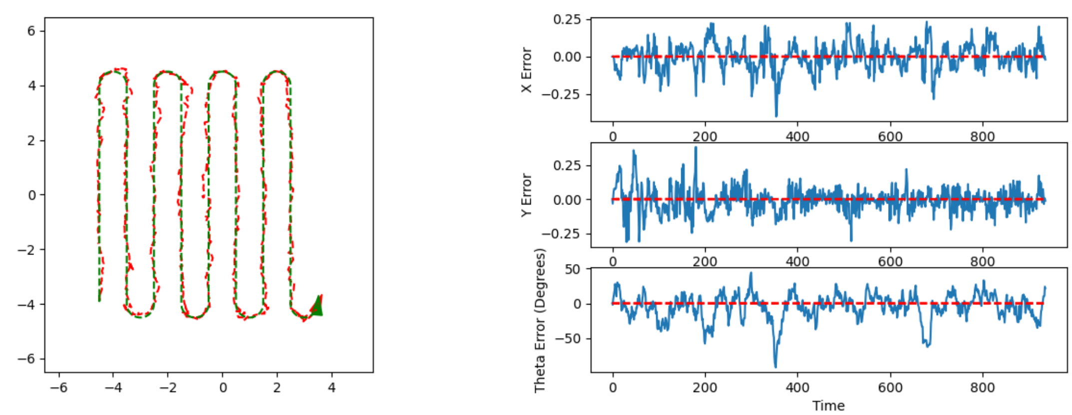
(d) **[4pts]** With the measurement step enabled, run the full EKF and provide the same plot that compares the estimated and ground-truth trajectories, as well as the three-component plot of the errors in x_t, y_t, and θ_t. Provide these plots for the default setting of the noise covariance matrices as well as one that suggests an overconfident estimate as discussed above.

With the default noise covariance matrices,





With much smaller noise covariance matrices,



5 Time and Collaboration Accounting

(a) **[1pts]** Did you work with anyone on this problem set? If so, who?

> I did not.

(b) **[1pts]** How long did you spend on this problem set?

> Roughly 10-15 hours.