TTK 4250 Assignment 2 Øystein Solbø

1 Given that

$x \in \mathbb{R}^n$, $\mathbf{x} \sim \mathcal{N}(\mu, C)$

a) $z_1 = C^{-\frac{1}{2}}(\mathbf{x} - \mu)$

Since $\mathbf{x}$ is a normal distribution, we know that a linear combination of $\mathbf{x}$ will also be a normal distribution

$$E[z_1] = E\left[C^{-\frac{1}{2}}(\mathbf{x} - \mu)\right] = C^{-\frac{1}{2}}(E[x] - \mu) = C^{-\frac{1}{2}}(\mu - \mu) = \underline{0}$$

$$\operatorname{cov}[z_1] = \operatorname{cov}\left[C^{-\frac{1}{2}}(\mathbf{x} - \mu)\right]$$

$$= C^{-\frac{1}{2}} \operatorname{cov}[\mathbf{x}] (C^{-\frac{1}{2}})^T - C^{-\frac{1}{2}} \underbrace{\operatorname{cov}[\mu]}_{=0} (C^{-\frac{1}{2}})^T$$

$$= C^{-\frac{1}{2}} C (C^{-\frac{1}{2}})^T$$

$$= C^{-\frac{1}{2}} C^{\frac{1}{2}} (C^{\frac{1}{2}})^T (C^{-\frac{1}{2}})^T = \underline{I}$$

$$\Rightarrow \underline{z_1 \sim \mathcal{N}(0, I)}$$

b) We know that $y_i = z_i^2$, ~~domain~~

This is a variable-transformation, and from basic statistics (and the book), we know that

$$p_y(y) = \sum_i p_{\mathbf{x}}(g_i^{-1}(y)) \, |\det(G_i^{-1}(y))|$$

with

$$Y = z_1^2 \Rightarrow g_1(y) = \sqrt{y}$$

$$g_2(y) = -\sqrt{y}$$

$$|\det(G_1^{-1}(y))| = \frac{1}{2\sqrt{y}}$$

$$|\det(G_2^{-1}(y))| = \frac{1}{2\sqrt{y}}$$

$$\Rightarrow p_y(y) = \sum_{i=1}^{2} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} y\right) \cdot \frac{1}{2\sqrt{y}}$$

$$= \underline{\frac{1}{\sqrt{2\pi y}} \exp\left(-\frac{1}{2} y\right)}$$

This implies that $Y \sim \chi^2$ with $r = 1$ DOF

c) $Y = (\mathbf{x}-\mu)^T C^{-1} (\mathbf{x}-\mu) = z_1^T z_1 = \sum_i Y_i$

From b) we know that $Y_i \sim \chi^2$ distribution with $r=1$

The MGF for a χ^2-distribution is given as

$$M_{Y_i}(t) = \left(\frac{1}{1-2t}\right)^{\frac{1}{2}}$$

$$\Rightarrow M_Y(t) = \prod_{i=1}^{n} M_{Y_i}(t) = \left(\frac{1}{1-2t}\right)^{\frac{n}{2}}$$

$\Rightarrow Y \sim \chi^2$ with $r = n$ DOF

$$p_Y(y) = \frac{1}{2^{\frac{n}{2}}\,\Gamma(\frac{n}{2})} y^{\frac{n}{2}-1} \exp\left(-\frac{1}{2}y\right)$$

---

2) Given that

z^r: measurement from radar, $z^r = H^r x + v^r = p(z^r|x)$

z^c: measurement from camera, $z^c = H^c x + v^c = p(z^c|x)$

State of the vessel: x

Prior state of the vessel: $\sim N(\bar{x}, P)$

$v^c \sim N(0, R^c)$

$v^r \sim N(0, R^r)$

Expected movement of the vessel: $x^+ = F x + w$, $w \sim N(0, Q)$

a) $\cancel{p(z^c|x) = \frac{p(z^c, x)}{p(x)}}$

$\cancel{= \frac{p(x|z^c)\,p(z^c)}{p(x)}}$

$p(z^c|x)$ is the likelihood of a measurement in the camera given a position in x.

This likelihood can be extracted from the information given in the task.

Given x, the likelihood

$p(z^c|x) = H^c x + v^c$

$\Rightarrow$ $p(z^c|x) \sim N(H^c x, R^c)$

b) $p(z^c, x) = p(z^c | x) \cdot p(x)$

$= \mathcal{N}(H^c x, R^c)\; p(\cancel{x_0}, \bar{P})$

I am a bit unsure if this is valid, however when looking at the equation ~~from~~ for the position x, I noticed that the position has the characteristics of "random walk" in brownian terms. Thus, we may assume that - given an initial position x_0 - the system will likely stay within a region of x_0

Size of the region is determined by Q, but I have written the covariance as $\bar{P}$, since I am a bit unsure on the transformation, since the differential equation will also depend on time

Therefore the likelihood for $p(x)$ has been written as a normal distribution: $p(x) \sim \mathcal{N}(x_0, \bar{P})$

Comments for task 2b)
After reading through the task more thoroughly, it is clear that we expect the estimate to be given as a gaussian on the form N(x; x_bar, P)

That means that I at least thought correctly, even though I did think a bit too complicated with the Brownian movement etc.

This gives us that

$P(z^c, x) = P(z^c|x)\, p(x)$

$= \mathcal{N}(H^c x, R^c)\; \mathcal{N}(x_0, \bar{P})$

Using the proof of 3.3.1, I will ~~[illegible]~~ only focus on the quadratic term

$$(x-x_0)^T \bar{P}^{-1}(x-x_0) + (z^c - H^c x)^T R^{-1}(z^c - H^c x)$$

$$= \begin{bmatrix} x - x_0 \\ z^c - H^c x \end{bmatrix}^T \begin{bmatrix} \bar{P}^{-1} & 0 \\ 0 & R^{-1} \end{bmatrix} \begin{bmatrix} x - x_0 \\ z^c - H^c x \end{bmatrix}$$

$$= \begin{bmatrix} x - x_0 \\ z^c - H^c x_0 \end{bmatrix}^T \begin{bmatrix} I & -(H^c)^T \\ 0 & I \end{bmatrix} \begin{bmatrix} \bar{P}^{-1} & 0 \\ 0 & R^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -H^c & I \end{bmatrix} \begin{bmatrix} x - x_0 \\ z^c - H^c x_0 \end{bmatrix}$$

$$= \begin{bmatrix} x - x_0 \\ z^c - H^c x_0 \end{bmatrix}^T \left(\begin{bmatrix} I & 0 \\ H^c & I \end{bmatrix} \begin{bmatrix} \bar{P} & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} I & (H^c)^T \\ 0 & I \end{bmatrix} \right)^{-1} \begin{bmatrix} x - x_0 \\ z^c - H^c x_0 \end{bmatrix}$$

$$= \begin{bmatrix} x - x_0 \\ z^c - H^c x_0 \end{bmatrix} \begin{bmatrix} P & P(H^c)^T \\ H^c P & H^c P (H^c)^T + R \end{bmatrix} \begin{bmatrix} x - x_0 \\ z^c - H^c x_0 \end{bmatrix}$$

This expression is the quadratic form of a gaussian in x and in z^c, which traps all dependence between x and z^c, which means that $p(x, z^c)$ is a gaussian.

c) Marginal distribution:

$$p(z^c) = N(Hx_0, H^c P (H^c)^T + R)$$

Conditional dependence:

$$p(x|z^c) = N(\hat{\mu}_{x|z^c}, \hat{P}_{x|z^c}) \quad \text{where}$$

$$\hat{\mu}_{x|z^c} = x_0 + P(H^c)^T (H^c P (H^c)^T + R)^{-1} (z^c - H^c x_0)$$

$$\hat{P}_{x|z^c} = P - P(H^c)^T (H^c P (H^c)^T + R)^{-1} H^c P$$

d) Marginal distribution of $p(x^+)$ should be possible to find.

Using the expression for the likelihood for x that I used in b), I get that

$$p(x) \sim N(x_0, \bar{P})$$

Since $x^+ = Fx + w$

we have a linear combination of a gaussian. This gives that

$$p(x^+) \sim N(Fx_0, F^T \bar{P} F + Q)$$

where this expression was found using normal rules for the gaussian distribution.

For the conditional distribution $p(x|z^r)$ we can use that the measurements z^r and z^c are independent. This means that

$$p(x|z^r) = p(x|z^c)\Big|_{\substack{H^c \to H^r \\ R^c \to R^r}}$$

$$\sim N(\hat{\mu}_{x|z^r}, \hat{P}_{x|z^r})$$

$$\hat{\mu}_{x|z^r} = x_0 + P(H^r)^T (H^r P (H^r)^T + R^c)^{-1} (z^r - H^r x_0)$$

$$\hat{P}_{x|z^r} = P - P(H^r)^T (H^r P (H^r)^T + R)^{-1} H^r P$$

e)

MMSE is the estimator that will try to minimize the MSE. Since we are operating with a gaussian distribution, this will become equal to the expected value.

MAP will try to find the value that maximizes the likelihood. Since we are operating with a gaussian, the maximum will occur at the expected value.

$$\widehat{x|z^c}_{MAP} = x_0 + P(H^c)^T (H^c P (H^c)^T + R^c)^{-1} (z^c - H^c x_0)$$

$$\widehat{x|z^c}_{MMSE} = \widehat{x|z^c}_{MAP}$$

f) g) and h) : python

```python
def condition_mean(x: ndarray, P: ndarray,
                   z: ndarray, R: ndarray, H: ndarray) -> ndarray:
    """compute conditional mean

    Args:
        x (ndarray): initial state
        P (ndarray): initial state covariance
        z (ndarray): measurement
        R (ndarray): measurement covariance
        H (ndarray): measurement matrix i.e. z = H @ x + error

    Returns:
        cond_mean (ndarray): conditioned mean (state)
    """
    PH = np.matmul(P, np.transpose(H)) # HP = (PH)^T
    W = np.matmul(PH, np.linalg.inv(np.matmul(H, PH) + R))
    z_HX = z - np.matmul(H, x)

    return x + np.matmul(W, z_HX)
```

```python
def condition_cov(P: ndarray, R: ndarray, H: ndarray) -> ndarray:
    """compute conditional covariance

    Args:
        P (ndarray): covariance of state estimate
        R (ndarray): covariance of measurement
        H (ndarray): measurement matrix

    Returns:
        ndarray: the conditioned covariance
    """
    PH = np.matmul(P, np.transpose(H)) # HP = (PH)^T
    W = np.matmul(PH, np.linalg.inv(np.matmul(H, PH) + R))
    W_HP = np.matmul(W, np.transpose(PH))

    return P - W_HP
```

```python
def get_task_2f(x_bar: ndarray, P: ndarray,
                z_c: ndarray, R_c: ndarray, H_c: ndarray,
                z_r: ndarray, R_r: ndarray, H_r: ndarray
                ):
    """get state estimates after receiving measurement c or measurement r

    Args:
        x_bar (ndarray): initial state estimate
        P (ndarray): covariance of x_bar
        z_c (ndarray): measurement c
        R_c (ndarray): covariance of measurement c
        H_c (ndarray): measurement matrix i.e. z_c = H_c @ x + error
        z_r (ndarray): measurement r
        R_r (ndarray): covariance of measurement r
        H_r (ndarray): measurement matrix i.e. z_r + H_c @ x + error

    Returns:
        x_bar_c (ndarray): state estimate after measurement c
        P_c (ndarray): covariance of x_bar_c
        x_bar_r (ndarray): state estimate after measurement r
        P_r (ndarray): covariance of x_bar_r
    """
    x_bar_c = condition_mean(x_bar, P, z_c, R_c, H_c)
    P_c = condition_cov(P, R_c, H_c)

    x_bar_r = condition_mean(x_bar, P, z_r, R_r, H_r)
    P_r = condition_cov(P, R_r, H_r)

    return x_bar_c, P_c, x_bar_r, P_r
```

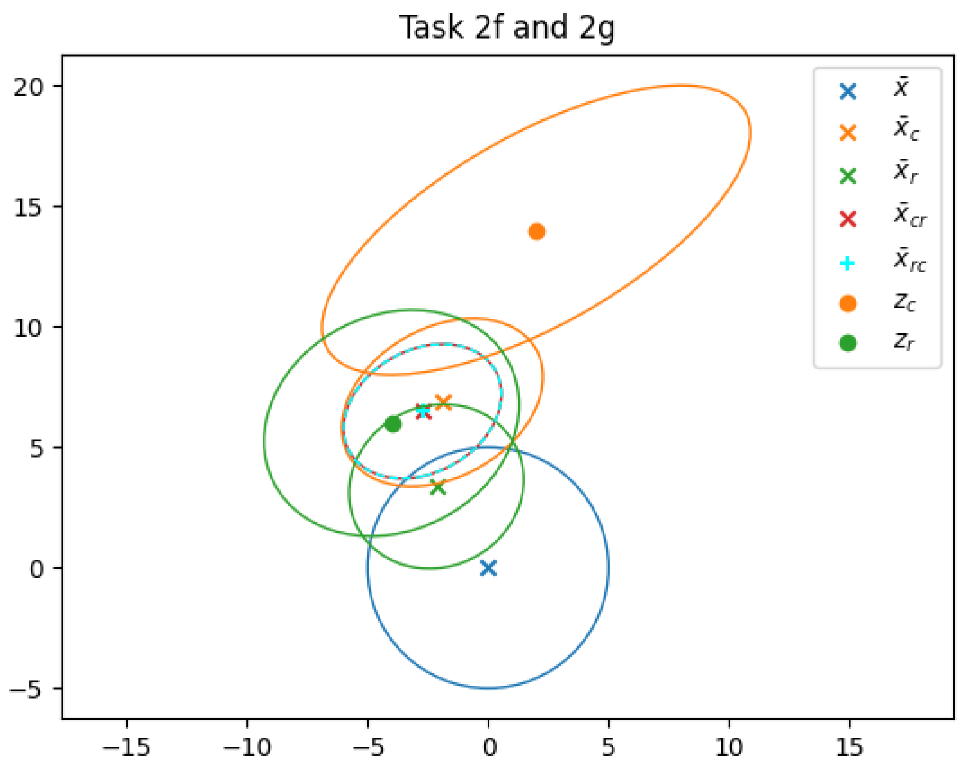
```python
def get_task_2g(x_bar_c: ndarray, P_c: ndarray,
                x_bar_r: ndarray, P_r: ndarray,
                z_c: ndarray, R_c: ndarray, H_c: ndarray,
                z_r: ndarray, R_r: ndarray, H_r: ndarray):
    """get state estimates after receiving measurement c and measurement r

    Args:
        x_bar_c (ndarray): state estimate after receiving measurement c
        P_c (ndarray): covariance of x_bar_c
        x_bar_r (ndarray): state estimate after receiving measurement r
        P_r (ndarray): covariance of x_bar_r
        z_c (ndarray): measurement c
        R_c (ndarray): covariance of measurement c
        H_c (ndarray): measurement matrix i.e. z_c = H_c @ x + error
        z_r (ndarray): measurement r
        R_r (ndarray): covariance of measurement r
        H_r (ndarray): measurement matrix i.e. z_r = H_r @ x + error

    Returns:
        x_bar_cr (ndarray): state estimate after receiving z_c then z_r
        P_cr (ndarray): covariance of x_bar_cr
        x_bar_rc (ndarray): state estimate after receiving z_r then z_c
        P_rc (ndarray): covariance of x_bar_rc
    """
    # In theory, these values should be identical
    x_bar_cr = condition_mean(x_bar_c, P_c, z_r, R_r, H_r)
    P_cr = condition_cov(P_c, R_r, H_r)

    x_bar_rc = condition_mean(x_bar_r, P_r, z_c, R_c, H_c)
    P_rc = condition_cov(P_r, R_c, H_c)

    return x_bar_cr, P_cr, x_bar_rc, P_rc
```



Comments for task 2f)
For task 2f), the mean and the covariance of the camera and the radar is the most important. When studying the results from the camera, we can clearly see that the new measurement "moves" the estimated position and its covariance to a place that is plausible for both the predicted and the measured state. The latter observation is to see that the covariance of the new estimate includes part of the covcariance from the camera and the original estimate. This means that it will not assume that the new data is totally correct, however it will use the new data in combination with the old data/estimate to get a new and hopefully better system estimate.

A similar observation could be made for the radar, where an estimate is closer to the initial prediciton. That means that the new estimate is inside of the covariance for the original estimate and the covariance of the radar measurment.

Comments for task 2g)

For task 2g), you can see that the estimate for x|z_rc is equivalent to the estimate for x|z_cr. This means that the order of the data has no effect on the end result, such that the data could be fed into a KF (or other bayesian filter) when it is available. As long as you either has a time-stamp of the data or can guarantee that it is not too old to show an outdated system state, it could be used directly in the filter.

```python
def get_task_2h(x_bar_rc: ndarray, P_rc: ndarray):
    """get the probability that the boat is above the line

    Args:
        x_bar_rc (ndarray): state
        P_rc (ndarray): covariance

    Returns:
        prob_above_line: the probability that the boat is above the line
    """

    """
    Explenation on how I thought about the process:
    We have a transformation where we would like to check
    x2 - x1 > 5

    which is equivalent to

    [-1 1] x > 5

    Could just check to find a Y that satisfies
    Y = [-1 1] X

    which due to X being a gaussian gives that Y is a gaussian satisfying
    Y ~ N(y; [-1, 1] E[x_hat], [-1, 1] Cov[x_hat] [-1, 1]^T)

    Using the linearity of the gaussian, we know that Y is then also
    a gaussian determined by
    Y ~ N([-1 1]x_hat, [-1 1] P [-1 1]^T)

    where one could find the likelihood of Y > 5 as

    P(Y > 5) = 1 - P(Y <= 5) = 1 - phi(5 / (sqrt(50)))

    but when calculating this by hand, I get a likelihood of roughly 0.2398
    which is wrong according to the solution.task2.get_task_2h which gives
    0.8710308654292277
    """
    # TODO replace this with your own code
    prob_above_line = solution.task2.get_task_2h(x_bar_rc, P_rc)

    return prob_above_line
```

3 a) $\mathcal{N}^{-1}(x; a, B) =$

$$= \exp\left\{-\frac{1}{2} n \ln(2\pi) + \frac{1}{2}\ln(|B|) - \frac{1}{2} a^T B^{-1} a + a^T x - \frac{1}{2} x^T B x\right\}$$

$\mathcal{N}^{-1}(y; Cx, D) =$

$$= \exp\left\{-\frac{1}{2} n \ln(2\pi) + \frac{1}{2}\ln(|D|) - \frac{1}{2} x^T C^T D^{-1} C x + x^T C^T y - \frac{1}{2} y^T D y\right\}$$

This gives that

$\mathcal{N}^{-1}(x; a, B)\, \mathcal{N}^{-1}(y; Cx, D)$

$$= \exp\left\{-n \ln(2\pi) + \frac{1}{2}\ln(|B|) + \frac{1}{2}\ln(|D|) - \frac{1}{2} a^T B^{-1} a\right\}$$

$$\cdot \exp\left\{a^T x - \frac{1}{2} x^T B x - \frac{1}{2} x^T C^T D^{-1} C x + x^T C^T y - \frac{1}{2} y^T D y\right\}$$

$$= K \exp\left\{-\frac{1}{2} x^T (B + C^T D^{-1} C) x + a^T x + x^T C^T y - \frac{1}{2} y^T D y\right\}$$

where $K = \exp\left\{-n \ln(2\pi) + \frac{1}{2}\ln(|B|) + \frac{1}{2}\ln(|D|) - \frac{1}{2} a^T B^{-1} a\right\}$

are the terms independent from x and y.

Based on the terms in x and y, we have that the expression can be simplified as the gaussian in canonical form

$$\mathcal{N}^{-1}\left(\begin{bmatrix} x \\ y \end{bmatrix}; \begin{bmatrix} a \\ 0 \end{bmatrix}, \begin{bmatrix} B + C^T D^{-1} C & -C^T \\ -C & D \end{bmatrix}\right)$$

b) From 3.4.1, we know that the marginal distribution of y is

$$\eta_* = \eta_b - \Lambda_{yx} \Lambda_{xx}^{-1} \eta_{ex}$$

$$= 0 + C (B + C^T D^{-1} C)^{-1} a$$

$$= \underline{C (B + C^T D^{-1} C)^{-1} a}$$

$$\Lambda_* = \Lambda_{yy} - \Lambda_{yx} \Lambda_{xx}^{-1} \Lambda_{xy}$$

$$= \underline{D - C (B + C^T D^{-1} C)^{-1} C^T}$$

$\Rightarrow \mathcal{N}^{-1}(y; C (B + C^T D^{-1} C)^{-1} a, D - C (B + C^T D^{-1} C)^{-1} C^T)$

is the marginal distribution for y.

Has to be an error in the task.

c) By using thrm. 3.4.1, we get that

$$\eta_{x|y} = \eta_a - \Lambda_{xy} y = a + C^T y$$

$$\Lambda_{x|y} = \Lambda_{xx} = B + C^T D^{-1} C$$

which gives that the conditional distribution of x given y is

$$N^{-1}(x;\ a + C^T y,\ B + C^T D^{-1} C)$$

d) From 3.21 we know that

$$N^{-1}(x; \eta, \Lambda) = N(x; \Lambda^{-1}\eta, \Lambda^{-1})$$

From c) we have that

$$\eta = a + C^T y$$

$$\Lambda = B + C^T D^{-1} C$$

From (6) we have that by comparing it to 3.10 that

$$\bar{P} = B^{-1} \qquad D = R^{-1}$$

$$\bar{x} = B^{-1} a \qquad C = R^{-1} H$$

$$\hat{P} = \Lambda^{-1} \quad \text{(end result)}$$

This gives us that

$$\hat{P}^{-1} = \Lambda$$

$$= B + C^T D^{-1} C$$

$$= \bar{P}^{-1} + H^T (R^{-1})^T R R^{-1} H$$

$$= \bar{P}^{-1} + H^T (R^{-1})^T H$$

$$R^T = R$$

$$\underline{= \bar{P}^{-1} + H^T R^{-1} H}$$