Started on	Friday, 29 October 2021, 12:00 PM
State	Finished
Completed on	Friday, 29 October 2021, 3:43 PM
Time taken	3 hours 42 mins
Grade	Not yet graded

Question **1**

Marked out of 6.00 Complete

Instructions: Please write your answers to the text fields, you can (but don't need to) use attachments to clarify your answers. Use equations to clarify your answers if necessary. Do not copy-paste equations or other things from say LibreOffice, instead use attachments or properly included figures. Finally submit your answers. Please note that "Finish attempt..." does not yet finish the exam, after pressing that you can still edit. But then finally, in the end, you need to press "Submit all and finish" -- that can be only done once. If you encounter problems during the exam, either email me at simo.sarkka@aalto.fi or call me: +358 40 757 0730.

Use equations to clarify your answers if necessary.

Describe briefly the following terms and what is their significance in sensor fusion:

a) measurement model (1p)

b) cost function (1p)

c) estimation algorithm (1p)

d) gradient descent (1p)

e) Gaussian distribution (1p)

f) line search (1p)

a) measurement model (1p)

The measurement model associates the readings(measurements) that you can get from a sensor with other factors like noise and the function of parameters.

The measurement model takes the form of the equation below(equation 1).

$$Measurement = Function\ of\ Parameter(s) + Noise$$

$$y_n = g_n(\boldsymbol{x}) + r_n$$

$$y_n \to Measurment, \quad g_n(\boldsymbol{x}) \to function\ of\ parameters, \quad r_n \to noise$$

Equation 1

In sensor fusion, it is the main equation used to evaluate the variable of interest.

b) cost function (1p)

A cost function associates performance to a model. In order to do this criterion to optimise the system needs to be established. For example, minimising the error. The goal of the cost function is to return a value of the error at a certain point. By choosing a value that is closest to 0, the minimum cost is chosen. This is important for sensor fusion to get the best predicted value of a certain variable of interest.

c) estimation algorithm (1p)

In sensor fusion, an estimation algorithm takes both the sensor values and the model of the system and produces an estimate of the quantities of interest.

This is the algorithm that can combine multiple sensor values, account for uncertainty and produce the optimum value of the variable of interest.

An example of an estimation algorithm is minimising the least-squares cost function.

d) gradient descent (1p)

While we can use least squares to estimate an optimum value for a linear model this approach can’t be taken for nonlinear ones. The goal of gradient descent is to minimise nonlinear functions. In sensor fusion its importance is within minimising a cost function of a nonlinear sensor model. This is done by repeatedly taking steps, in the negative gradient direction, of the cost function until a minimum is reached.

In summary, gradient descent is an iterative algorithm that can be used to find the minima of a cost function.

e) Gaussian distribution (1p)

Gaussian distribution is another term of normal distribution is mathematics. It is a function which represents the distribution of many random variables. In sensor fusion it relates to modelling noise of a sensor. Noise following the probability density function.

f) line search (1p)

Line search is an optimization technique used to find the optimum value to pick next in an iterative algorithm. In sensor fusion, it is used in iterative algorithms such as gradient descent and the gauss newton method. In those algorithms, the step size needs to be chosen carefully so it isn't too small that a minimum is never reached and not too big that is skips over the minima. Line search iteratively takes different values of the step size and finds a step size big enough to quickly converge, but still small enough to not overshoot the value.

Question **2**

Marked out of 6.00 Complete

a) Explain what are (ordinary) least squares, weighted least squares, and regularized least squares methods and what are their differences. (2p)

b) Assume that we have the following model:

$$y_1 = p_1 + p_2 + c_1 + c_2 + r_1,$$
$$y_2 = p_1 - p_2 + 2c_3 + r_2,$$

where y_1,y_2 are sensor measurements, p_1,p_2 are the unknowns, c_1,c_2,c_3 are known constants, and r_1,r_2 are measurement noises. Rewrite the model in form

$$\mathbf{y} = \mathbf{G}\mathbf{x} + \mathbf{b} + \mathbf{r},$$

where **x** is a vector of the unknowns. (2p)

c) Write down the (ordinary) linear squares solution of **x** in terms of **G**, **y**, and **b**. (1p)

d) Assume that r_1,r_2 are independent zero mean-Gaussian noises with variances s^2. What kind of weighting matrix **R** would you choose for weighted least squares and why? (1p)

A)

Ordinary Least Squares

Ordinary least squares is a method used to approximate a solution for a set of data values that gives the best fit. It does this but minimising the sum of the square residuals made for each data point. The residuals represent how far the data points are from the predicted value. So by minimising the sum of them you are finding the best-predicted value to represent the data.

$$J_{LS}(x) = \sum_{n=1}^{N} (y_n - g_n(x))^2$$

Weighted least squares

Weighted least squares adds another term to least squares. The objective is to include confidence in sensor readings. For example, if some measurements were more accurate than others then you want to make it so that more accurate readings have more of an effect than the less accurate readings.

$$J_{WLS}(x) = \sum_{n=1}^{N} w_n (y_n - g_n(x))^2$$

Regularized least squares

Regularised least squares are adding even more information to the system. If for example the result of weighted least squares gave two possible solutions for a set of data points and you had prior information that one of those solutions is incorrect. Then regularised least squares is a way of including that. It chooses a preferred value. This is done by adding a penalty term to the estimate. The equation looks like this:

B)

$$y_1 = p_1 + p_2 + c_1 + c_2 + r_1$$

$$y_2 = p_1 - p_2 + 2c_3 + r_2$$

$$\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{b} + \boldsymbol{r}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \begin{bmatrix} c_1 + c_2 \\ 2c_3 \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$$

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}, \; b = \begin{bmatrix} c_1 + c_2 \\ 2c_3 \end{bmatrix}, \quad r = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$$

C)

$$J_{LS}(x) = \sum_{n=1}^{N} (y_n - gx))^2$$

$$\tilde{y} = \begin{bmatrix} y_1 - b_1 \\ y_2 - b_2 \end{bmatrix},$$

$$\sum_{n=1}^{N} (\tilde{y}_n - Gx))^2$$

$$\sum_{n=1}^{N} (y_n - b_n - Gx))^2$$

D)

If r1 and r2 are zero mean gaussian noises, then it means their expectation is 0. This means noise shouldn’t affect the system. Therefore, the weighting R should be 1 so that it does not affect the system.

R = 1

Question **3**

Marked out of 6.00 Complete

a) Explain the principle of Gauss-Newton algorithm. What kind of models can it be applied to? What kind of approximation is formed at each step? (2p)

b) Explain the principle of Levenberg-Marquardt algorithm. What kind of models can it be applied to? What kind of approximation is formed at each step? What is the difference to Gauss-Newton? (2p)

c) Explain the principles of exact and inexact line search methods. Which methods can they be applied to? What are the benefits and disadvantages of using line search methods? (2p)

a)

Gauss-Newton algorithm linearizes a nonlinear measurement model around a single point. This is then minimised using least-squares like a linear model to find the next best value. It then uses the next best values in the function and iteratively does the same process over again. Of course, this algorithm also suffers from the same problem as gradient descent at the step size/ scaling factor needs to be chosen appropriately to avoid overestimation.

It can be applied to nonlinear models.

b)

Levenberg-Marquardt algorithm is the regularised version of gauss-newton. You could describe it as trying to get the best of both worlds: gradient descent and gauss newton. Where gradient descent is good at getting to the target area quickly and gauss newton is good at getting straight to the minimum once in the area.

It can be applied to nonlinear problems.

c)

Exact linear search finds the point at which the cost function is at its minimum value given a range of step size /scaling factors. Inexact line search is saying that the exact minimum doesn't actually need to be determined so long as the cost function is decreasing by a sufficient amount.

These methods can be applied to iterative algorithms, often used to solve nonlinear problems such as gradient descent, gauss newton, Levenberg–Marquardt.

Benefits of line search methods:

· Eliminates the possibility to overestimate a cost function and never get the minimum

· More reliable

· Possibility of taking fewer iterations

Disadvantages of line search methods:

· Algorithm speed processing speed per iteration will be slower

· Increased complexity

· Could take more iterations

Question **4**

Marked out of 6.00 Complete

Consider the cost function

$$J(x) = (1.1 - x - \sin(x))^2,$$

where *x* is a scalar.

a) Write down a model of the form

$$y = g(x) + r,$$

that is, define *y*, *g*(*x*), and *r* such that the cost function *J*(*x*) would be the nonlinear least-squares cost function for this model. (3p)

b) Write down the pseudo-code for minimizing the cost function *J*(*x*) by using the Gauss-Newton algorithm. (3p)

a)

$$y = g(x) + r$$

$$J(x) = (1.1 - x - \sin(x))^2$$

$$J_{LS}(x) = \sum_{n=1}^{N} (y_n - g_n(x))^2$$

$$y_n - g_n(x)$$

$$y_n = y, \qquad g(x) = -1.1 - x - \sin(x)$$

$$y = -1.1 - x_n - \sin(x_n) + r$$

b)

Require initial parameter $\hat{x}^{(0)}$, data y, function g(x) and Jacobian Gx

1. Set i <- 0
2. Repeat
3. Calculate the update direction
4. $\Delta\hat{x}^{(i+1)} = \left(G_x^T\left(\hat{x}^{(i)}\right)R^{-1}G_x\left(\hat{x}^{(i)}\right)\right)^{-1} G_x^T\left(\hat{x}^{(i)}\right)R^{-1}(y - g(\hat{x}^{(i)}))$
5. Calculate
6. $\hat{x}^{(i+1)} = \hat{x}^{(i)} + \Delta\hat{x}^{(i+1)}$
7. Set i <- i+1
8. Until converged
9. Set $\hat{x}_{WLS} = \hat{x}^{(i)}$

Question **5**

Marked out of 6.00 Complete

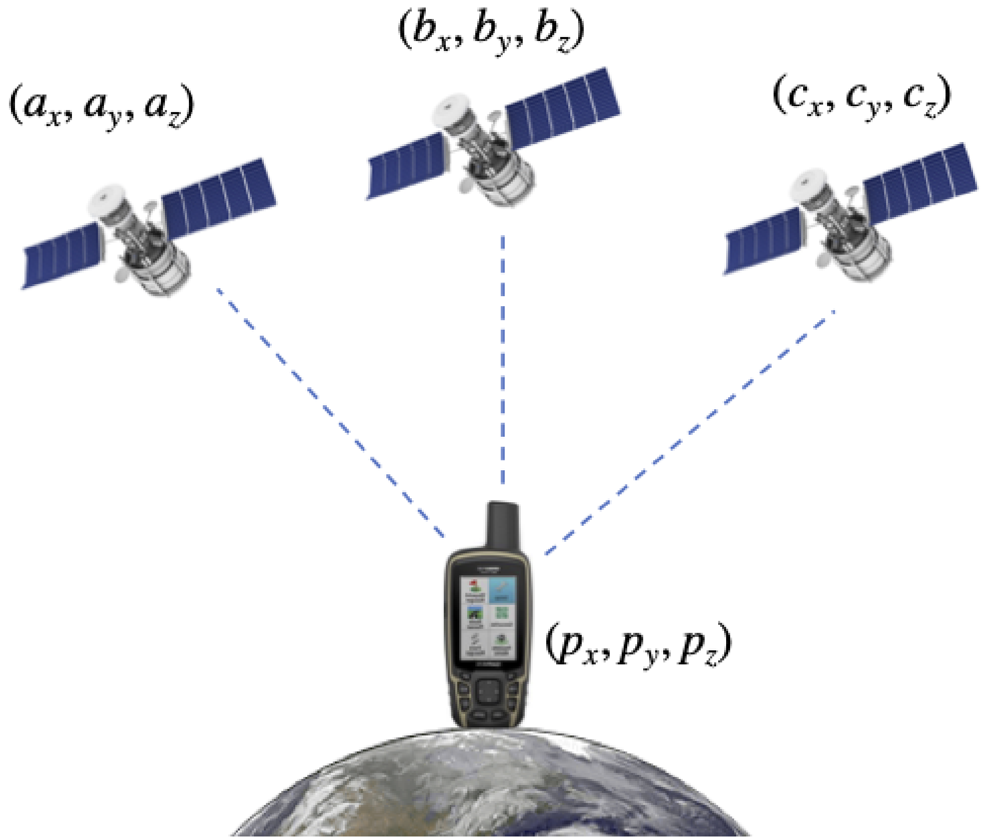
Suppose that we wish to determine the position (p_x,p_y,p_z) of a satellite receiver (see Figure below) by measuring the distances from the receiver to three satellites with positions (a_x,a_y,a_z), (b_x,b_y,b_z), and (c_x,c_y,c_z).

a) Write the model in a vector notation with the following form:

$$\mathbf{y} = \mathbf{g}(\mathbf{x}) + \mathbf{r}.$$

What should the vector of unknowns **x** contain? What form does function **g**(**x**) take? What kind of noise **r** assumption would be suitable in this case? (4p)

b) Identify a suitable cost function for estimating the receiver location from the noisy measurements. Which optimization algorithms would be suitable for determining the position estimate? (2p)



a)

Measurement models take the form

$$y_1 = y_{ap} = \sqrt{(a_x - p_x)^2 + \left(a_y - p_y\right)^2 + (a_z - p_z)^2} + r_1$$

$$y_2 = y_{bp} = \sqrt{(b_x - p_x)^2 + \left(b_y - p_y\right)^2 + (b_z - p_z)^2} + r_2$$

$$y_3 = y_{cp} = \sqrt{(c_x - p_x)^2 + \left(c_y - p_y\right)^2 + (c_z - p_z)^2} + r_3$$

So in the form

$$y = g(x) + r$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \sqrt{(a_x - p_x)^2 + \left(a_y - p_y\right)^2 + (a_z - p_z)^2} \\ \sqrt{(b_x - p_x)^2 + \left(b_y - p_y\right)^2 + (b_z - p_z)^2} \\ \sqrt{(c_x - p_x)^2 + \left(c_y - p_y\right)^2 + (c_z - p_z)^2} \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$

$$y = \sqrt{(\alpha_m^x - p_x)^2 + \left(\alpha_m^y - p_y\right)^2 + (\alpha_m^z - p_z)^2} + r$$

Where m is 1 to 3 since there are three satellites.

The vector of unknowns x should contain positions (x,y and z) of all three satellites.

The assumption is that the noise will be high in this system.

b)

A cost function using weighted least squares to account for the noise.

$$J(x) = W_n(y_n - g_n(x))^2$$

Where:

$$g_n(x) = \sqrt{(\alpha_m^x - p_x)^2 + (\alpha_m^y - p_y)^2 + (\alpha_m^z - p_z)^2} + r$$

And Wn is inversely proportional to the variance of the noise which is

$$W_n = \frac{1}{{\sigma_{r,n}}^2}$$

It's a nonlinear cost function therefore either gradient descent, gauss newton and Levenberg-Marquardt could be used. Line search could be used with them.

◄ Rehearsal Exam

Exam grades ►