Filtering and Identification notes

Pradeep Janakiraman [*]
3ME department
TU Delft, The Netherlands
pradeepjanakiraman@student.tudelft.nl

9 November 2020

[*] TU Delft student from 2019-2021, notes written in Quarter 2 of AY 2020-2021

Contents

1 Motivation behind the course

1.1 Identifying linear static models

The first part of the course will involve trying find the parameters of a function $f(\theta)$ that will be used to model a real system with measurements y. The course will focus on linear models however, so $f(\theta) = F\theta$. The problem is then reduced to:

$$\min_{\theta} \|y - F\theta\|^2 \tag{1}$$

1.2 Filtering

In the second part of the course, instead of identifying static parameters of a model, we will be identifying states of a system that vary with time. In this scenario, we are aware of the models that define the system and we want to make a prediction of the next state. We are also given the current state and current output, and have access to the history of states and outputs. The problem can be written as:

$$x_{k+1} = Ax_k + Bu_k \tag{2}$$

$$y_k = Cx_k + Du_k \tag{3}$$

1.3 Identification

In the last part of the course, we face the situation where we do not have any knowledge of the model that defines the system. We are then tasked to find the (A, B, C, D) matrices or parametric models in the form of $f(\theta)$.

2 Lecture 1

2.1 Sensor fusion

The first part of the course covers the identification of static models based on output data and filtering to estimate the state of a system through time. An application that is closely related to these topics is sensor fusion. Sensor fusion is the process of using different sensors to obtain more information about a quantity, which could've been obtained from just one sensor. THis is done to get better estimates and information about that quantity. Some examples of sensor fusion include:

- Motion capture of a ice skater.
- Motion capture of the actor playing Ted.
- Sensors on a drone to find the position and orientation.

An important problem is the issue of find the information on the exact position and orientation of an object as it moves through time. Theoretically, the figure below shows how this can be done by using inertial sensors that measure angular acceleration and linear acceleration.

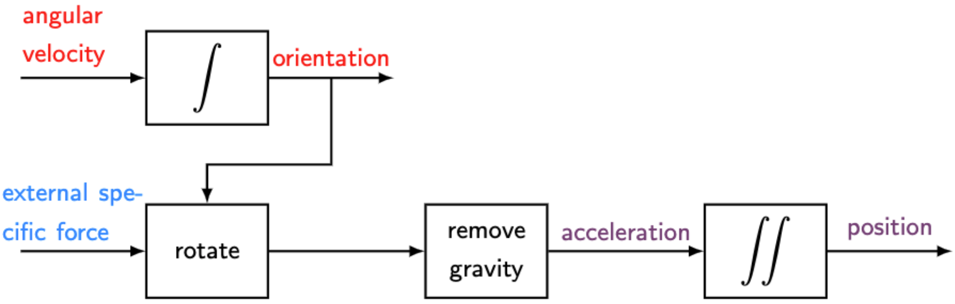

Figure 1: Process to obtain position and orientation of an object

The problem with this apporach is that since integration is involved, any errors, no matter how small, will eventually get summed up and lead to large errors in the position of the object. This phenomenon is called *integration drift*. To prevent such an eventuality, more information is needed about the system. This can be provided through additional sensors, information about the operation of the sensor itself or physical knowledge about the system. An example of how this can be done is seen in the following example.

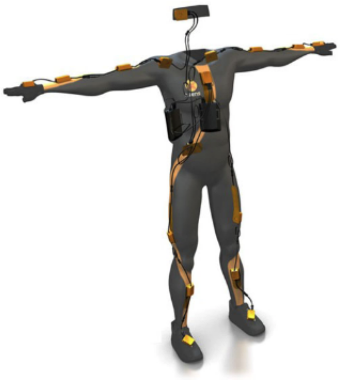

Figure 2: Example of sensor fusion

From the above setup there could a few issues that arise. The first as already discussed is the integration drift that might occur due to errors in measurement by the sensor. The second is that we have to assume that the sensors placed on the body are rigidly fixed in place when in reality that is not the case. Since the body is made of soft, deformable tissue, it is likely that the sensors have some relative motion with respect to the body. To counter some of these problems we can do the following:

- Realise that the joints of the body are fixed in place and they have no relative motion with respect to the body. Adding this as a constraint to estimating the position of each limb would then lead to more realistic results.

- Idefintiy parameters for the model that are currently unknown. How much to the sensors move by when a particular movement is executed? Could that be modeled? Could the bias in sensor be modelled by certain unknown parameters?

Soliving these problems would require the knowledge of both filtering technoiques and static model parameter identification.

2.2 System identification

Subspace identification is the process of finding mathematical models for dynamical systems based on input-output behavior of the system. There are two classes of the identification problem, which are white-box identification and black-blox identification.

With white box identification we are already aware of the differential equations behind the physical process. We can think of textbook examples of pendulum, mass-spring-damper systems, electrical ciruits to which we can apply newton's laws or kirchkoff's laws to find the mathematical model. However, in order to control the system and be precise, we also need to know the model parameters. These are things like mass, spring stiffness, resistance, voltage, etc. White box identification is then the science of looking at the input-output behaviour to identify these physical parameters.

There are also other kinds of systems for which we cannot write down a differential equation because either because they are too complex or there are many variables that affect the dynamics of the system. Black box identification then involves assuming a certain structure for the model of these systems and trying to find the parameters that best suit the input-output behavior.

2.3 Parametric system identification

Both the two cases above then have the following set up: You have input and output data given by $(u_n, y_n)_{n=1}^N$. We have a function $f(.,.,.|\theta)$ that we suspect models the system (black box), or know for sure it models the system (white box) and we have to identify the parameters in θ such that,

$$y_n \approx \hat{y}_n = f(u, y, e|\theta) \tag{4}$$

We see from the above, that f can depend on past inputs and outputs. And are not neccesarily confined to just the previous input-output pair. The question then becomes:

- What kind of function should f be? There are so many options.
- Should f be linear or non-linear?
- How much data from the past should be part of f?

2.4 Linear system identification

In this course, the focus will be on linear system identification. Under this there are two main approaches.

2.4.1 Prediction error methods

For this method we assume that the input-output and error-output relationship can be modeled with transfer funcitons. They would take the form of,

$$\hat{y}_n = \hat{G}(q|\theta)u_n + \hat{H}(q|\theta)e_n \tag{5}$$

Finding the parameters in θ would then require solving a non-linear optimisation problem. The problem is also likely to be non-convex which means there could be multiple local minima and the solution we obtain could largely depend on our initial guess.

2.4.2 Subspace methods

For subspace methods, instead if system identification we use state space methods to model the system. Particular, this course will use the following formulation:

$$x_{n+1} = Ax_n + Bu_n + Ke_n \tag{6}$$
$$y_n = Cx_n + Du_n + e_n \tag{7}$$

The advantage of the subspace identification method is that solving a non-linear optimisation problem is not required. Just through the use of linear algebra (SVD decomposition, RQ factorisation) we can arrive at the (A, B, C, D) matrices and identify the system.

3 Math preliminaries

3.1 Some linear algebra revision

We first consider the least squares problem which is

$$\min_{\theta} \epsilon^\top \epsilon \qquad \text{Where,} \quad y = F\theta + \epsilon \tag{8}$$

y is a vector of outputs, F is the parameter matrix and θ is the parameter vector that is to be found. ϵ is the error between the model $F\theta$ and the true system behavior y. Suppose that, $y \in \mathbb{R}^3, F \in \mathbb{R}^{3\times 2}, \theta \in \mathbb{R}^2$.

$$F = \begin{bmatrix} f_1 & f_2 \end{bmatrix} \tag{9}$$

where $f_1, f_2 \in \mathbb{R}^3$. We can then visualise $F\theta$ as the linear combinations of the vectors f_1, f_2 and all the possible linear combinations would fill up an entire plane. The solution to the least squares problem is then to combine f_1, f_2 such that the resultant vector $\theta_1 f_1 + \theta_2 f_2$ is the projection of the measurement vector y onto the plane F. The figure below shows this visualisation.

There are some interesting properties of the solution to the least squares problem that is worth nothing.

Lemma 3.1 *$F^\top F$ is a symmetric matrix and positive (semi-)definitie.*

To prove the above, we can first write out F to be the following,

$$F = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ f_{m1} & f_{m2} & \cdots & f_{mn} \end{bmatrix} \qquad F^\top = \begin{bmatrix} f_{11} & f_{21} & \cdots & f_{m1} \\ f_{12} & f_{22} & \cdots & f_{m2} \\ \vdots & \vdots & \vdots & \vdots \\ f_{1n} & f_{2n} & \cdots & f_{mn} \end{bmatrix} \tag{10}$$

We then consider the first column and row of $F^\top F$, and compare the results. Multiplying from the left, we have the column picture, where we multiply the $F^\top$ with the first column of F.

$$F^\top f_1 = \begin{bmatrix} f_{11}^2 + f_{21}^2 + \cdots + f_{m1}^2 \\ f_{11}f_{12} + f_{21}f_{22} + \cdots + f_{m1}f_{m2} \\ \vdots \\ f_{11}f_{1n} + f_{21}f_{2n} + \cdots + f_{m1}f_{mn} \end{bmatrix} \tag{11}$$

We can then look at the row picture where multiply F from the left with the first row of $F^\top$,

$$f_1^\top F = \begin{bmatrix} f_{11}^2 + f_{21}^2 + \cdots + f_{m1}^2 & f_{11}f_{12} + f_{21}f_{22} + \cdots + f_{m1}f_{m2} & \cdots & f_{11}f_{1n} + f_{21}f_{2n} + \cdots + f_{m1}f_{mn} \end{bmatrix} \tag{12}$$

We observe that $(f_1^\top F)^\top = F^\top f_1$, extending this to all columns we can conclude that,

$$(F^\top F)^\top = F^\top F \tag{13}$$

Which is the definition of a symmetric matrix. To see that $F^\top F$ is at the very least postive semi-definition, we go back to the definition of postive semi-definiteness,

$$\text{If,} \qquad A \geq 0 \tag{14}$$

$$z^\top A z \geq 0 \tag{15}$$

If we take $A = F^\top F$, thwn we for positive semi-definitess we need to satisfy,

$$z^\top F^\top F z \geq 0 \tag{16}$$

$$\|Fz\|_2^2 \geq 0 \tag{17}$$

Which is certainly true, since we are talking about the 2-norm of a vector. We also know from the discussion we had earler that of F is not full rank, there is no unique soltion for θ since the columns of F that are linearly dependent do not contribute anything. That implies that $(F^\top F)^{-1}$ is not inverible and that $F^\top F$ is for sure positive semi-definite and has 0 as an eigenvalue. From the eigenvalue decomposition, we know that:

$$A = S\Omega S^{-1} \tag{18}$$

Where,

$$\Omega = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \qquad \text{Where,} \quad Ae_i = \lambda_i e_i \tag{19}$$

If we again take $A = F^\top F$ and take the inverse,

$$(F^\top F)^{-1} = S\Omega^{-1}S^{-1} \tag{20}$$

If there is a $\lambda_i = 0$ then Ω^{-1} would not exist which corresponds to the least squares problem having no solution.

3.2 Random variables and processes

A random variable θ is a variable whose values depend on a random phenomenon and can take on unpredictable values. There are fortunately some properties of random variables that allow us to work with them despite thier unpredictability. We usuually work with the *expected* value of the random variable's properties.

Mean: We observe the random variable in N number of experiments and record the values of θ. Summing he values up and taking the average out of all the experiments provides the mean of the random variable θ.

$$E\Big[\theta\Big] = \frac{1}{N}\sum_{i=1} N\theta_i \tag{21}$$

Covariance: It is the measure of how spread out the values of theta are likely to be relative to the mean. The correct way to visualise this is to picture an ellipse where the size of the ellipse is a measure of covariance. The off diagnal terms play a role in shaping the matrix.

$$E\Big[(\theta-\mu_\theta)(\theta-\mu_\theta)^\top\Big] = P_\theta = \frac{1}{N}\sum_{i=1}^{N}(\theta_i-\mu_\theta)(\theta_i-\mu_\theta)^\top \tag{22}$$

Correlation: It is a special case of the measure of covariance when the mean is zero.

$$E\Big[\theta\theta^\top\Big] = R_\theta = \frac{1}{N}\sum_{i=1}^{N}\theta\theta^\top \tag{23}$$

A random process $x(k)$ comprises of a collection of random variables $(k = 1, 2, ..., N)$ after the completion of one experiement. In particular a random process is said to be stationary, if:

- The mean of $x(k)$ remains constant over the entire process.

$$\mu_x(k) = \mu_x \tag{24}$$

- The auto-correlation function only depends on the lag between two random variables.

$$R_x(k,l) = R_x(k-l) = R_x(\tau) = E\Big[x(k)x(k-\tau)\Big] \tag{25}$$

- The variance $E\Big[(x(k)-\mu_x)^2\Big]$ is finite.

The difference between variance, covariance, auto-covariance, cross-covariance needs to be clarified. Variance applies to only the random variable itself (that is, $E\Big[(x(k)-\mu_x)^2\Big]$ while covariance is a measure of how the different components of the random variable depend on each other (shape of the ellipse) $E\Big[(x(k)-\mu_x)(x(k)-\mu_x)^\top\Big]$. Auto-covariance is the measure of how the random variables depend on the lag between when they were measures (that is, $E\Big[(x(k)-\mu_x)(x(k-\tau)-\mu_x)^\top\Big]$ and cross-covariance measure how two random variables depend on each other (that is, $E\Big[(x(k)-\mu_x)(y(k-\tau)-\mu_y)^\top\Big]$.

A special kind of random process is the white noise signal. It has the property that it's mean is 0 and that the the signals at received at a certain time do not depend on the signals recieved at nother time. We can summarise the property of white noise signals as $e \backsim \mathcal{N}(0, \mathcal{I})$.