Linearization of vector systems

Overview of the flow

Assuming you are already somewhat comfortable with scalar linearization (lecture and discussion 7A), vector linearization is just a matter of some more detail, and the use of a basic and simple result from multivariate calculus. The basic idea is to express every vector in terms of scalar components, do scalar linearization of functions with multiple scalar arguments, and re-arrange in vector/matrix form (which then looks a great deal cleaner and gives us a linear state-space equation system).

The flow (which we went over in lecture 7B) is summarized below.

(a) The real action can be boiled down to simply understanding how to linearize a scalar function of two scalar arguments: $f(x,y)$ around (x^*,y^*), where x^* is an expansion point for x, and y^* is an expansion point for y. Everything later emerges from just extending and using this. The linearization is

$$f(x^*+\delta x, y^*+\delta y) \simeq f(x^*,y^*) + \left.\frac{\partial f}{\partial x}\right|_{(x^*,y^*)} \delta x + \left.\frac{\partial f}{\partial y}\right|_{(x^*,y^*)} \delta y. \tag{1}$$

(The above is a result from multivariate calculus known as the total derivative formula. See the video links in **Piazza post @667** if you want to dive into this in more detail.)

Just as for the scalar case, the smaller δx and δu are, the more accurate the above linear[1] approximation becomes. It is perfect if $\delta x = \delta u = 0$.

(b) Now, on to linearizing $\vec{f}(\vec{x})$. Say $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, and $\vec{f}(\vec{x})$ is expanded out as

$$\vec{f}(\vec{x}) \equiv \begin{bmatrix} f_1(\vec{x}_1,\vec{x}_2) \\ f_2(\vec{x}_1,\vec{x}_2) \end{bmatrix}. \tag{2}$$

Say you want to linearize this around $\vec{x}^* \triangleq \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix}$. All you need to do is expand each of $f_1(\cdot,\cdot)$ and $f_2(\cdot,\cdot)$ using (1), and put the expansion in vector form. The result is

$$\vec{f}(\vec{x}^*+\delta\vec{x}) \simeq \vec{f}(\vec{x}^*) + J_{\vec{x}}\,\delta\vec{x}, \tag{3}$$

[1] More precisely, (1) is **affine**, which simply means linear + constant.

where $J_{\vec{x}}$ is termed the **Jacobian matrix**[2] (of $\vec{f}$ with respect to $\vec{x}$ at $\vec{x}^*$) and is given by

$$J_{\vec{x}} \triangleq \begin{bmatrix} \left.\frac{\partial f_1}{\partial x_1}\right|_{(x_1^*,x_2^*)} & \left.\frac{\partial f_1}{\partial x_2}\right|_{(x_1^*,x_2^*)} \\ \left.\frac{\partial f_2}{\partial x_1}\right|_{(x_1^*,x_2^*)} & \left.\frac{\partial f_2}{\partial x_2}\right|_{(x_1^*,x_2^*)} \end{bmatrix}. \tag{4}$$

(c) Finally, the above can be applied to linearizing a nonlinear state-space system

$$\frac{d\vec{x}(t)}{dt} = \vec{f}(\vec{x},\vec{u}). \tag{5}$$

The procedure is exactly analogous to the scalar case: first choose (or you are given) a DC input $\vec{u}^*$, using which you find a DC operating point $\vec{x}^*$, by solving[3] $\vec{0} = \vec{f}(\vec{x}^*,\vec{u}^*)$. Then, extend (3) to include a second vector argument $\vec{u}$ and replace $\vec{f}(\cdot,\cdot)$ in (5) with the linearization to get

$$\frac{d\delta\vec{x}(t)}{dt} \simeq J_{\vec{x}}\ \delta\vec{x}(t) + J_{\vec{u}}\ \delta\vec{u}(t), \tag{6}$$

where $J_{\vec{u}}$ is the Jacobian of $\vec{f}$ with respect to $\vec{u}$.

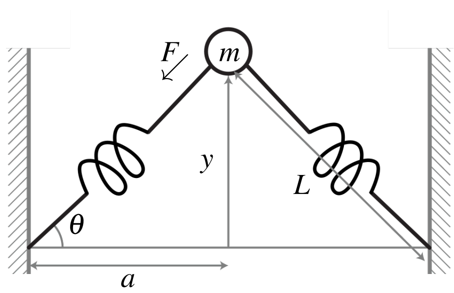
The above is an overview. All of the approximate equalities above can be made into exact equalities by the introduction of the fudge-factor disturbance $\vec{w}(t)$ into which all approximation errors get folded in, along with actual disturbances that come from nature.

It might not make much sense unless you have attended/watched lecture 7B and generally followed what happened there. If you haven't, please look through the lecture video first before going on further, using the notes in **Piazza posts @649 and @673**, which expand on the above summary.

Questions

1. Linearization

Consider a mass attached to two springs:



We assume that each spring is linear with spring constant k and resting length L. We want to build a state space model that describes how the displacement y of the mass from the spring base evolves. The differential equation modeling this system is $\frac{d^2y}{dt} = -\frac{2k}{m}(y - L\frac{y}{\sqrt{y^2+a^2}})$.

[2]also called the gradient of $\vec{f}$ (with respect to $\vec{x}$ at $\vec{x}^*$) and denoted alternatively as $\nabla_{\vec{x}}\vec{f}$. Confusingly, the gradient in many contexts is the *transpose* of this Jacobian matrix. The only way to realistically make sense of what is going on is to remember what you are trying to do — find a matrix that represents the impact on the function of a bunch of small changes in the components of $\vec{x}$.

[3]Solving for the DC operating point $\vec{x}^*$ is usually quite difficult, indeed largely impossible by hand. In fact it is pretty difficult even numerically, even after at least 50 years of research trying to solve the problem. Especially for big/practical systems.

(a) Write this model in state space form $\frac{d}{dt}\vec{x} = \vec{f}(\vec{x})$.

(b) Find the equilibrium of the state-space model. You can assume $L < a$.

(c) Linearize your model about the equilibrium.

Contributors:

- Jaijeet Roychowdhury.
- John Maidens.