State-space analysis of control systems: Part I

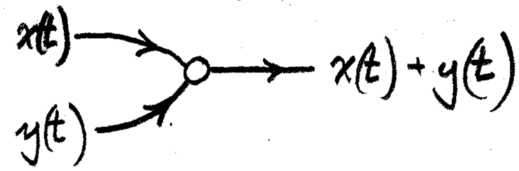
Why a different approach?

- Using a state-variable approach gives us a straightforward way to analyze MIMO (multiple-input, multiple output) systems.
- A state variable model helps us understand some complex general concepts about control systems, such as controllability and observability.

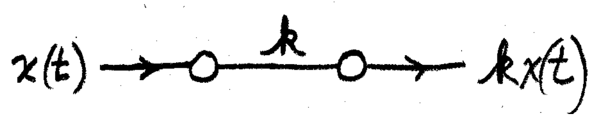
Signal-flow graphs

In order to introduce some key ideas in state-variable system modeling, we need to use signal-flow graphs. These graphs allow for only three types of operations:

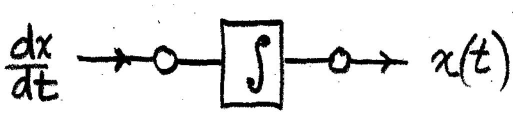
- Addition of incoming signals at a node: Here the node is a small circle. Any signal flowing out of a node is the sum of all the signals flowing in.



- Amplification by a fixed factor, positive or negative: the gain is just written above the signal path.



- Integration: This is denoted by a box containing the integral sign, or 1/s.



The state variable model for any linear system is a set of first-order differential equations. Therefore, ***the outputs of each integrator in a signal-flow graph of a system are the states of that system***. For any system, an infinite number of signal graphs are possible, but only a few are of interest. Let's look at some processes for obtaining a signal-flow graph for a given system. This is best done by means of a specific example. Consider the transfer function, and its equivalent differential equation:

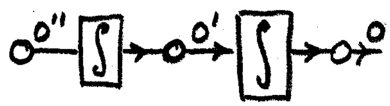
$$G(s) = \frac{\beta_0 + \beta_1 s}{\alpha_0 + \alpha_1 s + \alpha_2 s^2} \Leftrightarrow \alpha_0 o(t) + \alpha_1 \frac{do}{dt} + \alpha_2 \frac{d^2 o}{dt^2} = \beta_0 i(t) + \beta_1 \frac{di}{dt}$$

since this is a second-order system, its state model will have two states, which will appear at the outputs of the two integrators in any signal flow graph. Next, we will consider three forms of the state model for this system, each of which results from a slightly different approach:

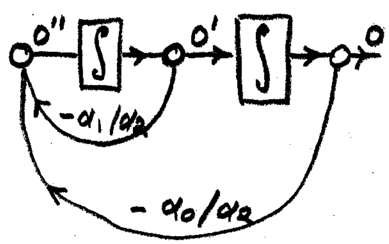
Control canonical form: This form gets its name from the fact that all of the states are fed back to the input in the signal flow graph. For this state-variable model, solve the differential equation for the highest-order derivative of the output as

$$\frac{d^2o}{dt^2} = -\frac{\alpha_1}{\alpha_2}\frac{do}{dt} - \frac{\alpha_0}{\alpha_2}\,o + \frac{\beta_0}{\alpha_2}\,i + \frac{\beta_1}{\alpha_2}\frac{di}{dt}$$

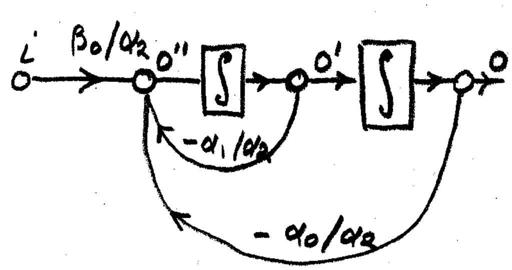
(This solution is for a particular second order system, but you can see how to extend this idea to a higher-order system) To begin to draw the signal graph, we connect two integrators in cascade, and identify the output and its two derivatives (using primes to denote differentiation), which gives



Note that the signals are drawn immediately to the right of (on the output side of) the nodes. The differential equation for the highest derivative of *o(t)* identifies this derivative as the sum of several terms. Two of these terms depend on lower-order derivatives of the output, and one depends on the input. You draw the signal paths corresponding to the lower output derivatives as feedback loops as shown here



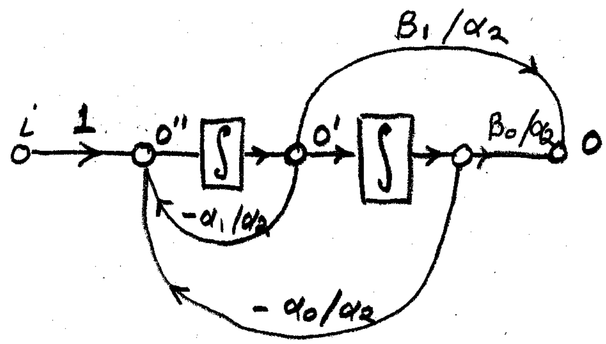
This diagram obviously represents, in a graphic way, the first two terms on the right side of the equation for *o''(t)*. Just like any diagram of a "signal-processing" system, the input should be on the left and the output should be on the right. Putting the input into the diagram gives



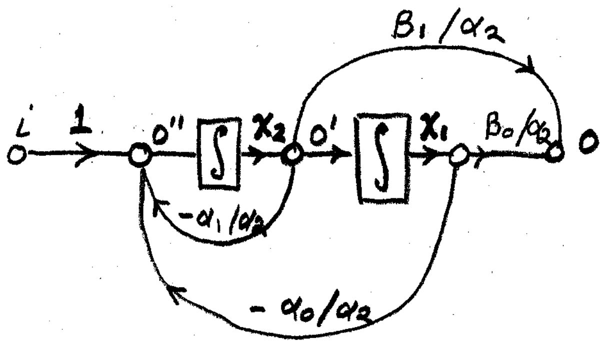
All terms of the equation for *o''(t)* but the last one are now represented in the diagram. In order to use only integration, addition and multiplication in out signal graph, we have to represent terms which are proportional to first (and higher) derivatives in the following way: suppose we rewrite the transfer function as

$$\frac{\beta_0 + \beta_1 s}{\alpha_0 + \alpha_1 s + \alpha_2 s^2} = \frac{\beta_0}{\alpha_0 + \alpha_1 s + \alpha_2 s^2} + \frac{\beta_1}{\alpha_0 + \alpha_1 s + \alpha_2 s^2} s$$

This clearly shows that the output arises from two terms, and the first of these terms could be obtained from the signal graph we have so far. The second term is proportional to the derivative of the first one. The signal graph has a node from which we can get the derivative of the output, namely *o'(t)*. To finish our signal graph, we just move the input gain to the output side, and take an additional signal proportional to *o'(t)* to the new output via a feed-forward loop with the required proportionality constant. The result is



You can now identify each state with an integrator output, to yield the states x_1 and x_2, as shown next:



The first derivative of each state is the signal just back on the upstream side of each integrator. Thus, we can write two differential state equations and an additional equation called the "output equation", which relates the states to the system output, as

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = -\frac{\alpha_1}{\alpha_2} x_2 - \frac{\alpha_0}{\alpha_2} x_1 + 1i$$

$$o(t) = \frac{\beta_0}{\alpha_2} x_1 + \frac{\beta_1}{\alpha_2} x_2$$

These equations can be organized into a compact set of matrix equations which look like this:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \overset{\mathbf{A}}{\begin{bmatrix} 0 & 1 \\ -\alpha_0/\alpha_2 & -\alpha_1/\alpha_2 \end{bmatrix}}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \overset{\mathbf{B}}{\begin{bmatrix} 0 \\ 1 \end{bmatrix}} i$$

$$o = \underset{\mathbf{C}}{\begin{bmatrix} \frac{\beta_0}{\alpha_2}, & \frac{\beta_1}{\alpha_2} \end{bmatrix}}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underset{\mathbf{D}}{\begin{bmatrix} 0 \\ 0 \end{bmatrix}} i$$

And have the general form,

$$\frac{d\mathbf{x}}{dt} = \mathbf{Ax} + \mathbf{Bi}$$
$$\mathbf{o} = \mathbf{Cx} + \mathbf{Di}$$

In this general form for the state equation model, if there are *n* states, *r* inputs, and *p* outputs, then the matrices will have the following names and forms (rows x columns):

- System matrix, **A**: $n \times n$,
- Input matrix, **B**: $n \times r$,
- Output matrix, **C**: $p \times n$,
- Feed-forward matrix, **D**: $p \times r$.

Note on transfer function normalization: Notice how the highest-order α for this transfer function keeps appearing in denominators everywhere. Transfer function coefficients are not unique, and you can always divide numerator and denominator of any transfer function by the highest-order α to obtain a normalized transfer function of the form

$$G(s) = \frac{\beta_0 + \ldots \beta_m s^m}{\alpha_0 + \alpha_1 s + \ldots s^n},$$

where the highest- order α is unity. This obviously makes for cleaner matrices. If you normalize the transfer function first, the control canonical form state equations look like this (for a normalized 4th-order system. Extension to higher order is straightforward):

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & -\alpha_3 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \beta_0 & \beta_1 & \beta_2 & \beta_3 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

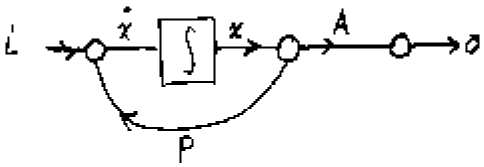
and **D** contains only zeros.

<u>Modal (or modal canonical) form</u>

Suppose you converted the second order transfer function we are using as an example here into pole residue form,

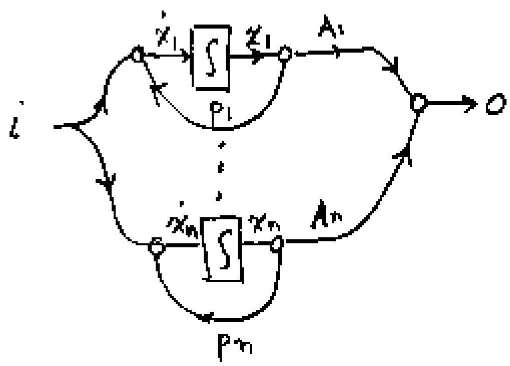
$$G(s) = \frac{A_1}{s-p_1} + \frac{A_2}{s-p_2} \qquad \text{poles at} \quad s = p_1, p_2$$

You could convert any one of the terms into the following simple signal flow graph and state- and output equation:



$$\frac{dx}{dt} = px + i$$
$$o = Ax$$

For a transfer function with *n* distinct poles (two poles with the same values can be artificially separated by some very small difference), you would get the following graph and state- and output equation:



$$\frac{d\bar{x}}{dt} = \begin{bmatrix} p_1 & & \\ & \ddots & \\ & & p_n \end{bmatrix}\bar{x} + \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} i$$

$$o = \begin{bmatrix} A_1 & \cdots & A_n \end{bmatrix}\bar{x}$$

The interesting thing about this form is the appearance of the system poles as the elements in a diagonalized system matrix. This says that the eigenvalues of the system matrix (regardless of what form it is in) are the poles of the transfer function. Also, note that in this form, the coefficients in the equations will generally be complex. This was not the case for the control canonical form earlier, since the coefficients in the equations there were ratios of (real) transfer function coefficients.

Here is the general matrix modal form for a fourth-order system:

$$\mathbf{A} = \begin{bmatrix} p_1 & 0 & 0 & 0 \\ 0 & p_2 & 0 & 0 \\ 0 & 0 & p_3 & 0 \\ 0 & 0 & 0 & p_4 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} A_1 & A_2 & A_3 & A_4 \end{bmatrix}$$

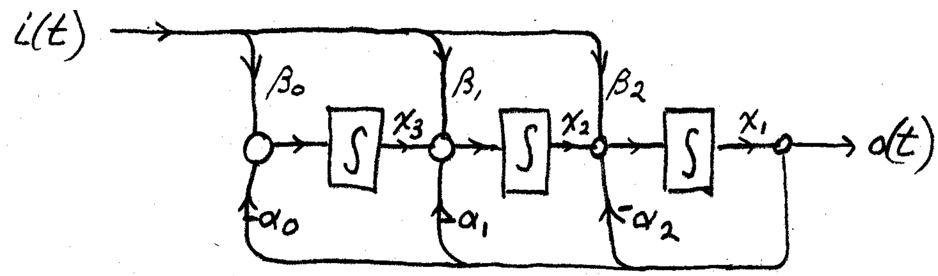
with **D** containing all zeros.

<u>Observer canonical form</u>

There is one more special form of the state equations that is of interest. In this case the feedback is from the output to the state variables. For this form, we start with a normalized , 3rd-order transfer function,

$$G(s) = \frac{\beta_0 + \beta_1 s + \beta_2 s^2}{\alpha_0 + \alpha_1 s + \alpha_2 s^2 + s^3},$$

and draw the following signal-flow graph:



which leads to the matrix forms,

$$\mathbf{A} = \begin{bmatrix} -\alpha_2 & 1 & 0 \\ -\alpha_1 & 0 & 1 \\ -\alpha_0 & 0 & 0 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} \beta_2 \\ \beta_1 \\ \beta_0 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix},$$

and **D** contains all zeros.

For the control canonical form, we justified the form of the signal-flow graph by solving the differential equation for the highest-order derivative of the output. For the modal form, we did this by first looking at a single term of the residue-pole form of the transfer function, then adding similar terms. For the observer canonical form, suppose we multiply the normalized transfer function by s raised to that power, thereby creating a rational polynomial in (1/s) as follows:

$$G(s) = \frac{O(s)}{I(s)} = \frac{\beta_0 s^{-3} + \beta_1 s^{-2} + \beta_2 s^{-1}}{\alpha_0 s^{-3} + \alpha_1 s^{-2} + \alpha_2 s^{-1} + 1} \quad .$$

This leads to the following LaPlace transform equation relating the input, *I(s)* to the output *O(s)*:

$$O(s) = (\beta_0 I(s) - \alpha_0 O(s))s^{-3} + (\beta_1 I(s) - \alpha_1 O(s))s^{-2} + (\beta_2 I(s) - \alpha_2 O(s))s^{-1}.$$

This equation corresponds exactly to the signal-flow graph: The first term on the right side gets integrated three times, the second twice, and the third once.

<u>Transformation to other state-space representations</u>

How are the different state-space representations related, other than in representing the same physical system? If a linear system can be represented by two state vectors, **u** and **v**, the two vectors must be related through a transformation **T** by

u=**Tv**, and **v**=$\mathbf{T}^{-1}$**u**

The inverse of **T** must exist, that is **T** must be non-singular.

We can use this relation to transform the state and output equations as well, for example, if with one state vector,

$$\dot{\mathbf{u}} = \mathbf{Gu} + \mathbf{Hi}, \mathbf{o} = \mathbf{Pu} + \mathbf{Qi},$$

then using the transformation, **T**,

$$\dot{\mathbf{u}} = \mathbf{GTv} + \mathbf{Hi} \text{ and } \mathbf{o} = \mathbf{PTv} + \mathbf{Qi}.$$

Pre-multiplying by the inverse of T gives a new set of state equations,

$$\dot{\mathbf{v}} = \mathbf{T}^{-1}\dot{\mathbf{u}} = \underbrace{\mathbf{T}^{-1}\mathbf{GT}}_{\mathbf{A}}\mathbf{v} + \underbrace{\mathbf{T}^{-1}\mathbf{H}}_{\mathbf{B}}\mathbf{i} \text{ and } \mathbf{o} = \underbrace{\mathbf{PT}}_{\mathbf{C}}\mathbf{v} + \mathbf{Qi}.$$

where **A**, **B**, and **C** are respectively the new system, input, and output matrices for the system using the state vector **v**.

The availability of the transformation, **T**, means that an infinite number of state representations for a system are possible. Only a few of these are interesting.

Solving State Equations: LaPlace domain

Since we've gone to all this work to develop a wide variety of available state equations, it might be interesting at this point to actually solve one! If we assume zero initial conditions (usually the case in control system design) and take the LaPlace transform of both sides of the state equation, we get

$$s\mathbf{X}(s) = \mathbf{AX}(s) + \mathbf{BI}(s),$$

which is solvable as

$$\mathbf{X}(s) = [\mathbf{sI} - \mathbf{A}]^{-1}\mathbf{BI}(s)$$

(note here that **I**(s) is the LaPlace transform of the input(s), while I is the identity matrix). This almost looks like a transfer function, but one more thing is needed: the output(s) in terms of the states, which we get from the output (lower) equation,

$$\frac{d\mathbf{x}}{dt} = \mathbf{Ax} + \mathbf{Bi}$$

$$\mathbf{o} = \mathbf{Cx} + \mathbf{Di}$$

LaPlace transforming the output equation and substituting in the result for **X**(s) above gives a matrix version of an input-output relationship,

$$\mathbf{O}(s) = \left\{\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}\right\}\mathbf{I}(s).$$

The quantity in {} is really a transfer function in matrix form. For example, in a system with three outputs and two inputs, it says

$$\begin{bmatrix} O_1(s) \\ O_2(s) \\ O_3(s) \end{bmatrix} = \begin{bmatrix} H_{11}(s) & H_{12}(s) \\ H_{21}(s) & H_{22}(s) \\ H_{31}(s) & H_{32}(s) \end{bmatrix} \begin{bmatrix} I_1(s) \\ I_2(s) \end{bmatrix}.$$

The elements of the matrix, **H**, show the effect that each input has on each output. Since the system is linear, superposition applies and the effects of each input add. This way of organizing the treatment of multiple-input, multiple-output (MIMO) systems is one characteristic that makes state variables so useful.