Vehicle Dynamics and Simulation

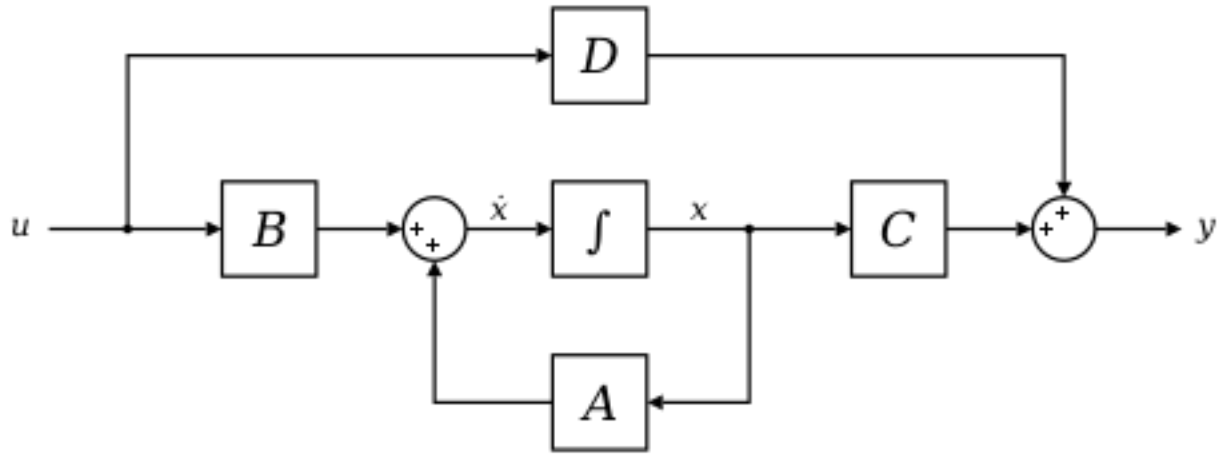
Linearity and State Space

Dr B Mason

Loughborough University

Lecture overview

- Linear systems
- State-space representation



Linear Systems

- In mathematics a system is said to be linear if*;

$$y = ax$$

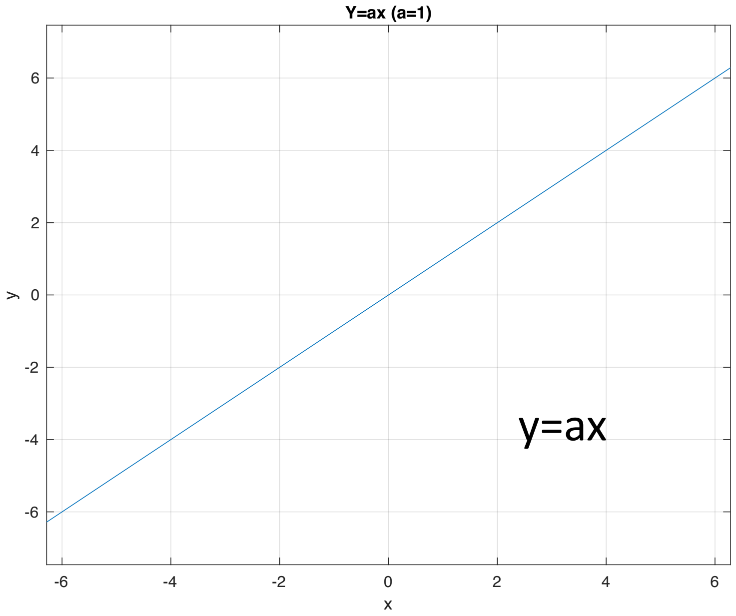
Or more generally

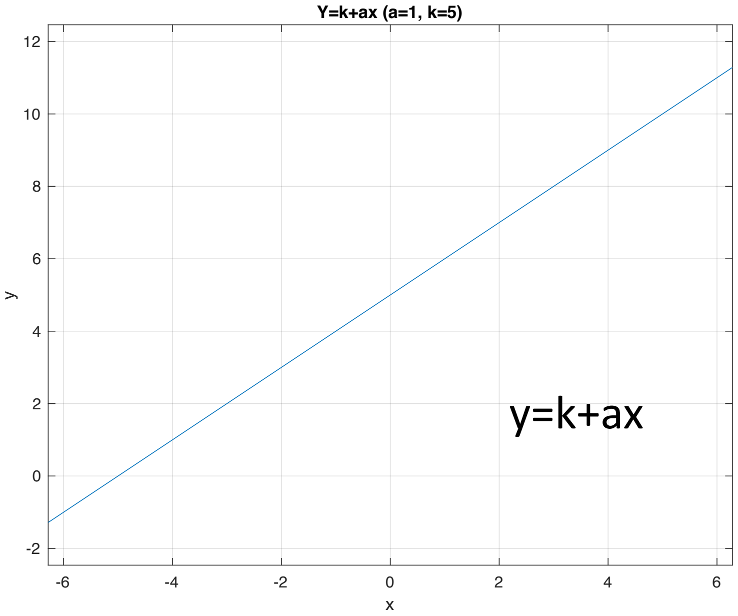
$$y = ax_1 + bx_2 + cx_3$$

- Technically, including a constant term makes the system non-linear;

$$y = k + ax_1 + bx_2 + cx_3$$

- This is known as an affine function (system) and can be dealt with in much the same way as a truly linear system.





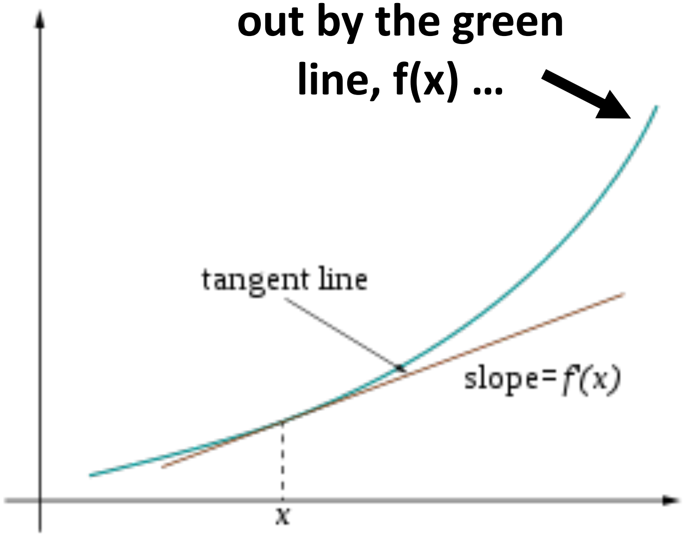
*the actual definition is a bit more technical and is written in terms of the principle of superposition, we will focus on linear systems so the exact definition doesn't matter (but please look it up!)

Loughborough University

Linear Systems

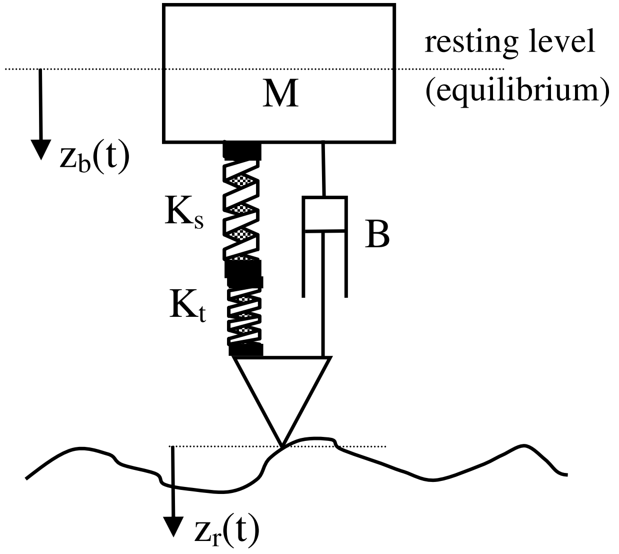
- Linear systems are important since the mathematics around them is more fully developed.
- There are many open questions around non-linear systems including finding of more general solutions.
- Most non-linear systems can be linearised (around a point) which provides a means of making some intractable nonlinear system problems tractable.

correction…traced out by the green line, f(x) …



Linearisation around x for f(x). Note that the approximation is good close to x but less good as one moves away from x to f(x+h) for example.

Linear or non-linear??



<u>System Equations</u>

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{K}{M}(x_3 - x_2) + \frac{B_s}{M}(u - x_2)$$

$$\dot{x}_3 = u$$

Linear Systems

- The mathematical description of the system states is written as a set of n coupled first order equations;

$$\dot{x}_1 = f_1(\boldsymbol{x}, \boldsymbol{u}, t)$$
$$\vdots$$
$$\dot{x}_n = f_n(\boldsymbol{x}, \boldsymbol{u}, t)$$

- If we restrict our type of system to those that are (very nearly) linear and time invariant, we can write the system state change i.e. *state equations* as a linear combination of inputs, u_i and states, x_i

$$\dot{x}_1(t) = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + b_{11}u_1 + \cdots + b_{1r}u_r$$
$$\vdots$$
$$\dot{x}_n(t) = a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n + b_{n1}u_1 + \cdots + b_{nr}u_r$$

Where n is the number of states and r the number of inputs.

Loughborough University

State Space Representation

- Writing in matrix form;

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1r} \\ b_{21} & b_{22} & \cdots & b_{2r} \\ \vdots & \vdots & \cdots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nr} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_r \end{bmatrix}$$

$$or$$

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

This tells us how the states, $\dot{\boldsymbol{x}}$ changes but not what the outputs are.

State Space Representation

- The states don’t include information necessary for engineering purposes.
- In addition states can be arbitrarily chosen and therefore may not represent anything physically meaningful.
- An output variable (arbitrary) can be written as linear combination of states x_i and inputs u_i;

$$y_1(t) = c_1x_1 + C_1x_2 + \cdots + c_nx_n + d_1u_1 + \cdots + d_ru_r$$

Loughborough University

State Space Representation

- If we are interested in m output variables, we can write m equations as;

$$y_1(t) = c_{11}x_1 + c_{12}x_2 + \cdots + c_{1n}x_n + d_{11}u_1 + \cdots + d_{1r}u_r$$
$$\vdots$$
$$y_m(t) = c_{m1}x_1 + c_{m2}x_2 + \cdots + c_{mn}x_n + d_{m1}u_1 + \cdots + d_{mr}u_r$$

- Or in matrix form;

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1r} \\ d_{21} & d_{22} & \cdots & d_{2r} \\ \vdots & \vdots & \cdots & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mr} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_r \end{bmatrix}$$

State Space representation

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u}$$

- Note how the output $\boldsymbol{y}$ (a vector) is a combination of inputs and current state after a transformation (multiplication by $\boldsymbol{C}$ and $\boldsymbol{D}$ respectively)
- For many real systems $\boldsymbol{D}\boldsymbol{u}$ is not necessary, so;

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$$

- The output is just the states multiplied by some vector $\boldsymbol{C}$ (a property of the system)

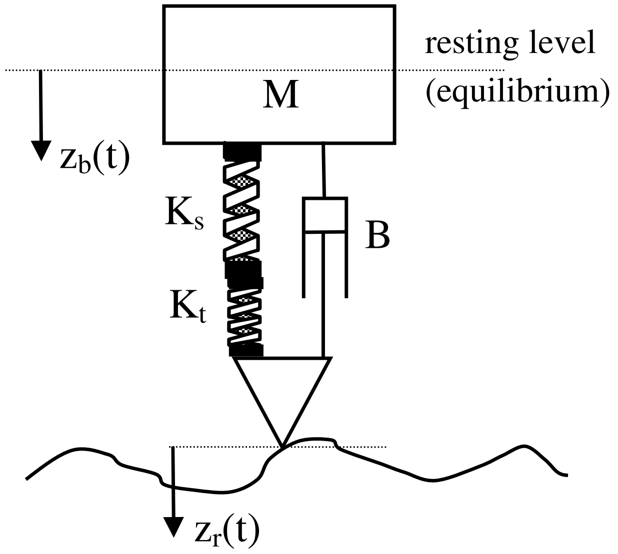
* Matrix transformation

Conclusion

- Linear systems
- State space representation

Tutorial

Tutorial - State Space Representation



Sign convention: +ve direction indicated by arrow heads

- Try for yourself;
  - Spring stiffness; $\frac{1}{K} = \frac{1}{K_s} + \frac{1}{K_t}$

$$F_s = \cdots$$