State-Space Modelling

"Vectors and matrices are the very language of state-space methods"

– Friedland

METR 4202: Advanced Control & **Robotics**

Dr Surya Singh -- Lecture # 11

October 14, 2015

metr4202@itee.uq.edu.au

http://robotics.itee.uq.edu.au/~metr4202/

Schedule

Week	Date	Lecture (W: 12:05-1:50, 50-N201)
1	29-Jul	Introduction
2	5-Aug	Representing Position & Orientation & State (Frames, Transformation Matrices & Affine Transformations)
3	12-Aug	Robot Kinematics Review (& *Ekka Day*)
4	19-Aug	Robot Dynamics
5	26-Aug	Robot Sensing: Perception
6	2-Sep	Robot Sensing: Multiple View Geometry
7	9-Sep	Robot Sensing: Feature Detection (as Linear Observers)
8	16-Sep	Probabilistic Robotics: Localization
9	23-Sep	Quiz
	30-Sep	*Study break*
10	7-Oct	Motion Planning
11	**14-Oct**	**State-Space Modelling**
12	21-Oct	Shaping the Dynamic Response
13	28-Oct	LQR + Course Review

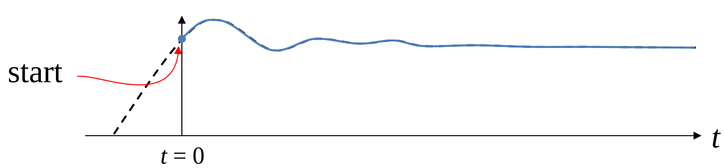
State-Space Modelling (from 2013 – Sorry!)

("Hear Ye! It be stated")

Affairs of state

- Introductory brain-teaser:
  - If you have a dynamic system model with history (ie. integration) how do you represent the instantaneous state of the plant?

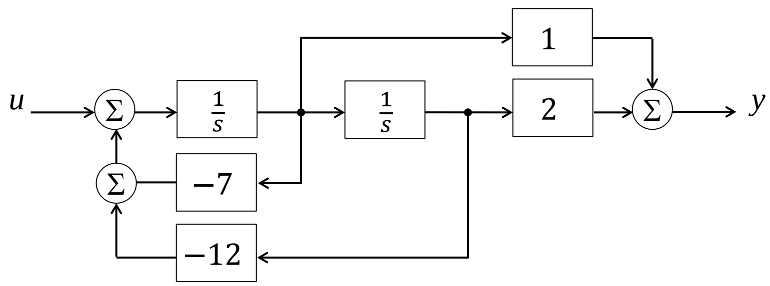
Eg. how would you setup a simulation of a step response, mid-step?



Introduction to state-space

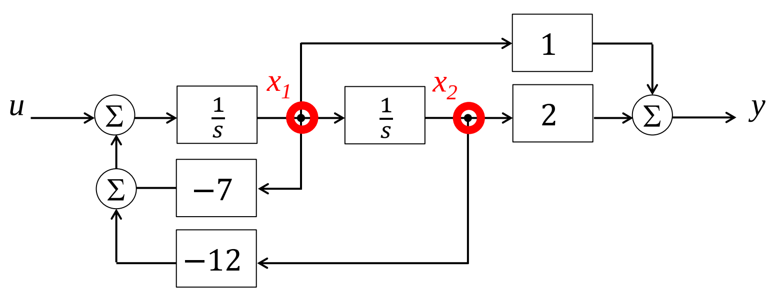
- Linear systems can be written as networks of simple dynamic elements:

$$H = \frac{s+2}{s^2+7s+12} = \frac{2}{s+4} + \frac{-1}{s+3}$$



Introduction to state-space

- We can identify the nodes in the system
  - These nodes contain the integrated time-history values of the system response
  - We call them "states"



Linear system equations

- We can represent the dynamic relationship between the states with a linear system:

$$\begin{aligned} \dot{x_1} &= -7x_1 - 12x_2 + u \\ \dot{x_2} &= x_1 + 0x_2 + 0u \\ y &= x_1 + 2x_2 + 0u \end{aligned}$$

State-space representation

- We can write linear systems in matrix form:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \begin{bmatrix} -7 & 12 \\ 1 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \\ \boldsymbol{y} &= \begin{bmatrix} 1 & 2 \end{bmatrix} \boldsymbol{x} + 0u \end{aligned}$$

Or, more generally:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \mathbf{A}\boldsymbol{x} + \mathbf{B}u \\ y &= \mathbf{C}\boldsymbol{x} + Du \end{aligned}$$

"State-space equations"

State-space representation

- State-space matrices are not necessarily a unique representation of a system
  - There are two common forms
- Control canonical form
  - Each node – each entry in $\boldsymbol{x}$ – represents a state of the system (each order of s maps to a state)
- Modal form
  - Diagonals of the state matrix **A** are the poles (“modes”) of the transfer function

State variable transformation

- Important note!
  - The states of a control canonical form system are not the same as the modal states
  - They represent the same dynamics, and give the same output, but the vector values are different!
- However we can convert between them:
  - Consider state representations, $\boldsymbol{x}$ and $\boldsymbol{q}$ where

$$\boldsymbol{x} = \mathbf{T}\boldsymbol{q}$$

T is a “transformation matrix”

State variable transformation

- Two homologous representations:

$$\dot{\boldsymbol{x}} = \mathbf{A}\boldsymbol{x} + \mathbf{B}u \qquad \text{and} \qquad \dot{\boldsymbol{q}} = \mathbf{F}\boldsymbol{q} + \mathbf{G}u$$
$$y = \mathbf{C}\boldsymbol{x} + Du \qquad\qquad\qquad y = \mathbf{H}\boldsymbol{q} + Ju$$

We can write:

$$\dot{\boldsymbol{x}} = \mathbf{T}\dot{\boldsymbol{q}} = \mathbf{AT}\boldsymbol{z} + \mathbf{B}u$$
$$\dot{\boldsymbol{q}} = \mathbf{T}^{-1}\mathbf{AT}\boldsymbol{z} + \mathbf{T}^{-1}\mathbf{B}u$$

Therefore, $\mathbf{F} = \mathbf{T}^{-1}\mathbf{AT}$ and $\mathbf{G} = \mathbf{T}^{-1}\mathbf{B}$

Similarly, $\mathbf{C} = \mathbf{HT}$ and $D = J$

Controllability matrix

- To convert an arbitrary state representation in **F**, **G**, **H** and *J* to control canonical form **A**, **B**, **C** and *D*, the "controllability matrix"

$$\mathcal{C} = [\mathbf{G} \quad \mathbf{FG} \quad \mathbf{F}^2\mathbf{G} \quad \cdots \quad \mathbf{F}^{n-1}\mathbf{G}]$$

must be nonsingular.

Why is it called the "controllability" matrix?

Example:
(Back To) Robot Arms

Slides 17-27 Source: R. Lindeke, ME 4135, “Introduction to Control”

Remembering the Motion Models:

- Recall from Dynamics, the Required Joint Torque is:

$$\tau_i = D_i\left(q\right)\ddot{q}_i + C_i\left(q, \dot{q}_i\right) + h\left(q\right) + b\left(\dot{q}_i\right)$$

- $D_i(q)\ddot{q}_i$: Dynamical Manipulator Inertial Tensor – a function of position and acceleration
- $C_i(q, \dot{q}_i)$: Coupled joint effects (centrifugal and coriolis) issues due to multiple moving joints
- $h(q)$: Gravitational Effects
- $b(\dot{q}_i)$: Frictional Effect due to Joint/Link movement

Lets simplify the model

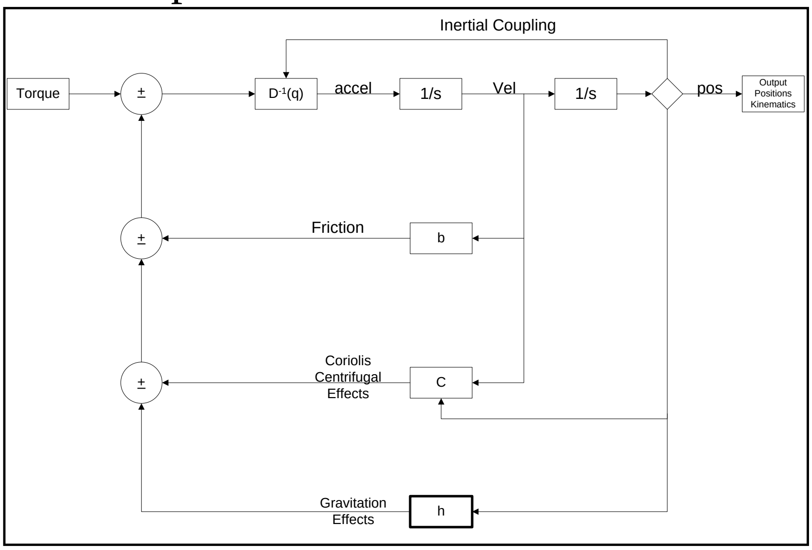
- This torque model is a 2nd order one (in position) lets look at it as a velocity model rather than positional one then it becomes a system of highly coupled 1st order differential equations
- We will then isolate Acceleration terms (acceleration is the 1st derivative of velocity)

$$a = \dot{v} = \ddot{q} = D_i^{-1}(q)\left(\tau_i - C_i(q,\dot{q}_i) - h(q) - b\left(\dot{q}_1\right)\right)$$

Considering Control:

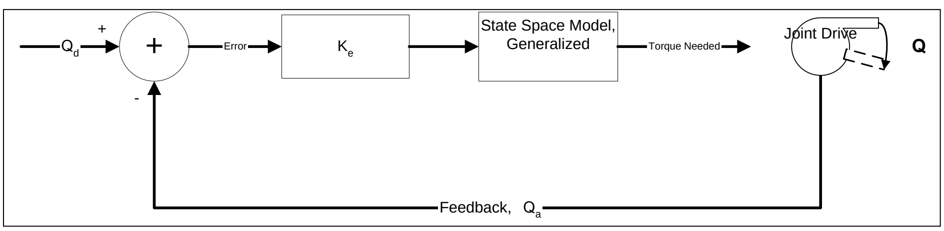
- Each Link's torque is influenced by each other links motion
  - We say that the links are highly coupled
- Solution then suggests that control should come from a simultaneous solution of these torques
- We will model the solution as a "State Space" design and try to balance the torque-in with *positional control*-out – the most common way it is done!
  - But we could also use 'force control' to solve the control problem!

The State-Space Control Model:



Setting up a Real Control

- We will (start) by using positional error to drive our torque devices



- This simple model is called a PE (proportional error) controller

PE Controller:

- To a 1st approximation, $\tau = K_m * I$
  - Torque is proportional to <u>motor current</u>

- And the Torque required is a function of 'Inertial' (Acceleration) and 'Friction' (velocity) effects as suggested by our L-E models

$$\tau_m \simeq J_{eq}\ddot{q} + F_{eq}\dot{q}$$

→ Which can be approximated as:

$$K_m I_m = J_{eq}\ddot{q} + F_{eq}\dot{q}$$

Setting up a "Control Law"

- We will use the <u>positional error</u> (as drawn in the state model) to develop our torque control
- We say then for PE control:

$$\tau \propto k_{pe}(\theta_d - \theta_a)$$

- Here, k_{pe} is a "gain" term that guarantees sufficient current will be generated to develop appropriate torque based on observed positional error

Using this Control Type:

- It is a representation of the physical system of a mass on a spring!
- We say after setting our target as a ‘zero goal’ that:

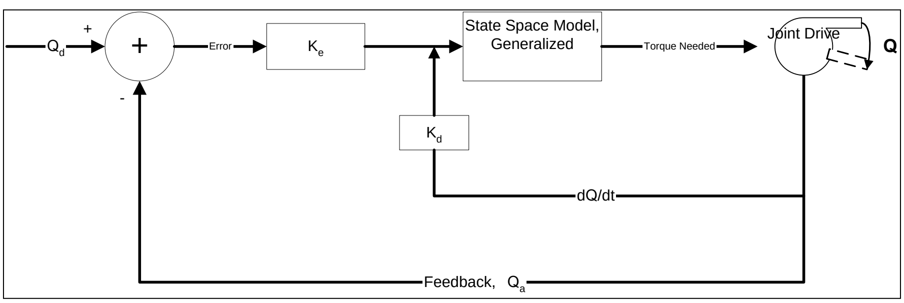
$$-k_{pe} * \theta_a = J\ddot{\theta} + F\dot{\theta}$$

the solution of which is:

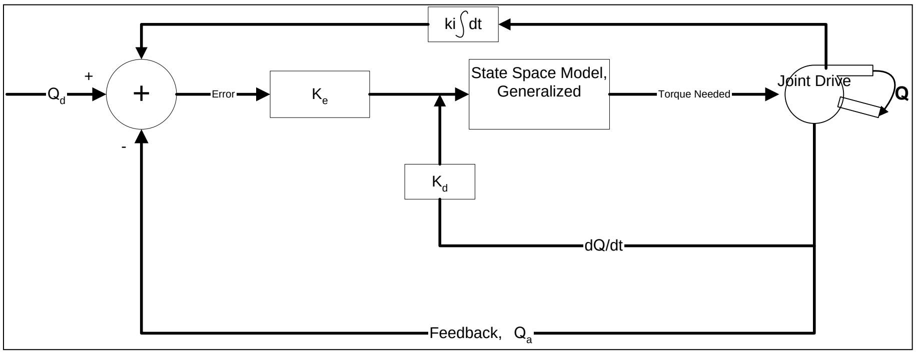
θ_a is a function of the servo feedback as a function of time!

$$\left(\theta_a = e^{-\left(F/2J\right)t}\left[C_1 e^{\left(1/2\right)\omega t} + C_2 e^{-\left(1/2\right)\omega t}\right]\right.$$

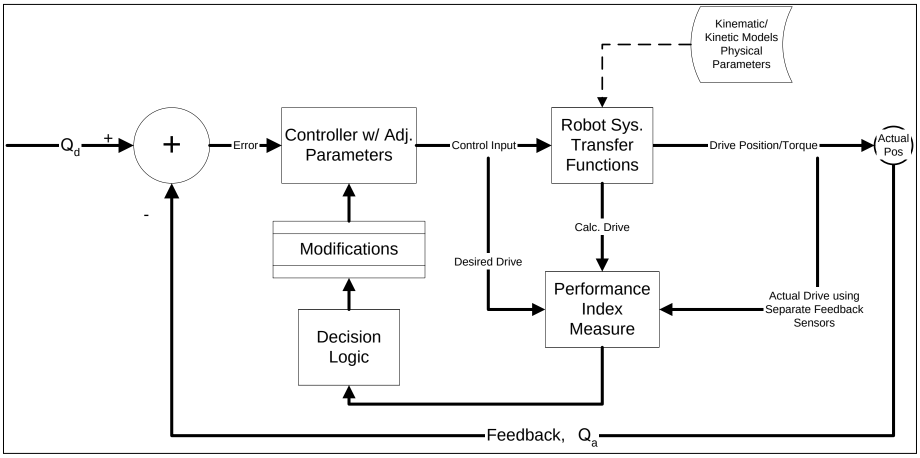
State Space Model of PD:



PID State Space Model:



State Model of Adjustable Controller



Controllability

Controllability matrix

- If you can write it in CCF, then the system equations must be linearly independent.
- Transformation by any nonsingular matrix preserves the controllability of the system.
- Thus, a nonsingular controllability matrix means $\boldsymbol{x}$ can be driven to any value.

State evolution

- Consider the system matrix relation:

$$\dot{\boldsymbol{x}} = \mathbf{F}\boldsymbol{x} + \mathbf{G}u$$
$$y = \mathbf{H}\boldsymbol{x} + Ju$$

The time solution of this system is:

$$\boldsymbol{x}(t) = e^{\mathbf{F}(t-t_0)}\,\boldsymbol{x}(t_0) + \int_{t_0}^{t} = e^{\mathbf{F}(t-\tau)}\,\mathbf{G}u(\tau)d\tau$$

If you didn't know, the matrix exponential is:

$$e^{\mathbf{K}t} = \mathbf{I} + \mathbf{K}t + \frac{1}{2!}\mathbf{K}^2t^2 + \frac{1}{3!}\mathbf{K}^3t^3 + \cdots$$

Stability

- We can solve for the natural response to initial conditions $\boldsymbol{x_0}$:

$$\boldsymbol{x}(t) = e^{p_i t}\boldsymbol{x}_0$$
$$\therefore \dot{\boldsymbol{x}}(t) = p_i e^{p_i t}\boldsymbol{x}_0 = \mathbf{F}e^{p_i t}\boldsymbol{x}_0$$

Clearly, a system will be stable provided
$\text{eig}(\mathbf{F}) < 0$

Characteristic polynomial

- From this, we can see $\mathbf{F}\boldsymbol{x}_0 = p_i\boldsymbol{x}_0$

$$\text{or, } (p_i\mathrm{I} - \mathbf{F})\boldsymbol{x}_0 = 0$$

$$\text{which is true only when } \det(p_i\mathrm{I} - \mathbf{F})\boldsymbol{x}_0 = 0$$

Aka. the characteristic equation!

- We can reconstruct the CP in s by writing:

$$\det(s\mathrm{I} - \mathbf{F})\boldsymbol{x}_0 = 0$$

Great, so how about control?

- Given $\dot{\boldsymbol{x}} = \mathbf{F}\boldsymbol{x} + \mathbf{G}u$, if we know $\mathbf{F}$ and $\mathbf{G}$, we can design a controller $u = -\mathbf{K}\boldsymbol{x}$ such that

$$\mathrm{eig}(\mathbf{F} - \mathbf{GK}) < 0$$

- In fact, if we have full measurement and control of the states of $\boldsymbol{x}$, we can position the poles of the system in arbitrary locations!

(Of course, that never happens in reality.)

Example: PID control

- Consider a system parameterised by three states:
  - x_1, x_2, x_3
  - where $x_2 = \dot{x}_1$ and $x_3 = \dot{x}_2$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & -2 \end{bmatrix} \boldsymbol{x} - \mathbf{K}u$$

$$y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \boldsymbol{x} + 0u$$

x_2 is the output state of the system;

x_1 is the value of the integral;

x_3 is the velocity.

- We can choose **K** to move the eigenvalues of the system as desired:

$$\det \begin{bmatrix} 1 - K_1 & & \\ & 1 - K_2 & \\ & & -2 - K_3 \end{bmatrix} = \mathbf{0}$$

All of these eigenvalues must be positive.

It's straightforward to see how adding derivative gain K_3 can stabilise the system.

Just scratching the surface

- There is a lot of stuff to state-space control

- One lecture (or even two) can't possibly cover it all in depth

Go play with Matlab and check it out!

State-space control design

- Design for discrete state-space systems is just like the continuous case.
  - Apply linear state-variable feedback:

$$u = -\mathbf{K}\boldsymbol{x}$$

such that $\det(z\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{K}) = \alpha_c(z)$

where $\alpha_c(z)$ is the desired control characteristic equation

Predictably, this requires the system controllability matrix $\boldsymbol{\mathcal{C}} = [\mathbf{\Gamma} \quad \mathbf{\Phi}\mathbf{\Gamma} \quad \mathbf{\Phi}^2\mathbf{\Gamma} \quad \cdots \quad \mathbf{\Phi}^{n-1}\mathbf{\Gamma}]$ to be full-rank.

Solving State Space (optional notes) …

Time-invariant dynamics The simplest form of the general differential equation of the form (3.1) is the "homogeneous," i.e., unforced equation

$$\dot{x} = Ax \tag{3.2}$$

where A is a constant k by k matrix. The solution to (3.2) can be expressed as

$$x(t) = e^{At}c \tag{3.3}$$

where e^{At} is the matrix exponential function

$$e^{At} = I + At + A^2\frac{t^2}{2} + A^3\frac{t^3}{3!} + \cdots \tag{3.4}$$

and c is a suitably chosen constant vector. To verify (3.3) calculate the derivative of $x(t)$

$$\frac{dx(t)}{dt} = \frac{d}{dt}(e^{At})c \tag{3.5}$$

and, from the defining series (3.4),

$$\frac{d}{dt}(e^{At}) = A + A^2t + A^3\frac{t^2}{2!} + \cdots = A\left(I + At + A^2\frac{t^2}{2!} + \cdots\right) = A\,e^{At}$$

Thus (3.5) becomes

$$\frac{dx(t)}{dt} = Ae^{At}c = Ax(t)$$

Solving State Space (optional notes)

which was to be shown. To evaluate the constant c suppose that at some time τ the state $x(\tau)$ is given. Then, from (3.3),

$$x(\tau) = e^{A\tau}c \tag{3.6}$$

Multiplying both sides of (3.6) by the inverse of $e^{A\tau}$ we find that

$$c = (e^{A\tau})^{-1}x(\tau)$$

Thus the general solution to (3.2) for the state $x(t)$ at time t, given the state $x(\tau)$ at time τ, is

$$x(t) = e^{At}(e^{A\tau})^{-1}x(\tau) \tag{3.7}$$

The following property of the matrix exponential can readily be established by a variety of methods—the easiest perhaps being the use of the series definition (3.4)—

$$e^{A(t_1+t_2)} = e^{At_1}e^{At_2} \tag{3.8}$$

for any t_1 and t_2. From this property it follows that

$$(e^{A\tau})^{-1} = e^{-A\tau} \tag{3.9}$$

and hence that (3.7) can be written

$$x(t) = e^{A(t-\tau)}x(\tau) \tag{3.10}$$

Solving State Space (optional notes)

The matrix $e^{A(t-\tau)}$ is a special form of the *state-transition matrix* to be discussed subsequently.

We now turn to the problem of finding a "particular" solution to the nonhomogeneous, or "forced," differential equation (3.1) with A and B being constant matrices. Using the "method of the variation of the constant,"[1] we seek a solution to (3.1) of the form

$$x(t) = e^{At}c(t) \tag{3.11}$$

where $c(t)$ is a function of time to be determined. Take the time derivative of $x(t)$ given by (3.11) and substitute it into (3.1) to obtain:

$$Ae^{At}c(t) + e^{At}\dot{c}(t) = Ae^{At}c(t) + Bu(t)$$

or, upon cancelling the terms $A\,e^{At}c(t)$ and premultiplying the remainder by e^{-At},

$$\dot{c}(t) = e^{-At}Bu(t) \tag{3.12}$$

Thus the desired function $c(t)$ can be obtained by simple integration (the mathematician would say "by a quadrature")

$$c(t) = \int_T^t e^{-A\lambda}Bu(\lambda)\,d\lambda$$

The lower limit T on this integral cannot as yet be specified, because we will need to put the particular solution together with the solution to the

Solving State Space (optional notes)

homogeneous equation to obtain the complete (general) solution. For the present, let T be undefined. Then the particular solution, by (3.11), is

$$x(t) = e^{At}\int_T^t e^{-A\lambda}Bu(\lambda)\,d\lambda = \int_T^t e^{A(t-\lambda)}Bu(\lambda)\,d\lambda \tag{3.13}$$

In obtaining the second integral in (3.13), the exponential e^{At}, which does not depend on the variable of integration λ, was moved under the integral, and property (3.8) was invoked to write $e^{At}e^{-A\lambda} = e^{A(t-\lambda)}$.

The complete solution to (3.1) is obtained by adding the "complementary solution" (3.10) to the particular solution (3.13). The result is

$$x(t) = e^{A(t-\tau)}x(\tau) + \int_T^t e^{A(t-\lambda)}Bu(\lambda)\,d\lambda \tag{3.14}$$

We can now determine the proper value for lower limit T on the integral. At $t = \tau$ (3.14) becomes

$$x(\tau) = x(\tau) + \int_T^\tau e^{A(t-\lambda)}Bu(\lambda)\,d\lambda \tag{3.15}$$

Thus, the integral in (3.15) must be zero for any $u(t)$, and this is possible only if $T = \tau$. Thus, finally we have the complete solution to (3.1) when A and B are constant matrices

$$x(t) = e^{A(t-\tau)}x(\tau) + \int_\tau^t e^{A(t-\lambda)}Bu(\lambda)\,d\lambda \tag{3.16}$$

Solving State Space (optional notes)

This important relation will be used many times in the remainder of the book. It is worthwhile dwelling upon it. We note, first of all, that the solution is the sum of two terms: the first is due to the "initial" state $x(\tau)$ and the second—the integral—is due to the input $u(\tau)$ in the time interval $\tau \leqq \lambda \leqq t$ between the "initial" time τ and the "present" time t. The terms initial and present are enclosed in quotes to denote the fact that these are simply convenient definitions. There is no requirement that $t \geqq \tau$. The relationship is perfectly valid even when $t \leqq \tau$.

Another fact worth noting is that the integral term, due to the input, is a "convolution integral": the contribution to the state $x(t)$ due to the input u is the convolution of u with $e^{At}B$. Thus the function $e^{At}B$ has the role of the impulse response[1] of the system whose output is $x(t)$ and whose input is $u(t)$.

If the output y of the system is not the state x itself but is defined by the observation equation

$$y = Cx$$

then this output is expressed by

$$y(t) = Ce^{A(t-\tau)}x(t) + \int_{\tau}^{t} Ce^{A(t-\lambda)}Bu(\lambda)\,d\lambda \qquad (3.17)$$

Solving State Space (optional notes)

and the impulse response of the system with y regarded as the output is $Ce^{A(t-\lambda)}B$.

The development leading to (3.16) and (3.17) did not really require that B and C be constant matrices. By retracing the steps in the development it is readily seen that when B and C are time-varying, (3.16) and (3.17) generalize to

$$x(t) = e^{A(t-\tau)}x(\tau) + \int_{\tau}^{t} e^{A(t-\lambda)}B(\lambda)u(\lambda)\,d\lambda \qquad (3.18)$$

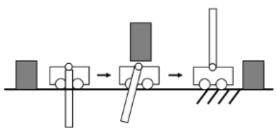
and

$$y(t) = C(t)\,e^{A(t-\tau)}x(\tau) + \int_{\tau}^{t} C(t)\,e^{A(t-\lambda)}B(\lambda)u(\lambda)\,d\lambda \qquad (3.19)$$

Example:
Inverted Pendulum

Digital Control

Wikipedia,
Cart and pole



$$L = \frac{1}{2}Mv_1^2 + \frac{1}{2}mv_2^2 - mg\ell\cos\theta$$

where v_1 is the velocity of the cart and v_2 is the velocity of the point mass m. v_1 and v_2 can be expressed in terms of x and θ by writing the velocity as the first derivative of the position;

$$v_1^2 = \dot{x}^2$$
$$v_2^2 = \left(\frac{d}{dt}(x - \ell\sin\theta)\right)^2 + \left(\frac{d}{dt}(\ell\cos\theta)\right)^2$$

Simplifying the expression for v_2 leads to:

$$v_2^2 = \dot{x}^2 - 2\ell\dot{x}\dot{\theta}\cos\theta + \ell^2\dot{\theta}^2$$

The Lagrangian is now given by:

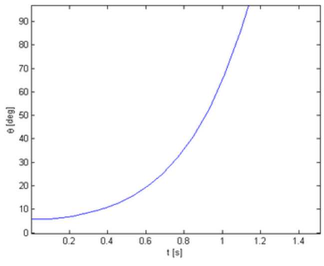
$$L = \frac{1}{2}(M+m)\dot{x}^2 - m\ell\dot{x}\dot{\theta}\cos\theta + \frac{1}{2}m\ell^2\dot{\theta}^2 - mg\ell\cos\theta$$

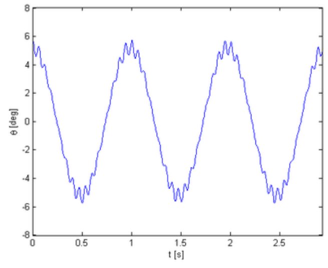
and the equations of motion are:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial\dot{x}} - \frac{\partial L}{\partial x} = F$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial\dot{\theta}} - \frac{\partial L}{\partial\theta} = 0$$

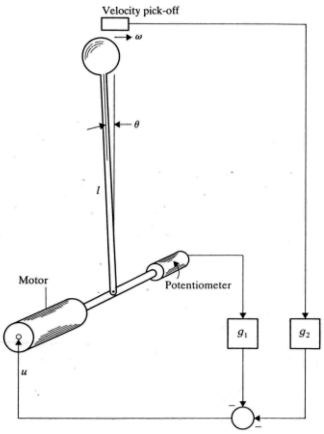
substituting L in these equations and simplifying leads to the equations that describe the motion o

$$(M+m)\ddot{x} - m\ell\ddot{\theta}\cos\theta + m\ell\dot{\theta}^2\sin\theta = F$$
$$\ell\ddot{\theta} - g\sin\theta = \ddot{x}\cos\theta$$





Inverted Pendulum



$$L = \frac{1}{2}Mv_1^2 + \frac{1}{2}mv_2^2 - mg\ell\cos\theta$$

where v_1 is the velocity of the cart and v_2 is the velocity of the point mass m. v_1 and v_2 can be expressed in terms of x and θ by writing the velocity as the first derivative of the position;

$$v_1^2 = \dot{x}^2$$

$$v_2^2 = \left(\frac{d}{dt}(x - \ell\sin\theta)\right)^2 + \left(\frac{d}{dt}(\ell\cos\theta)\right)^2$$

Simplifying the expression for v_2 leads to:

$$v_2^2 = \dot{x}^2 - 2\ell\dot{x}\dot{\theta}\cos\theta + \ell^2\dot{\theta}^2$$

The Lagrangian is now given by:

$$L = \frac{1}{2}(M+m)\dot{x}^2 - m\ell\dot{x}\dot{\theta}\cos\theta + \frac{1}{2}m\ell^2\dot{\theta}^2 - mg\ell\cos\theta$$

and the equations of motion are:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = F$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = 0$$

substituting L in these equations and simplifying leads to the equations that describe the motion o

$$(M+m)\ddot{x} - m\ell\ddot{\theta}\cos\theta + m\ell\dot{\theta}^2\sin\theta = F$$

$$\ell\ddot{\theta} - g\sin\theta = \ddot{x}\cos\theta$$

Inverted Pendulum – Equations of Motion

- The equations of motion of an inverted pendulum (under a small angle approximation) may be linearized as:

$$\dot{\theta} = \omega$$

$$\dot{\omega} = \ddot{\theta} = Q^2\theta + Pu$$

Where:

$$Q^2 = \left(\frac{M+m}{Ml}\right)g$$

$$P = \frac{1}{Ml}.$$

If we further assume unity ***Ml*** ($Ml \approx 1$), then $P \approx 1$

Inverted Pendulum –State Space

- We then select a state-vector as:

$$\boldsymbol{x} = \begin{bmatrix} \theta \\ \omega \end{bmatrix}, \text{ hence } \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \omega \\ \dot{\omega} \end{bmatrix}$$

- Hence giving a state-space model as:

$$A = \begin{bmatrix} 0 & 1 \\ Q^2 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- The resolvent of which is:

$$\Phi(s) = (sI - A)^{-1} = \begin{bmatrix} s & -1 \\ -Q^2 & s \end{bmatrix}^{-1} = \frac{1}{s^2 - Q^2} \begin{bmatrix} s & 1 \\ Q^2 & s \end{bmatrix}$$

- And a state-transition matrix as:

$$\Phi(t) = \begin{bmatrix} \cosh Qt & \frac{\sinh Qt}{Q} \\ Q\, sinh\, Qt & \cosh Qt \end{bmatrix}$$

Stability

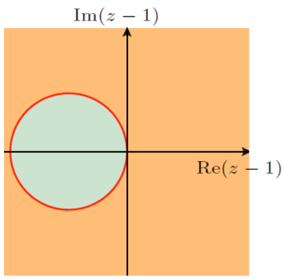
Fast sampling revisited

- For small T:

$$z = e^{sT} = 1 + sT + \frac{(sT)^2}{2} + \ldots \approx 1 + sT$$

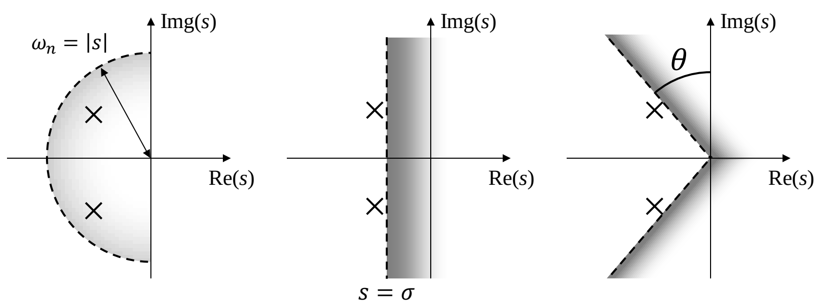
$$\to z \approx 1 + sT \to s = \frac{z-1}{T}$$

- Hence, the unit circle under the map from z to s-plane becomes:



Specification bounds

- Recall in the continuous domain, response performance metrics map to the s-plane:



$|s| = \frac{1.8}{t_r}$ $\quad s = \frac{4.6}{t_s}$ $\quad \theta = \sin^{-1}\zeta$

Cool Robotics Share