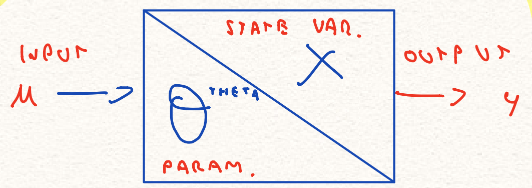
ALTAIR CLASS

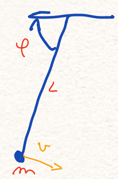
1) WHAT IS A SYSTEM?

- SPREAD OF A VIRUS
- DEVICE
- EN. CONSUMPTION

ETC...

THEY HAVE IN COMMON TO HAVE A MATH DESCRIPTION





AN EXAMPLE: A PENDOLUM

WITH $\theta(m,L)$ and $x(\varphi)$

CAN WE PREDICT THE MASS SPEED?

NO! WE NEED TO KNOW THE WHOLE STATE VECTOR $x(\varphi, \boxed{\dot{\varphi}})$ → if we know the speed $\dot{\varphi}$, WE CAN PREDICT THE SPEED OF THE MASS!

JUST LIKE THIS, WE CAN SAY THE SAME THING FOR A CAR.

THIS KIND OF RAPPRESENTATION OF A SYSTEM COULD BE DIFFERENT ACCORDING TO WHAT I'M STUDYING, SO I CAN CLASSIFY SYSTEMS BY THEYR CHARACTERISTIC

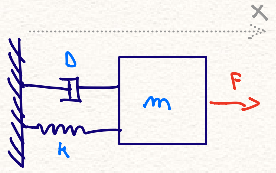
PARAMETERS
- TIME INVARIANT: PARAMETERS ARE COSTANT OVER TIME
- TIME VARIANT: PARAMETERS ARE NOT CONSTANT OVER TIME (SHUTTLE CONSUMING FUEL)

STATE
- LINEAR: MATH. FUNCTION IS A LINEAR COMBINATION OF STATE VARIABLES
- NON LINEAR: IF THE SYS. IS DESCRIBED BY NONLINEAR EQUATION

INPUT/OUTPUT
- SISO
- SIMO
- MISO
- MIMO

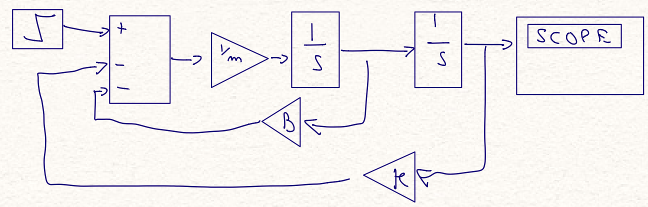
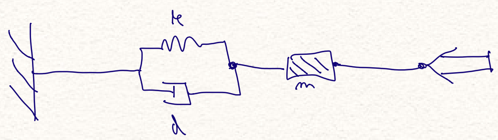
TIME
- CONTINUOUS TIME: STATE VARIABLES CHANGE CONTINUOUSLY OVER TIME
- DISCRETE TIME: STATE VARIABLES CHANGE BY DESCRETE QUANTITIES ONLY AT DESCRETE SET OF POINT IN TIME.

AN EXAMPLE: SPRING/DUMPER MASS OSCILLATION:



MATH. EQ: $m\ddot{x}(t) + b\dot{x}(t) + k\,x(t) = F(t)$

WE CAN MODEL THIS IN DIFF. WAYS

SCRIPT (OO)	GRAPHICAL SIGNAL-BASED (1D)
SCRIPT.m	
	GRAPHICAL PHYSICAL BASED
	 modelica — WE CAN FORGET ABOUT MATH!

ALTAIR COMPOSE — ALTAIR ACTIVATE

usually we solve this kind of problem with ordinary DIFF. EQUATION

$$f\left(t, x(t), \frac{dx(t)}{dt}, \frac{d^2x(t)}{dt^2}, \ldots, \frac{d^{\boxed{n}}x(t)}{dt^n}, u(t), \theta(t)\right) = 0$$

ORDER (of n); INDIP. VARIABLE (t); INPUT ($u(t)$); PARAMETER ($\theta(t)$)

STATE VARIABLE
X = STATE VECTOR

IMPLICIT FORMULATION

WHAT WE WILL NEED TO USE A SIGNAL-BASED APPROACH?

1) OBTAIN THE IMPLICIT FORMULATION

2) FIND IT'S EXPLICIT FORMULATION:

$$\frac{d^n x(t)}{dt^n} = f\left(t, x(t), \frac{d^2(t)}{t^2}, \ldots \frac{d^{n-1}(t)}{t^{n-1}}, u(t), \theta(t)\right)$$

3) IDENTIFY THE ORDER

WE APPLY ALL THE STEPS

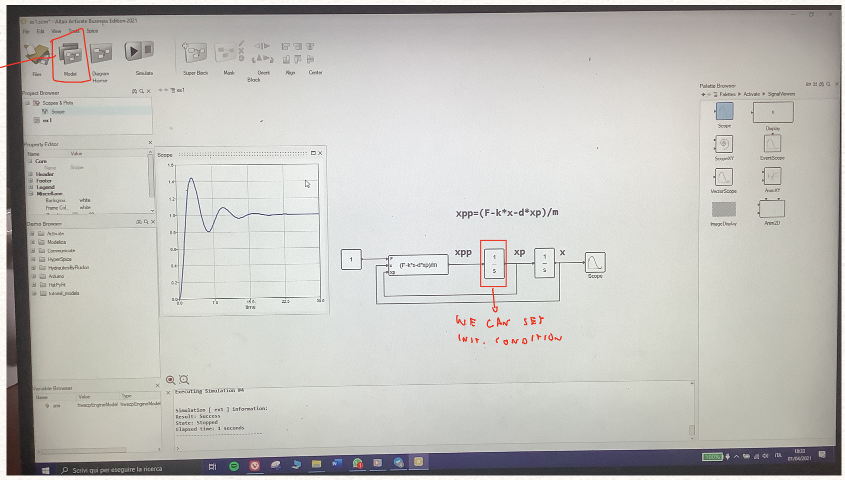
1) $m\ddot{x}(t) + d\dot{x}(t) + k\,x(t) - F = 0$

2) $\ddot{x}(t) = \frac{1}{m}\left(F - d\dot{x}(t) - k\,x(t)\right)$

3) 2^{ND} ORDER

NOW LET'S GO TO ACTIVE AND DO THE FIRST EXERCISE: EX1.SCM

MODEL PARAMETERS



NOW WE NEED TO FIND THE PHYSICAL BASED APPROACH → AT LEAST REMEMBER THE ORDER

WE DO NOT FORGET EVERYTHING, AND WE OPEN MODELICA

it is useful in modelica to add all the initial condition,

THE NUMBER IS THE ORDER N°

FOR EXAMPLE: I SET TRUE TO POSITION AND SPEED OF THE MASS.

AND THAT IS OKAY!

THEN when I simulate, everything works!

AFTER I DID THIS, I SET THE INITIAL POSITION IN THE SENSOR = 1

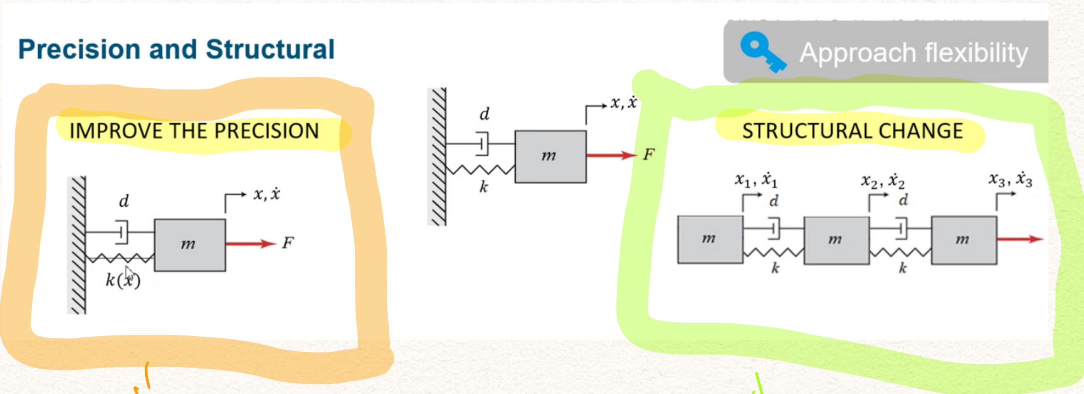
THERE ARE 2 CASE

- SET POSITION = FALSE :) → THIS POSITION IS IGNORED
- SET POSITION = TRUE :(→ ERROR! WE HAVE 2 INCONSISTENT STARTING POINT!!

PHYSICAL APPROACH HAS A LOTS OF ADVANTAGES !!

	Signal Approach	Physical Approach
Time required for modeling		X
Approach simplicity		X
Usability/sharing		X
Mistakes during modeling		X
System understanding	X	
Time for model simulation	X	

WHAT WE COULD DO NOW IS SOME MODIFICATION AT OUR MODEL:



SIMPLER TO USE A SIGNAL-BASED

SIMPLER TO USE PHYSICAL-BASED

NOW GO ON

2_SYST. MODELLING / GENERAL-APPROACH. SCM

IMPROVE_PRECISION: SIGNAL BASED WE SIMPLY ADD $k(x) = k + x$

PHYS. BASED -> WE NEED TO CREATE A NEW ~~MO~~ MODELICA MODEL

STRUCTURAL-CHANGE: PHYS. APPROACH WE JUST NEED CRCT-C CRCT-V

LINEAR THEORY

WE HAVE AS ALWAYS A LINEAR SYSTEM

$$\underset{[M]}{u} \rightarrow \boxed{LIN} \rightarrow \underset{[A]}{y}$$

$$a_n \frac{d^n x(t)}{dt^n} + \dots + a_i \frac{d^i x(t)}{dt^i} + \dots + a_1 \dot{x}(t) + a_0 x(t) =$$

STATE VARIABLES

$$= b_m \frac{d^m u(t)}{dt^m} + \dots + b_j \frac{d^j u(t)}{dt^j} + \dots + b_0 u(t)$$

$\boxed{\text{with } m < n}$

if $m = n$ -> the sys is acting instantly, and in real thinks is not possible

WE KNOW THAT THE SOLUTION OF THIS LINEAR SYST. WILL BE LIKE THIS:

$$X(t) = \underbrace{\sum_{i=1}^{m} A_i e^{\lambda_i t} + \cancel{\sum_{h=1}^{M-1} C_h t^r e^{\lambda_h t}} + B}_{\text{NATURAL RESPONSE}} + \underbrace{X_*(t)}_{\text{FORCED RESPONSE}}$$

USUALLY NEGLECTED

- NATURAL IS DUE TO THE RESPONSE IF NO INPUT IS ADDED
- THE FORCED RESPONSE IS THE ONE THAT YOU CAN SEE WHEN WE GIVE SOMETHING IN INPUT

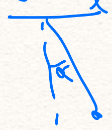
why THOSE ARE IMPORTANT? THINK THAT THE VAST MAJORITY OF PROBLEM ARE NON LINEAR! :C

↳ SO 2 MAIN REASON

1) A NON LINEAR SYS COULD BE LINEARIZED → SIMPLER FORM

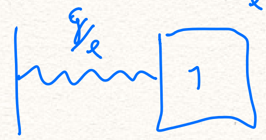
$\boxed{\text{LYUPANOV INDIRECT METHOD}}$ → NON-LINEAR SYSTEM IS VERY CLOSE TO A LINEAR ONE LOCALLY AROUND AT ITS EQUILIBRIUM POINTS

EXAMPLE: PENDULUM $\ddot{\alpha} + \frac{g}{l}\boxed{\sin\alpha} = 0$



$\alpha < 10°$

~ LINEARIZATION ~> $\ddot{\alpha} + \frac{g}{l}\alpha = 0$

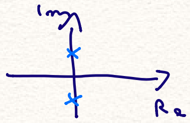


YOU CAN FIND IT IN COMPARISON.SCM

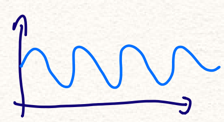
THE 2 SYST. ARE VERY SIMILAR! (if $\alpha < 10°$)

2) ALSO BECAUSE LINEAR SYST. ARE MUCH MORE SIMPLE AND EASY TO UNDERSTAND THE NATURE OF THE SYSTEM!! → MANY COOL PEOPLE HAS STUDIED LINEAR MODELS BEFORE US

$\boxed{\text{LINEARIZED}}$: WE CAN FIND THE ZERO AND POLES:



→



NOW, IF I HAVE A LTI SYSTEM, I CAN USE LAPLACE TRANSFORM

LTI EQUATIONS

$$\frac{d^m x(t)}{dt^m} \xrightarrow{\mathcal{L}} s^m \cdot X(s) \quad \text{WHERE } s = \sigma + j\omega$$

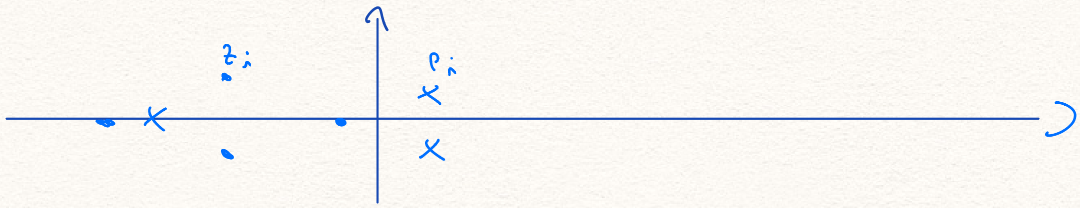
$$(a_m s^m + a_{m-1} s^{m-1} + \dots + a_1 s + a_0) X(s) = (b_m s^m + \dots + b_1 s + b_0) U(s)$$

↳ LAPLACE DOMAIN (FREQ. DOMAIN) → THOSE ARE ALGEBRIC FORMULAT. MUCH MORE EASIER TO DEAL WITH

ONCE WE DO THAT:

TRANSF. FUNCTION: $\frac{\text{OUTPUT}}{\text{INPUT}} = \frac{Y(s)}{U(s)} = \frac{b_m s^m + \dots + b_1 s + b_0}{a_m s^m + \dots + a_1 s + a_0}$ } ZEROS • } POLES X

WE CAN DRAW ROOTS LOCUS



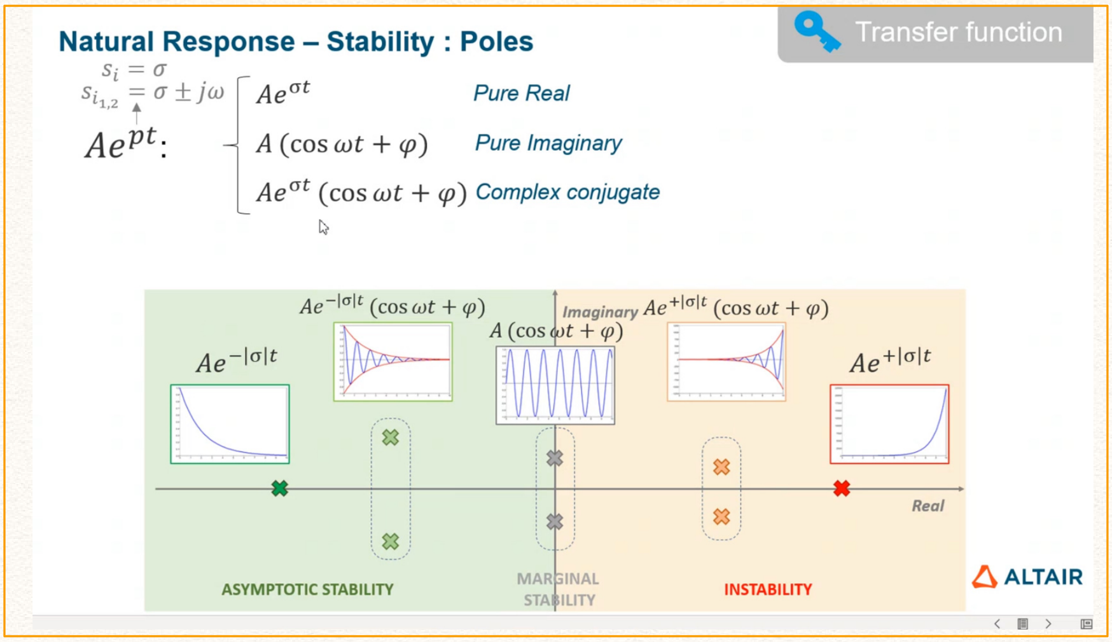
SOME PROPERTIES

POLES OF THE FCN AFFECTS THE STABILITY OF THE SYSTEM!
BECAUSE THEY ARE IN THE NATURAL RESPONSE OF THE SYST.

$$x(t) = \boxed{\sum_{i=1}^{n} A_i e^{\lambda_i t} + B} + x_p(t)$$

POLES

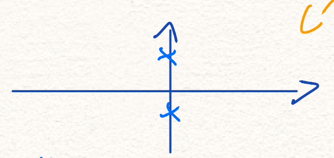
AND THE POLES ARE IN THE DENOMINATOR OF THE T.F.



IN OUR CASE, WITH OUR PENDULUM?

$$\ddot{\alpha} + \frac{g}{L}\alpha = 0 \rightarrow \text{LAPLACE} \rightarrow s^2 A(s) + \frac{g}{L}A(s) = 0$$

$$\left(s^2 + \frac{g}{L}\right)A(s) = 0 \Big\} \; s = \pm\sqrt{-\frac{g}{L}} = \begin{cases} ① \; \boxed{j\sqrt{\frac{g}{L}}} \\ ② \; \boxed{-j\sqrt{\frac{g}{L}}} \end{cases}$$



MARGINAL STABILITY →

PROPERTY OF FORCED RESPONSE

$$x(t) = \sum_{i=1}^{n} A_i e^{\lambda_i t} + B + \boxed{x_*(t)} \rightsquigarrow x_*(t) = x_*(\lambda_i, (z_i), t)$$

FORCED

when we have an input $u(t)$, WE CAN SPLIT IT IN A FOURIER SERIES:

$$\boxed{u(t) = \sum_{i=1}^{m} B_i \left(\cos[\Omega_i t + \varphi_i]\right)}$$

THEN WE CAN USE SUPERPOSITION PRINCIPLE - AND - SIMILARITY METHOD

$$\boxed{x_*(t) = \sum_{i=1}^{m} C_i \left[\cos(\Omega_i t + \varphi_i + \psi_i)\right]}$$

IT IS AGAIN A SUM OF COSIN, WITH the same FREQUENCIES, BUT WITH A PHASE SHIFT PSI ψ_i

THEN THE TRICK IS TO CONSIDER ONLY THE FORCED RESPONSE, AND SO THE ONLY IMPORTANT PART OF OUR STATE VARIABLE S, is the IMG PART! $S = J\Omega$

if we use only $J\Omega$ in place of S, WE WILL OBTAIN THE INFORMATION OF OUR SYSTEM AT REGIME CONDITION!

↳ TF is telling info about the forced response

$$\frac{X(s)}{U(s)} = \frac{X(J\Omega)}{U(J\Omega)} = \frac{b_m (J\Omega)^m + \dots + b_1 J\Omega + b_0}{a_m (J\Omega) + \dots + a_1 J\Omega + a_0} = \boxed{\text{CONSTANT COMPLEX NUMBER}}$$

so, for a given $J\Omega^*$, the ratio between $\frac{y}{u}$ is a constant, so it is INDIPENDENT BY THE INPUT AMPLITUDE!!

this mean that we can write the BODE PLOT! $\frac{y}{u} = |TF| e^{J\omega}$ COMPLEX NUMBER

LITTLE RECAP:

1) T.F. gives us the TRANSIENT & STATIONARY behaviour of the system

2) STABILITY DEPENDS ONLY ON POLES P_i

3) POLES P_i + zeros Z_i are linked togheter and affect the FORCED RESPONSE

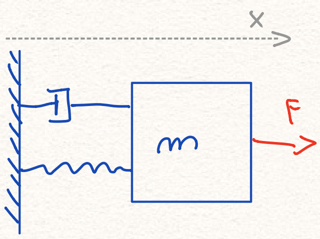
4) FOR LTI system we can draw the bode plot, wich represents the frequency response of the system in REGIME CONDITION

STATE SPACE FORMULATION OF A SYSTEM

$$\begin{cases} \dot{x}(t) = A\,x(t) + B\,u(t) \\ y(t) = C\,x(t) + D\,u(t) \end{cases}$$

- A → STATE MATRIX
- C → OUTPUT MATRIX
- B → INPUT MATRIX
- D → FREEFORWARD MATRIX

WE GO BACK TO THE MASS + DUMPER EXAMPLE



IMPLICIT FORMULATION:

$$m\ddot{x}(t) + d\,\dot{x}(t) + k\,x(t) - F = 0$$

$$\boxed{x(s) = \frac{F(s)}{ms^2 + ds + k}}$$

$\Downarrow$

EXPLICIT FORMULATION

$$\begin{cases} \ddot{x}(t) = -\frac{d}{m}\dot{x}(t) - \frac{k}{m}x(t) + \frac{F}{m} \\ \dot{x}(t) = \dot{x}(t) \end{cases}$$

⤳ WE HAVE AN IDENTITY: NICE TRICK

another trick: we use $\bar{x}(t) = \begin{bmatrix} \dot{x} \\ x \end{bmatrix}$ ⤳ the new state vector

$$\begin{cases} \dot{x}(t) = \overset{[A]}{\begin{bmatrix} -d/m & -k/m \\ 1 & 0 \end{bmatrix}} x + \overset{[B]}{\begin{bmatrix} 1/m \\ 0 \end{bmatrix}} F \\ y(t) = \overset{[C]}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}} x + \overset{[D]}{\begin{bmatrix} 0 \\ 0 \end{bmatrix}} F \end{cases}$$

SIMILARITIES BETWEEN THIS STATE SPACE FORMULATION → AND TRANSF. FUNCTION ⤳ $\frac{y(s)}{u(s)} = \frac{x(s)}{F(s)} = \frac{1}{ms^2 + bs + k}$

$\Downarrow$

EIGENVALUE OF [A]
(AUTOVALORI)

$$\det(A - \lambda I) = \det\left(\begin{bmatrix} -d/m - \lambda & -k/m \\ 1 & -\lambda \end{bmatrix}\right) =$$

$$= \lambda^2 + \lambda\frac{d}{m} + \frac{k}{m} = \boxed{m\cdot\lambda^2 + d\cdot\lambda + k} = 0$$

$\lambda = s$

↳ the eigen vallue of A, are the poles of our system we can do the laplace transformation of our state space system:

$$\begin{cases}\dot{x}(t) = Ax(t) + Bu(t)\\ y(t) = Cx(t) + Du(t)\end{cases} \xrightarrow{\mathcal{L}} \begin{cases} s\cdot x(s) = Ax(s) + BU(s) \\ y(s) = Cx(s) + BU(s)\end{cases} = \begin{cases}[sI - A]\,x(s) = BU(s)\\ y(s) = Cx(s)+DU(s)\end{cases}$$

$$= \begin{array}{l} x(s) = [sI-A]^{-1}\cdot B\cdot U(s) \\ y(s) = \left(C\cdot[sI-A]^{-1}\cdot B + D\right)\cdot U(s)\end{array}$$

$$\boxed{\frac{y(s)}{U(s)} = \left[C\cdot[sI-A]^{-1} + D\right]} = \begin{bmatrix} TF_{11} & TF_{12} & \cdots & TF_{1q} \\ & & & \\ TF_{m1} & & \cdots & TF_{mq}\end{bmatrix}$$

THIS MEAN THAT IT IS VERY SIMPLE TO GO FROM

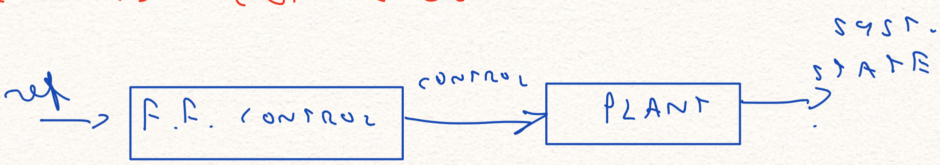
$\boxed{\text{STATE SPACE} \xrightarrow{\text{TO}} \text{TRANSFER FUNCTION}}$ AND VICE VERSA

IN TF, SS, SCM YOU CAN FIND SOME EXAMPLE OF HOW THE MODEL CAN BE REPRESENTED:

1) initial state set
2) initial output set
3) STEADY STATE → WE START IN ORDER THAT EVERYTHING IS IN EQ

A BIT OF CONTROL THEORY

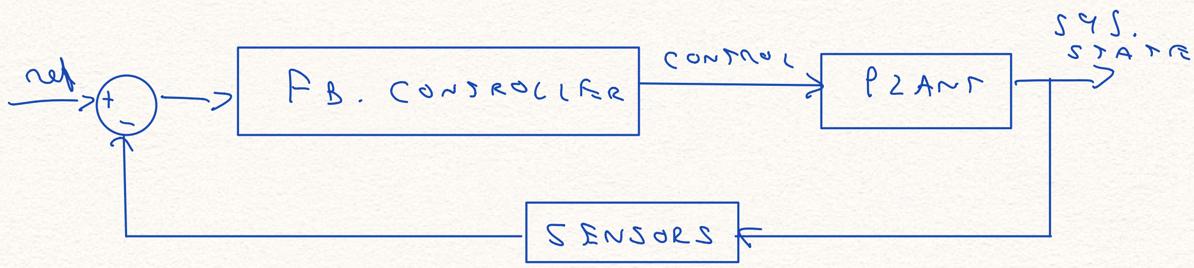
1) FREE FORWARD CONTROLLER



- VERY CHEAP ...
- NO FEEDBACK → $\boxed{\text{NO REACTION TO DISTURBANCE}}$

(YOU CAN SEE IT IN $FF_{CONTROLLER}$.SCM)

2) FEED BACK CONTROLLER!



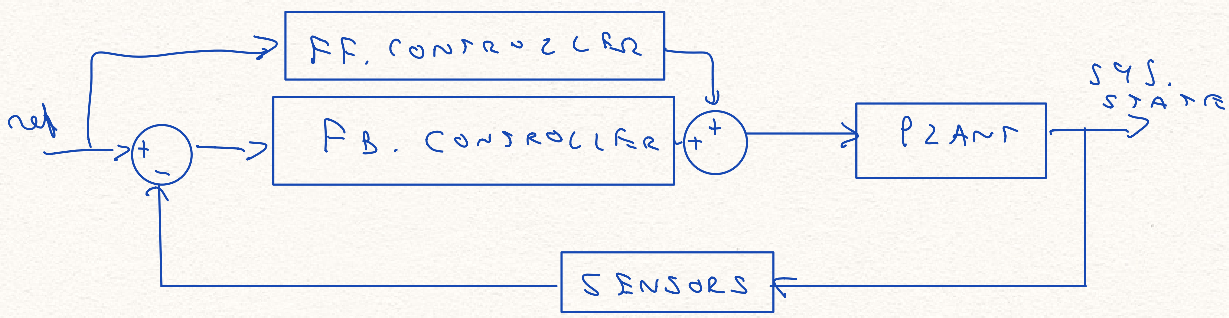
+ ROBUST

- WE NEED SENSORS

(YOU CAN SEE IT IN FB CONTROLLER)

3) FEEDBACK + FEEDFORWARD

THE BEST RESULTS! AND ALSO <u>FASTER</u>



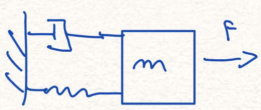
- FF USED TO FOLLOW THE REF
- FB. USED TO COMPENSATE DISTURBANCE

(FB+FF controller . scm)

PID CONTROLLER

P) PROPORTIONAL ACTION

in our system mass - damper



$$m\ddot{x} + b\dot{x} + kx = F_c = \boxed{k_p (x_{ref} - x)}$$ ⇝ FORCE PROPORTIONAL TO THE ERROR (FEEDFORWARD)

$$m\ddot{x} + b\dot{x} + (k + k_p)x = k_p x_{ref}$$

so now, a proportional controller is changing the stiffness of our system...

WHAT DOES HAPPEN IN REGIME CONDITION?

$$\cancel{m\ddot{x}} + \cancel{b\dot{x}} + (k + k_p)x = k_p x_{ref}$$

$$x = \frac{k_p}{k + k_p} x_{ref} \rightarrow x \simeq x_{ref} \text{ if } \boxed{k_p \to \infty}$$

P, D) proportional AND DERIVATIVE:

$$m\ddot{x} + b\dot{x} + kx = F_c = k_p (x_{ref} - x) + k_d (\dot{x}_{ref} - \dot{x})$$

$$m\ddot{x} + (b + k_d)\dot{x} + (k + k_p)x = k_p x_{ref} + k_d \dot{x}_{ref}$$

here we also changed the equivalent damping!!

in steady state:

$$(k + k_p)x = k_p x_{ref} \rightarrow x = \frac{k_p}{k + k_p} x_{ref}$$

again, we will follow the reference only if $k_p \to \infty$

P, D, I) we add also the INTEGRAL:

here we change the variable: $\boxed{y = \int_0^t x \, dt}$

$$m\ddot{x} + b\dot{x} + kx = F_c = k_p (x_{ref} - x) + k_d (\dot{x}_{ref} - \dot{x}) + k_i \left(\int_0^t (x_{ref} - x) \, dt \right)$$

$$m\dddot{y} + (b + k_d)\ddot{y} + (k + k_p)\dot{y} + k_i y = k_d \ddot{y}_{ref} + k_p \dot{y}_{ref} + k_i y_{ref}$$

the effect is that we have increased the order of the SYSTEM !!

IN STEADY STATE:

$$m \cancel{\dddot{y}} + \cancel{(b+k_d)\ddot{y}} + \cancel{(k+k_p)\dot{y}} + k_i y = \cancel{k_d \ddot{y}_{ref}} + \cancel{k_p \dot{y}_{ref}} + k_i y_{ref}$$

$$k_i y = k_i y_{ref} \rightarrow \boxed{y = y_{ref}}$$:)

WE DELETE REGIME ERROR,

BUT WE COULD MAKE OUR SYSTEM UNSTABLE

SO WHY PID ARE USEFUL??

1) INCREASE PERF. OF THE SYSTEM

2) MOVE THE ROOT LOCUS, SO WE CAN MAKE OUR SYSTEM STABLE!

3) CANCEL THE REGIME ERROR

• TAKE CARE THAT FOR AN HIGH K_P, the ACTUATOR COULD GO IN SATURATION, AND THUS WE CAN HAVE CLAMPING

AUTOMATIC GAIN OPTIMIZER !!

OPEN 1dof_opt.scm

the next hour we will see the optimizer for the suspension

OPTIMIZER:

REFERENCE (P,I,D) → → OUTPUT (P,I,D)

COST FUNCTION (SYS. ERROR) → → ERROR

there are many ways to fix the cost function in our case!

if(e > 0)

cost = e^2

else

cost = $(100 \cdot e)^2$ → PENALTY, BECAUSE WE WANT TO AVOID THIS CONDITION! when e < 0 we are having an OVERSHOOT

$$cost = \int_0^t cost \, dt$$

→ THIS IS BECAUSE WE NEED TO EVALUATE ALONG ALL THE TIME!

INSIDE THE OPTIMIZATOR:

- WE CAN SET UP AND DOWN BOUND [0] LOWER, [20, 20, 20] UPPER
- INITIAL AND FINAL VALUE OF THE TRUST REGION
- MAX ITERATION

NOTE: NOW PARAMETERS P, I, D are not DEFINED IN THE MODEL

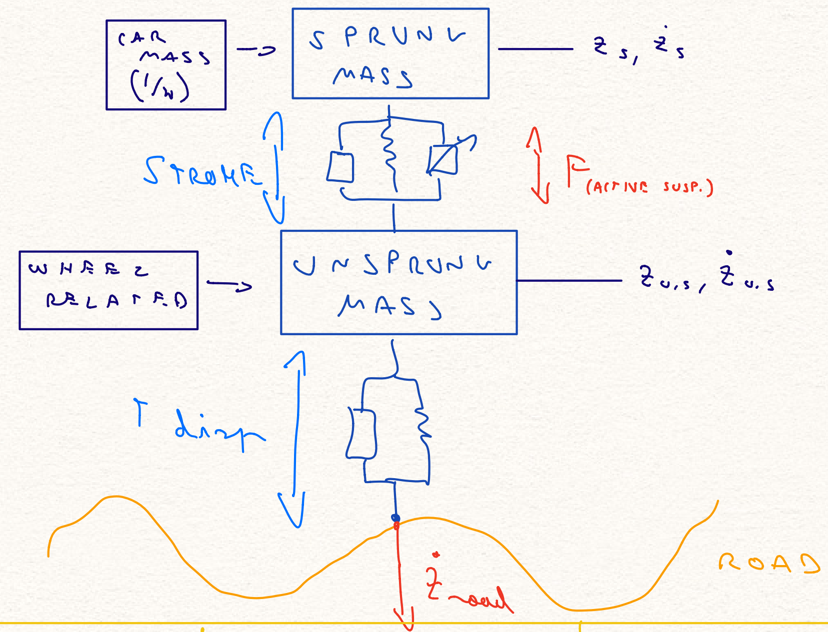
EVERYTIME THAT OPTIMIZATOR WORKS, WE WILL SET NEW PID IN THE BASE, AND THE NEXT RUN

$$\begin{bmatrix} P \\ I \\ D \end{bmatrix} = \text{GETFROMBASE}\left(\begin{bmatrix} P \\ I \\ D \end{bmatrix}, 10\right)$$

→ if they are not in base, get 10 (FIRST RUN)

4 ACTIVE SUSPENSION

- based on the Quarter car suspension:



DESIGN DECISIO

SYST. INPUT:

- ACTUATION FORCE →
- ROAD VELOCITY →

SYST. OUTPUT

→ TIRE DISPLACEMENT

→ UNSPRING MASS VELOCITY

→ SUSP. STROKE

→ SPRUNG MASS VELOCITY

↳ COMFORT OF THE DRIVER

WE START TO MODEL THE SYSTEM IN ALTAIR ACTIVATE

EXE 2

1) we create the model with mass ... etc

TIP:

VARIABLES IN "MODEL" → GLOBAL

VARIABLES IN "DIAGRAM HOME" → LOCAL INSIDE SUPERBLOCK

2) ADD A SIGNAL GEN + LPF

$$LPF = \frac{1}{\frac{1}{\omega}\cdot s+1} = \boxed{\frac{1}{\frac{s}{100}+1}} \rightarrow \text{FILTERS AT } f = \frac{\omega}{2\pi} = \frac{100}{2\pi} = 16\,Hz$$

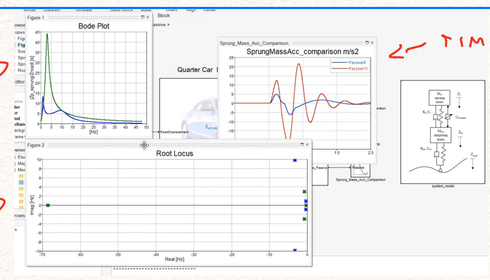
if we increase the SPRING DAMPING → WE ARE ADDING RIGIDITY TO THE SYSTEM, SO THE ACCELERATION WILL BE HIGHER
↳ SUSPENSION WILL NOT DEFORMATE TOO MUCH

NOW, IF YOU LOOK PROF'S FILE
ACTIVE_ SUSP._ PASSIVE . SCM

this is a 2^{nd} order 2 DOF SYSTEM EACH.
↳ SO WE HAVE 2 RESONANCE IN THE BODE PLOT
→ WE CAN ALSO SEE 4 - ROOTS ON THE ROOT LOCUS (2 DoF, each 2^{nd} order)
↳ THIS CAN BE DONE BECAUSE WE CAN LINEARIZE AUTOMATICALLY WITH MODELICA!

NOW, WE INCREASE AGAIN SPRING DUMPING!
WHAT HAPPEN?

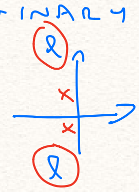
- IN BODE I WILL HAVE ONLY ONE HIGHER PEAK,
- THE TIME RESPONSE WILL BE WORSE
- IN THE ROOT LOCUS, 2 roots have Img part, and the other 2 have only real → this is why we have only ONE PEAK IN BODE



3RD ATTEMPT → WE DECREASE THE SPRUNG DUMPING

↳ we are making softer DAMPERS! WHAT WILL HAPPEN?

- we are getting less accellerations → BETTER
  ↳ BUT THE SYSTEM WILL HAVE VERY LONG OSCILLATIONS
- IN BODE PLOT we will have high spike in lower frequencies (BUT ALSO A SMALLER ONE IN HIGHER FREQ)
- in root locus I will see 2 ROOTS WITH VERY HIGH IMMAGINARY respect to real (less dumped behavior)



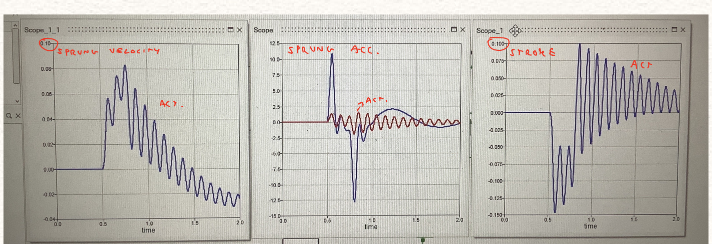
$\boxed{\text{3) NOW WE ADD ACTIVE SUSPENSION!!}}$

we add ACTION-REACTION FORCES AND WE ADD A CONTROLLER!

WE MAKE A SORT OF FEEDBACK LOOP WHERE THE REFERENCE IS ∅, AND THE FEEDBACK IS THE SPEED OF THE SPRUNG MASS! this mean that the car vertical speed = 0 → ACC = 0 :)

WE ADJUST THIS CONTROLLER BY ADDING A PID REGULATOR $k_p = 10^4$, $k_i = 0$, $k_D = 10^4$

THE RESULT IS PRETTY GOOD!



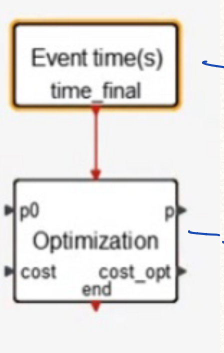
4) WE ADD THE OPTIMIZER

• FIRST OF ALL WE ADD A SAT. BLOCK AFTER THE PID
↳ BECAUSE PERF. OF ACTUAT. IS LIMITED

→ NOW WE HAVE WORST BEHAVIOR ☹ (MORE SIMILAR TO PASSIVE SUSP.)

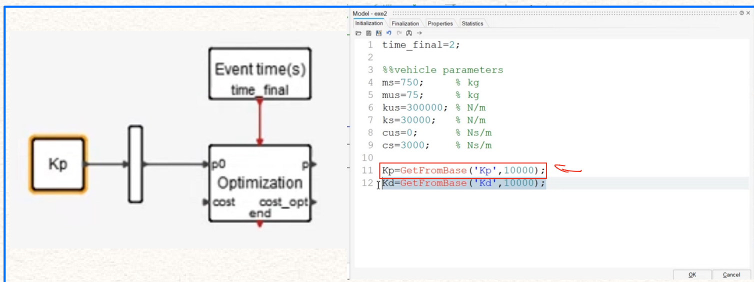
AFTER THIS, WE HAVE ALSO ADDED NON LINEARITIES!!

NOW, LET'S START TO OPTIMIZE THE K_p

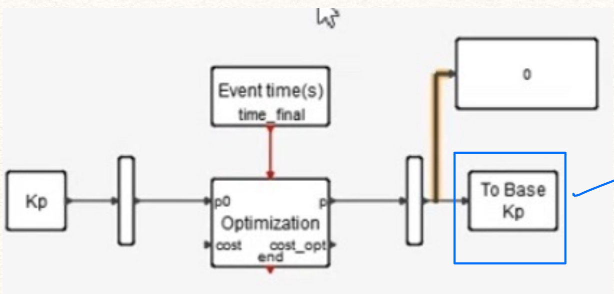


→ OPT. IS TRIGGERED WHEN FINAL_TIME IS ARRIVED (AT THE END)

→ OPTIMIZATION

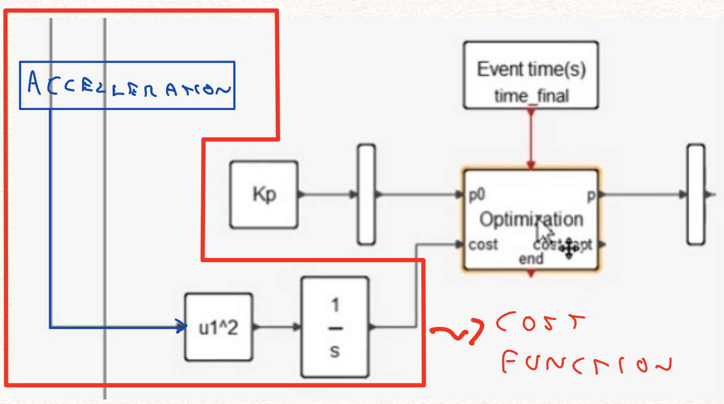


AND WE NEED TO ADD K_p IN THE WORKSPACE



→ THEN WE NEED SOMETHING TO OUTPUT THE K_p PARAM.

NOW THE COST FUNCTION:



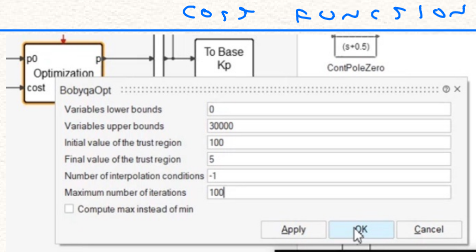
→ COST FUNCTION

WE HAVE BUILD THE COST FCN IN ORDER TO MINIMIZE THE ACC.

$$\text{COST} = \int_0^t acc^2 \cdot dt$$

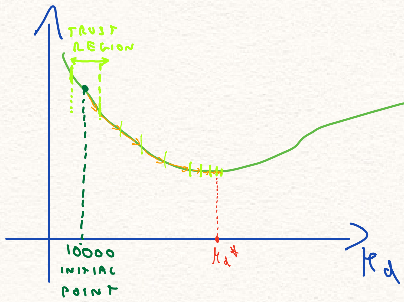
• why the squared? because bobyqa opt uses the gradient for the cost. so if i use the square I'm sure that I'm building a continuous derivative for cost function → simpler for optimizer

WE THEN SET ALL THE BOUNDARIES FOR THE OPTIMIZER...



HOW OPTIMIZATION IS WORKING?

WE HAVE ONLY A PARAM. (K_p) and I want to minimize a cost function ($\int_0^t acc^2 \cdot dt$)



the alg. starts by APPROXIMATING THE COST VALUE INSIDE THE TRUST REGION, THEN DECIDES WHERE TO MOVE KP IN ORDER TO MINIMIZE THE COST FCN

WHEN I'M NEAR THE MIN., MY ALG IS GOING TO SRINK THE TRUST REGION

in K_d^* we reached a very little value of TRUST REGION AND THE ALG. STOPS → THIS MEAN THAT WE REACHED THE CONVERGENCE

the same is if we have 2 parameter, but trust region now is a RADIOUS

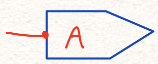
IN OUR CASE: START TRUST REG: 100

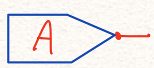
FINAL TRUST REG: 5

WE RUN THE SIMULATION, AND IT IS GOING TO REPEAT UNTIL IT REACH $K_p = 30'000$

WE CAN ADD ALSO A COOL VIEWER FOR THE COST FCN

NOW FOR HOMEWORK WE CAN ADD K_p, K_i OPTIMIZER!

TIP: SET-SIGNAL  an remember to set "GLOBAL"

GET-SIGNAL 

- ACTUATOR CAN BE ALSO IMPLEMENTED BY CHANGING THE DAMPER OPENING! → REAL CASE → IT IS CHEAPER, MINIMUM MODIFICATION, BUT WE ONLY CHANGE VELOCITY BEHAVIOUR

- WE CAN ALSO ADJUST THE SPRING CONSTANT THAT IS NOT ACTIVE

FIND SPRUNG MASS OF A REAL CAR TO USE
MAYBE STIFFNESS ARE NOT SIMPLE TO FIND

WHAT IF THERE ARE MORE THAN 1 RELATIVE MINIMUM?

1) I CAN START FROM DIFFERENT INITIAL CONDITION
2) I CAN USE GENETIC OPTIMIZER! BUT A BIT OVERKILL