Consultant: Geoff | Project: Edu-101 [page: 1/2] | Date: 20018_08_06

Mass spring damped system

Damped system in continuous time ... η => damping coefficient.
f(2), g(2) go in pair with "init"

```
dynSys(f(2), g(2), init, NumIt, dt) := | t := init
                                       | u := t
                                       | for k ∈ 1..NumIt
                                       |     | t := eval(t + dt·f(t, g(t, r)))
                                       |     | u := augment(u, t)
                                       | u
```

Linear system with 2 states and 1 input: unit step response 'K=1'.

$M:=1$, $\eta:=0.75$, $K:=0.5$, $init:=\begin{bmatrix}0\\0\end{bmatrix}$

$n:=150=$ "number of samples"
$step:=0.1=$ "increment size"
$K=$ "step input"

For "zoom detail" of the front end
<= decrease step size.
<= experiment various 'K'

$$MasSprDmp(X, u) := \begin{bmatrix}\frac{1}{M}\cdot\left(u_1-\eta\cdot X_1-\frac{1}{K}\cdot X_2\right)\\ X_1\end{bmatrix}$$

The mass-spring differential system

$Ctrl(t, r) := [1]$

Constant input, i.e. no state feedback in this case

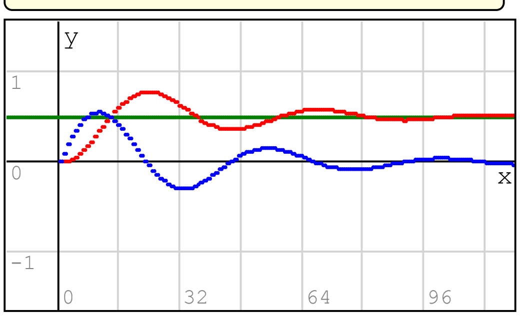
Observe: Ctrl(x,r):=[1]...step input
[1] is a unit vector from matrix panel

Sol := dynSys(MasSprDmp(any1, any2), Ctrl(any1, any2), init, n, step)

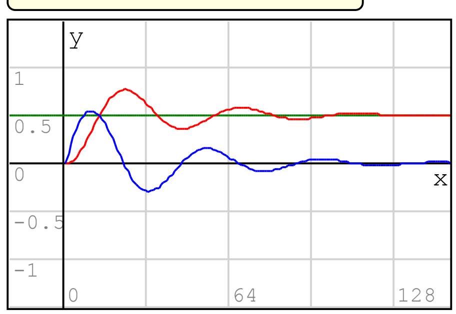
$v := row(Sol, 1)^T$ $\quad$ $s := row(Sol, 2)^T$

```
Response := | sol := dynSys(MasSprDmp(v1, v2), Ctrl(v1, v2), init, n, step)
            | U := | for i ∈ 1..n+1
            |      |     idx_i := i
            |      | augment(idx, sol^T)
            | [ augment(col(U, 1), col(U, 2)) ]
            | [ augment(col(U, 1), col(U, 3)) ]
```

Auto-index from the canvas var 'x'



Export discrete response



Consultant: Geoff	Project: Edu-101 [page: 2/2]	Date: 20018_08_06

```
dynSysDiscr(f(2), g(2), xo, NumIt):= | "initialise 't & x' with xo"
                                     | [t:= xo  x:= xo]
                                     | for k ∈ 1 .. NumIt
                                     |   | t:= eval(f(t, g(t, r)))
                                     |   | x:= augment(x, t)
                                     | x
```

EXAMPLE 2 - Linear Feedback Stabilization

$$\mathrm{LinSys1}(x, u) := \begin{bmatrix} 0.5 \cdot x_1 + x_2 + u_2 \\ 0.5 \cdot x_1 + x_3 \\ -2 \cdot x_1 + u_1 + u_2 \end{bmatrix}$$

Linear system with 3 states and 2 inputs

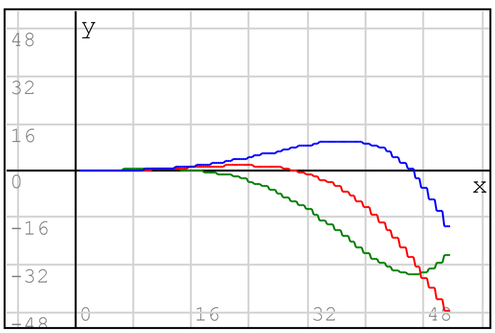
$$\mathrm{Ctrl2}(x, r) := \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

Constant input

$$xo := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Sol2:= dynSys(LinSys1(any1, any2), Ctrl2(any1, any2), xo, 50, 0.1)

$x1 := \mathrm{row}(Sol2, 1)^T$ $x2 := \mathrm{row}(Sol2, 2)^T$ $x3 := \mathrm{row}(Sol2, 3)^T$



$$K := \begin{bmatrix} 1.4 & -0.5 & -1.4 \\ -3.3 & -1.9 & 0 \end{bmatrix}$$

The state grows indefinitely, i.e. the system is unstable.

$$\mathrm{Ctrl3}(x, r) := K \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

Try to stabilize with proportional state feedback matrix:

Sol3:= dynSys(LinSys1(any1, any2), Ctrl3(any1, any2), xo, 100, 0.05)

$x1 := \mathrm{row}(Sol3, 1)^T$ $x2 := \mathrm{row}(Sol3, 2)^T$ $x3 := \mathrm{row}(Sol3, 3)^T$