## Today's EEE 151 Lecture

- Some more math tools.
- Laplace transform.
- Forcing functions.
- Familiarization with Octave.




## Laplace Transform

- Suppose $f(t)$ satisfies

$$\int_0^\infty |f(t)e^{-\sigma t}|dt \; < \; \infty$$

for some finite real $\sigma$

- Define the Laplace transform as

$$F(s) \; = \; \int_0^\infty f(t)e^{-st}dt$$
$$\Updownarrow$$
$$F(s) \; = \; \mathscr{L}[f(t)]$$




## Laplace Transform

- Comparison

| Time domain | Frequency domain |
| --- | --- |
| $f(t)$ | $F(s)$ |
| $t \in \mathbb{R}^+$ | $s \in \mathbb{C}$ |
| differential equation | algebraic equation |

- We will apply Laplace transforms to LTI continuous-time systems to make our lives easier.

  Looking to the future, $z$-transforms will be used in LTI discrete-time systems.




## Laplace Transform

- Laplace transform theorems.
  - multiplication by a constant

$$\mathscr{L}[kf(t)] \; = \; kF(s)$$

  - sum and difference

$$\mathscr{L}[f_1(t) \; \pm \; f_2(t)] \; = \; F_1(s) \; \pm \; F_2(s)$$

  - differentiation

$$\mathscr{L}\left[\frac{d}{dt}f(t)\right] \; = \; sF(s) \; - \; f(0)$$

$$\mathscr{L}\left[\frac{d^n}{dt^n}f(t)\right] = s^nF(s) \; - \; s^{n-1}f(0) \; - \; \ldots$$
$$- \; sf^{n-2}(0) \; - \; f^{n-1}(0)$$

- Laplace transform theorems.
  - integration

$$\mathscr{L}\left[\int_0^t f(\tau)d\tau\right] = \frac{F(s)}{s}$$

  - shift-in-time

$$\mathscr{L}[f(t-T)u_s(t-T)] = e^{-Ts}F(s)$$

  - initial-value theorem

$$\lim_{t\to 0} f(t) = \lim_{s\to\infty} sF(s)$$

  - final value theorem

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s)$$








- Laplace transform theorems.
  - complex shifting

$$\mathscr{L}\left[e^{\mp at}f(t)\right] = F(s\pm a)$$

  - real convolution

$$F_1(s)F_2(s) = \mathscr{L}\left[\int_0^t f_1(\tau)f_2(t-\tau)d\tau\right] = \mathscr{L}[f_1(t) * f_2(t)]$$

- Other important Laplace stuff.
  - inverse Laplace transform.
  - partial fraction expansions.

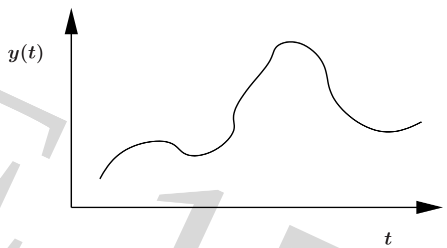







- Why do we want to study forcing functions?

Do we really want to deal with something like the following?




- Why not use standard inputs?
  - step input.
  - ramp input.
  - parabolic input (sometimes).
  - sinusoids.

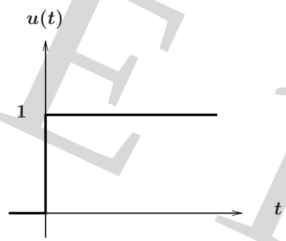







- Step function.

$$f(t) = u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

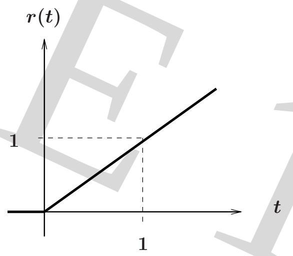



$$F(s) = \mathscr{L}[u(t)] = \int_0^\infty u(t)e^{-st}dt = -\frac{1}{s}e^{-st}\Big|_0^\infty = \frac{1}{s}$$

## Forcing Functions

- **Ramp function.**

$$f(t) = r(t) = \begin{cases} t, & t \geq 0 \\ 0, & t < 0 \end{cases}$$




- **Laplace transform of a ramp function?** $\mathscr{L}[r(t)] = ?$



## Forcing Functions

- **Sinusoids**

$$f(t) = sin(\omega t)$$
$$f(t) = cos(\omega t)$$

- **Laplace transforms of sine and cosine functions?**

$$\mathscr{L}[sin(\omega t)] = ?$$
$$\mathscr{L}[cos(\omega t)] = ?$$



## Familiarization with Octave

- **What is Octave?**
  - mathematical package suited for matrix operations.
  - functionality has been extended by toolboxes.
  - we are primarily interested in the control toolbox.

- **Help**

```
>> help
>> help [command name]
>> help plot
```



## Familiarization with Octave

- **Variables**

```
>> x = 3, y = 2*x
>> x = [1, 2, 3]
>> y = [1+j, 2+pi*i, -sqrt(-1)]
>> x(3)
```

- **Vector operations**

```
>> z = [4, 5, 6]
>> w = x + z
>> x'*z
>> x.*z
```

- **Matrix operations**

```
>> A = [1 2 3; 4 5 6; 7 8 9]
>> A(1,2)
>> A(1:2, 3)
>> A(3,:), A(:,2), A(1:2,:)
```

- **Generating a series of numbers**

```
>> t = 0:0.1:10
>> t = linspace(5,20,30)
>> t = logspace(1,100,30)
```






Familiarization with Octave

- **Special names**

```
>> eye(3, 3)
>> ones(2, 4)
>> zeros(2, 3)
```

- **Plotting**

```
>> t = 0:0.1:2*pi;
>> f = sin(t);
>> plot(t, f)
>> g = cos(t);
>> plot(t, f, t, g);
>> clf; plot(t, f), hold, plot(t, g)
```






Familiarization with Octave

- **Simulations**

```
>> [t,x] = ode23('xprime', [t0 tfinal], x0);
>> [t,x] = ode23('xprime', [0 10], [5; 0]);
```

- **What is 'xprime'?**
  **This is the m-file 'xprime.m' that contains the function for computing the derivative of the state variable, $\dot{x}$, based on the current state $x$ and the input $f$.**






Familiarization with Octave

- **Recall state-space representation.**
  $\dot{x} = \textit{state term} + \textit{input term}$

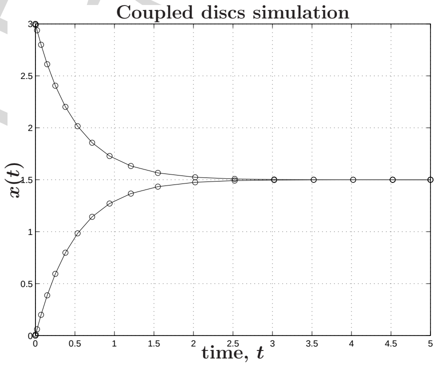
- **From state-space representation, one can easily extract $\dot{x}$ and compose 'xprime.m.'**

- **Example. Coupled discs model.**

```
function xp = xprime(t, x);
B = 1; J1 = 1; J2 = 1;
xp = B*[-1/J1 1/J1; 1/J2 -1/J2]*x;
```

- **Example. Coupled discs model.**

```
>> ode23('xprime', [0 5], [3; 0]);
```










- **Laplace transforms, an essential tool for control.**

- **What typical forcing functions do we encounter?**

- **How does Octave help us?**