Today's EEE 101 Lecture

- Some more tools.
- Laplace transform.
- functions.
- Familiarization with Matlab.

Laplace Transform

- Suppose $f(t)$ satisfies

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

for some real σ

- Define the as

$$F(s) \ = \ \int_0^\infty f(t)e^{-st}dt$$

$$\Updownarrow$$

$$F(s) \ = \ \mathscr{L}[f(t)]$$

Laplace Transform

- Comparison

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

- We will apply Laplace transforms to LTI 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

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

  -

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

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

Laplace Transform

- Laplace transform theorems.
  - –

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

  - shift-in-time

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

  - initial-value theorem

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

  - theorem

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

Laplace Transform

- Laplace transform theorems.
  - complex shifting

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

  - real

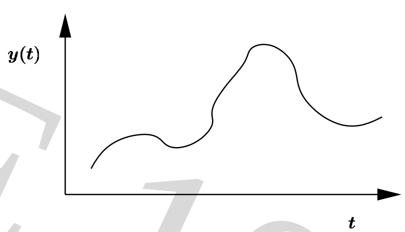
$$F_1(s)F_2(s) = \mathscr{L}\left[\int_0^t f_1(\tau)f_2(t-\tau)d\tau\right] = \mathscr{L}[f_1(t) * f_2(t)]$$

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

Forcing Functions

- Why do we want to study forcing functions?

Do we really want to something like the following?

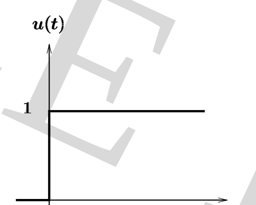


- Why not use inputs?
  - step input.
  - ramp input.
  - parabolic input (sometimes).
  - sinusoids.

Forcing Functions

- function.

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

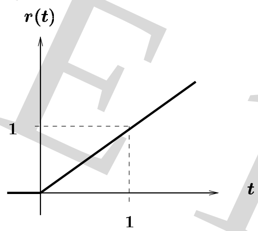


$$F(s) = \mathscr{L}[u(t)] = \int_0^\infty u(t)e^{-st}dt = -\frac{1}{s}e^{-st}\Big|_0^\infty = \frac{1}{s}$$

Forcing Functions

- function.

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



- 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 functions?

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

Familiarization with

- What is Matlab?
  - mathematical package suited for operations.
  - functionality has been extended by toolboxes.
  - we are primarily interested in the control toolbox.
  - (not P100 as some people might think).
- Octave, a Matlab alternative.
  - almost same syntax as Matlab.
  - runs on different platforms.
  - comes bundled with RH Linux 7.2.
  - free. no feeling.

Familiarization with

- Help

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

- Variables

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

Familiarization with

- **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,:)
```


Familiarization with

- **Generating a of numbers**

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

- **Special names**

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


Familiarization with

- **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)
```

-

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


Familiarization with

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

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

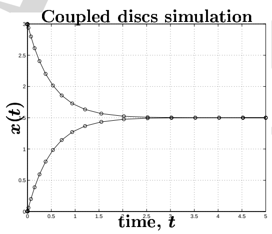
- **From state-space representation, one can easily extract $\dot{x}$ and compose 'xprime.m.'**

Familiarization with

- **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;

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



Summary of Today's EEE 101 Lecture

- **Laplace transform comes back to haunt us.**

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

- **How does Matlab help us?**

- **Can it be true?**
  Two/three more lectures until the first exam.