

# EECE 301

## Signals & Systems

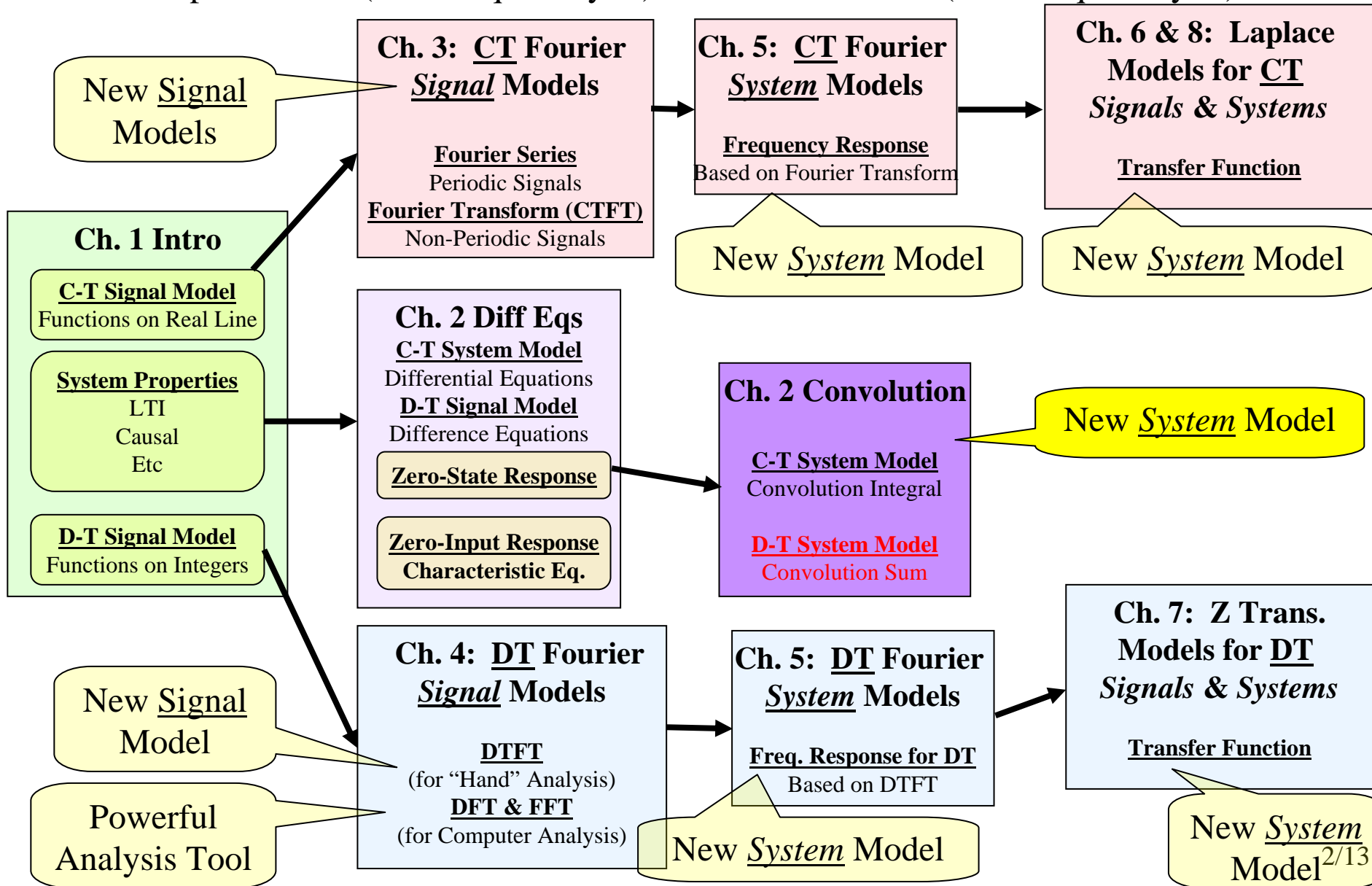
Prof. Mark Fowler

### Note Set #9a

- Example: Using D-T Convolution in an Application of Measuring the Speed of a Motor

# Course Flow Diagram

The arrows here show conceptual flow between ideas. Note the parallel structure between the pink blocks (C-T Freq. Analysis) and the blue blocks (D-T Freq. Analysis).



## Example: D-T Processing for Motor Speed

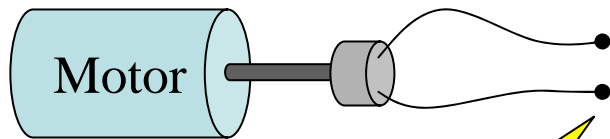
The m-file `motor_speed.m` is available for download from the course website.

To run it all you need to do is:

1. Download it to a folder on your computer
  - a. Make sure Matlab's current directory is the folder where you have put the m-file (change it if needed)
2. At the Matlab command line type: `motor_speed`
3. The m-file will run and create several figures

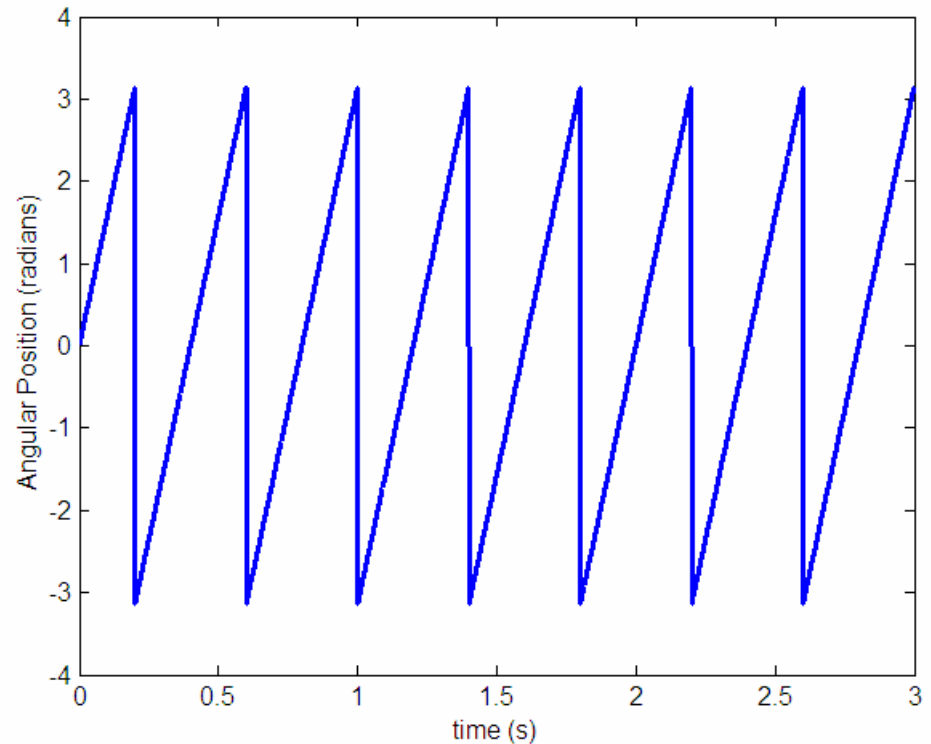
## Physical Set-Up of Motor and Sensor

Suppose you have a motor whose shaft is connected to a sensor that can measure the angular position of the shaft... its values will range between  $-\pi$  and  $\pi$ .



Assume that the sensor generates  
1 volt/radian

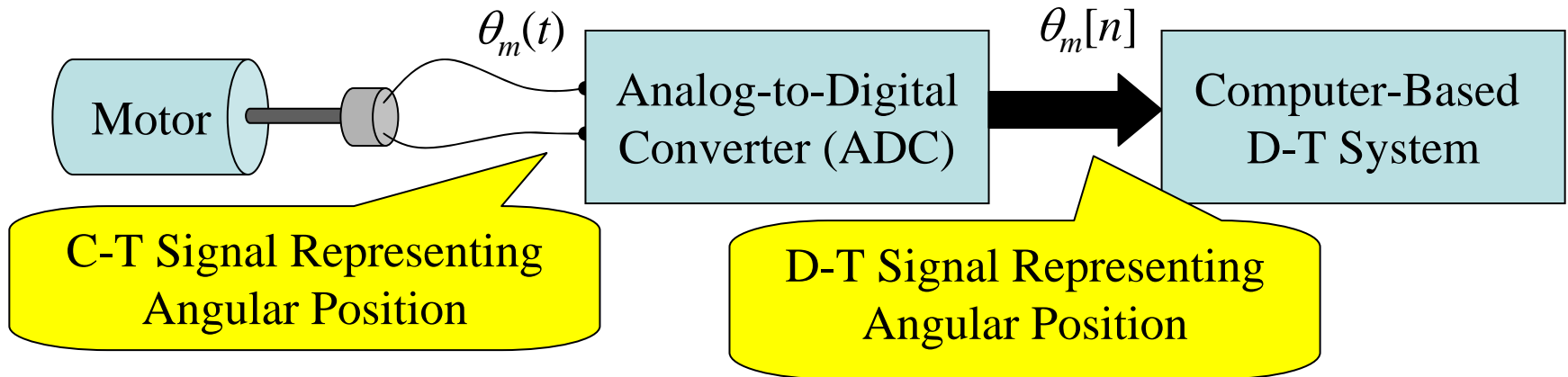
**Output of Sensor when Motor spins  
at constant angular rate of  $5\pi$  rad/sec**



**Q: How do we measure the speed of the motor?**

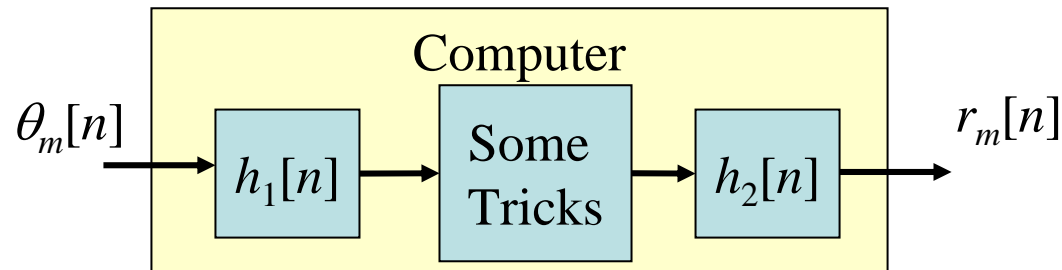
# Motor and D-T System for Processing

First we could sample the sensor's signal at a fast enough rate to get a D-T signal that accurately conveys the C-T signal coming from the sensor:



The resulting samples could be put into a computer (as binary numbers) and processed using D-T convolution.

Now... the trick is to determine how to design the D-T system (i.e., choose the system's impulse response  $h[n]$ ) to get a measure of the angular rate  $r_m[n]$  !!!



**We can use matlab to explore how to do this.**

# Simulate the Angular Position's D-T Signal in Matlab

```
% Create time samples spaced 1 ms apart
```

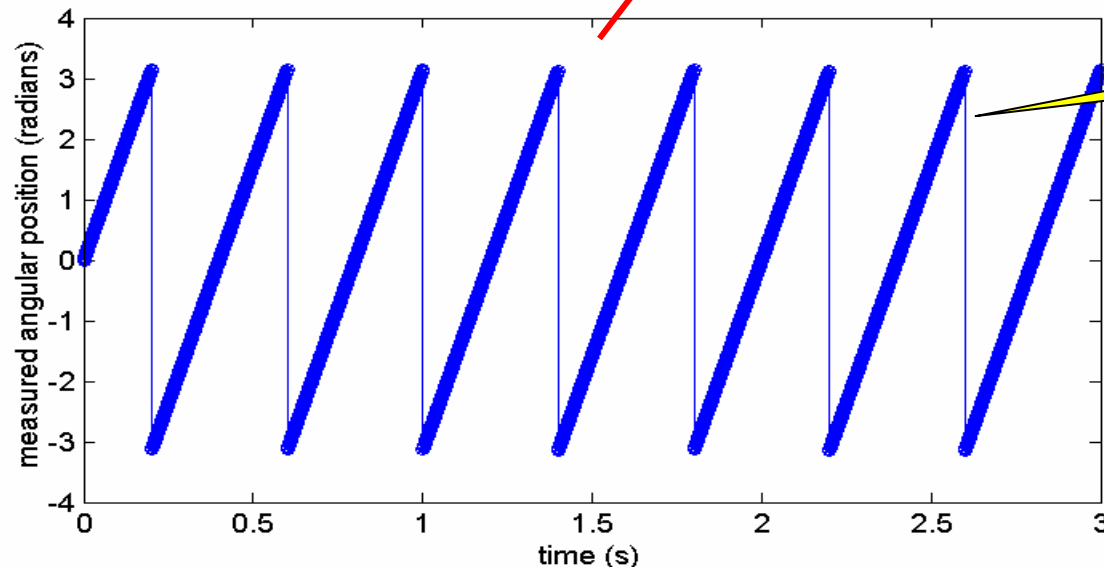
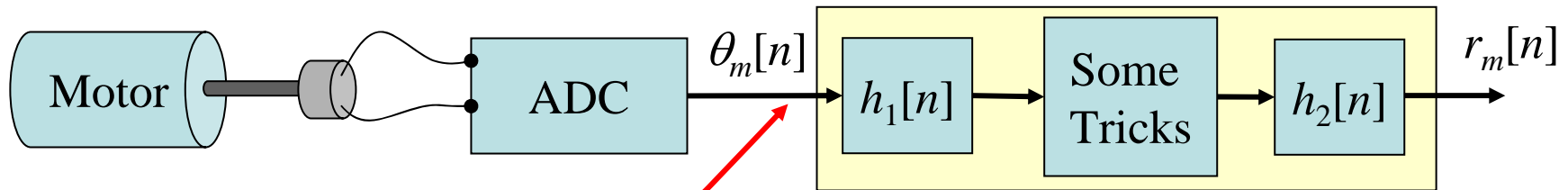
```
del_t=0.001;
```

```
t=0:del_t:3;
```

```
theta=5*pi*t; %% simulates angular position of motor shaft spinning at a constant rate
```

```
%% use some matlab "tricks" to simulate the fact that measured angle lies in  $-\pi$  to  $\pi$ 
```

```
theta_m=angle(exp(j*theta));
```



Note: Discontinuities

Figure 1 from  
motor\_speed.m

# 1<sup>st</sup> D-T Convolution Approximates Computing a Derivative

```
% specify impulse response of D-T system that computes  
%% the time derivative of the input  
h_1=[-3 16 -36 48 -25]/(12*del_t);
```

Comes from Branch of Math  
known as “Numerical Methods”

```
%%% Compute D-T system's output using convolution  
x=theta_m; % just rename position measurements to "typical" input symbol  
y=conv(x, h_1); % Matlab has a command for convolving two signals
```

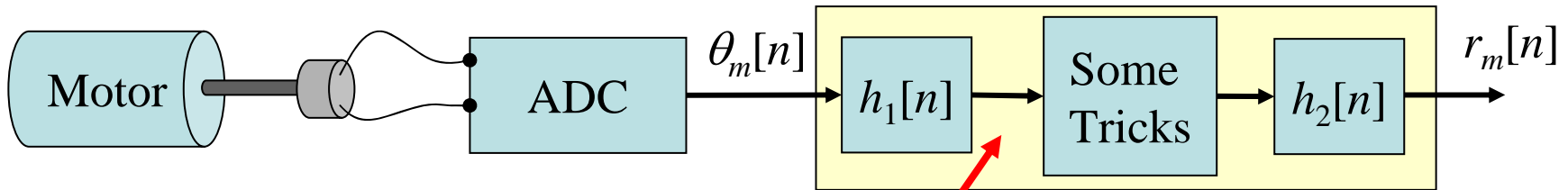
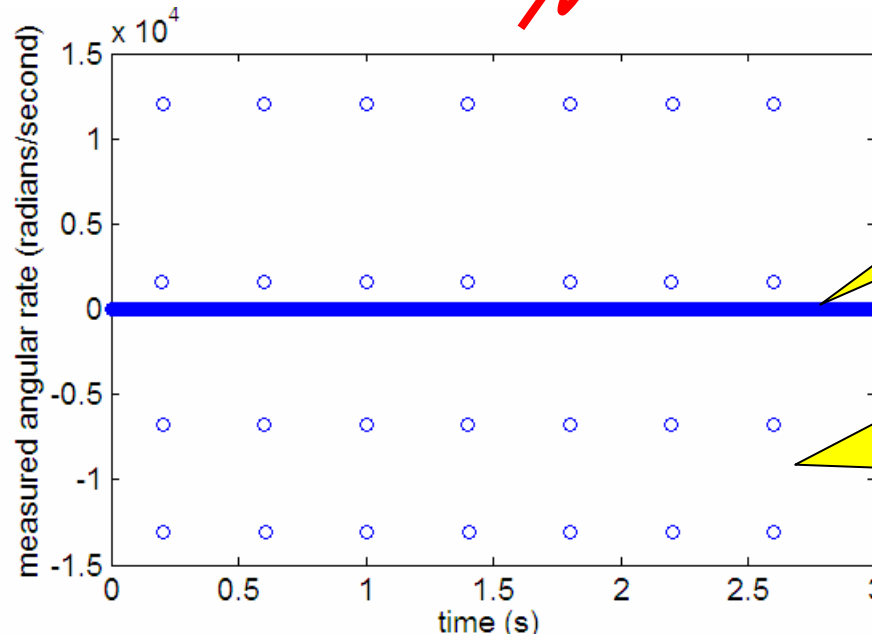


Figure 2 from  
motor\_speed.m



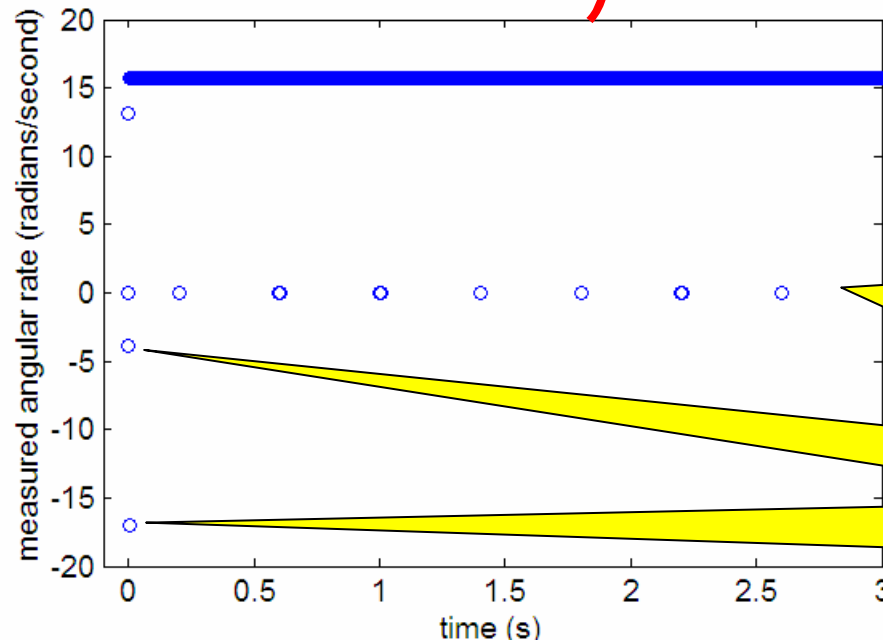
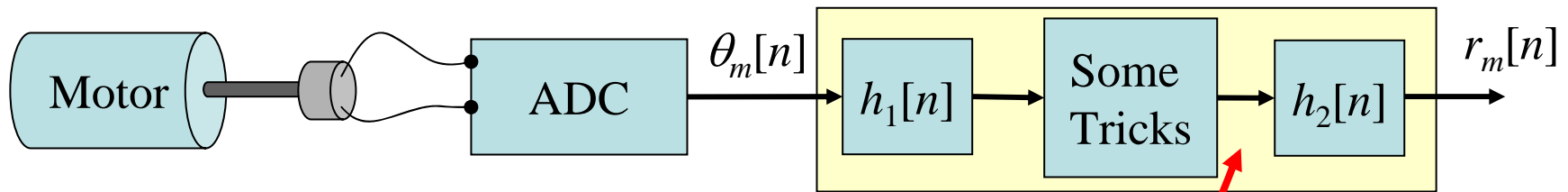
≈ Correct Values  
For Rate

Large Spurious  
Values due to  
Discontinuities

# D-T Tricks to Fix the Effects of the Discontinuities

```
%% we can do some computer-based "exception processing" to get rid of those  
%% by throwing away output values whose absolute value is bigger than we would expect  
%% the rotational rate to be
```

```
I=find(abs(y)>500); % find the indices of such large values  
y(I)=0; % replace the values with zeros at those locations
```



≈ Correct Values  
For Rate

Zeros Inserted to  
Replace Spurious  
Values from  
Discontinuities

“Transient” values  
due to convolution

Figure 3 from  
motor\_speed.m



## D-T Convolution to Further Fix the Discontinuities

In the previous stage we inserted zeros at the discontinuity points... so what we end up with (ignoring the transient) is many correct values with a few zero values included.

So... imagine taking a small block of  $N$  such data points and taking the data average (add them up and divide by  $N$ )... the small number of zero values will have a small effect and we should end up with the average being close to the true value.

If we imagine “sliding the block” used for averaging ahead by one sample each time... then we would get an average for each block that should be a good estimate of the angular rate

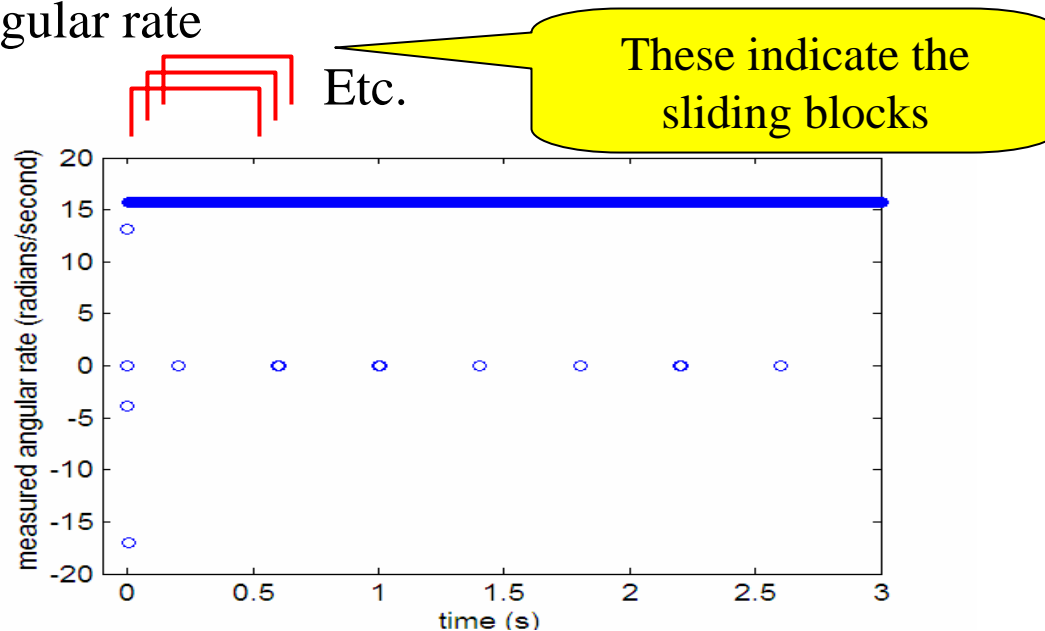
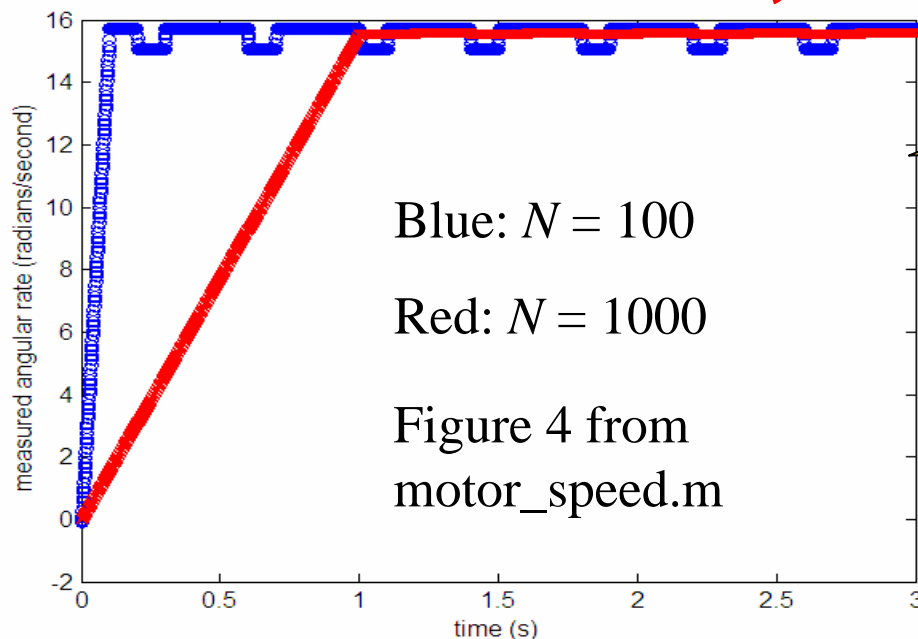
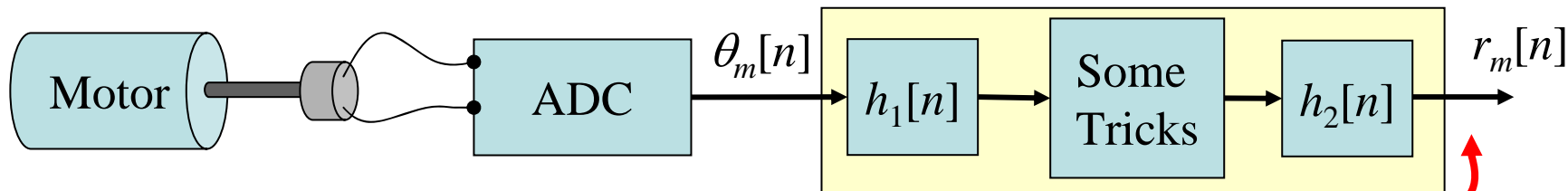


Figure 3 from  
motor\_speed.m

With a little thought we can convince ourselves that this sliding average is nothing more than convolving the “zero-corrected” signal with a D-T signal that has  $N$  ones and zeros elsewhere and then dividing by  $N$ :

```
y2a=conv(y,ones(1,100))/100; % do sliding average of 100 pts to reduce effect of inserted zeros
```

```
y2b=conv(y,ones(1,1000))/1000; % do sliding average of 1000 pts to explore effect of N value
```



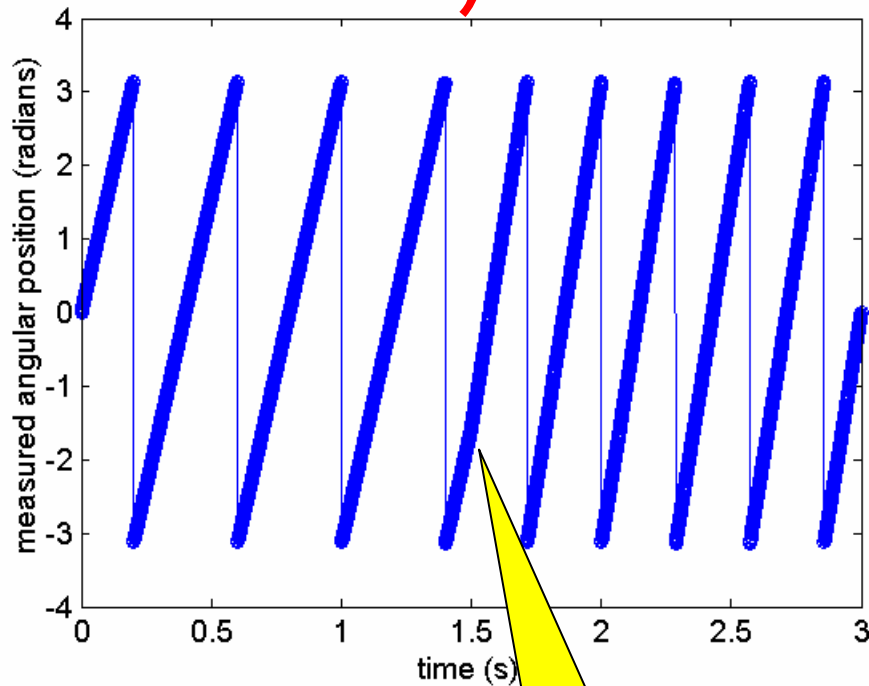
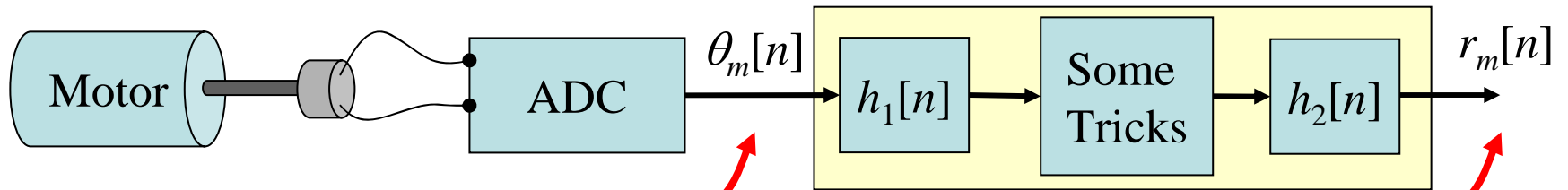
This indicates the following trade-off for the size of the averaging block:

	Small N	Large N
Pro	Rapid Response Time	Small Fluctuation in Values
Con	Large Fluctuation in Values	Slow Response Time

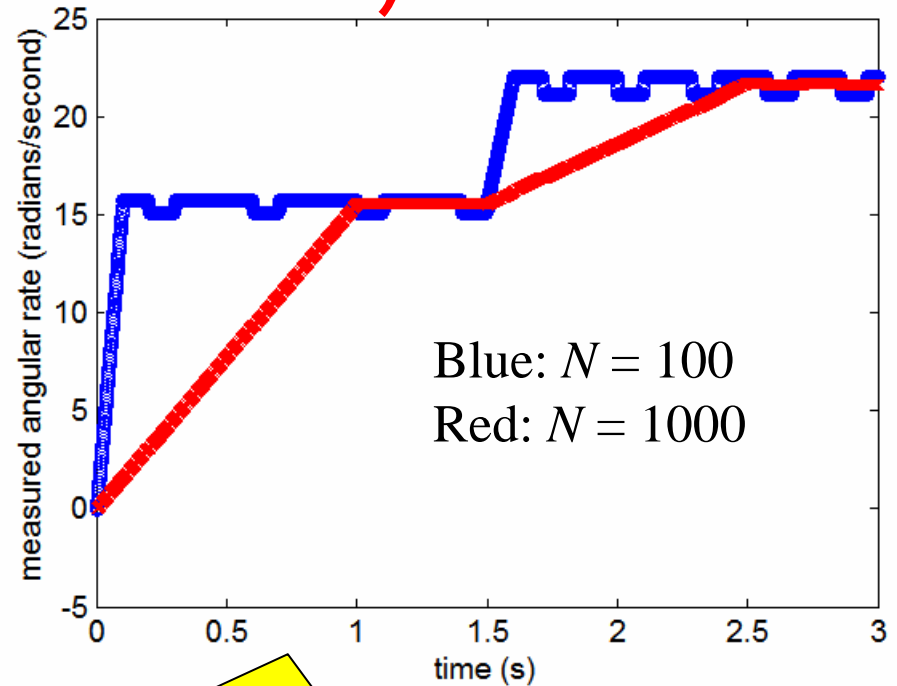
**Admission:** The way we did it (replace anomalies with zeros, smooth with convolution) is probably not the best way to do this:

A better “Trick” would be to replace each anomaly with the non-anomaly value that precedes it. This would reduce the jumps due to the inserted zeros and reduce the need for the second convolution.

# Exploring Impact of Abrupt Rate Change



Rate Change



**Bigger  $N$  gives:**

- More accurate measure...
- But... slower response to changes

Figures 5&6 from motor\_speed.m

## Things We've Seen Here

- Using Matlab to simulate D-T signals that will be obtained from an ADC
- Use D-T convolution to approximately compute the derivative of a signal
  - The D-T signal represents samples of some “underlying” C-T signal
- Use tricks to correct anomalies and spurious results
  - In practice this kind of thing is pretty common
- Use D-T convolution to “smooth out” jumps
  - Here the jumps were due to physical characteristics of angle measurement
- We have seen a fundamental trade-off that often arises in linear systems:
  - A system that provides more “smoothing” capability typically responds slowly to signal changes
  - So... if we have a noisy signal, we can remove a lot of the noise but only at the expense of smoothing out some of the signal's rapid changes