

Introduction to System Identification

1.1: Background concepts

- The goal of system identification is to develop an “appropriate” mathematical model of a specific dynamic system using observed data combined with:
 1. Basic mechanics and dynamics, and
 2. Prior knowledge of relationships between signals.
- Generally, models can be of different types
 - White box: A physical/analytical description of the dynamics of a system for which a lot of information is known *a priori*. For example, $F(t) = ma(t)$, or $V(t) = RI(t)$, and so forth.
 - ◆ Once the model description is determined, system identification comprises measuring the unknown parameters directly, or performing experiments designed to infer the parameter values indirectly.
 - ◆ We do not look at white-box system identification in this course.
 - Black box: A completely empirical description of the dynamics of a system for which essentially no information is known *a priori*.
 - ◆ This is the focus of this course.
 - Gray box: A combination of the two.
 - ◆ If the “known” white-box part can be subtracted out, the unknown part may be identified using methods from this course.

KEY RULE: Don't estimate what you know.

- System identification is not the end of the story.
- It is best for preliminary analysis to get initial quick-and-dirty solutions to problems, or as a last resort.

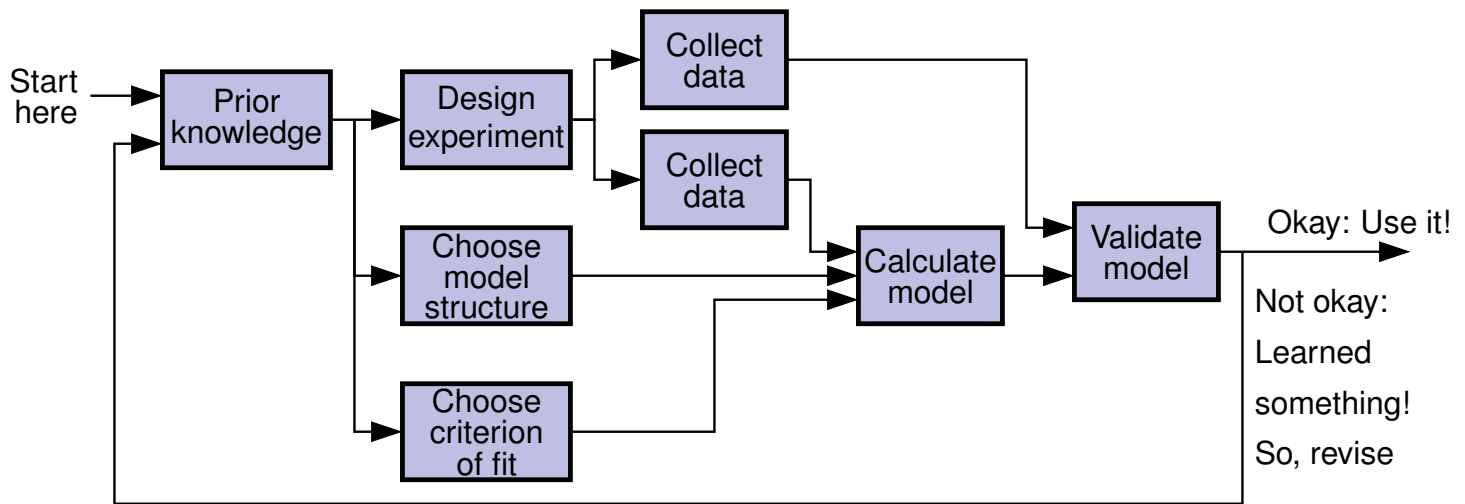
- The models we identify will be primarily input/output models.
 - Even though we will look at state-space methods, the identified state may not correspond directly to any physical state description.
 - If that is desired, prior knowledge must be incorporated into the methods to enforce this relationship via constraints.
 - System ID, in general, tends to be very experimental, heuristic.
 - We will develop some essential tools required to perform the task

BUT

It will take many more tries (years?) to develop the intuition necessary to get **good, low-order** models.

 - Need to work with real data as much as possible.
- Why do system ID?
 - Can develop models for systems with very complex dynamics and/or systems with unknown physical parameter values.
 - Really should be done in parallel with the development of an analytic model (well-formed formula, WFF).
- End use of model dictates model accuracy requirements:
 - For use in control system design,
 - Estimation (of states not available),

- Prediction (of response to different inputs).
- System identification process: Iterative on many levels



EXPERIMENT: Need to design experiment well to get useful data.

MODEL STRUCTURE: Many choices, pick one based on our understanding of system dynamics.

FIT CRITERION: Used by optimization to determine model parameters.

CALCULATE MODEL: Optimization to select model parameters.

EVALUATION: Validate model to make sure that the fit is reasonable.

Issues encountered when performing system ID

Some issues with experiment design

- Open loop or closed loop?
 - Often no choice (e.g., if system is open-loop unstable), but,
 - Closed loop introduces many complicating factors.
- What is the input sequence?
 - Frequency content has big impact!

- Actuator limits interfere (slew rates, saturation).
- What sampling rate and data length (memory)?
- Data filtering: Must deal with drifts, biases, outliers; attenuate noise.

Some issues with model structure

NON-PARAMETRIC: Frequency response plot, or impulse response. We focus on these models during first weeks of the course.

PARAMETRIC: Capture dynamics in a simple parametrized structure. For example, transfer-function form

$$G(s) = \frac{s + \alpha}{s^2 + \beta_1 s + \beta_2}.$$

We focus on these for the rest of the course (first: transfer functions; next: state space).

SYSTEM CLASS: For example, linear versus nonlinear (we focus on linear models, but text also talks about nonlinear).

MODEL SIZE: For example, number of poles and zeros of system transfer function and disturbance transfer function. Also, how many delays between input and output?

Some issues with fitting the model

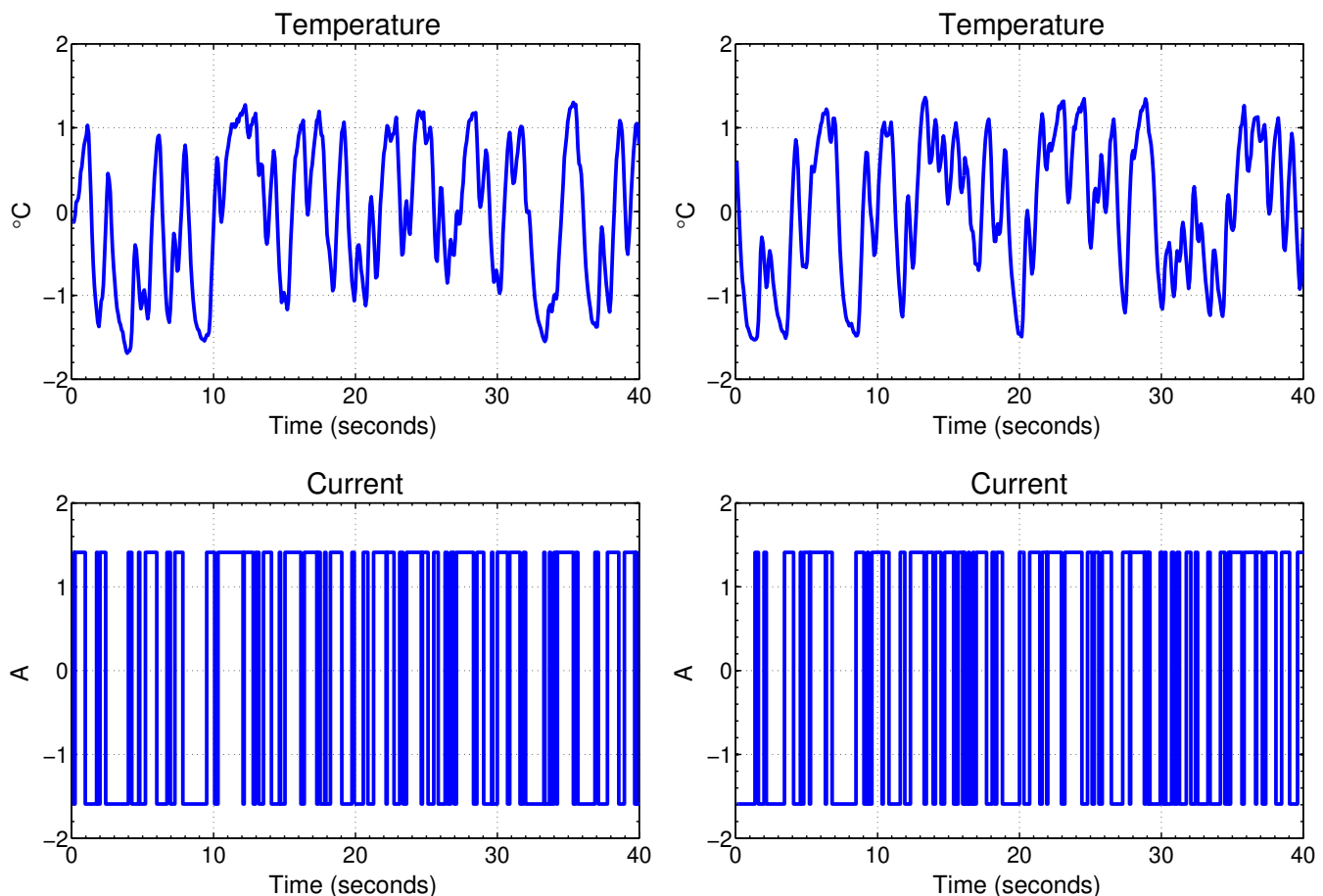
- Big tradeoff between accuracy of final model and ease of solution in finding a decent model.
- Significant degree of user input required.
- Does the process always work?

Some issues with validation

- Different data sets used to create model and validate model.
- Validation method chosen depends on ultimate use of model (*e.g.*, prediction versus estimation).
 - Time- and frequency-domain analysis of the error.
 - Stochastic analysis of the residual error.
- Does result imply that we should make changes to:
 - Experiment (input sequence),
 - Model choice (order, type, . . .),
 - Objective function for fit,
 - Optimization procedure,
 - All of the above?

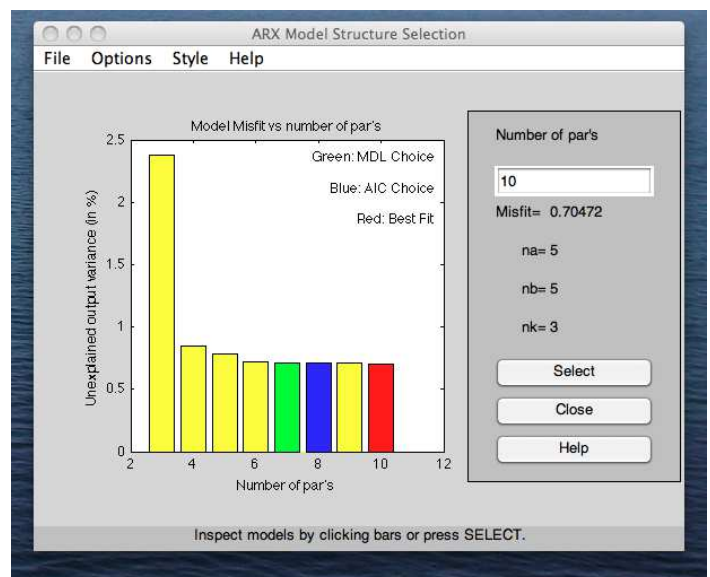
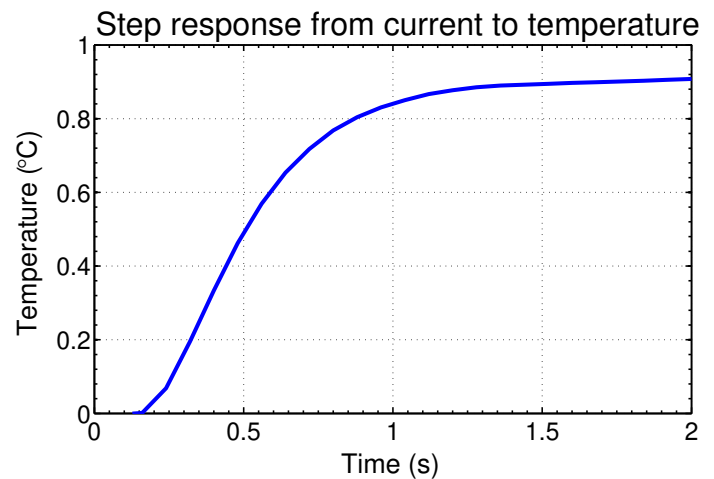
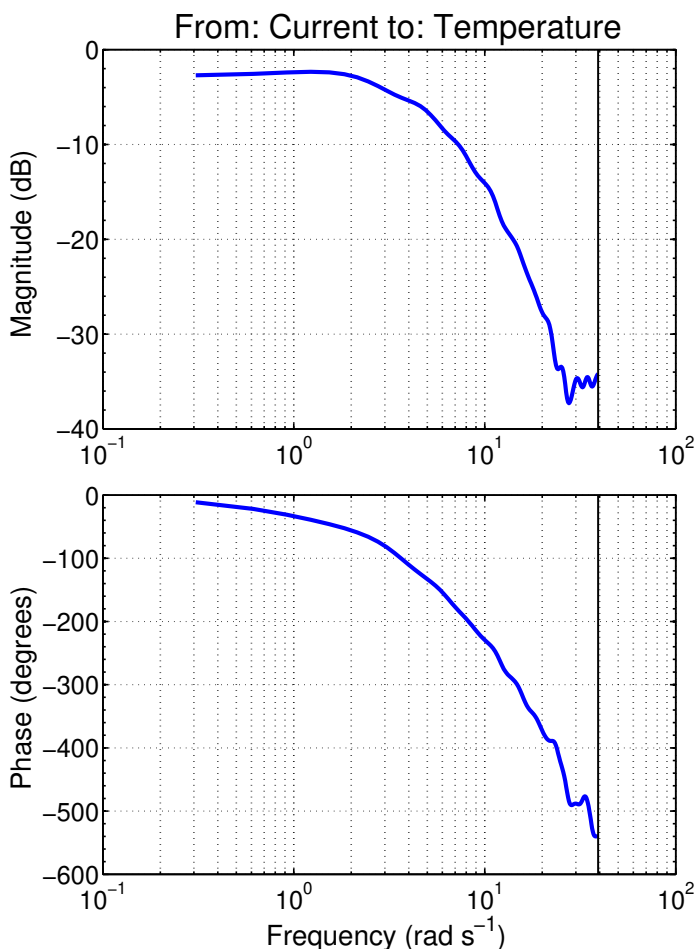
1.2: Dryer example

- Data from a laboratory model of a hair dryer.
 - Input: Current applied to some heating elements (resistors).
 - Fan blows the air past the heating coils.
 - Output. Thermistor measures the temperature variations.
 - Below: Main I/O data set (left), and validation I/O data set (right).



- Making a physical model would require that we understand:
 - How the coils respond to applied current (heat source efficiency).
 - How the air and coils interact (heating/cooling).
 - Air flow propagation through nozzle.
- Can try to make a model from the measured data only

- Rich input sequence, and fairly complex response.
- By the processes that we will develop in this course, we can develop a discrete time model for this system.
 - Typically start by plotting “Bode” plot of data to get an idea of likely system order, number of poles/zeros.
 - ◆ Note that the rapidly decreasing phase is an indicator of pure delay terms $e^{-s\tau}$ in the transfer function.
 - So, also plot estimated step response to get an idea of how many delays. In this case, around three samples.
 - ◆ Can also use GUI tool to select model “size.”



- We use some built-in routines in MATLAB to estimate the model, which gives us the following output (“ q^{-1} ” = unit delay):

```

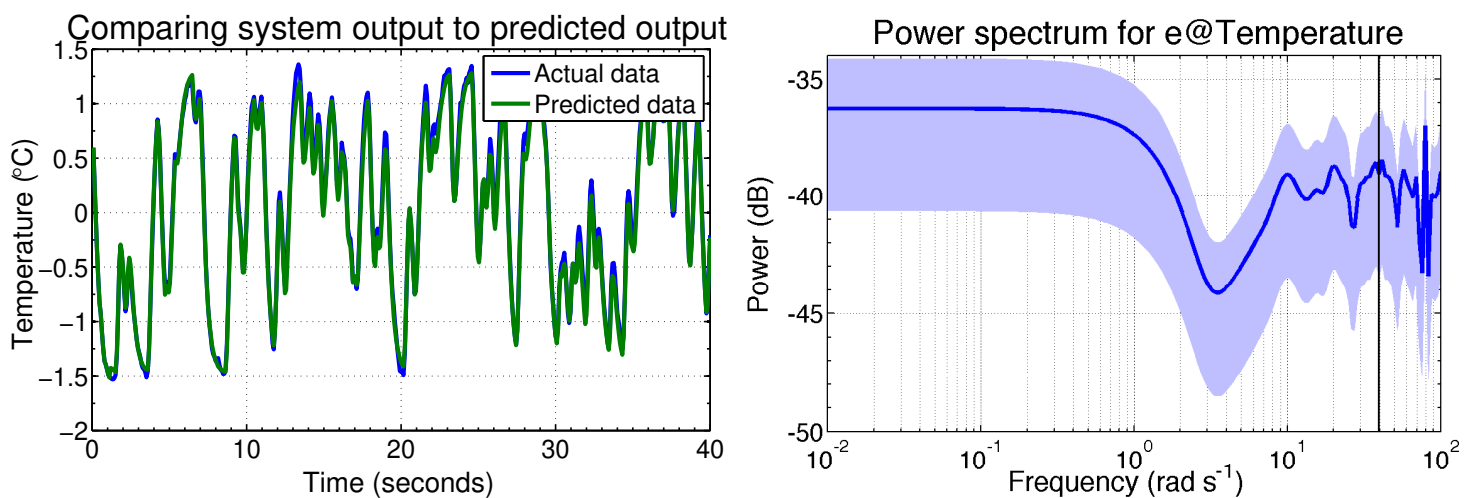
Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + e(t)
A(q) = 1 - 0.9938 (+-0.04529) q^-1 + 0.05448 (+-0.06351) q^-2
      - 0.04001 (+-0.06316) q^-3 + 0.1896 (+-0.05791) q^-4
      - 0.06689 (+-0.0229) q^-5

B(q) = 0.06668 (+-0.001582) q^-3 + 0.05999 (+-0.003462) q^-4
      + 0.01748 (+-0.004437) q^-5 - 0.00555 (+-0.004403) q^-6
      - 0.005821 (+-0.003185) q^-7

Estimated using ARX on data set ze
Loss function 0.0015097 and FPE 0.00157009
Sampling interval: 0.08 sec
Created:          08-Jan-2011 17:10:44
Last modified:   08-Jan-2011 17:10:44

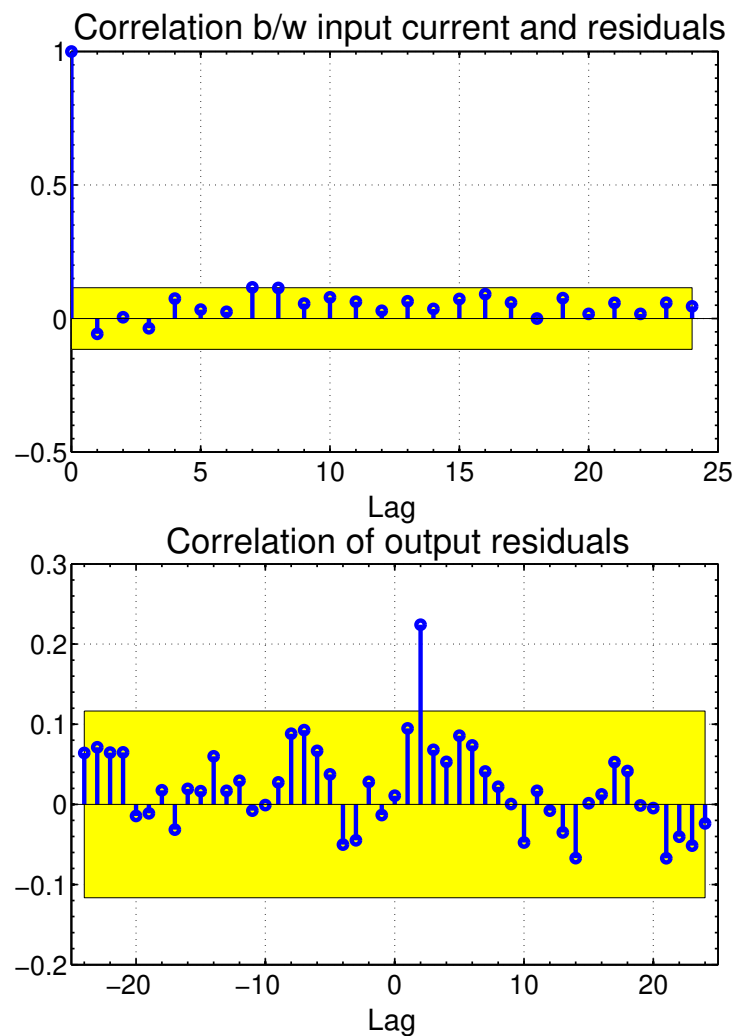
```

- In terms of performance, we care about how well the model predicts and simulates.
 - Plots show prediction and prediction errors in time and frequency domain (the latter with 3σ error bounds):



- We can also do “residual analysis” to see if we’ve squeezed all the information from the data.

- Model provides reasonable fit.
 - No obvious structure to the residual errors, which will be one of our discriminators.
- So, to proceed in this course, we will need to develop tools in:
 - Analysis of linear systems,
 - Stochastic processes and random variables,
 - Discrete model structures and nonlinear optimization,
 - And more. . .



Where from here?

- We first focus our attention on some background topics, which blend well with nonparametric system identification.
- We then look at transfer-function models, and methods to identify the transfer function parameters.
- Our final topic is MIMO state-space models and system identification.
- Will try to blend techniques into (nearly) every lecture, so that theory and practice have good blend.
- Course goal is a practical survey of the most important methods used.

Appendix: Code for example

```

load dryer2; % load example data from sysid toolbox
Ts = 0.08;    % assumed sample period for data

% Remove equilibrium values from primary data and validation data
u2 = u2 - mean(u2); y2 = y2 - mean(y2);
ue = u2(1:length(u2)/2); uv = u2(length(u2)/2+1:end);
ye = y2(1:length(y2)/2); yv = y2(length(y2)/2+1:end);

% Create iddata objects for sysid toolbox to work with
% If you want to see "inside", use "get(ze)" or "get(zv)"
ze = iddata(ye,ue,Ts); zv = iddata(yv,uv,Ts);

% Set some properties of the objects
ze.InputName = 'Current';    ze.InputUnit = 'A';
ze.OutputName = 'Temperature'; ze.OutputUnit = '\circC';
ze.TimeUnit = 'sec';        zv.TimeUnit = 'sec';
zv.InputName = 'Current';    zv.InputUnit = 'A';
zv.OutputName = 'Temperature'; zv.OutputUnit = '\circC';

% Plot input/output data for main and validation data sets
figure(1); clf; plot(ze); figure(2); clf; plot(zv);

% Estimate frequency-response model to get a feel for how many poles/zeros
Ge = spa(ze); figure(3); clf; bode(Ge);

% Estimate step-response model to get a feel for how many input delays
figure(4); clf; step(impulseest(ze),2); xlim([-1 2]);

% Can also do this to estimate how many delays
nk = delayest(ze); % responds with "3"

% Select the most likely structure size for this data (ARX model)
NN = struc(2:5,1:5,nk);
theStruc = selstruc(arxstruc(ze(:,:,1),zv(:,:,1),NN)); % use [5 5 3]

% Do the ID, using this ARX model type
Marx = arx(ze,theStruc);
present(Marx) % display identified model, with uncertainties
marxPoles = roots(Marx.a) % e.g., get some info from the model

figure(5); clf; compare(zv,Marx)

```

```
title('Comparing data to prediction');
legend('location','best');

figure(6); clf; resid(zv,Marx)
h = get(gcf,'Children');
title(h(1),'Correlation of output residuals');
title(h(2),'Correlation b/w input current and residuals');
xlabel(h(1),'Lag'); xlabel(h(2),'Lag');

figure(7); clf;
yp = predict(Marx,zv,10); % how good at predicting 10 samples ahead?
plot(Ts*(1:length(ue)),[zv.OutputData yp.OutputData]);
legend('Actual data','Predicted data');
xlabel(strcat('Time (',zv.TimeUnit,')'));
ylabel(strcat(zv.OutputName,' (',zv.OutputUnit,')'));
title('Comparing system output to predicted output');

figure(8); clf;
err = pe(Marx,zv); % compute prediction error
h = bodeplot(spa(err,[],logspace(-2,2,200))); showConfidence(h,3);
title('Power spectrum for e@Temperature');
```