Sinusoidal-Input Describing Function (SIDF) Analysis Methods

Prof. James H. Taylor
Department of Electrical & Computer Engineering
University of New Brunswick
Fredericton, NB CANADA E3B 5A3
telephone: +506.453.5101
internet: jtaylor@unb.ca

16 July 2002

JUV RESCCE'02 Summer School
Danang, Viet Nam
05-09 August 2002

Topic Outline

- Introduction
- Basic Concepts
- SIDF Calculations
- Qualitative Behavior of SIDFs; Heuristics
- Classical Harmonic Balance (Limit Cycle Conditions)
- Classical SIDF I/O Analysis ("Transfer Functions")
- Examples
- Systems with Multiple Nonlinearities
- Modern Algebraic SIDF Methods
- Examples
- SIDF Methods for Control System Design

"Classic" references:

- D. P. Atherton, *Nonlinear Control Engineering*, Van Nostrand, 1975 (Reprinted as Student Edition, 1982).
- A. Gelb & W. Vander Velde, *Multiple-Input DF's and Nonlinear System Design*, McGraw-Hill, 1968.
- J. E. Gibson, *Nonlinear Automatic Control*, McGraw-Hill, 1963.
- J. H. Taylor, *Describing Functions*, an article in the *Electrical Engineering Encyclopedia*, John Wiley & Sons, Inc., New York, 1999.

Introduction

- Problem to be addressed: Analyzing Periodic Phenomena
  - Nonlinear oscillations (limit cycles)
  - Response to periodic forcing functions
- Considerations:
  - Simulation is often too time-consuming and cumbersome, especially for parametric (trade-off) studies.
  - There are situations in which simulation is almost useless for studying periodic behavior
  - Few other methods handle high-order systems of systems with multiple nonlinearities with ease

Importance of Periodic Effects

- A limit cycle may be **desired**, with a specified frequency and amplitude – can you design the system?

- A limit cycle may be **unwanted but unavoidable** – is it small enough or slow enough to be acceptable?

- An unstable limit cycle is a **stability boundary** – is it large enough?

- A nonlinear system may be driven by **sinusoidal inputs** – how will it respond?

  - SIDF I/O models for different amplitudes $\rightarrow$ diagnosis

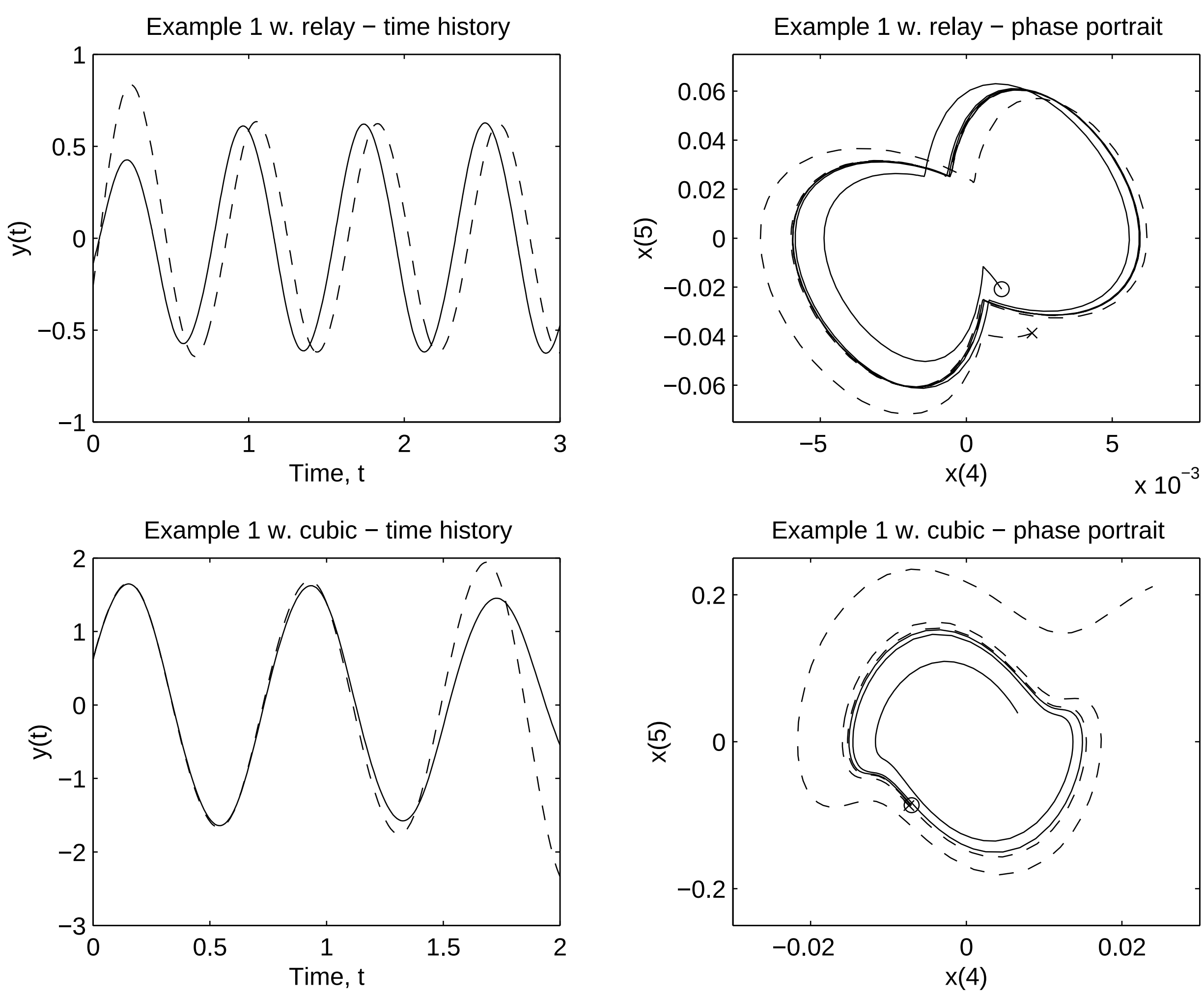
  - SIDF I/O models exhibit interesting phenomena, e.g., "jump resonance" – later

  - SIDF I/O models form an excellent basis for control system design – later

Definition of Limit Cycles

A simple limit cycle is a periodic trajectory in the state space, $x^*(t+T) = x^*(t)\,,\ \forall t$ where T is the period, such that all nearby trajectories

- asymptotically approach $x^*(t)$ (a **stable** limit cycle) **or**
- diverge from $x^*(t)$ (an **unstable** limit cycle)



Limit cycles only occur in nonlinear systems

Definition of Nonlinear System "Frequency Response"

- Input: $u(t) = u_0 + a\cos(\omega t)$
- Output: **may be** periodic:

$$y = \sum_{k=0}^{\infty} b_k \cos(k\omega t + \psi_k)$$

- "Transfer function" for the fundamental component:

$$G(j\omega; u_0, a) = \frac{b_1}{a} \exp(j\psi_1) \qquad (1)$$

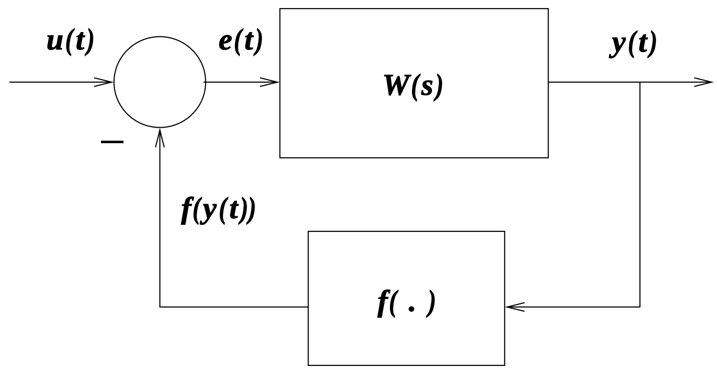
- Operating point ("DC level"): $b_0(u_0, a)$

Note that the "transfer function" and operating point are coupled

Hereafter we will call $G(j\omega; u_0, a)$ an *SIDF Input/Output Model* (SIDF I/O Model)

Basic System Models

Classical Case:



$$
\begin{aligned}
Y(s) \triangleq \mathcal{L}\left(y(t)\right) &= \frac{p(s)}{q(s)}\, E(s) \\
&\triangleq W(s)\, \mathcal{L}\left(e(t)\right) \qquad (2) \\
e(t) &= u(t) - f(y(t))
\end{aligned}
$$

Multi-variable Case:

$$
\begin{aligned}
\dot{x} &= f(x, u(t)) \\
& \qquad\qquad (3) \\
y(t) &= h(x, u(t))
\end{aligned}
$$

For limit cycle analysis: $u(t) = u_0$

For forced response: $u(t) = u_0 + \mathrm{Re}\left[a \exp(j\omega t)\right]$

Basic System Model in MATLAB

- Given: $W(s) = 2/(s^2 + 3s + 7)$ and $f(y) = 4y^3$
- In ODE form: $\ddot{y} + 3\dot{y} + 7y = 2e = 2[u(t) - 4y^3]$
- In state-space form (one realization): $x^T = [\, y \;\; \dot{y} \,]$

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -7x_1 - 3x_2 + 2[\, u(t) - 4y^3 \,] \end{aligned}$$

- In MATLAB:

```
function xdot = basic(t,x)
% Example in controllable canonical form:
% JH Taylor - 9 July 2002

num = 2; den = [ 1 3 7 ];
u = 3.5 * sin(10*t);
xdot(1) = x(2);
xdot(2) = - den(3)*x(1) - den(2)*x(2) + num*(u - 4*x(1)^3);
xdot = xdot(:);
```

- To run a simulation:

```
tspan = [ 0 6*pi/10 ]; % three cycles of sin(10*t)
x0 = [ 1.2 -3.4 ];     % arbitrary initial condition
[t,x] = ode45('basic',tspan,x0); % model is in basic.m
plot(t,x(:,1));        % plot first state only
```

Basic Idea of the Describing Function Method

- Knowledge of signal **form** and **amplitude** is essential in understanding the behavior of a nonlinear system
- Linear system approaches are the most powerful tools we have for analysis
- Replacing nonlinearities with **signal-dependent linear gains** ("quasilinearization") provides the best way to take advantage of linear system approaches to understand the behavior of a nonlinear system
- You will see examples that use the machinery of Nyquist plots, Routh-Hurwitz, root locus, … but the **underlying theory** is entirely different

Classical Definition of a Describing Function

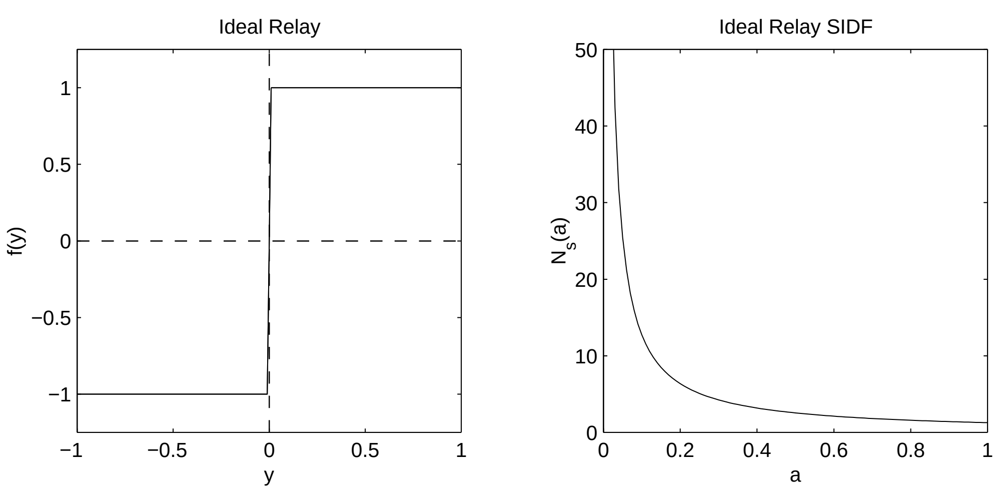
- Given: a specific nonlinearity $f(v)$ and an input signal form, $v(t) = v_0 + \mathrm{Re}\,(a\exp(j\omega t))$
- Find: the quasilinear model $f(v) \cong f_0(v_0, a) + N(v_0, a) \cdot a\exp(j\omega t)$ such that **mean square approximation error** is minimized
- Method 1: $f_0(v_0, a)$ and $N(v_0, a)$ are determined by Fourier analysis (constant plus first harmonic terms)
- Method 2: $f_0(v_0, a)$ and $N(v_0, a)$ are determined by using trigonometric identities (for power-law and product-type nonlinearities

Calculating SIDFs – Piece-Wise-Linear Case

Ideal relay: $f(y) = D \cdot \mathrm{sgn}(y)$ where we assume no DC level, $y(t) = a\cos(\omega t)$

Set up and evaluate the integral for the first Fourier coefficient divided by a as follows:

$$\begin{aligned} N_s(a) &= \frac{1}{\pi\, a}\int_0^{2\pi} f(a\,\cos(x))\cdot\cos(x)\,dx \\ &= \frac{4\,D}{\pi\, a}\int_0^{\pi/2}\cos(x)\,dx \quad \text{(by symmetry)} \\ &= \frac{4\,D}{\pi\, a} \end{aligned} \tag{4}$$



This makes good, intuitive sense.

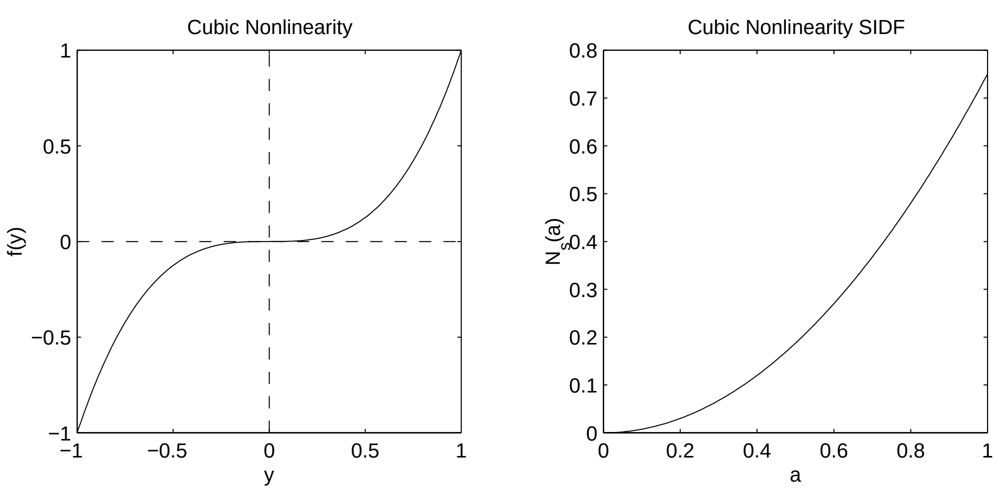
Calculating SIDFs – Power Law Case

Cubic nonlinearity: $f(y) = K\,y^3(t)$; again, assuming $y(t) = a\cos(\omega t)$

Directly write the Fourier expansion using trigonometric identities:

$$\begin{aligned} f(a\cos(\omega t)) &= K[\,a\cos(\omega t)\,]^3 \\ &= K\,a^3\,[\,\frac{3}{4}\cos(\omega t) + \frac{1}{4}\,\cos(3\omega t)\,] \\ &\cong \frac{3\,K\,a^2}{4}\cdot a\cos(\omega t) \end{aligned} \tag{5}$$

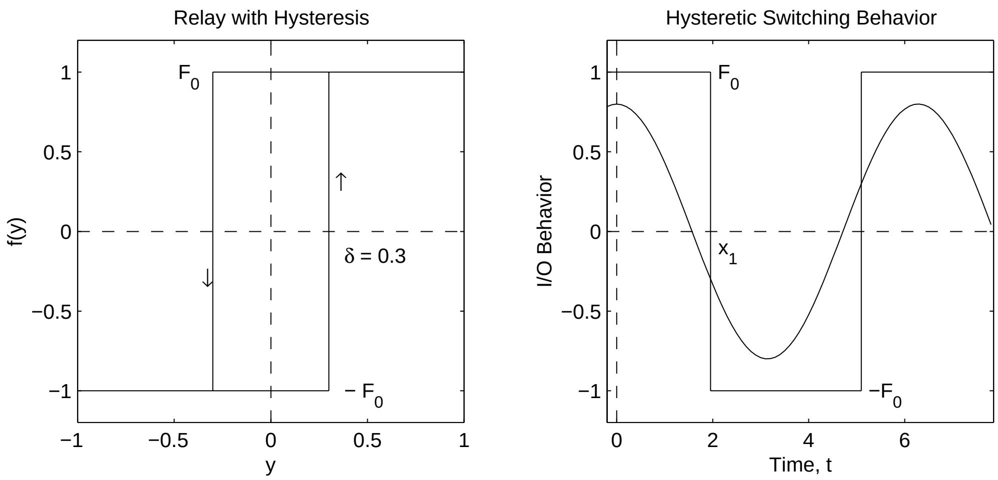
so $N_s(a) = 3\,K\,a^2/4$. Trigonometric identities are a shortcut to formulating and solving Fourier integrals; use for any power-law element.



This also makes good, intuitive sense.

Calculating SIDFs – Multi-valued Case

Setting up the Fourier integrals requires care:



$$N(a) = \frac{1}{\pi\, a}\int_0^{2\pi} f(a\,\cos(x))\cdot\exp(-jx)\,dx$$

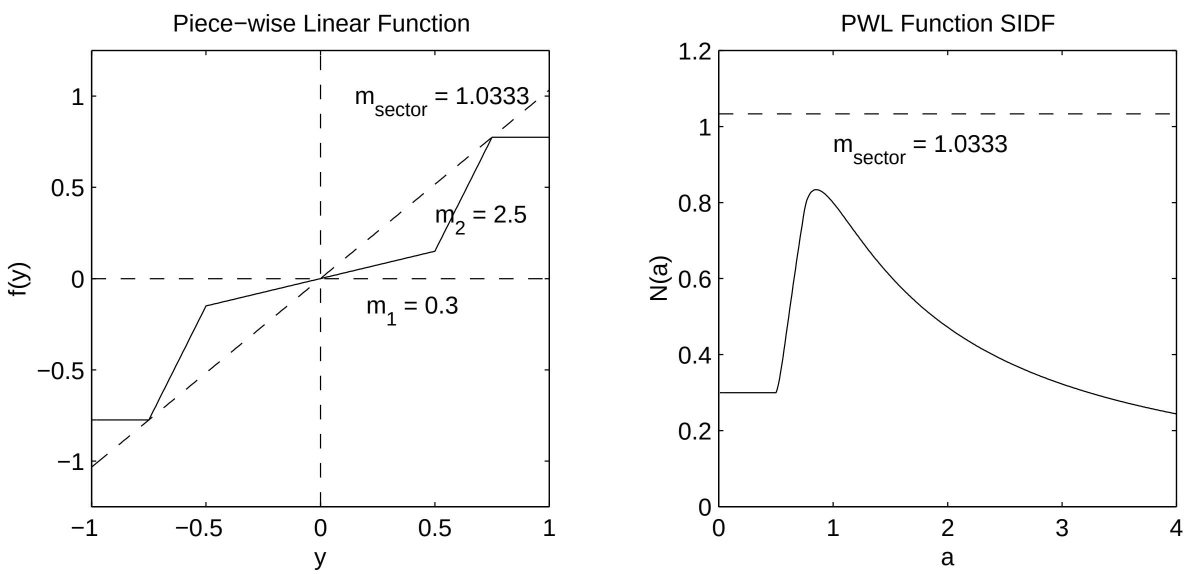
$$= \frac{2\,F_0}{\pi\, a}\left\{\int_0^{x_1}\exp(-jx)dx - \int_{x_1}^{\pi}\exp(-jx)dx\right\} \quad \text{(by symmetry)}$$

where $x_1 = \cos^{-1}(-h/a)$;

$$= \begin{cases} \frac{4\,F_0}{\pi\, a}\left\{\sqrt{1-(\delta/h)^2} - j\,\delta/h\right\} & a > \delta \\ 0 & a \le \delta \end{cases} \qquad (6)$$

Note that $N(a) \triangleq 0$ if $a \le h$ – the relay does not switch $\Rightarrow$ output is not periodic

Qualitative Behavior of SIDFs



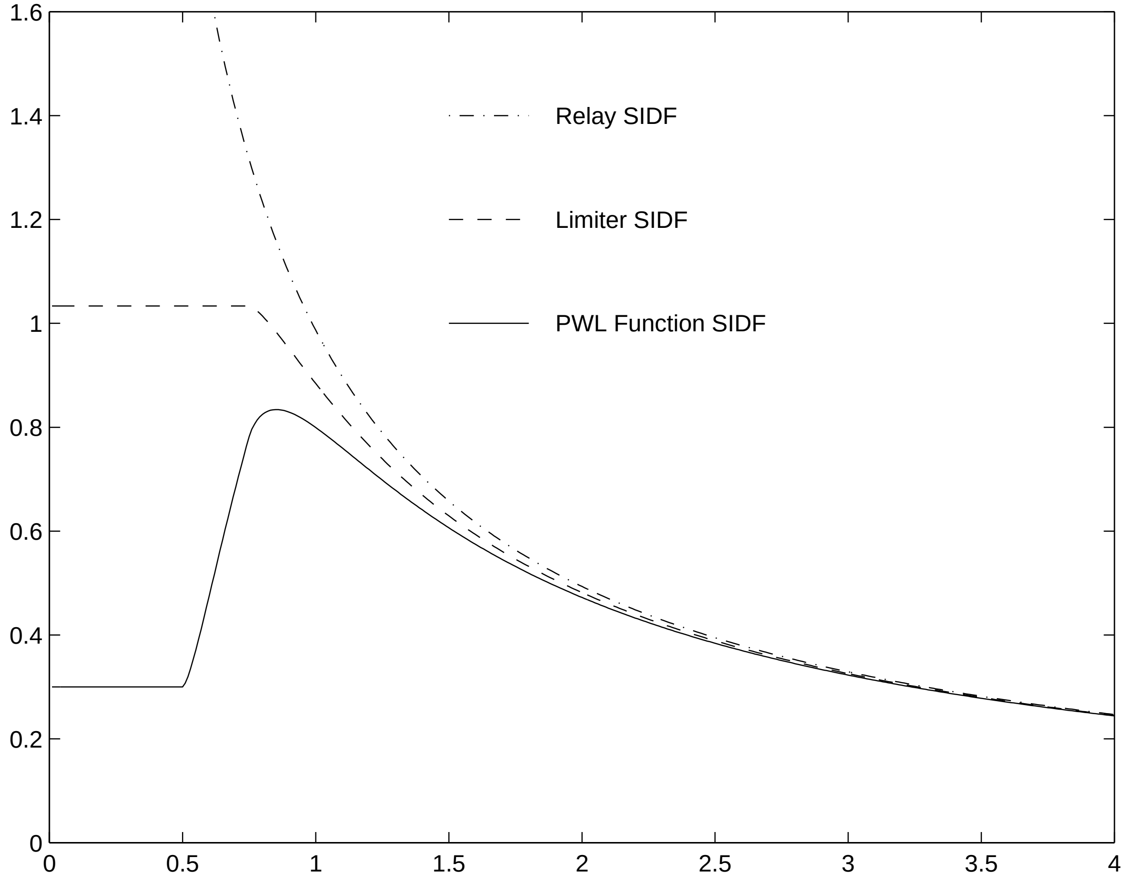
- For small signals $N(a) = [\,df/dv\,]_{v=0} = m_1$ (if the derivative exists)

- The SIDF cannot lie outside the slopes of the enclosing sector

- The SIDF is always continuous, even though the nonlinearity derivative is discontinuous

- The SIDF always approaches the ultimate slope of the non-linearity as $a \to \infty$ (zero for this example)

Qualitative Behavior of SIDFs (Cont'd)



For large signals the "details" near the origin do not make much difference

Calculating SIDFs in MATLAB

- First, define the basic "saturation function" used in calculating SIDFs for piece-wise-linear functions:

$$f_{\text{sat}} = \begin{cases} \text{sign}(x)\,, & |x| \geq 1 \\ 2\,[\,sin^{-1}(x) + x\sqrt{1-x^2}\,]/\pi\,, & |x| < 1 \end{cases} \qquad (7)$$

- The SIDF for a general limiter is $N_{LIM}(a) = m f_{\text{sat}}(\delta/a)$

- The SIDF for the piece-wise-linear example is $N_{PWL}(a) = m_1 f_{\text{sat}}(\delta_1/a) + m_2\,[\,f_{\text{sat}}(\delta_2/a) - f_{\text{sat}}(\delta_1/a)\,]$

- Therefore the previous plots are obtained as follows:

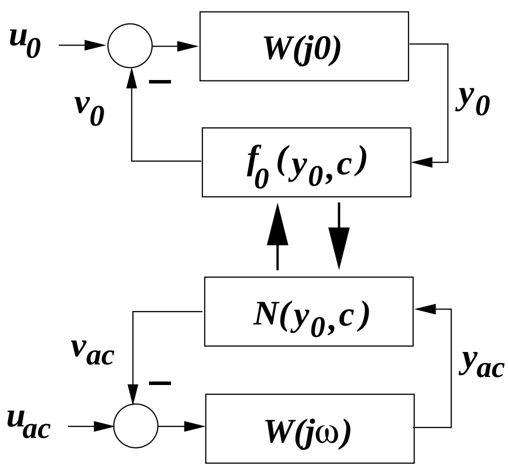
```
D = m1*d1 + m2*(d2 - d1); m_sect = D/d2;
av = 0.01:0.01:4.0;
for i = 1:length(av);
  DFqual(i) = m1*f_sat(d1/av(i)) + m2*(f_sat(d2/av(i))-f_sat(d1/av(i)));
  DFlim(i) = m_sect*f_sat(d2/av(i)); % limiter
  DFrel(i) = 4*D / (pi*av(i)); % relay
end
plot(av,DFqual,av,DFlim,'--',av,DFrel,'-.');
axis([0 4 0 1.6]);
```

where:

```
function f_sat = f_gvdv(x)
% saturation function "f" for calculating SIDFs for PWL functions
% Gelb & Vander Velde, Appendix B, p. 519
% JH Taylor - 18 June 2002
if abs(x) >= 1,
  fdf = sign(x);
else
  fdf = 2*(asin(x) + x*sqrt(1 - x*x))/pi;
end
```

Harmonic Balance – Limit Cycle Conditions

1. Classical Case:



$$u = u_0 + a\cos(\omega t) \triangleq u_0 + u_{ac}$$

$$y = y_0 + \mathrm{Re}\,[\,c \cdot \exp(j\omega t)\,] \triangleq y_0 + y_{ac}$$

$$v = f_0(y_0, c) + \mathrm{Re}\,[\,N(y_0, c) \cdot c \cdot \exp(j\omega t)\,] \triangleq v_0 + v_{ac}$$

DC Harmonic Balance: $y_0 = W(j0)[\,u_0 - f_0(y_0, c)\,]$

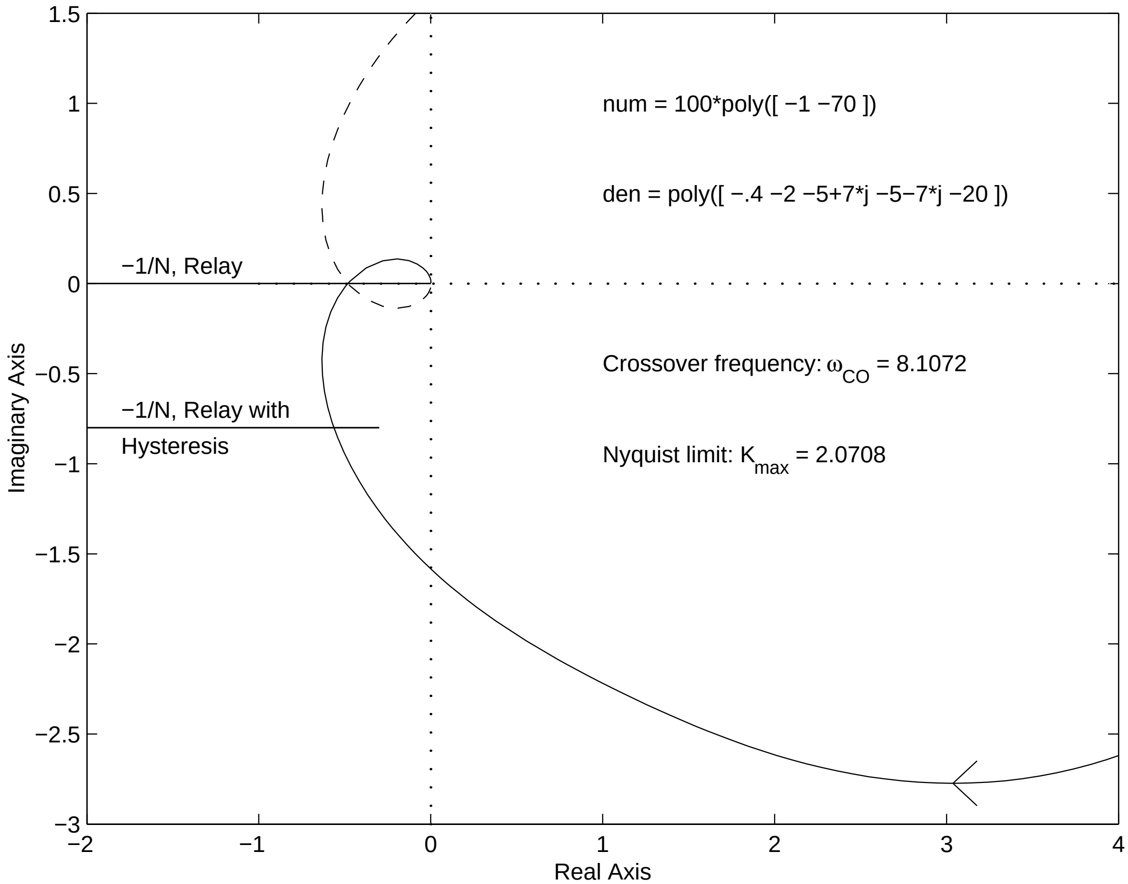
AC Harmonic Balance:

- Limit Cycles: $a = 0$; $W(j\omega) \cdot N(y_0, c) = -1$ must be satisfied for some $\{y_0,\ c,\ \omega\}$ for limit cycle prediction

- Forced Response: $a \neq 0$; $c = \frac{W(j\omega)}{1+N(y_0,c)\cdot W(j\omega)} = -1$; solve for $c(j\omega;\ u_0,\ a)$ to obtain the "transfer function" $G(j\omega;\ u_0,\ a)$

Classical Limit Cycle Analysis

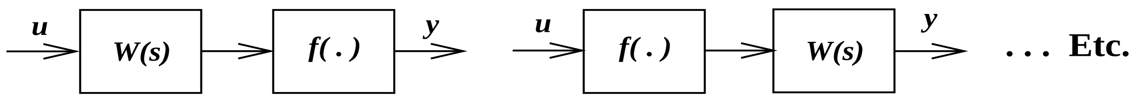
The condition $G(j\omega) \cdot N = -1$ (or $G(j\omega) = -1/N$) is easily investigated on a Nyquist plot:



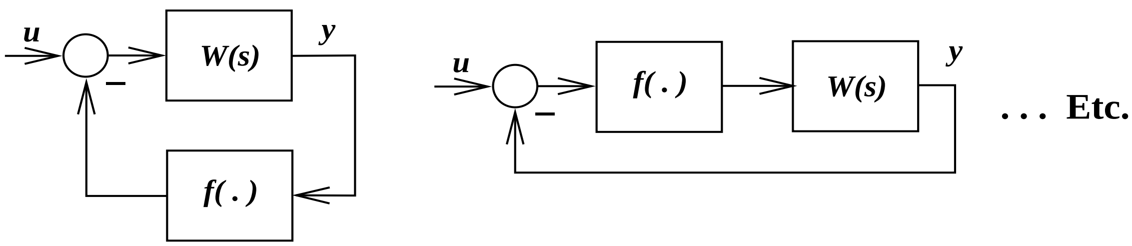
Note: This is **not** the Nyquist test for stability!

Limitations of SIDF Analysis

- Situations when SIDFs are exact:



- Situations when SIDFs are **not** exact:

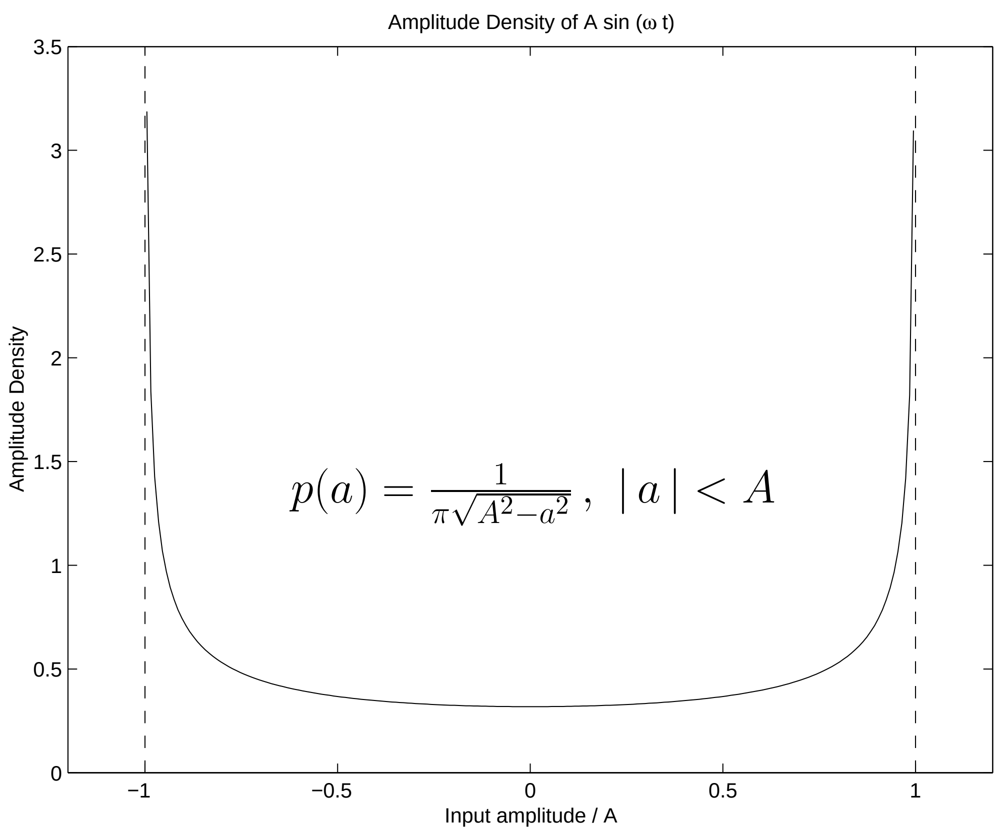


- How to deal with inexact situations:
  - Consider the validity of the "low-pass filter hypothesis" (the nonlinearity input is essentially sinusoidal due to the filtering of higher harmonics by $W(j\omega)$)
  - Consider how well-behaved the system nonlinearity is
  - Look at simulation results, assess the importance of higher harmonics (distortion)

Limitations of SIDF Analysis (Cont'd)

Except for multi-valued nonlinearities (hysteresis, backlash etc.) the DF is not dependent on the assumption of periodicity – only the **amplitude distribution** matters

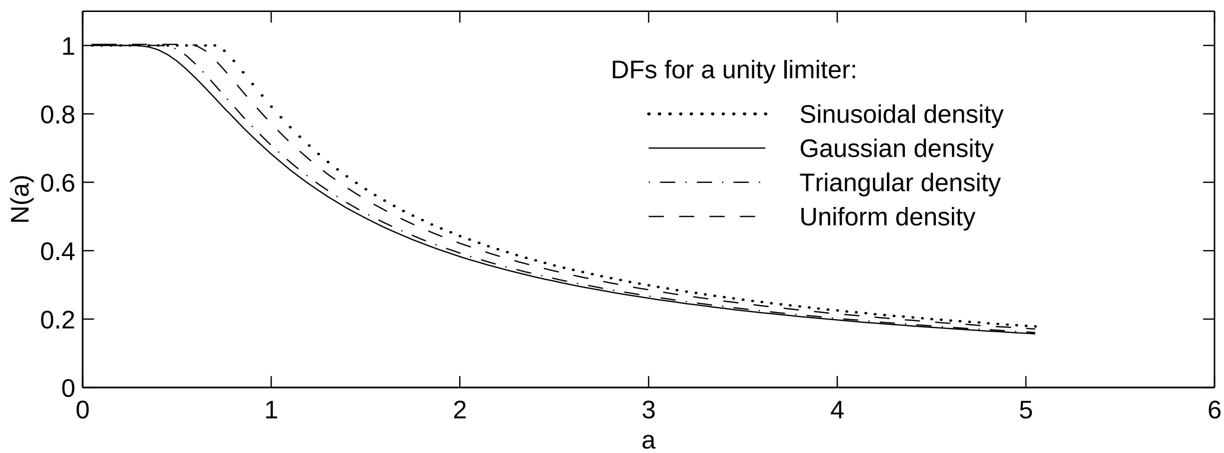
- For a triangular ("saw-tooth") wave the DF is the same as that for a uniformly distributed random variable
- In many control applications the sine-wave distribution,

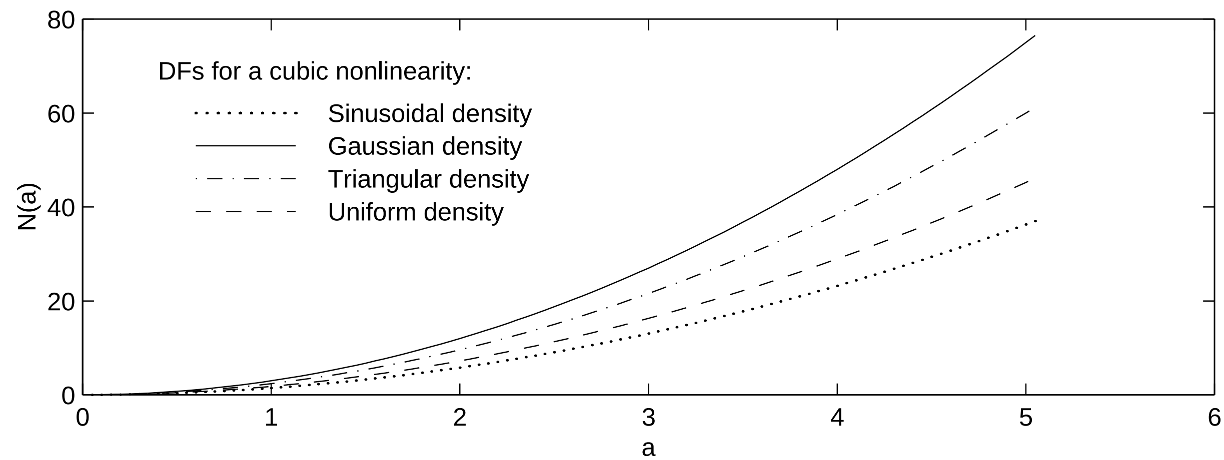


is a good approximation

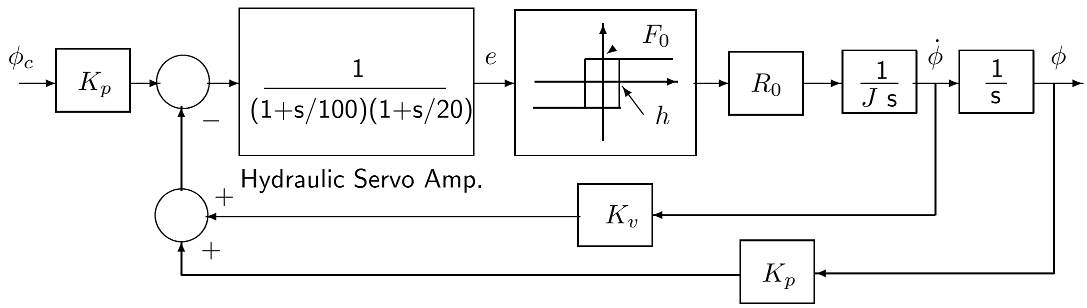
Limitations of SIDF Analysis (Cont'd)

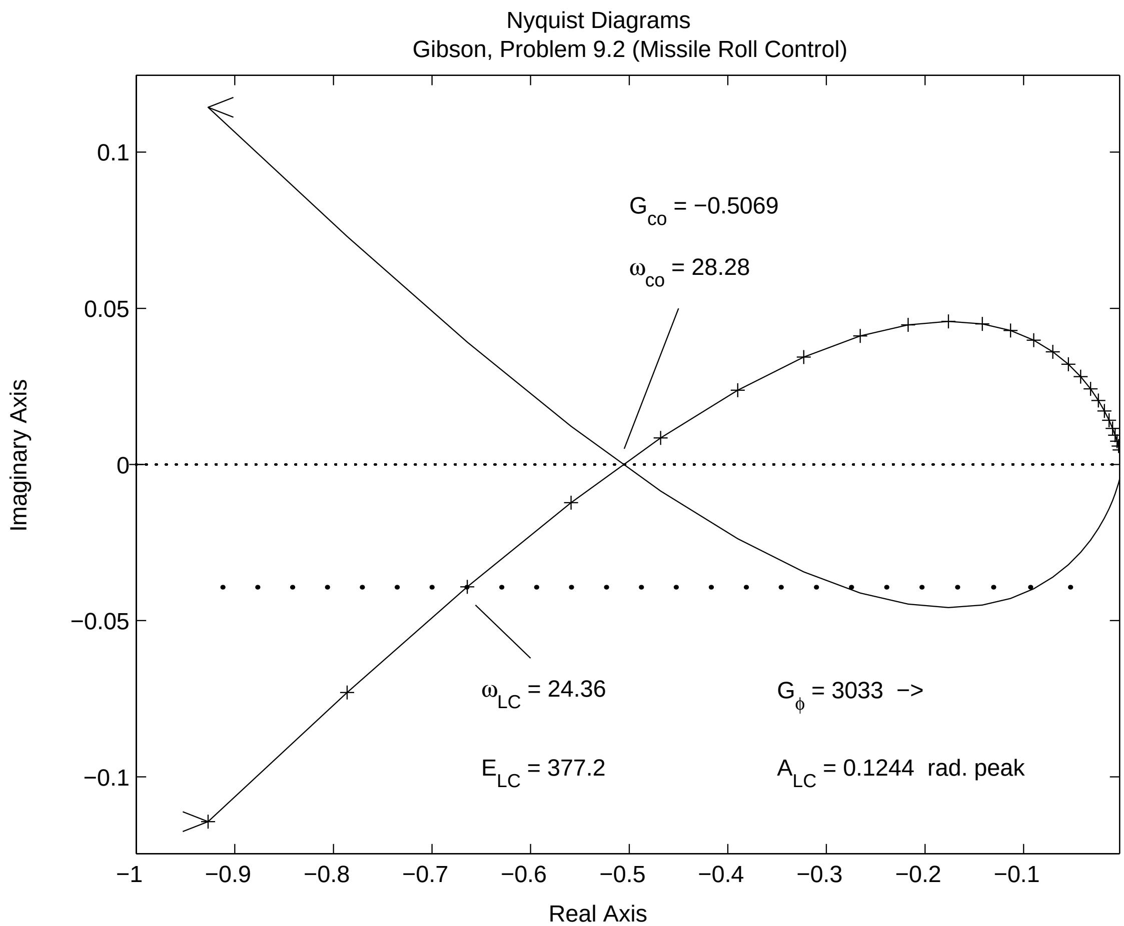
For many nonlinearities the DF is not particularly sensitive to the amplitude distribution:





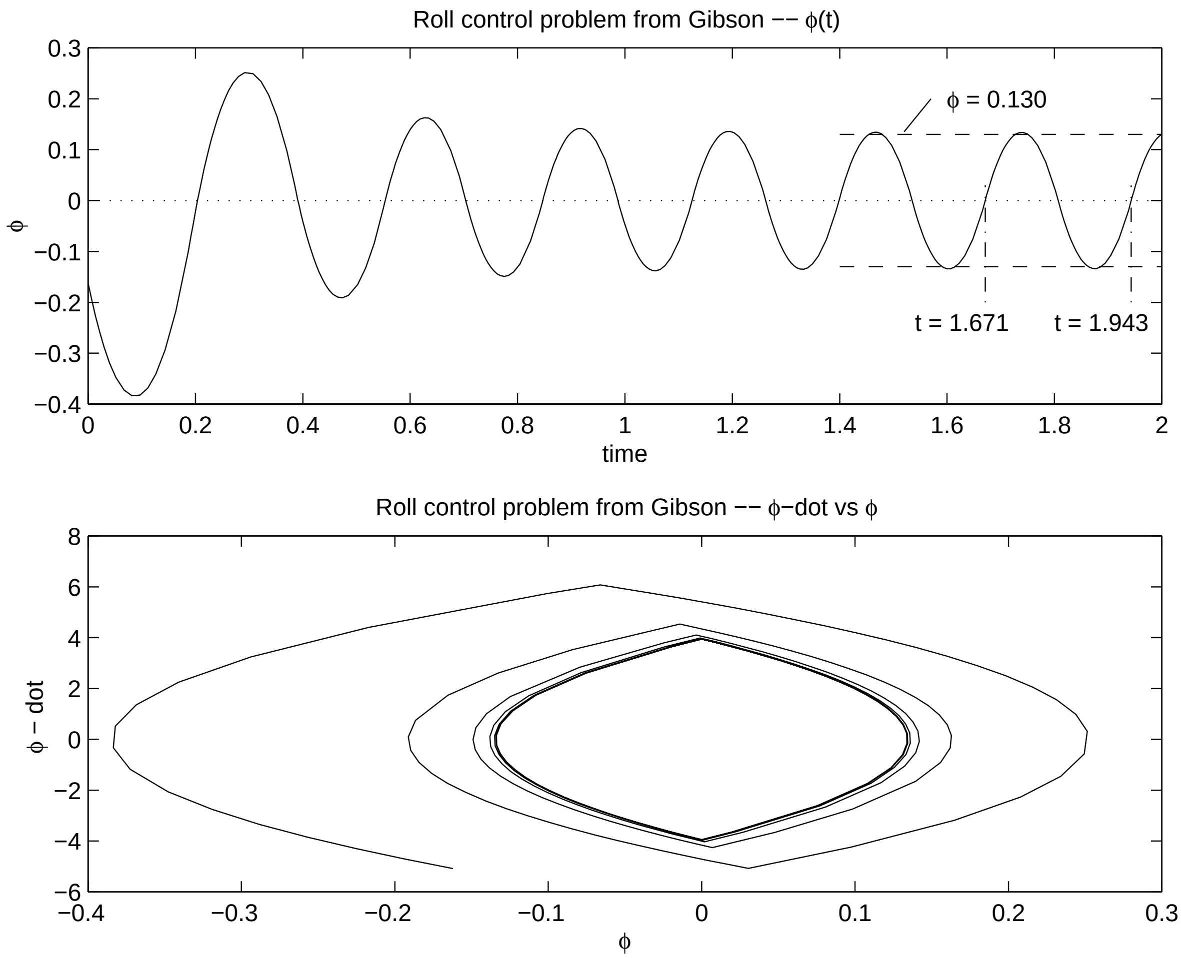
Example: Limit Cycle Analysis, Missile Roll-Control Loop





Limit Cycle Verification

Simulation provides a good verification:



$$T = 0.272 \text{ sec} \rightarrow \omega_{LC} = 23.1 \text{ rad/sec}$$

Harmonic Balance "Transfer Functions"

Two methods for generating the SIDF I/O model $G(j\omega; u_0, a)$:

1. Analytic approach: solve the AC Harmonic Balance equation for $c(j\omega; u_0, a)$, divide by a
   (a) Advantage: you can tell, for example, when solutions do not exist
   (b) Disadvantage: it's difficult to carry out if the nonlinear system is at all complicated
2. Simulation approach: develop a simulation model for the nonlinear dynamic system with a sinusoidal input, simulate to obtain the steady-state response, perform Fourier analysis of the result
   (a) Advantages: No need to assume that the input to each nonlinearity is sinusoidal, the number of system states and nonlinearities is relatively unimportant
   (b) Disadvantages: May be quite time consuming, may be difficult to interpret the results

Harmonic Balance "Transfer Function" – Classical Duffing's Equation

Duffing's Equation: $\ddot{x} + 2\zeta\dot{x} + x + x^3 = a\cos(\omega t)$

This represents, for example, a normalized mass-spring-damper system with a hardening spring; in the standard form $W(s) = 1/(s^2 + 2\zeta s + 1)$, $u(t) = a\cos(\omega t)$ and $f(\cdot) = x^3$

Let b be the amplitude of the fundamental component of x; then quasilinearize Duffing's equation to obtain:

$$b^2\left[(1 + \frac{3}{4}b^2 - \omega^2)^2 + (2\zeta\omega)^2\right] = a^2$$

or, if we let $B = b^2$,

$$B\left[(1 + \frac{3}{4}B - \omega^2)^2 + (2\zeta\omega)^2\right] = a^2$$

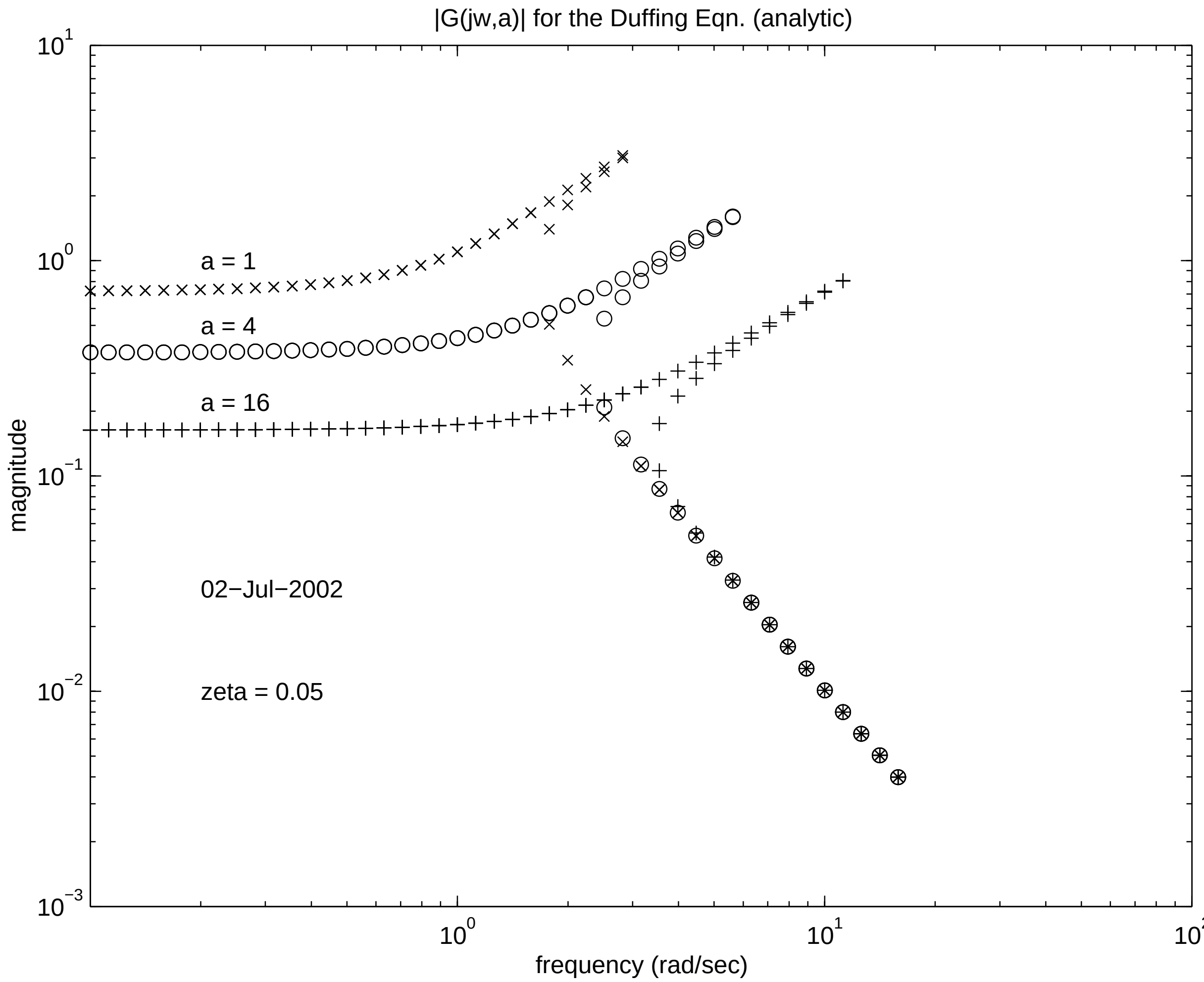
finally,

$$\frac{9}{16}B^3 + \frac{3}{2}(1 - \omega^2)B^2 + \left[(1 - \omega^2)^2 + (\zeta\omega)^2\right]B - a^2 = 0 \quad (8)$$

The last simple polynomial equation may have 1 or 3 real roots, depending on a and ω:

Duffing's Equation "Transfer Function"

The results for several values of a are as follows:



Here we see a **jump resonance** phenomenon

Solving the Duffing Problem in MATLAB

- First, define the polynomial (Eqn. 8 multiplied by 16/3):

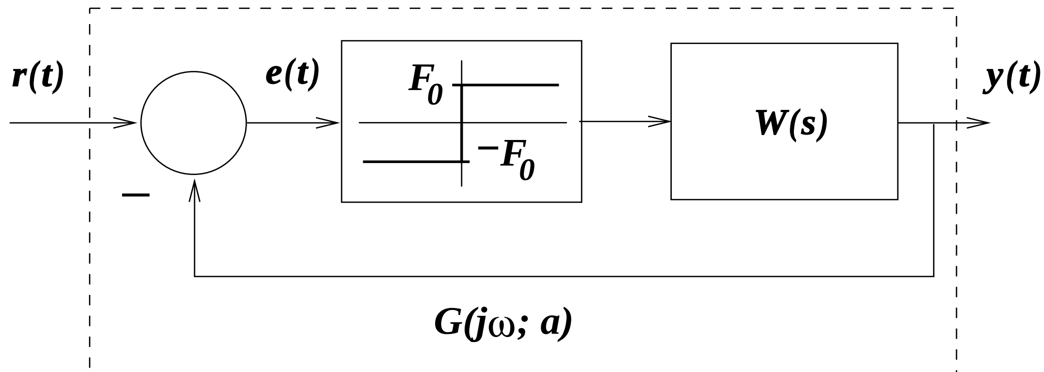
```
function soln = duff_poly(a,w,zeta)
% polynomial to be solved for Duffing's Equation
beta = 1 - w*w; gamma = 2*zeta*w; K = 16/3;
C(1) = 3; C(2) = 8*beta; C(3) = K*(beta^2 + gamma^2);
C(4) = - K*a*a;
soln = sqrt(roots(C)./(a*a));
```

- Now, set up loops for 3 amplitudes and 45 frequencies:

```
zeta = 0.050;
for jj=1:3 %% amplitude loop
  a = 4^(jj-1) %% a = 1, 4, 16
  av(jj) = a;
  for ii=1:45 %% frequency loop
    w = 10^((ii-21)/20)  %% w_min = 0.1, w_max = 10
    wv(ii) = w;
    G = duff_poly(a,w,zeta);
    % discard any complex conjugate
    if imag(G(1)) ~= 0 | imag(G(2)) ~= 0,
      for iii=1:3
        if imag(G(iii)) == 0, RG = G(iii); end
      end
      G(1) = RG; G(2) = RG; G(3) = RG;
    end
    for iii=1:3
        GM(ii,3*jj-2) = G(1); GM(ii,3*jj-1) = G(2); GM(ii,3*jj) = G(3);
    end
  end % frequency loop
end % amplitude loop
%% plotting
loglog(wv,GM(:,1),'x',wv,GM(:,2),'x',wv,GM(:,3),'x', ...
      wv,GM(:,4),'o',wv,GM(:,5),'o',wv,GM(:,6),'o', ...
      wv,GM(:,7),'+',wv,GM(:,8),'+',wv,GM(:,9),'+');
title('|G(jw,a)| for the Duffing Eqn. (analytic)')
xlabel('frequency (rad/sec)');
ylabel('magnitude');
```

Harmonic Balance "Transfer Functions" (Cont'd)

Closed-loop system with relay:



$u(t) = a\cos(\omega t)$ $y(t) = \mathrm{Re}\,[\,c\exp(j\omega t)\,]$

Harmonic Balance Relation:

$$c = (a - c)\cdot\frac{4F_0}{\pi|\,a - c\,|}W(j\omega)$$

- Magnitude part:

$$\begin{aligned} M(j\omega) &\triangleq |\,W(j\omega)\,|; \\ |\,G(j\omega;a)\,| &\triangleq \frac{|\,c\,|}{a} = \frac{4F_0}{\pi|\,a-c\,|}M(j\omega) \end{aligned}$$

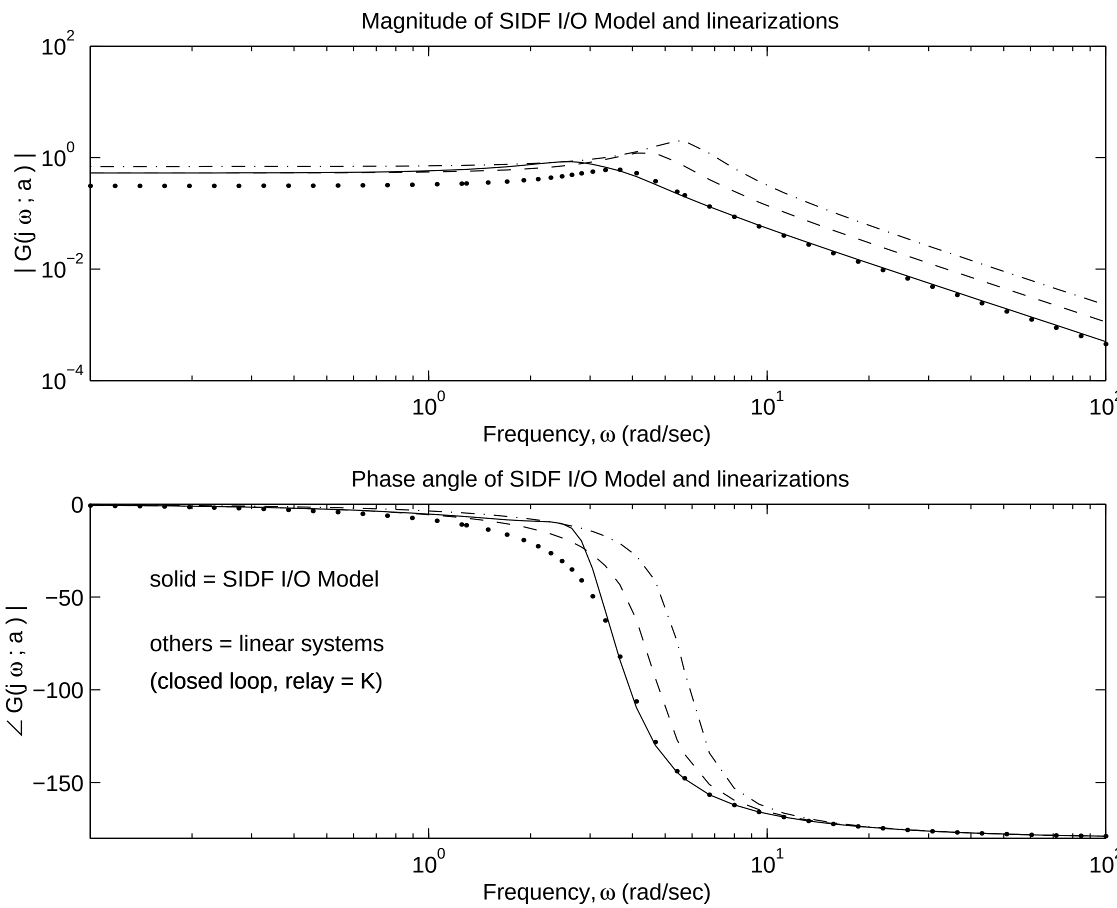
- Phase part:

$$\begin{aligned} \psi &\triangleq \angle W(j\omega)\ ; \\ \angle G(j\omega) &= \psi - \sin^{-1}\left(\frac{4F_0}{\pi a}M(j\omega)\sin(\psi)\right) \end{aligned}$$

Harmonic Balance "Transfer Functions" (Cont'd)

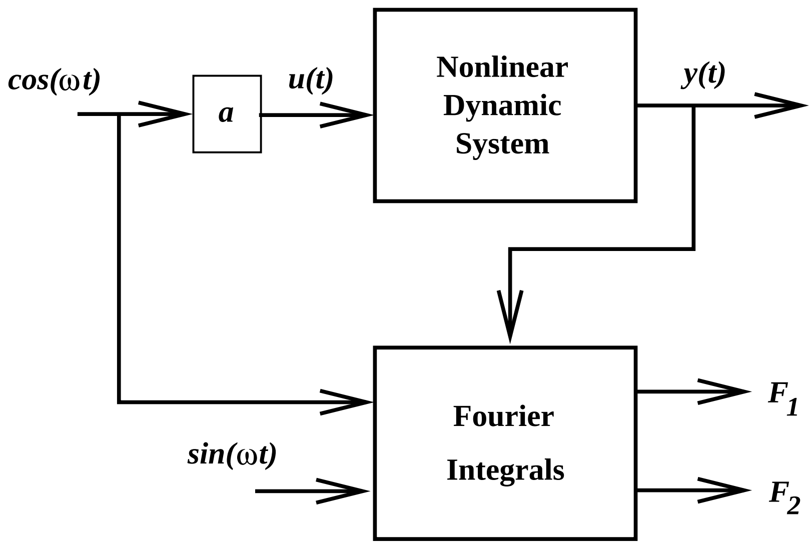
Closed-loop system with relay (cont'd)

- The magnitude relation is quite straightforward (but it appears that the feedback disappears)
- The phase relation can only be met if the input amplitude a is large enough that the argument of $\sin^{-1}$ is less than one at all frequencies
- Example: $W(s) = 45/(s^2 + 2s + 9)$, $F_0 = \pi/2$, $a = 18 \rightarrow$



SIDF I/O Models by Simulation

The most efficient approach is to simulate and perform Fourier analysis simultaneously:



$$F_1^k = \int_{(k-1)T}^{kT} y(t) \cdot \cos(\omega t) dt$$
$$F_2^k = \int_{(k-1)T}^{kT} y(t) \cdot \sin(\omega t) dt$$

from which we obtain:

$$\operatorname{Re} G(j\omega; u_0, a) = \frac{\omega}{\pi a} F_1^k$$
$$\operatorname{Im} G(j\omega; u_0, a) = -\frac{\omega}{\pi a} F_2^k$$

Integrate for k cycles where k is sufficiently large that the magnitude and phase of $G(j\omega; u_0, a)$ have converged to your satisfaction

SIDF I/O Model by Simulation in MATLAB

1. Add the Fourier integral states to your model:

```
function xdot = lim_filt2(t,x)
%  Second-order linear model with limiter; model is
%  augmented with Fourier integrals, to obtain G(jw,a)
%  JH Taylor, 10 July 2002
%
zeta = 0.15; global Ampl Freq
u = Ampl*sin(Freq*t);
xdot(1) = x(2);
xdot(2) = u - x(1) - 2*zeta*x(2);
%% define Y and set up the Fourier integrals:
if abs(x(1)) < 1
   y = x(1);
else
   y = sign(x(1));
end
xdot(3) = y*sin(Freq*t);
xdot(4) = y*cos(Freq*t);
xdot = xdot(:); %% end of model lim_filt2
```

2. Run a simulation to steady state and extract $G(j\omega)$:

```
function [mag,phase] = ggen(Model,MAGTOL,PHASETOL)
%% ggen(model,MAGTOL,PHASETOL) returns the magnitude
%% and phase of an ODE model defined in the file 'Model'.m
%% JH Taylor - University of New Brunswick - 7 July 2002

% Initialize:
global Ampl Freq Xdim;
k = 0; T = 2*pi/Freq; tspan = [ 0 T ]; x0 = zeros(Xdim,1);
[t,x] = ode45(Model,tspan,x0);
[nrows,ncols] = size(x);
```

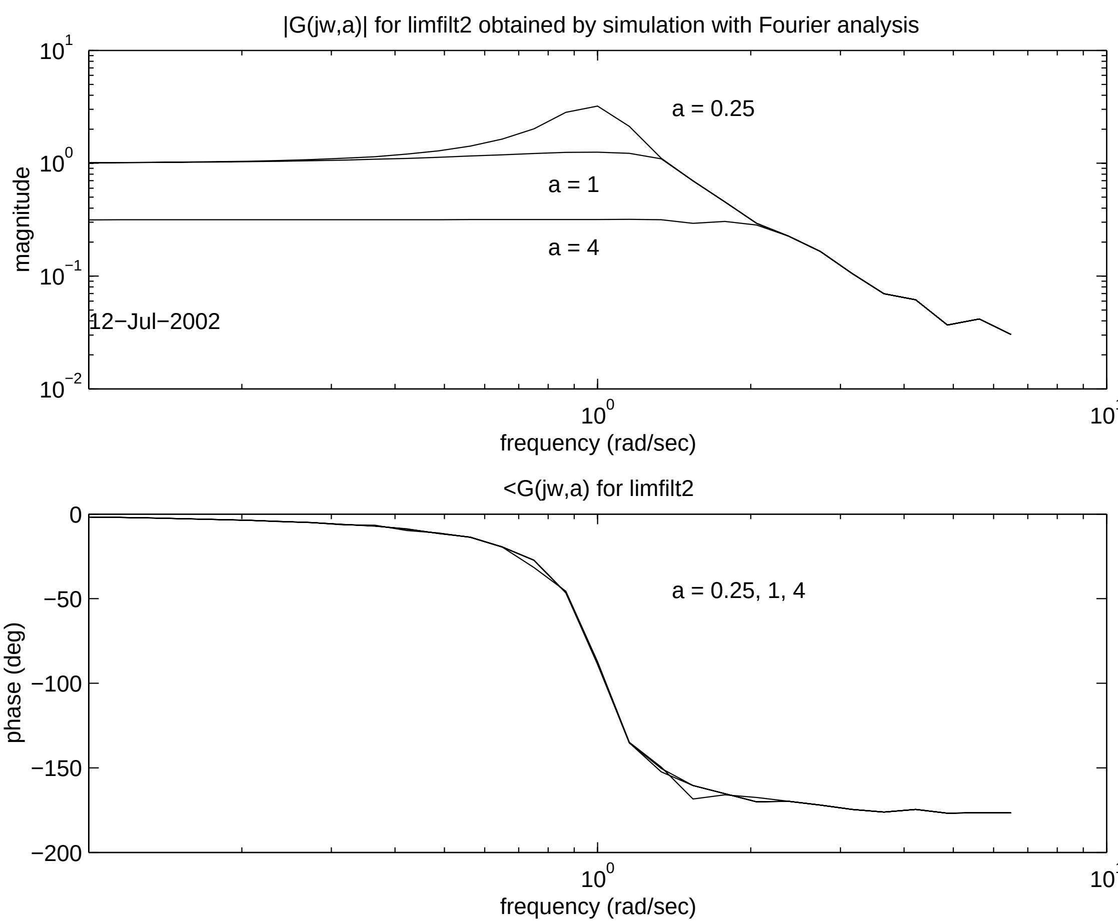
```
xf = x(nrows,:);
mag0 = Freq/(pi*Ampl)*abs(xf(ncols-1)+j*xf(ncols));
phase0 = atan2(xf(ncols),xf(ncols-1));
% Simulate cycle-by-cycle until convergence obtained:
while (k >= 0)
   k = k+1;
   x0 = xf; % initial condition from last cycle
   x0(ncols-1) = 0; % reset the Fourier states
   x0(ncols) = 0;
   [t,x] = ode45(Model,tspan,x0);
   [nrows,ncols] = size(x);
   xf = x(nrows,:);
   mag = Freq/(pi*Ampl)*abs(xf(ncols-1)+j*xf(ncols));
   phase = atan2(xf(ncols),xf(ncols-1));
   magdiff = abs(20*log10(mag/mag0));
   phasediff = (180/pi)*abs(phase-phase0);
   if ((magdiff >= MAGTOL) | (phasediff >= PHASETOL))
      mag0 = mag;
      phase0 = phase;
   else
      k = -1;
   end
end;
```

3. Here is the main executive:

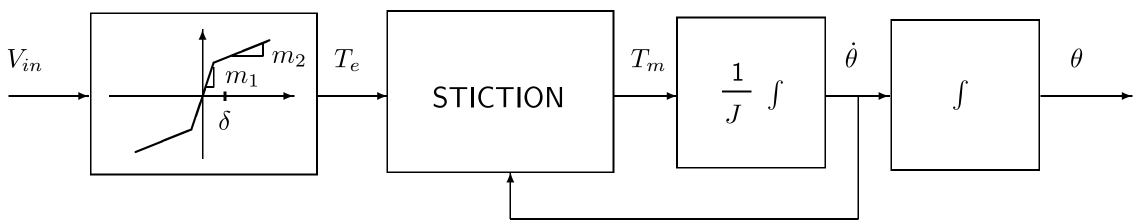
```
%% script for generating a set of G(jw,a) for model "mdl"
%%  JH Taylor  5 July 2002

global Ampl Freq Xdim; dpr = 180/pi; % degrees/radian
mtol = 1;  % magnitude tolerance (dB)
ptol = 5;  % phase tolerance (deg)
mdl = 'lim_filt2' % model = lim_filt2.m (2nd order filter + limiter
Xdim = 4;  % # states, **including Fourier integrals**
%
% amplitude loop
```

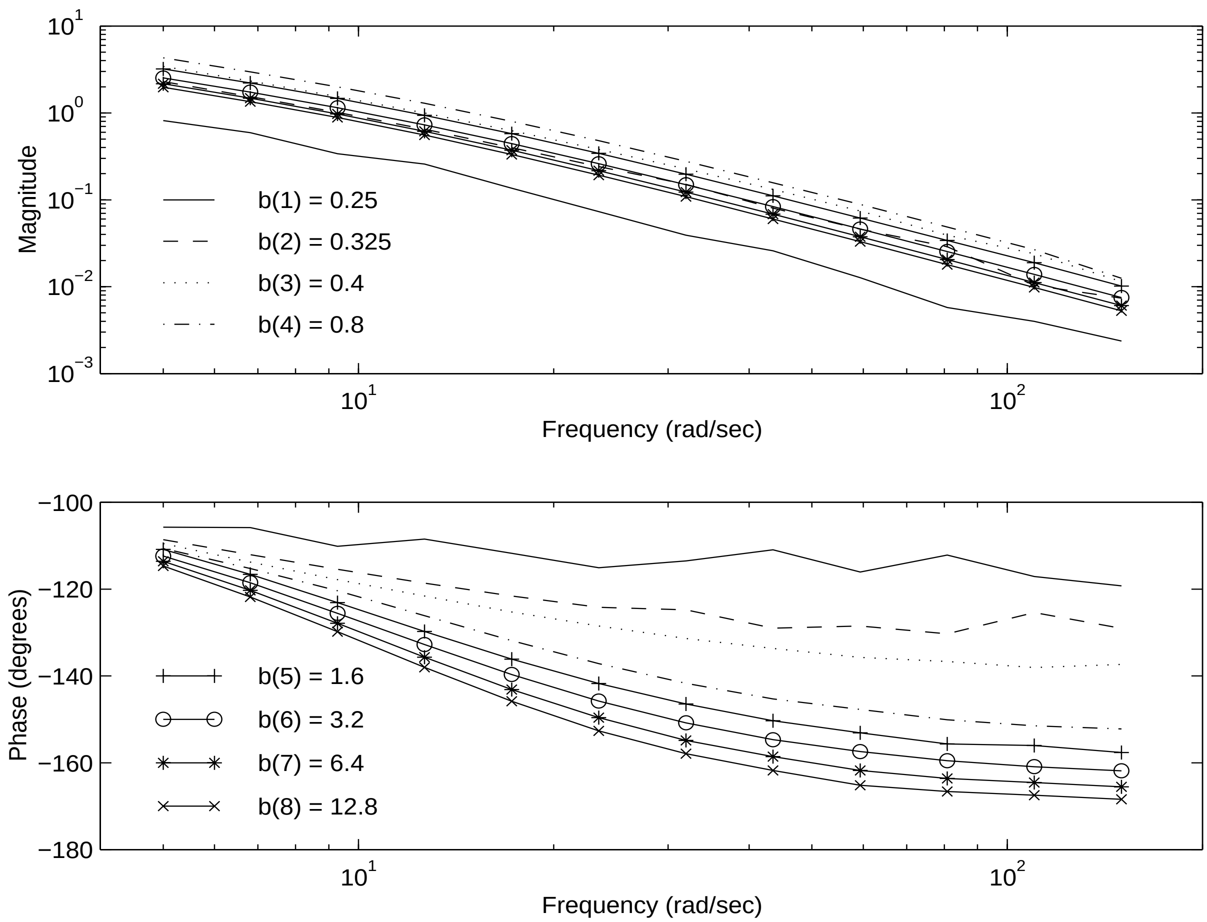
```
for jj=1:3
   Ampl = 4^(jj-2) %% Ampl = .25, 1, 4
   av(jj) = Ampl;
   % frequency loop
   for ii=1:30
      Freq = 10^((ii-17)/16)  %% w_min = 0.1, w_max ~= 6.5
      wv(ii) = Freq;
      [mag(ii,jj),phase(ii,jj)] = ggen(mdl,mtol,ptol);
   end % frequency loop
end % amplitude loop
phase = phase .*dpr;  %% change radians to  degrees
%% routine plotting commands for "Bode plots" omitted
```

4. Finally, here is the main result:



Example: SIDF I/O Model, Electromechanical System, by Simulation





Power of Classical SIDF Approach

When will SIDF limit cycle predictions be "good"?

- When $-1/N(a)$ definitely cuts $G(j\omega)$ (not a near miss or near hit)
- When only one limit cycle is predicted (no "nesting")
- When $G(3j\omega_{LC})$ is far from $-1/N(a)$ where ω_{LC} is the predicted limit cycle frequency

When will SIDF I/O models be "good"?

- When the nonlinear system is not highly resonant
- When higher harmonics are not dominant predicted limit cycle frequency

Modern SIDF Analysis

- Given: $\dot{x} = f(x,u)$ with $u(t) = u_0 + \mathrm{Re}\,[\,a\exp(j\omega t)\,]$
- Assume: $x(t) \stackrel{\sim}{=} x_c + \mathrm{Re}\,[\,b\exp(j\omega t)\,]$
- Quasilinearize the entire state-space system:

$$\begin{aligned} f(x,u) = \;& f_B(u_0,a,x_c,b) \\ &+\; \mathrm{Re}\,[\,A_{DF}(u_0,a,x_c,b)\cdot b\exp(j\omega t)\,] \\ &+\; \mathrm{Re}\,[\,B_{DF}(u_0,a,x_c,b)\cdot a\exp(j\omega t)\,] \end{aligned} \tag{9}$$

- Therefore DC harmonic balance is given by $0 = f_B(u_0,a,x_c,b)$
- ... and AC harmonic balance is given by:
  - Nonlinear Oscillations: $a = 0$, find $b \neq 0$ such that $[\,j\omega_{LC}I - A_{DF}\,]^{-1}b = 0$ ("A_{DF} has pure imaginary eigenvalues and b is the corresponding eigenvector"), i.e., limit cycles are predicted if solutions b, ω_{LC} exist
  - Forced Response: $b = [\,j\omega_{LC}I - A_{DF}\,]^{-1}B_{DF}\cdot a$

SIDFs for Multivariable Functions

- Single-input nonlinearities $f(v)$ are quasilinearized as before
- Multi-variable nonlinearities $f(v_1, v_2, \ldots)$ are more complicated; products and powers of states are easiest to do:

$$\begin{aligned}
\text{Given: } f(x) &= x_1 x_2^2 \\
&= (x_{10} + \mathrm{Re}[\, a_1 \exp(j\omega t)\,])(x_{10} + \mathrm{Re}[\, a_1 \exp(j\omega t)\,])^2 \\
&= \ldots \\
&\cong [\, x_{10}\, x_{20}^2 + \frac{1}{2} x_{10} \,|\, a_2 \,|^2 + x_{20}\, a_1 \bullet a_2 \\
&\quad +[\, x_{20}^2 + \frac{1}{4} |\, a_2 \,|^2 \,] \cdot x_{1,AC} \\
&\quad +[\, 2 x_{10}\, x_{20} + \frac{1}{2} a_1 \bullet a_2 \,] \cdot x_{2,AC} \qquad (10)
\end{aligned}$$

(via trigonometric identities and eliminating higher harmonic terms), where $\bullet$ denotes dot product, $a_1 \bullet a_2 = \mathrm{Re}\, a_1 \cdot \mathrm{Re}\, a_2 + \mathrm{Im}\, a_1 \cdot \mathrm{Im}\, a_2$

Handling multivariable functions represents a **significant generalization** over the classical approach

Multivariable Limit Cycle Example

- Given: $D^3y + D^2y + 2(1 + Ky^2)Dy + 3(1 + y^2)y = u_0$
- We assume: $y \cong y_c + a\sin(\omega t)$, so $Dy \cong a\omega\cos(\omega t)$
- Quasilinearize the system nonlinearities:

$$\begin{aligned} y^3 &\cong y_c\,(y_c^2 + \frac{3}{2}a^2) + 3(y_c^2 + \frac{1}{4}a^2)\cdot a\sin(\omega t) \\ y^2 Dy &\cong (y_c^2 + \frac{1}{4}a^2)\cdot a\omega\cos(\omega t) \end{aligned}$$

- DC harmonic balance: $3y_c\,(1 + y_c^2 + \frac{3}{2}a^2) = u_0$
- "Trick" for AC harmonic balance: the "quasilinear characteristic equation" is

$$\begin{aligned} 0 &= s^3 + s^2 + 2[\,1 + K(y_c^2 + \frac{1}{4}a^2)\,]s + 3[\,1 + 3(y_c^2 + \frac{1}{4}a^2)\,] \\ &\triangleq s^3 + s^2 + \beta s + \alpha \end{aligned}$$

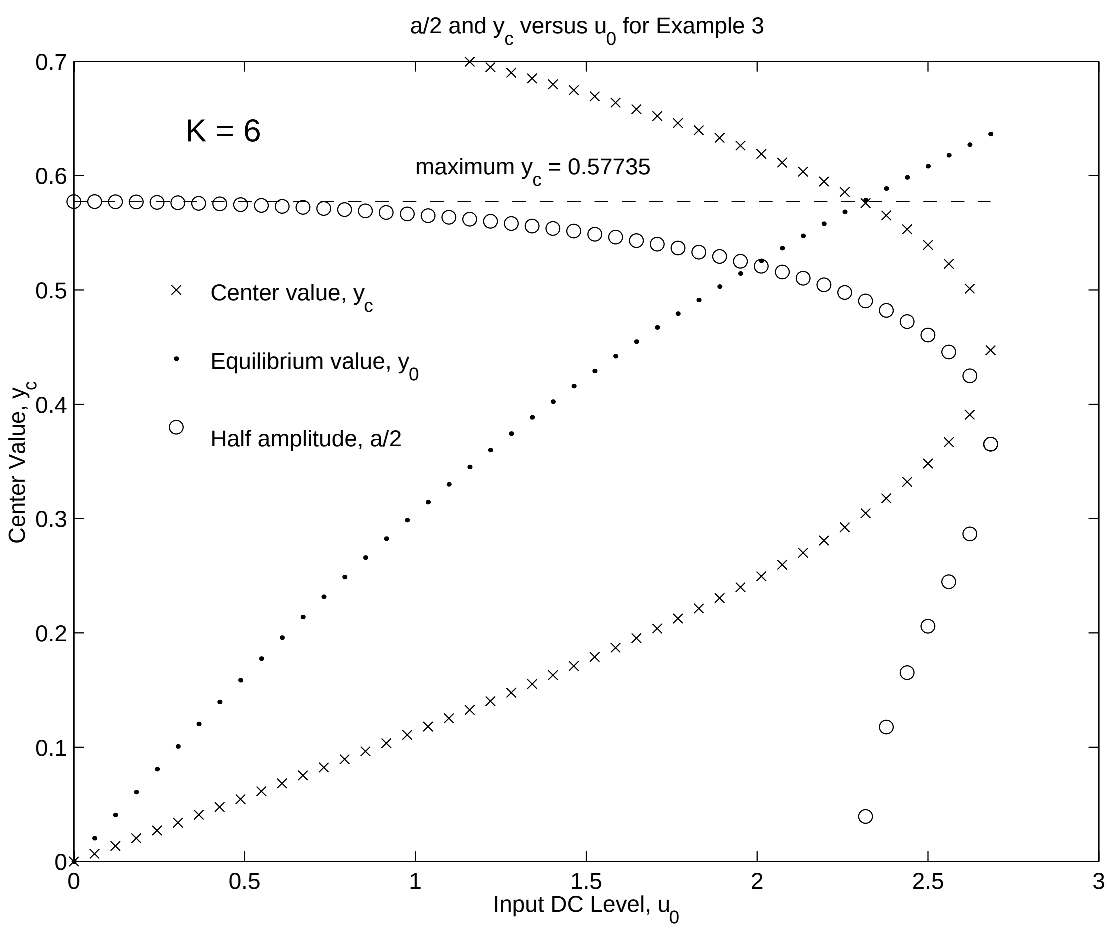
- Limit cycles are predicted if $\beta = \alpha$ (the "quasilinear characteristic equation" has pure imaginary roots) $\rightarrow$ $(2K - 9)\cdot(y_c^2 + \frac{1}{4}a^2) = 1$
- The simultaneous equations can be separated (let $K = 6$):

$$\begin{aligned} u_0 &= 3y_c(3 - 5y_c^2) \\ a &= 2\sqrt{1/3 - y_c^2} \end{aligned}$$

Multivariable Example (Cont'd)

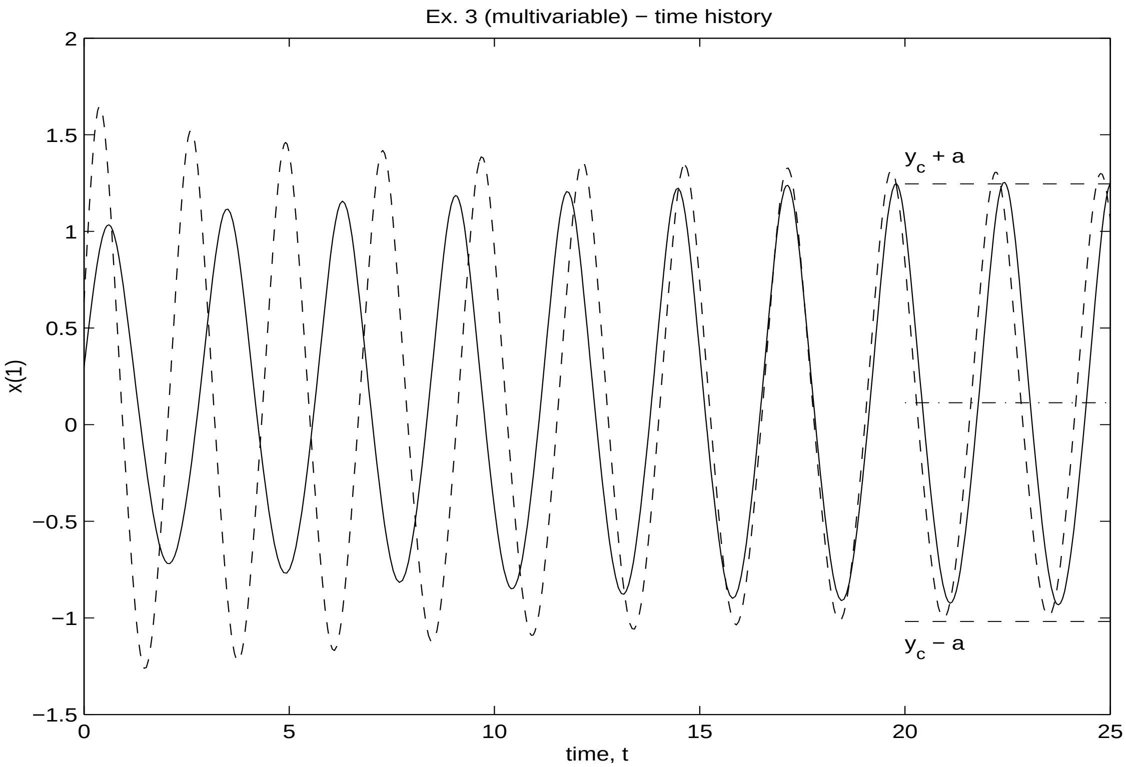
Key analysis results:

1. Limit cycles cannot exist for $K < 9/2$
2. For $K = 6$,
   (a) No limit cycles exist for $|\, u_0 \,| > 6/\sqrt{5} = 2.68$
   (b) Two limit cycles exist for $2.31 < |\, u_0 \,| < 2.68$
   (c) One limit cycle exists for $|\, u_0 \,| < 2.31$
3. Simulations for $u_0 = 1$ provided excellent verification, for $u_0 = 2$ results were good, but for $u_0 = 2.5$ all simulations died out



Multivariable Example (Cont'd)

How good are the SIDF predictions? For $u_0 = 1 \;\rightarrow$



How to solve in MATLAB:

```
max_u0 = 6/sqrt(5); % solutions don't exist for u_0 > 6/sqrt(5)
% input DC level loop
   for ii=1:45
      u0 = max_u0*(ii-1)/44;
      C1 = [ 15 0 -9 u0 ]; % 3 y_c (3 - 5 y_c^2) = u_0
      rts1 = roots(C1);
     for iii=1:3
         YC(ii,iii) = rts1(iii);
         ao2(ii,iii) = sqrt(1/3 - rts1(iii)^2);
      end
   end % DC level loop
plot(uv,YC(:,1),'x',uv,YC(:,2),'x',uv,YC(:,3),'x');
hold on; plot(uv,ao2(:,1),'o',uv,ao2(:,2),'o',uv,ao2(:,3),'o');
```

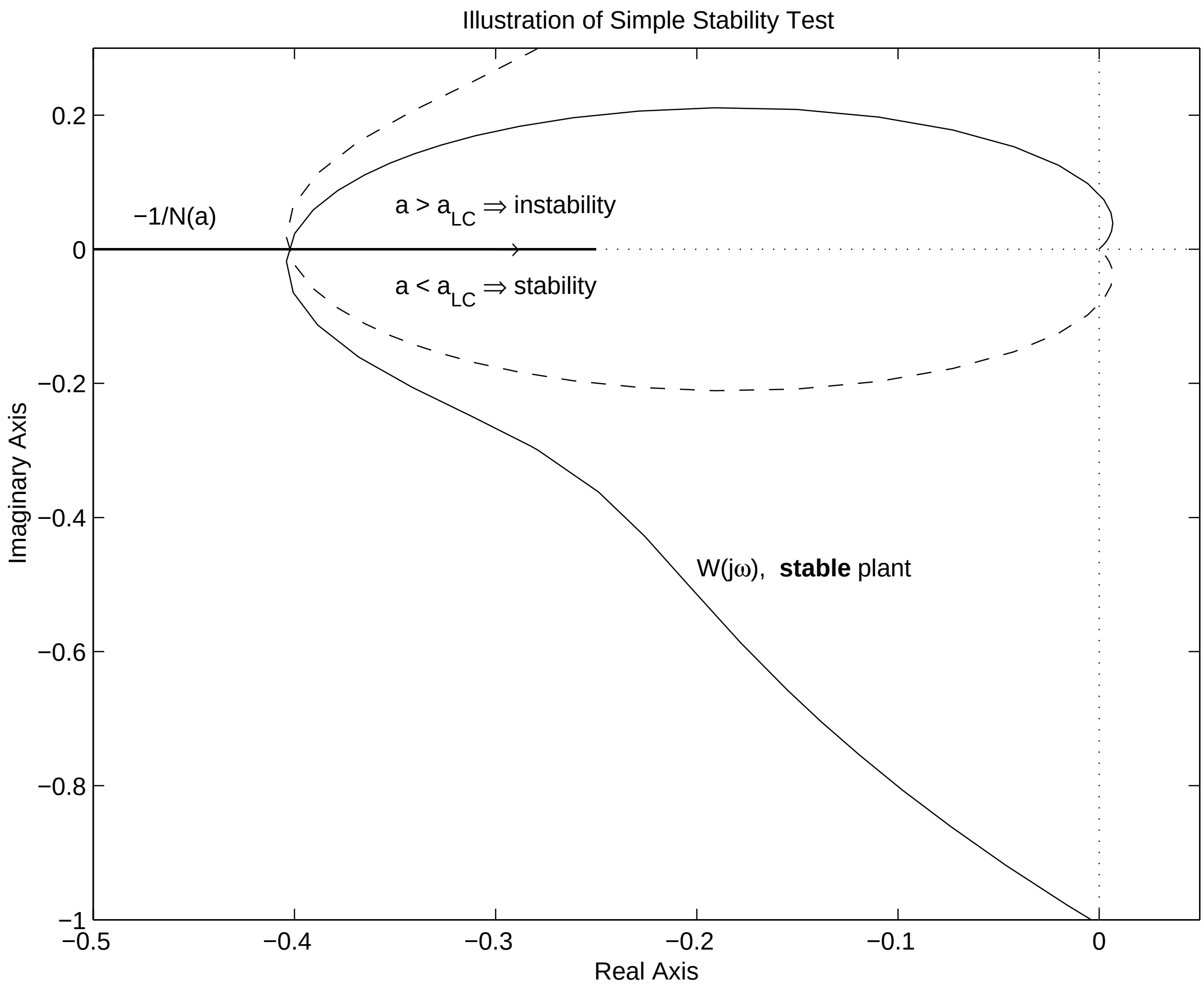
Limit Cycle Stability

The SIDF approach can also yield information on **limit cycle stability** – assume a limit cycle is predicted with amplitude a_{LC}, then:

- The limit cycle is **stable** if $a > a_{LC}$ moves the pure imaginary eigenvalues into the *left* half plane and $a < a_{LC}$ moves the pure imaginary eigenvalues into the *right* half plane
- The limit cycle is **unstable** if the converse is true
- Otherwise the limit cycle is **structurally unstable** (this is an uncommon "borderline" case)
- These conditions are easy to check in cases where there is no bias (DC level), otherwise the coupling between the center value and amplitude (y_c, a) must be taken into account

Limit Cycle Stability (Cont'd)

Here is a limit cycle stability test in the no bias case:



Another test works if there is no bias and there is only one limit cycle predicted: The limit cycle is stable if the enclosed equilibrium is unstable, and conversely.

SIDF Methods: Conclusions

- SIDF techniques are very powerful for studying periodic behavior (nonlinear oscillations, forced response), even in high order and highly nonlinear dynamic system models, even where discontinuous and multi-valued functions exist
- One of the key uses of this approach is exploration:
  - Finding areas in parameter space where limit cycles exist and boundaries where bifurcations occur
  - Determining how a nonlinear system's response to sinusoidal inputs changes as model parameters change
- SIDF analysis and simulation are highly complementary; both have important roles to play