Recording of Model Frequency Responses and Describing Functions in Modelica

Tilman Bünte

German Aerospace Center (DLR)

Institute of Robotics and Mechatronics, Germany

Tilman.Buente@dlr.de

Abstract

An assistant Modelica package is introduced which supports the determination of model frequency responses or describing functions of Modelica models, as the case may be. The result is frequency response data which can be used for further analysis such as stability properties of the system in closed loop control or the derivation of linear time invariant (LTI) model approximations. The paper addresses inter alia proper scheduling of excitation frequency and amplitude, a brief theory of describing functions (harmonic linearization), the Modelica classes implemented in the package, and some application examples.

Keywords: Frequency response; describing function; system identification.
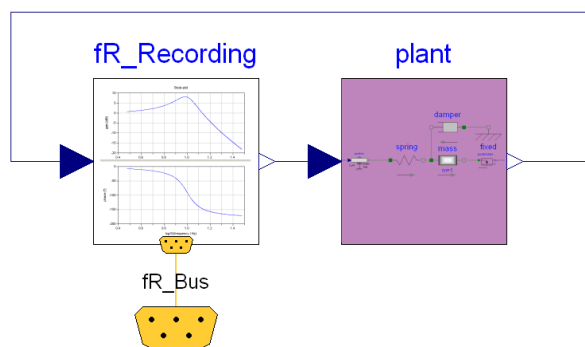
Figure 1: Total Modelica model for frequency response data acquisition of a plant model.

1 Introduction

Frequency responses are applied both for analysis and design of control systems. Multiple useful graphical representations of the frequency response exist like the Bode plot, the frequency response locus (also denoted Nyquist plot for open loop considera-tions), or the Nichols chart. A number of established criteria for evaluating the dynamic properties of linear systems refer to their frequency response. Also *open loop shaping* and *closed loop shaping* control design methods are based on frequency responses. Moreover, some methods for system identification are available which are based on frequency response data.

1.1 Approaches for frequency response determination

Two methods for frequency response determination exist which are fundamentally different. The first method is to develop a white box model based on physical insight. After operations such as lineariza-tion and further model transformation there is a Laplace transfer function which may be evaluated for any value of the frequency ω along the imaginary axis $s = j\omega$. With Modelica models, the first ap-proach is already half way done; the Bode plot of the linearized model can be used for comparison and assessment of the second one.

The second approach is a black box approach apply-ing correlation methods to experimentally acquired in- and output data. The focus of this paper is on the second method whereupon "experiment" is consid-ered the simulation of a Modelica model. The tool provided with the presented package yields fre-quency response data sampled over frequency being well comparable with *FRD models* in Matlab.

1.2 Frequency response vs. model linearization

Linearization of Modelica models may not be suit-able to obtain the aimed result in all cases. Modelica modeling and simulation tools like Dymola or Ma-pleSim provide methods for model linearization. If a linearized model exists then computing and plotting the frequency response is straightforward, for exam-ple by using methods from the LinearSystems library

[1]. Nevertheless, standard linearization may yield unsatisfactory results. Imagine the case of a narrow dead zone characteristic at the system's input. Then the gain of the linearized system is zero. However, it is evident that *zero gain transfer behavior* is a deficient system description. Other examples of nonlinearities which yield problematic linearization results include hysteresis, breakaway force, slackness, and so on. In other cases system specifications directly refer to frequency responses which were determined using sinusoidal inputs with certain amplitude. An example from the domain of vehicle dynamics will be given in section 5. Again, standard linearization is not appropriate.

On the other hand recording the frequency response data in a real world or simulation experiment requires that the regarded system is stable.

1.3 Frequency response of nonlinear systems

Generalizing the frequency response from linear to nonlinear systems there is the notion of a *describing function* which will be explained in section 4. For ease of presentation, in this paper we will understand the term *frequency response* in the expanded sense of the describing function: It is the quotient of two phasors. The denominator of this quotient is the phasor of the sinusoidal input function with preassigned amplitude. The numerator is the phasor of the first harmonic of the steady state system response after all transient portions have decayed. Generally spoken, the frequency response hence is not only a function of the frequency. It may depend also on the input amplitude, cf. section 4.

1.4 Contribution of the Modelica package and this paper

Frequency response related topics having been addressed in the Modelica context so far include the investigation of powertrain oscillations in the SimulationX environment [2] and the modal approach for flexible bodies [3]. Due to the lack of a generic Modelica frequency response data acquisition tool, it occurs so far that alternative environments such as Matlab are used for post processing simulation results of Modelica models [4]. The new package presented in this article is a contribution to close the gap.

The paper is organized as follows. The issue of adequate system stimulation is addressed by section 2. The procedure used for frequency response and describing function determination is explained in 3. For convenience, in section 4 the theory of describing functions is briefly described. Section 5 is dedicated to the Modelica specific implementation and the display of application examples.

2 Plant stimulation

For simplicity, in this paper the term *plant* stands for a Modelica model whose frequency response is searched for. Contrary to real world experiments, in this case we do not need to consider the effect of disturbances or noise. Of course, the approach is explicitly not restricted to controlled systems.

Basically, the experimental frequency response determination of a plant presumes that it is stimulated in an adequate manner. Excitation signals can be distinguished discrete valued vs. continuous, deterministic vs. random, periodic vs. step or impulse, etc. The appropriateness depends on the class of systems to be identified and the method applied for identification. In any case all system modes of interest should be stimulated to be apparent significantly enough in the output signal. For details, interested readers are referred to the technical literature, e.g. [12].

2.1 Quasi-harmonic plant excitation

Here, we confine our considerations to quasi-harmonic excitation signals, i.e. the stimulus is based on a sinusoidal function. Moreover, both the amplitude and the frequency may depend on time. Here, we follow up the concept of *sinus sweep* or *chirp* signals. The function

$$u(t) = A(t, f(t)) \cdot \sin\left(2\pi \cdot F(t)\right) \qquad (1)$$

is used as plant input where

$$F(t) = \int_0^t f(\tau)\, \mathrm{d}\tau \qquad (2)$$

is the integral of the instantaneous frequency *f*. Incidentally, $F(t)$ indicates the number of elapsed periods of the sine function and is therefore called *period function* in the sequel. The amplitude *A* may depend on time and/or frequency.

In the simplest case, the frequency is constant:

$$f_{\mathrm{const}}(t) = f_{\mathrm{Start}}$$

$$F_{\mathrm{const}}(t) = f_{\mathrm{Start}} \cdot t \qquad (3)$$

However, in this case the stimulus consists of only one single frequency. A frequency response determination covering a frequency range thus needs several separate experiments each with its own frequency value along a sufficiently fine grid. Each of the ex-

periments should take long enough such that the plant gets steady state before any serviceable response data can be collected.

More generally, with (1), (2) it is possible to stimulate the plant continuously over a range [f_{Start}, f_{End}] of frequencies $f(t)$, i.e.

$$f_{\text{Start}} \leq f(t) \leq f_{\text{End}} \tag{4}$$

each with adjusted amplitude in one single experiment. However, the amplitude and frequency should be varying so slowly, that the plant can be considered in a steady state oscillation at any time. In reality this assumption cannot be assured perfectly, otherwise the total experiment would take an infinite duration. Rather, a scheduling for amplitude and frequency is searched for such that accuracy of the result is traded off against efficiency of the experiment. We start from the conception that it is most efficient to change the frequency and/or amplitude just as much as is necessary such that the transient effects remain negligible. At the same time it should be kept in mind that after plant excitation the plant response will be recorded and processed (cf. section 4) for frequency response data acquisition. Ideally for efficiency, each period contributes serviceable i.e. non-redundant data.

For illustrating the frequency scheduling problem lets assume that the sinusoidal excitation signal sweeps the interval from $f_{\text{Start}} = 0.1$ Hz to $f_{\text{End}} = 10$ Hz with a total number of $n = 10$ periods. Commonly, for frequency sweeps either linear

$$f_{\text{linear}}(t) = f_{\text{Start}} + k_l \cdot t$$

$$F_{\text{linear}}(t) = f_{\text{Start}} \cdot t + \frac{k_l}{2} t^2 \tag{5}$$

or exponential

$$f_{\text{exponential}}(t) = f_{\text{Start}} \cdot k_e{}^t$$

$$F_{\text{exponential}}(t) = \frac{f_{\text{Start}}}{\ln(k_e)} \cdot k_e{}^t \tag{6}$$

frequency functions are used. Note that the parameters k_l, k_e can be chosen each such that f_{End} is reached after n periods by solving $f(t_{\text{End}}) = f_{\text{End}}$ and $F(t_{\text{End}}) = n$.

However, the plot of the period function over the logarithm of the frequency as shown in Figure 2 reveals, that in both cases, i.e. blue line for the linear case and red line for exponential the number of periods at upper frequencies is disproportionately high. In fact, too few periods are spent at low frequencies. As a result the excitation signal is warped during the

first couple of periods and cannot be considered *sinusoidal* (cf. Figure 3).
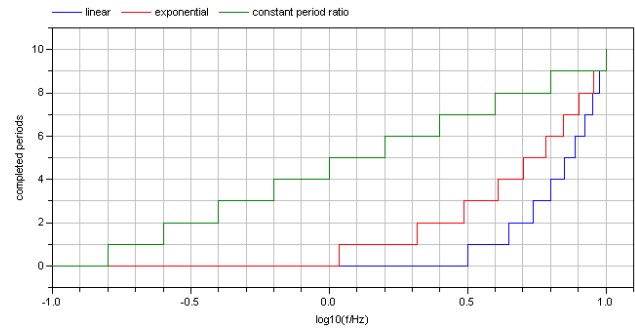

Figure 2: Plot of completed periods over logarithmic frequency.
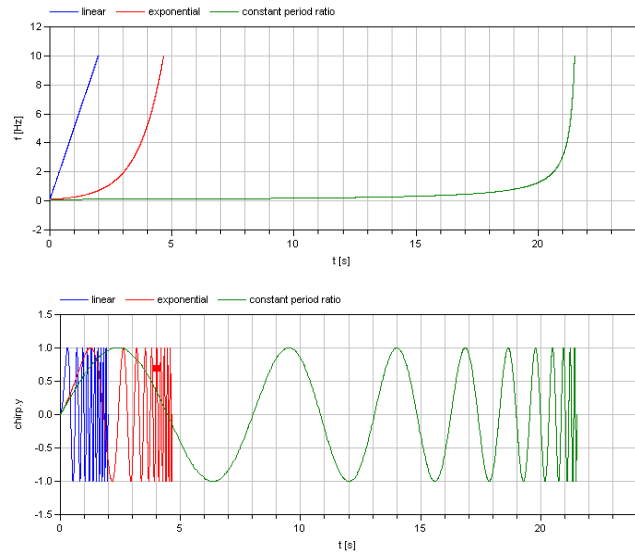

Figure 3: Comparison of sine sweep signals (lower) using different frequency functions of time (upper plot).

On the other hand, the Bode diagram as the most commonly adopted representation of the frequency response uses logarithmic frequency scaling. This suggests using a different frequency progression such that the periods are equidistantly distributed over the logarithmic frequency, corresponding to the green line in Figure 2. A frequency progression function can be derived which exhibits the just formulated property:

$$f_{\text{constant period ratio}}(t) = \frac{f_{\text{Start}}}{1 - f_{\text{Start}} \cdot \ln(k_c) \cdot t}$$

$$F_{\text{constant period ratio}}(t) = -\frac{\ln(1 - f_{\text{Start}} \cdot \ln(k_c) \cdot t)}{\ln(k_c)} \tag{7}$$

For proof let t_k be an arbitrary instant of time and t_{k+1} denote the time one period later then the correlation

$$F(t_{k+1}) = 1 + F(t_k) \qquad (8)$$

holds by definition (1). Solving (8) for t_{k+1} yields

$$t_{k+1} = \frac{1}{k_c}\left(\frac{k_c - 1}{f_{\text{Start}} \cdot \ln(k_c)} + 1\right). \qquad (9)$$

Evaluation reveals that

$$\frac{f(t_{k+1})}{f(t_k)} = k_c \qquad (10)$$

i.e. the frequency ratio over any period of $u(t)$ is constant independent of time, q.e.d. The frequency in fact increases exponentially with the *period function* rather than with time and each period does contribute a sample of the frequency response along an equidistant grid of the logarithmic frequency.

Figure 3 compares the corresponding time signals of the three variants of frequency progression during an experiment. In all simulations the total number of periods is *nPeriods* = 10 and the frequency sweeps from 0.1 Hz to 10 Hz, cf. Figure 2. The result reveals that with the frequency progression defined by (7) the comparatively high expense at low frequencies implicates far better sinusoidal shape but longer total simulation time.

2.2 Scheduling of excitation amplitudes

Using (1) as plant excitation signal, not only the frequency can change with time. Also the amplitude can be made varying according to one's needs as for example to reduce high output amplitudes at poorly damped system modes and thus to prevent damage to real world systems if applicable. Moreover, amplitude scheduling will be used later in the context of describing functions.

3 Frequency response calculation algorithm

As indicated in the introduction, the frequency response is understood as a *complex-valued gain* which is defined as the ratio of the complex-valued phasor of the output signal's first harmonic

$$\widetilde{y}_g(\omega \cdot t, \omega) = \hat{y}_g(\omega) \cdot \mathrm{e}^{\mathrm{j}\left(\omega \cdot t + \varphi_{y_g}(\omega)\right)} \qquad (11)$$

over the complex-valued phasor of the sinusoidal input

$$\widetilde{u}(\omega \cdot t) = A \cdot \mathrm{e}^{\mathrm{j} \cdot \omega \cdot t} \qquad (12)$$

that is

$$N(\omega, A) = \frac{\widetilde{y}_g}{\widetilde{u}} = \frac{\hat{y}_g(\omega)}{A} \mathrm{e}^{\mathrm{j} \cdot \varphi_{y_g}(\omega)} . \qquad (13)$$

The complex-valued gain $N(\omega, A)$ i.e. the frequency response data is tabled over sampled values of frequency and/or excitation amplitude. It contains the information about gain and phase of the plant's transfer behavior and may be displayed in a Bode diagram. Discrete Fourier transformation of input and output signals is used for its calculation. In detail, the algorithm used in the Modelica package at hand executes the following steps:

1. The plant's input is stimulated with a sinus sweep according to (1). Along the way the frequency and/or amplitude should be changing so slowly that the plant can approximately be assumed steady state all the time. Frequency progression according to one of the options (3), (5), (6), or (7) is assumed.

2. By definition periods begin when $F(t)$ takes on integer values.

3. An integer parameter *nSamples* defines how many samples of the plant response are taken per period. Triggered by corresponding events when $(F(t) \cdot nSamples - \lambda)$ becomes integer then plant response data is sampled and collected over the current period. (The Real parameter λ can be set arbitrarily in the interval $0 \leq \lambda < 1$).
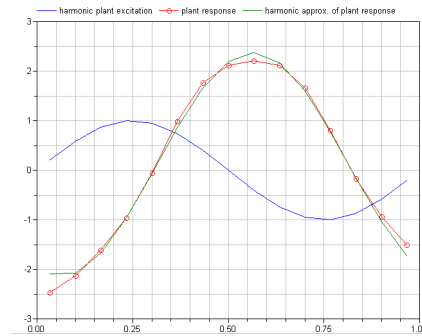

Figure 4: Exemplary signals of the elapsed period: Input signal (blue), plant response (red), harmonic approximation (green).

4. After the completion of a period the data is processed in order to identify the *complex-valued gain* assigned to the current frequency, cf. to Figure 4 for illustration. Therefore, the complex–valued Fourier coefficient of the first harmonic of the plant response is computed. It results from simple discrete Fourier transform of the time series formed by the collected data of the elapsed

period. Also the complex–valued Fourier coefficient of the first harmonic of the input signal time series is computed. (The latter depends only on A, *nSamples,* and λ and can therefore be computed beforehand without the need for sampling.) The *complex-valued gain* is the quotient of these two Fourier coefficients.

5. Procedure steps 1-4 need to be repeated for each of the grid frequencies and/or amplitudes. The *complex-valued gains* representing the frequency response data are successively stored in a table.

6. Finally, the total frequency response may be displayed in a Bode diagram and used for any of the purposes discussed in the introduction.

4 Describing functions und dual locus method

The matter of this section is adopted from [5], [6] and presented here for convenience only and to support the understanding of the application example in section 5.6.

4.1 Assumptions

Limit cycles are periodic oscillations performed by non-linear systems. Without external input signals, the oscillations sustain with a certain frequency and amplitude. Since in many cases they are not desired, criteria that enable non-linear stability analysis are useful. An approximation method that can be applied for analyzing the existence and properties of limit cycles for a class of non-linear systems is the dual locus method [9], [10]. This method requires several assumptions. The first assumption is that it is feasible to represent the open-loop system as a series connection of a single nonlinearity n and the remaining linear part $G(s)$. Therefore, the total system consists of a single loop as depicted in Figure 5.
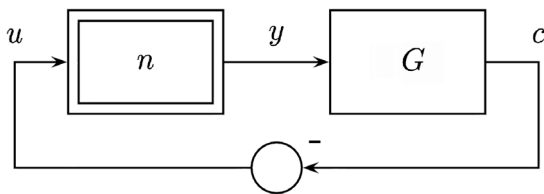
Figure 5: Closed loop formed by a nonlinear part n and a linear filter G.

The analysis of limit cycles starts with the assumption that the system is in the state of a sustained oscillation. Another assumption is necessary for the application of the dual locus method: the linear part G needs to have distinct low-pass properties in the frequency range of the considered oscillation and at higher frequencies. As a rule of thumb, a drop of -40 dB per decade is required.

4.2 Harmonic linearization

With these assumptions it is reasonable to assume that the signal u being present at the input of the non-linearity is approximately sinusoidal. This holds because u is equal (except for the sign) to the output signal c of the linear part. Since the higher harmonics are attenuated by the low-pass effect of G, the output of the non-linearity can be approximated by its first harmonic. Thus, the consideration of the non-linearity can be restricted to its transmission of sinusoidal input signals, and a linear approximation of the non-linear system can be obtained. This approach is called *harmonic linearization*.

Furthermore, a *describing function N* of the nonlinear system n is defined according to (13) as the frequency response from a sinusoidal input signal to the first harmonic of the output signal. For static characteristics, the describing function depends only on the input amplitude A. Input and first harmonic of the output are in phase and thus $N(A)$ is real-valued. The describing function may also be applied to non-linear dynamic elements that produce a frequency- and amplitude-dependent phase shift. Then the describing function $N(\omega, A)$ is complex-valued. For some elementary non-linearities, the describing functions can be derived analytically [10] by Fourier series expansion of the periodic signal y.

4.3 Dual locus method

If the system shown in Figure 5 is in a sustained oscillation and the abovementioned assumptions hold then the transmission properties of the non-linearity can be approximated by its describing function. This leads to the condition

$$N(\omega, A) \cdot G(j\omega) = -1$$

or $\qquad G(j\omega) = \dfrac{-1}{N(\omega, A)}$ (14)

which is denoted *harmonic balance*. If this equation holds for a pair (ω, A) then the system is capable of performing an oscillation with this frequency and amplitude. The lower representation of (14) provides the foundation for the graphical *dual locus method*: limit cycles are possible if there exist intersection points between the locus $G(j\omega)$ of the linear part and the locus of the *negative-inverse describing function* $-1/N(\omega, A)$. From the parameterization of both loci at

the intersection, the values of ω and A can be determined as properties of the corresponding limit cycle. For a criterion reflecting the stability of limit cycles the reader is referred to [6].

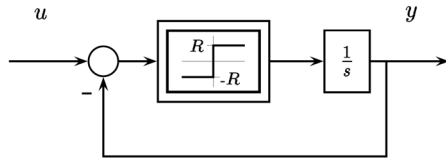### 4.4 Example: Describing function of a rate limiter

Figure 6: Ideal rate limiter.

As an example Figure 6 shows an ideal realization of a rate limiter. (Note that this realization is not suitable for numeric simulation since the infinite gain of the switch induces chattering. A remedy is replacement of the switch by a limiter with high gain.)

In Figure 7 some time responses of the rate limiter are shown. Various sinusoidal input signals are applied with different frequency ω and amplitude A.
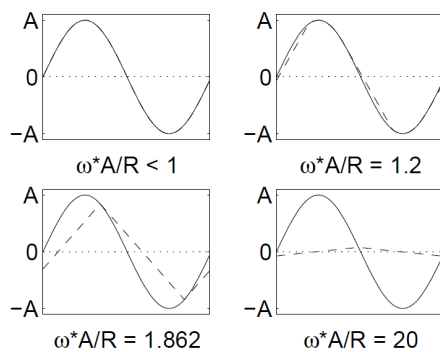
Figure 7: Input and output of a rate limiter at various values of $\omega{\cdot}A/R$.
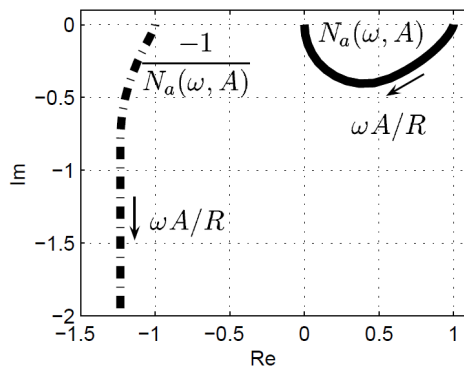
Figure 8: Locus (solid line) and negative inverse locus (dashed line) of the rate limiter describing function.

Due to the memory of the integrator this dynamic non-linearity is not representable by a static characteristic. Nevertheless, a describing function can be derived. The shape of the output signal in relation to the input signal only depends on the ratio $\omega{\cdot}A/R$. Therefore, the describing function only depends on this composed parameter.

Figure 8 shows both the locus and negative-inverse locus of $N(\omega, A)$. More details of the rate limiter describing function derivation can be found in [11], [6].

5 Application of the Modelica package and examples

In this section, the Modelica package application is shown in the context of Dymola 7.4 used as modeling and simulation environment.

5.1 Stimulus signal generator implementation in Modelica

A model class *Chirp* which can be used to create the plant stimuli described in section 2.1 is included in the Modelica package. By parameter settings it allows for choosing adequate frequency progression as well as signal amplitude scheduling according to one's needs.

5.2 Implementation of the algorithms for frequency response data recording

In Figure 1 the standard application is shown. The model class *FR_Recording* covers the total functionality necessary for recording the frequency response data of the block *plant*. To avoid confusion note that the interconnection shown here is *not* a closed loop in the classical meaning, since the output of *FR_Recording* does not depend on its input. Rather, stimulation and analysis functionalities are combined in one model class for easier use. The output of *FR_Recording* is internally connected to the output of a *Chirp* instance thus providing the stimulus for the plant. To complete the interconnection the plant response is input to the *FR_Recording* instance in order to make it accessible there for analysis. That is then performed according to the steps demonstrated in section 3.

Before starting the simulation, the Dymola *Experiment Setup* needs to be well defined. Only variable step solvers should be used since the events triggered by the procedure e.g. for data sampling are not equidistant in time. This is due to the continuously changing frequency but equidistant sampling along the *period function*.

Attention should be paid to the *Output→Store* settings. Depending on the model complexity storing all variables may easily exceed the available memory or disc space. Therefore, it is recommended to only store *Output* variables only at events (no *Equidistant time grid* storage). An instance of the *FR_Bus* model class should be used and connected as shown in Figure 1. On the bus all signals are available which are relevant for the frequency response evaluation. They all have the *output* prefix and will therefore be exclusively saved during simulation and be available in the Dymola variable browser, cf. Figure 9. In addition, for large models consider to set the *protected* attribute to both the plant and the *FR_Recording* instance to prevent excessive data storage.

Figure 9: Dymola variable browser showing signals on fR_Bus

Figure 10: Dymola dialog for setting parameters of the *FR_Recording* class.

Figure 10 shows the dialog window for setting the parameters of the frequency response data recording tool. With parameter *m* the signal dimension of the plant output is set, e.g. *m* = 1 for single output plants. For the chirp stimulus *fStart* and *fEnd* define the frequency interval, *nPeriods* is the total number of sinusoid periods. One of four frequency progression types corresponding to (3), (5), (6), or (7) can be selected. The dependency of the stimulus amplitude on time and/or frequency is provided by a replaceable model, which allows the user to define it according to his needs. *nSamples* is used to specify the number of data points per period to collect time series data for discrete Fourier transform. Finally, some parameters can be specified to control the saving of frequency response data to disc. After the simulation signals for plotting one of multiple representations of the frequency response are available in the Dymola variable browser as shown in Figure 9.

5.3 Example: Frequency response of a mass-spring-damper system

The mass-spring-damper plant depicted in Figure 11 is used as an example to demonstrate how frequency response data can be recorded by means of a simulation while using classes from the Modelica package.
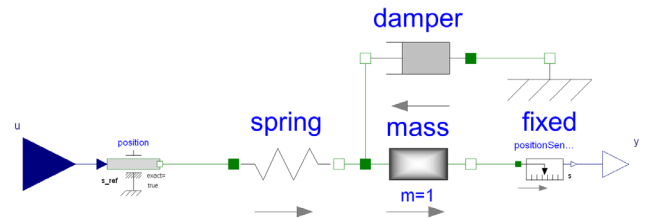

Figure 11: Mass-spring-damper system.

Therefore, the block *plant* in Figure 1 is made an instance of the mass-spring-damper model. It can just as well be modeled as a second order transfer function

$$G_{msd}(s) = \frac{y(s)}{u(s)} = \frac{\omega_0^2}{s^2 + 2 \cdot D \cdot \omega_0 \cdot s + \omega_0^2} \quad (15)$$

Thus, the frequency response can be calculated analytically by replacing $s = j\omega$ and the experimental result can be compared to this precise analytic reference (denoted *theory* in the plots below). The parameters of the mass-spring-damper plant are chosen such that the resonance frequency is $\omega_0/2\pi = 10$ Hz and the damping coefficient is $D = 0.2$.
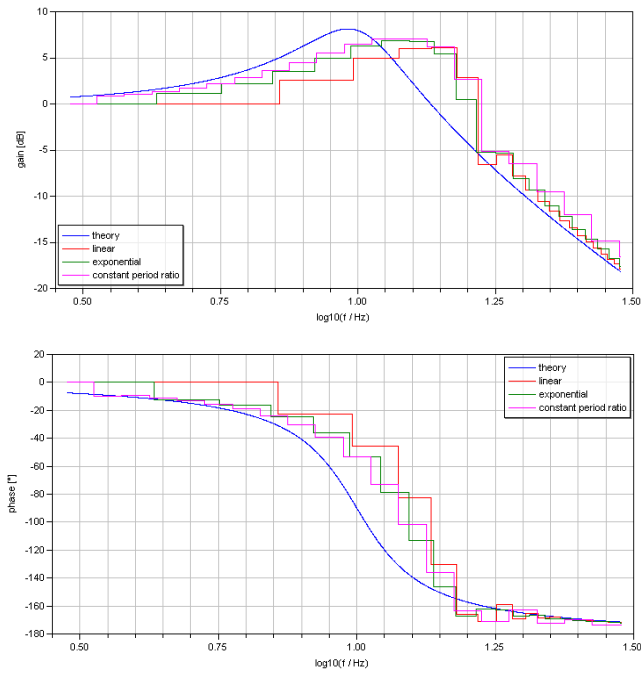
Figure 12: Influence of frequency progression on the resulting Bode plot

Figure 13: Influence of the number of periods *nPeriods* on the resulting Bode plot.

In Figure 12 first attempts are shown to determine the frequency response of the mass-spring-damper system using only twenty excitation periods. The simulation is repeated with each of the three frequency progression variants (5), (6), (7). With the *constant period ratio* frequency progression according to (7) the Bode plot frequency grid points are equally distributed along the logarithmic frequency axis which appears to be the better choice when compared to linear (5) or exponential (6) progression. Figure 13 continues the simulations shown in Figure 12. However, now the total number of periods *n* is increased from one simulation to the next whereas the frequency progression with *constant period ratio* is kept. The comparison shows that the accuracy of the resulting frequency response data clearly improves with increasing *nPeriods* and converges towards the analytic reference. The reason is that the transient portion of the plant response looses significance when the frequency is changing at a lower rate. In practice the precise reference normally is not known. Then, a reasonable number of periods can be found by gradual increase until the change of the result appears tolerable. In addition, at least for linear systems the plant response offset and/or the deviation between the plant response and its first harmonic can be observed to find out whether the system is sufficiently steady state. Both quantities are also calculated by the tool after each completed period.
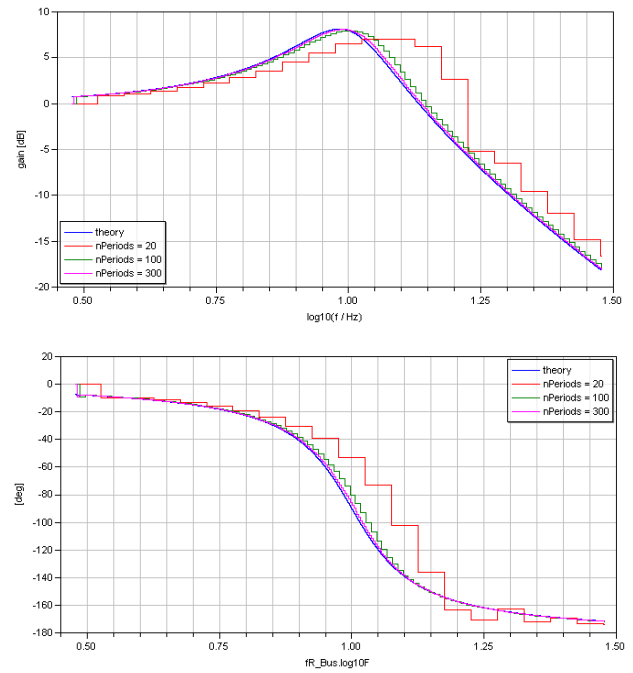
5.4 Example: Frequency response of a complex multi-body vehicle model

The recording of frequency response data using the Modelica package does not only work for simple academic models.
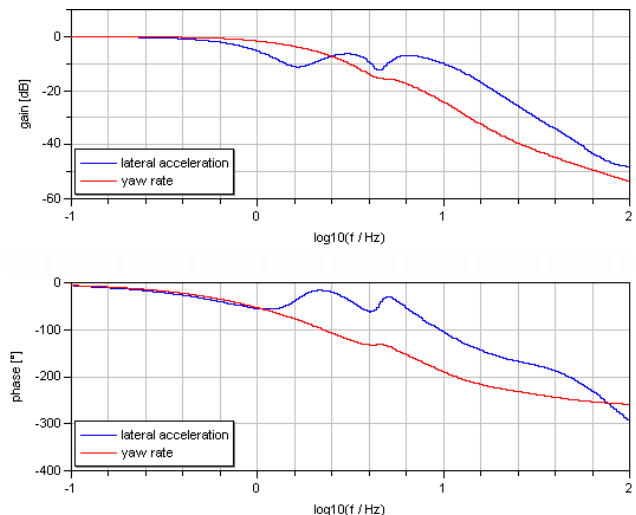

Figure 14: Bode plot of a complex multi-body vehicle model at 80 km/h; The input is the steering angle, outputs are lateral acceleration (blue curve) and yaw rate (red curve).

In Figure 14 the analysis of a multi-body vehicle model with 54 states from our *VehicleControls* library [7] is shown. The comparison of the vehicle

steering response w.r.t. gain and phase of yaw rate vs. lateral acceleration allows for assessment of respective criteria [13]. The gains are normalized with the respective steady state gains here.

The frequency response data is represented by nice smooth curves even in face of the complexity and nonlinearity of the plant. *Dassl* was used as integration method. The total simulation of 434 seconds took 169 seconds on a 3.0 GHz MS Windows PC.

5.5 Frequence response of multi-input multi-output (MIMO) models

Multiple plant outputs can be handled without any modification but setting the number m of plant outputs correctly, cf. Figure 10. The frequency response data for each frequency sample will then be vector valued, correspondingly.
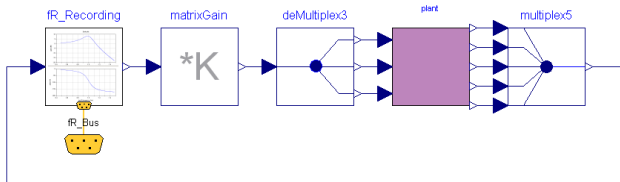

Figure 15: Total Modelica model for frequency response recording of a MIMO plant model.

Handling of multiple inputs is no difficulty either. Apparently a stimulation of multiple inputs at the same time is not expedient. Hence the analysis should be performed by stimulating one input only in one simulation. Figure 15 shows the frequency response data recording setup for an exemplary MIMO plant model exhibiting five outputs and three inputs. The active input can be selected by appropriate choice of the *matrixGain.K* parameter vector.

5.6 Application example for describing functions

The tool at hand can be applied to determine describing functions in exactly the same manner as for frequency response data recording. The example shown here examines the rate limiter from section 4.4. Therefore, the *plant* instance in Figure 1 is redeclared by the corresponding rate limiter Modelica class (using $R = 1$ here). The combined parameter $\omega{\cdot}A/R$ is the only relevant independent quantity, see section 4.4. Therefore we can choose the variation of either ω or A. In this example we choose ω constant by setting *fStart = fEnd* = 1Hz and specify a variation of A with time, instead. (This proceeding is suitable especially for all static non-linearities.) The re-

sults are shown in Figure 16 in terms of the Bode diagram. The describing function may as well be represented as a locus or negative inverse locus, which was already shown in Figure 8.
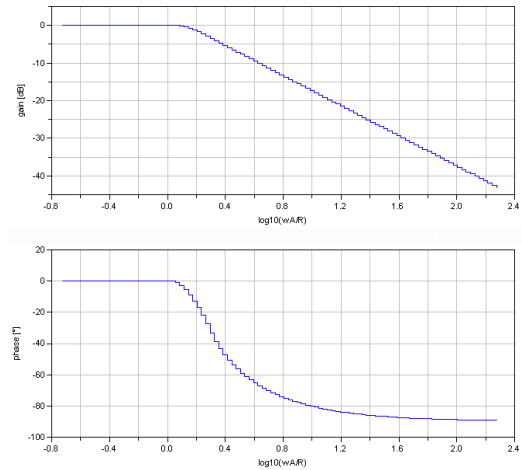

Figure 16: Bode plot of rate limiter describing function.

Now, the dual locus method is applied to investigate whether the closed loop shown in Figure 17 can perform limit cycles. Here, the linear system model is the second order lag system already considered in section 5.3, however, with a damping coefficient of $D = 0.1$ here.
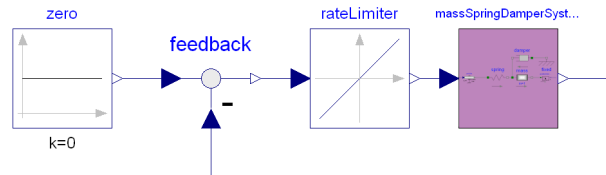

Figure 17: Closed loop with second order linear system and rate limiter in series connection.

Figure 18 shows both the negative inverse describing function locus of the rate limiter (green line) and the locus of the linear part. Again the analytical result (blue line) and the recorded frequency response data (red line) are compared. The discrepancy between them is noticeable. This is due to the specific graphical representation. With the locus, the highest gain (which occurs at the resonance peak, compare Figure 13) appears most prominent. Obviously, during frequency response data recording the system would need some more time here to become steady state and to produce a preciser locus.

Two intersection points between the Nyquist locus of the linear part and the locus of the negative inverse rate limiter describing function exist. Further analy-

sis yields that only the lower intersection point represents a *stable* limit cycle [6].
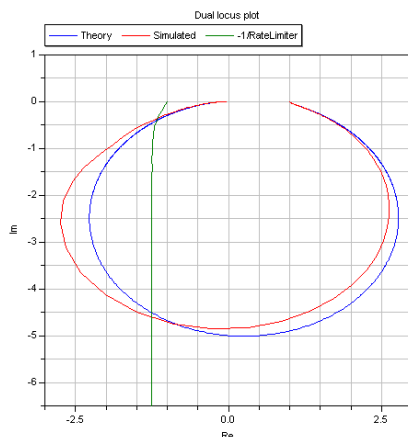

Figure 18: Dual locus plot for the detection of limit cycles of the system shown in Figure 17.

A time simulation of the total system from Figure 17 with dedicated initial condition in fact exhibits the limit cycle shown in Figure 19. The frequency and amplitude of the limit cycle correspond to the parameterization of the loci at the lower intersection point.
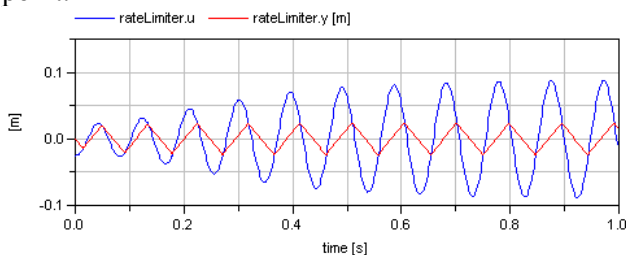

Figure 19: Limit cycle performed by the system shown in Figure 17.

5.7 System identification from frequency response data

To complete the application examples, briefly the option of identifying LTI models (linear time invariant models) from frequency response data is discussed. Imagine the task to derive a handy LTI representation of a rather complex nonlinear Modelica model when application of the linearization method (cf. section 1.1) is not reasonable. This may be the case e.g. due to one of the problems illustrated in section 1.2. The LTI representation can be used e.g. for model based control. Moreover, models represented in terms of frequency response data cannot be simulated and therefore, back-translation into a model which can be numerically time-integrated may be useful. For example, it may serve as an alternative

complexity reduced model in a simulation framework where scalable models are beneficial.

As stated in the introduction, the frequency response data generated with our package well complies with *FRD models* in Matlab. Hence, the methods available with the Matlab System Identification Toolbox (e.g. the *pem* function) can be used to derive LTI models of user-preassigned order. Another way which completely avoids Modelica foreign tools is the following: Choose an LTI model representation from the LinearSystems library [1] and an adequate system order. In the next step optimize (i.e. identifiy) the parameters of the LTI model. This can be done e.g. by using the model calibration feature from the Dymola Design Library such that the Bode diagram fits the frequency response data sufficiently well.

6 Conclusions

An easy-to-use tiny Modelica package for automatic recording of frequency response data and describing functions was presented. The frequency response representations which are resulting from quasi-single frequency harmonic stimulation are significantly smoother than what can be obtained from spectral analysis after stimulation with a multi-frequency (e.g. noise) input signal.

The question into which greater library this package will be usefully integrated and what needs to be done for seamless assembly still needs to be resolved.

7 Acknowledgement

References

[1] Baur, M., Otter, M., Thiele, B.: Modelica Libraries for Linear Control Systems. Proc. 7th Modelica Conference, Como, Italy, 2009.

[2] Abel, A., Nähring, T.: Frequency-Domain Analysis Methods for Modelica Models. Proc. 6th Int. Modelica Conference, Bielefeld, Germany, 2008

[3] Heckmann, A. et. al: The DLR FlexibleBodies library to model large motions of beams and of flexible bodies exported from finite element programs. Proc. 5th Int. Modelica Conference, Vienna, Austria, 2006.

[4] Edrén, J. et. al.: Modelica and Dymola for education in vehicle dynamics at KTH. Proc. 7th Int. Modelica Conference, Como, Italy, 2009.

[5] Ackermann, J., Bünte, T.: Robust prevention of limit cycles for robustly decoupled car steering dynamics. Kybernetika, 35(1):105-116, 1999.

[6] Ackermann, J. et. al.: Robust Control: The Parameter Space Approach. Springer, London, 2002.

[7] EUROSYSLIB Project Profile, http://www.itea2.org/public/project_leaflets/ EUROSYSLIB_profile_oct-07.pdf, 2007.

[8] FAIR Project profile http://www.forschungs-stiftung.de/index.php/Projekte/Details/FAIR-Fahrwerk-Antrieb-Integration-ins-Rad.html, 2009.

[9] Gelb, A., Vander Velde W.: Multiple-Input Describing Functions and Nonlinear System Design. New York: MacGraw-Hill, 1968.

[10] Siljak, D.: Nonlinear systems: the parameter analysis and design. New York: Wiley, 1969

[11] Duda, H.: Fliegbarkeitskriterien bei begrenzter Stellgeschwindigkeit. Ph.D. thesis, Technische Universität Braunschweig, April 1997. Forschungsbericht 97-15, Deutsche Forschungsanstalt für Luft- und Raumfahrt e.V., Köln.

[12] Pintelon, R., Schoukens, J.: System identification: a frequency domain approach. IEEE Press, New York, 2001.

[13] Mitschke, M., Wallentowitz, H.: Dynamik der Kraftfahrzeuge. 4. Auflage, Springer-Verlag, 2004.