Finite Order Approximation by Moment-Matching

Developed by:

Centre for Ocean Energy Research

FOAMM - Finite Order Approximation using Moment-Matching

Version 1.0

Centre for Ocean Energy Research, Maynooth University

February 20, 2019

Contents

1 General Information **2**
- 1.1 Brief Software Description . . . 2
- 1.2 Description of the Main Features Behind FOAMM . . . 2
- 1.3 Platform Requirements . . . 2
- 1.4 Installation . . . 3
- 1.5 Organization of the Manual . . . 3

2 Getting Started **3**
- 2.1 File List . . . 3
- 2.2 Main.m . . . 4
- 2.3 Data.mat . . . 6

3 Using the System **7**
- 3.1 Example case: Manual method . . . 7
- 3.2 Example case: Optimised-automatic method . . . 10
- 3.3 Example case: Force-to-motion dynamics approximation . . . 10

4 Recommendations and Contact **11**

Acknowledgments

The authors are grateful to Denis Buckley and John Maloco, from the Electronic Engineering department of Maynooth university, who have provided valuable advice.

This application is based upon works supported by **Science Foundation Ireland** under grant no. **13/IA/1886**.

Journal paper

This algorithm is based on the theoretical foundations of the following study:

Faedo, N, Peña-Sanchez, Y and Ringwood, J. V. *Finite-Order Hydrodynamic Model Determination Using Moment-Matching*. Ocean Engineering (In Press), 2018.

1 General Information

This document provides a simple user's guide to FOAMM. In the following sections, you will find guidelines to each of the software capabilities.

1.1 Brief Software Description

Finite **O**rder **A**pproximation by **M**oment-**M**atching is a MATLAB application that implements the moment-matching based frequency-domain identification algorithm described in *[Faedo, Peña-Sanchez and Ringwood, 2018]*. This procedure was primarily developed for marine applications (specifically, wave energy). Both the identification of the **radiation kernel frequency response** and the **force-to-motion dynamics** of a marine structure can be performed using this tool. Please, see *[Faedo, Peña-Sanchez and Ringwood, 2018]* for a detailed development of the theory behind this application and the hypothesis required on the target system to identify.

This software comprises different modes of operation and several options, which will be detailed in each corresponding section. Furthermore, this technical manual provides simple examples on how to use the application.

1.2 Description of the Main Features Behind FOAMM

The moment-matching based identification technique proposed in *[Faedo, Peña-Sanchez and Ringwood, 2018]*, has been developed to obtain a finite order approximation for both the **radiation kernel frequency response** and the **force-to-motion dynamics** of a marine structure. This technique has one main distinctive characteristic: it allows the user to select a set of frequencies to **exactly match** the behaviour of the system being identified. This means that the obtained finite order model will produce the **same** steady-state response as the target system for the chosen (interpolation) set. The final order of the finite approximation ν will explicitly depend on the number of frequencies β selected in this interpolation set i.e. $\nu = 2\beta$.

We clarify this aspect since this application provides different "methods" (options) to select this set of interpolation points: a **manual** method, **automatic** method and **optimised-automatic** method. These methods, along with other different characteristics of the algorithm, are explicitly detailed in Section 2.2.

Another feature of this technique is that it allows to specify a particular **frequency range** to perform the identification. Such a range is considered by the algorithm to assign the eigenvalues of the finite order approximation using an ℓ^2 criterion, and can be selected following Section 2.2.

1.3 Platform Requirements

The application was created using a MATLAB compiler. The main file is a MATLAB **".m"** script, which calls an executable file called **FOAMM**. Two versions of FOAMM are available at the moment, one for Windows, and another one for Linux. The compatibility of the Windows version has been tested over Windows 7,8 and 10 with different MATLAB versions (2012, 2015, 2017); while the Linux version has been tested over Ubuntu 18 and CentOS (6.1 and 7.4).

Note: this application runs in a stand-alone fashion on a plain MATLAB distribution. No other MATLAB toolboxes/applications are required. This is exactly why we have considered a MATLAB compiler in the first place.

1.4 Installation

The files can be downloaded from http://www.eeng.nuim.ie/coer/downloads/. The only requirement is to have the correct version of the Matlab Runtime, for which the installer is provided in the folder "Matlab Runtime". After installing the Matlab Runtime, it effectively runs in a stand-alone fashion on a plain MATLAB. It should be noted that, for the Linux version, administrative access is required for both the installation of the Matlab Runtime, and the use of the application.

Note: the provided Matlab Runtime installer needs internet connection to download the required files. Additionally, the first run of the application is considerably slower than the subsequent ones. This is indeed a known issue when using the MATLAB compiler.

1.5 Organization of the Manual

The remainder of this technical manual is organised in the following sections:

- Section 2 - **Getting Started**: This section provides details on the files structure and how to access each one of the possible options of the application. Particularly, a **File List** is provided in Section 2.1, while the technicalities behind each important file are discussed in Section 2.2 and 2.3.
- Section 3 - **Using the System**: This section discusses two different example cases (3.1 and 3.2), so that the user can have a technical case to follow.
- Section 4 - **Recommendations**: This is a short section to provide some recommendations and list the known issues of this **FOAMM** β-version.

2 Getting Started

2.1 File List

Here you will find the list of files compressed in **"FOAMM.rar"** and its corresponding usage.

Files

- **Main.m**: This is the main file and the first user interface to the application. The file loads the frequency-domain data contained in **"Data.mat"** (see Section 2.3) used to perform the identification and allows to change between the different modes of the system (see Section 2.2). **"Main.m"** set-ups the required variables accordingly and explicitly calls the executable file **"FOAMM"**.

  Note: This file provides the **ONLY** way to select between the different modes of the application.

- **Data.mat**: This file contains the frequency-domain data to identify (computed with any Boundary Element Method code) and it should be provided by the user, following the specific input format detailed in Section 2.3.

- **FOAMM**: This is the executable file of the application, and its explicitly called by **"Main.m"**.

2.2 Main.m

This file provides control over the options and modes of the application and loads the frequency-domain data from **"Data.mat"**. We provide in this section, a description of the possible identification "methods" of this application and each one of the customisable options.

2.2.1 Application Methods

The application counts with the following identification methods (we refer the reader to the brief description of the fundamentals behind moment-matching provided in Section 1.2).

Application methods

- **Manual method**: the user selects the desired set of frequencies to interpolate (achieve moment-matching). If we denote the number of frequencies as β, then the order of the finite order approximation is $\nu = 2\beta$. The selection of the frequencies can be done using a user interface (contained in **"FOAMM"**) or manually in the "Options" structure inside **"Main.m"**. This is further discussed in Section 2.2.2. This method is exemplified in Section 3.1.

- **Automatic method**: the user selects a subset of frequencies to interpolate (with size denoted by α) and a final number of interpolation points denoted by β. This method has an optimization process that selects an optimal combination of the $\beta - \alpha$ interpolation points on a given frequency range selected by the user using the "Options" structure (see Section 2.2.2) or with a graphical interface provided by the application). This optimisation is based on a minimisation of the ℓ^2-norm in the chosen frequency range.

  Note: if the user does not pre-select a set of frequencies i.e. $\alpha = 0$, all the β interpolation points are selected automatically under the same optimisation process.

- **Optimised-automatic method:** this method is essentially the automatic method, but it also selects the number of interpolation points β automatically. This method basically adds frequencies to match in the identification process until the approximated model satisfies both an absolute and a relative tolerance specified by the user in the "Options" structure (see Section 2.2.2). This method is exemplified in Section 3.2.

Note: Even though the application counts with different automated method, the author's highly recommend a sensible (manual) choice of the interpolation frequencies, based on the system dynamics to identify. In fact, this option to perform a suitable selection of points to interpolate in the frequency-domain is one of the most attractive characteristics behind moment-matching. For example, the user might want to select as one sensible choice the resonant frequency of the structure under identification.

2.2.2 Options Structure

Every option of the application can be changed using the structure **"Options"**. Such an structure can be accessed (and tuned) as follows:

Options structure

- (integer) **Options.Mode**

default **0** $\longrightarrow$ compute an approximated model for the radiation impulse response of the device.
1 $\longrightarrow$ compute an approximated model for the force-to-velocity dynamics of the device.

- (integer) **Options.Method** - see Section 2.2.1 for a description of each method.

default **0** $\longrightarrow$ Manual method.
1 $\longrightarrow$ Automatic method.
2 $\longrightarrow$ Optimised-automatic method.

- (string or float) **Options.FreqRangeChoice**

default **'G'** $\longrightarrow$ Select the frequency range from a plot provided by a graphical interface.
'C' $\longrightarrow$ Enter the frequency range manually when asked in the command window, as a vector containing the lower and upper bounds of the range i.e. $[\omega_{\text{min}}, \omega_{\text{max}}]$.
VEC $\longrightarrow$ Enter the frequency range manually as a vectorcontaining the lower and upper bounds of the range i.e. **VEC**$=[\omega_{\text{min}}, \omega_{\text{max}}]$.

- (string or float) **Options.FreqChoice**

default **'G'** $\longrightarrow$ Select the set of frequencies to achieve moment-matching from a plot provided by a graphical interface.
'C' $\longrightarrow$ Enter the set of frequencies to achieve moment-matching manually when asked in the command window in vector form i.e. $[\omega_1, \ldots, \omega_\beta]$.
VEC $\longrightarrow$ Enter the set of frequencies to achieve moment-matching manually in vector form i.e. **VEC**$=[\omega_1, \ldots, \omega_\beta]$.

- (string or integer) **Options.FreqNumChoice** (active if **Options.Method** $= 1$)

default **'C'** $\longrightarrow$ Enter the number of frequencies to interpolate $\beta > 0$ when asked in the command window.
INT $\longrightarrow$ Enter the number of frequencies to interpolate $\beta > 0$ manually.

- **Options.Optim** (optimisation-related options)

(integer) **Options.Optim.InitCond** $\longrightarrow$ Number of initial conditions used on the optimisation process.
(float) **Options.Optim.Tol** $\longrightarrow$ Tolerance value on the final value of the optimisation process.
(integer) **Options.Optim.maxEval** $\longrightarrow$ Maximum number of iterations used in the optimisation process.
(float) **Options.Optim.StepTol** $\longrightarrow$ Step tolerance value for the optimisation process.
(float) **Options.Optim.ThresRel** (active if **Options.Method** $= 2$) $\longrightarrow$ Relative error threshold.
(float) **Options.Optim.ThresAbs** (active if **Options.Method** $= 2$) $\longrightarrow$ Absolute error threshold.

Note: If any of the labels inside the structure "Options" is changed, the application will not recognize the variables and the default values will be used.

2.2.3 Input Variables

The frequency-domain data used to perform the moment-matching based identification is loaded from the user supplied file **"Data.mat"** (see Section 2.3).

2.2.4 Output Variables

We list the output variables in this section, using the following code:

- ν: order of the approximated model. If the number of frequencies selected is β, then $\nu = 2\beta$.

- n: length of the frequency vector used to compute the hydrodynamic coefficients of the structure under identification.
- f: frequency response of the target system.
- $\tilde{\mathsf{f}}$: frequency response of the finite order approximated model.

Output variables

- (matrix, float, $\nu \times \nu$) **A_ss**: Dynamic matrix of the finite order approximation.
- (matrix, float, $\nu \times 1$) **B_ss**: Input matrix of the finite order approximation.
- (matrix, float, $1 \times \nu$) **C_ss**: Output matrix of the finite order approximation.
- (float) **MAE**: Mean Absolute Error of the approximation, which is defined as,

$$\textbf{MAE} = \frac{1}{\mathsf{n}} \sum_{i=1}^{\mathsf{n}} \left| \frac{\mathsf{f}(i) - \tilde{\mathsf{f}}(i)}{\mathsf{f}(i)} \right|.$$

2.3 Data.mat

This file with ".mat" extension must contain all the information regarding the frequency-domain data considered for the identification process. In the following, we provide a consistent description of the required format so that the application can load the data correctly. We use the same code as in Section 2.2.4.

Input data format

- †(vector, float, $\mathsf{n} \times 1$) **A**: Radiation added mass.
- †(vector, float, $\mathsf{n} \times 1$) **B**: Radiation damping.
- †(vector, float, $\mathsf{n} \times 1$) **w**: Frequency vector.
- (scalar, float) **Mu** (active if **Options.Mode** $= 0$): Radiation added mass when $\omega \to \infty$. If this value is not supplied, the application will calculate its value using Ogilvie's relations.
- (scalar, float) **Mass** (active if **Options.Mode** $= 1$): Mass of the structure under analysis.
- (scalar, float) **K** (active if **Options.Mode** $= 1$): Sum of stiffness terms (in the wave energy case, this is usually specified as the sum of the hydrostatic stiffness of the device and the stiffness of the Power Take-Off system).
- (scalar, float) **D** (active if **Options.Mode** $= 1$): Sum of (additional) damping terms (in the wave energy case, this is usually specified as the damping of the Power Take-Off system).

Note: If any of the variables denoted with † is named differently the application will halt.

3 Using the System

In this section we provide a step-by-step example on how to use the **FOAMM** application to identify a finite order parametric model of the radiation impulse response of a particular structure, using different application modes.

Note that, in order to show a geometrically complex device, the hydrodynamic parameters used for this example are not the ones inside the **"Data.mat"** file provided with the application.

3.1 Example case: Manual method

The hydrodynamic coefficients are stored in the file **"Data.mat"** as specified in Section 2.3. The **"Main.m"** file for this example is as follows (please refer to Section 2.2.2 for the meaning of each variable):

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Finite Order Approximation using Moment-Matching (FOAMM) application
%
% Theoretical background for the application:
% - Nicolas Faedo, Yerai Pena-Sanchez and John V. Ringwood. Finite-Order
% Hydrodynamic Model Determination for Wave Energy Applications Using
% Moment-Matching, Ocean Engineering (under review), 2018.
%
% Centre for Ocean Energy Research, Maynooth University.
%
% Last update: 24/04/2018
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load('data.mat')                    % Load the frequency-domain data.

%% Options structure ----------------------------------------------------------
Options.Mode = 0;                   % Radiation impulse response mode.
Options.Method = 0;                 % Manual method (user picks frequencies).
Options.FreqRangeChoice = 'G';      % Frequency range from plot.
Options.FreqChoice = 'G';           % Interpolation frequencies from plot.
Options.FreqNumChoice = [];         % Not used for this example case.
Options.Optim.InitCond = 50;        % Default value.
Options.Optim.Tol = 1E-6;           % Default value.
Options.Optim.maxEval = 100;        % Default value.
Options.Optim.StepTol = 1E-6;       % Default value.
Options.Optim.ThresRel = 0.1;       % Default value.
Options.Optim.ThresAbs = 0.01;      % Default value.
%% ---------------------------------------------------------------------------


%% Run application ------------------------------------------------------------
save('temp_file.mat')               % Save all the structure values so that the
                                    % executable file "FOAMM" can read them

system('FOAMM');

load('temp_file.mat')               % Load the parametric model computed.
delete('temp_file.mat')             % Delete temporary file.
%% ---------------------------------------------------------------------------
```

Note on the Options.Optim structure: Since the assignment of the eigenvalues of the parametric model is carried out considering an optimisation formulation it is possible to change some of the parameters involved in such an optimised process (to make the application more flexible). However, in order to guarantee a correct performance of the application, we recommend not to change the default values of the optimisation parameters unless an atypical case arises.

Running this **"Main.m"** file executes the **"FOAMM"** application. Since for this example we chose to select the frequency range from the graphical interface, the following plot appears:

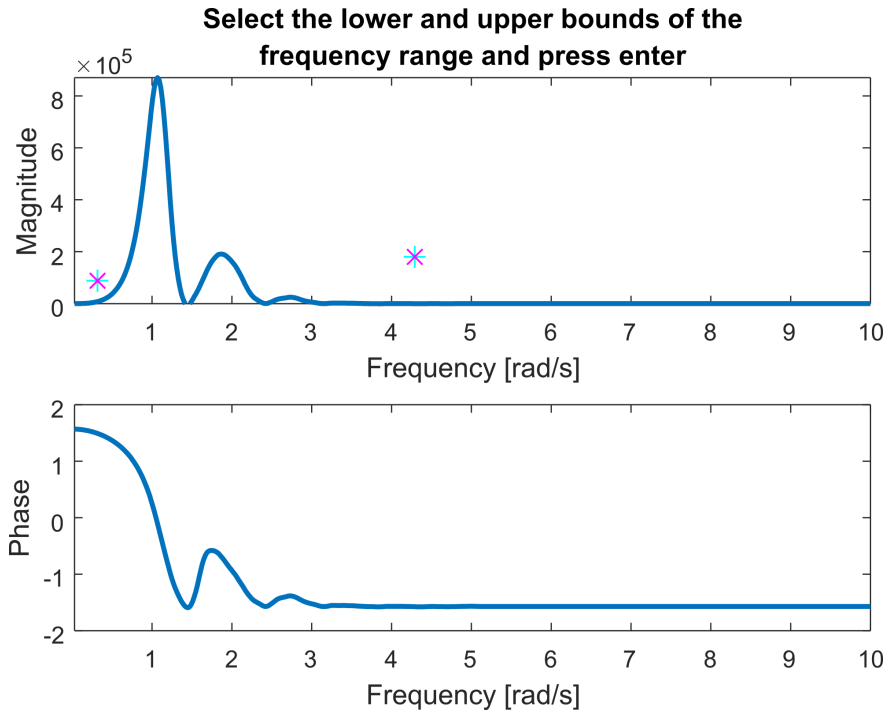
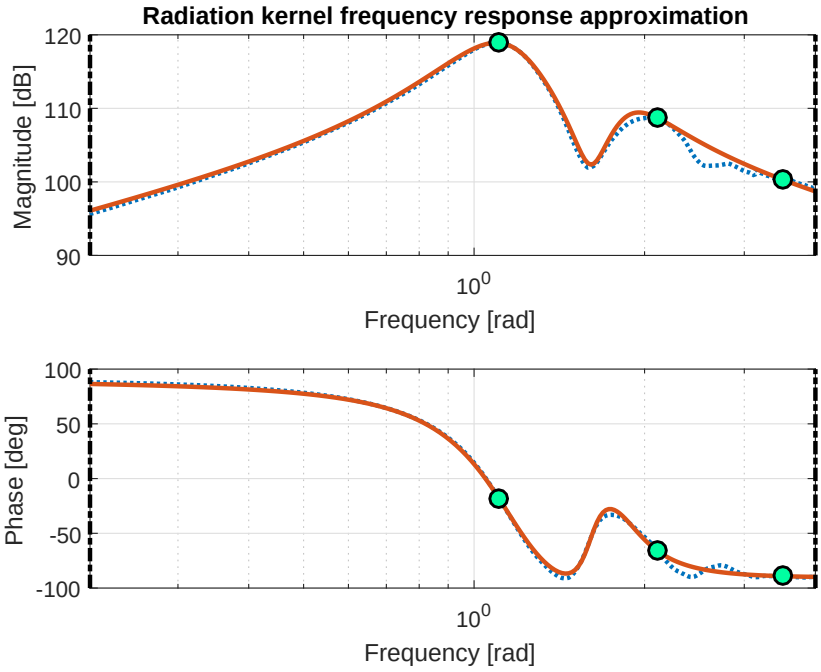


Figure 1: Graphical interface to select the frequency range for the approximation.

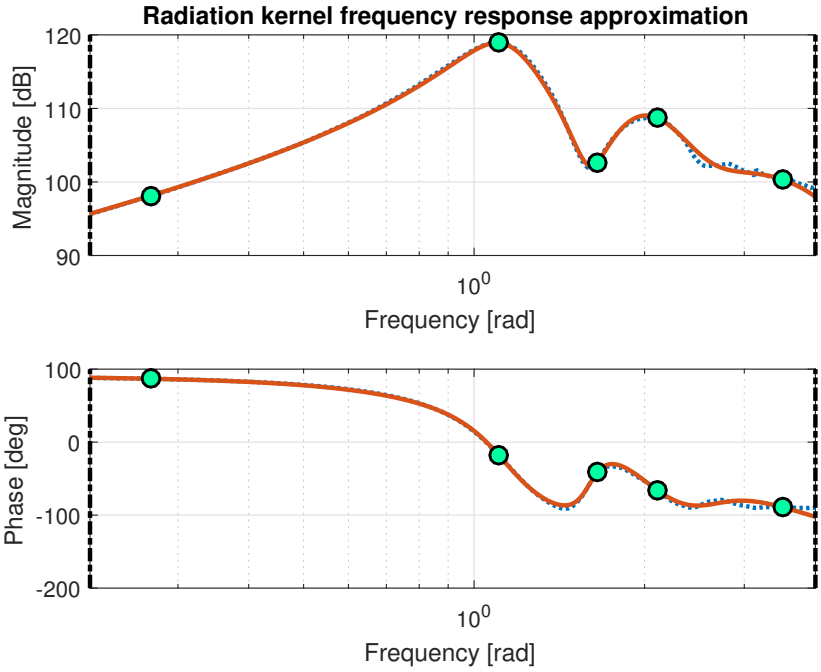
As can be seen in the plot of Figure 1, we select two points (a lower and an upper bound) to define the frequency range. In this case, such a range is given (approximately) by $[0.2, 4.3]$ [rad/s].

According to the options selected for this example, we select the frequencies manually, from a plot provided by the graphical interface. We omit this plot since it is extremely similar to Figure 1 but in this case the user "picks" the frequencies to interpolate from the graph. For this first example, we picked three frequencies: $[1.1, 2.1, 3.5]$ [rad/s]. The frequency response of the obtained parametric model interpolating these frequencies and the target frequency-domain data is depicted in Figure 2a. The mean absolute error computed for this case is 0.071. Note that the order of this model is 6.

We would like to improve this approximation by adding two more frequencies to the interpolation process. This time, the selected frequencies are $[0.2, 1.1, 1.6, 2.1, 3.5]$. The finite order model is shown in Figure 2b. For this last case, the mean absolute error is 0.040.



(a) Bode diagram of the finite order approximation using three interpolation frequencies (solid orange) and the target frequency-domain data (dashed blue).



(b) Bode diagram of the finite order approximation using five interpolation frequencies (solid orange) and the target frequency-domain data (dashed blue).

3.2 Example case: Optimised-automatic method

For this case, the **Optimised-automatic** method (see Section 2.2.1) is selected, setting **Options.Method** = 2. For this case, the user does not select neither the specific set of frequencies to achieve moment-matching nor the number of interpolation points for the final model. This method has an optimised process that selects both variables automatically, according to the thresholds defined in **Options.Optim.ThresRel** and **Options.Optim.ThresAbs**.

For this example case, and while the algorithm computes the necessary calculations, the following plot emerges:

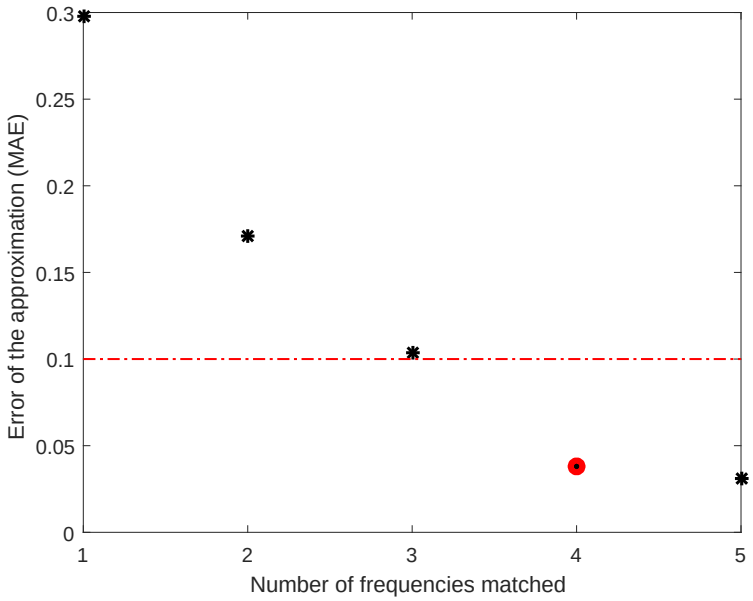


Figure 3: MAE vs. number of frequencies matched for the finite order approximation.

Figure 3 has to be interpreted as follows: the algorithm first seeks for an optimal first interpolation point, and computes a finite order approximation. For this case, the MAE with one frequency point is 0.3, which is higher than the absolute threshold of 0.1 (red horizontal line in plot) and, hence, it automatically seeks for an approximated model with one more frequency. One more time, it optimises the position of these new two interpolation points to obtain a new parametric model. As can be appreciated in Figure 3, and for this example case, the condition on the absolute threshold is met after computing a model with four frequencies. Nevertheless, the algorithm continues seeking for a higher order approximation, until the condition on the relative threshold is met. Since when moving from four to five frequencies the improvement on the approximation (in terms of the MAE) is less than the relative threshold, the algorithm assumes that the increase of model order does not have a strong impact on the approximation and, hence, selects as output the model with four interpolation points (as specified by the red dot in Figure 3).

3.3 Example case: Force-to-motion dynamics approximation

As mentioned in Section 1.1, this application can be used for the identification of both the radiation kernel frequency response and the complete force-to-motion dynamics. In the examples shown before (see Sections 3.1 and 3.2), two different ways to approximate the radiation kernel frequency response were introduced. It is also possible to identify a parametric model of the force-to-motion dynamics by simply changing the value of **Options.Mode** (see Section 2.2.2) from 0 to 1. Figure 4 depicts the approximation of the force-to-motion dynamics when choosing 3 particular frequencies, i.e. [1.1, 1.4, 1.8], achieving a mean absolute error of 0.005.

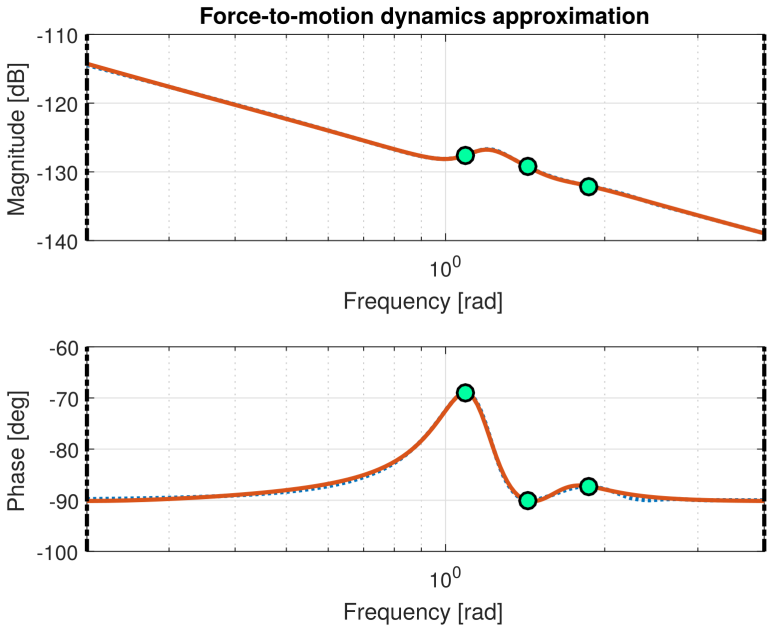


Figure 4: Bode diagram of the finite order approximation of the force-to-motion dynamics using three interpolation frequencies (solid orange) and the target frequency-domain data (dashed blue).

4 Recommendations and Contact

This section is devoted to offer some recommendations on this **FOAMM** β-version. We offer a list of the known issues and how to potentially solve them.

- **Known issue 1** - *Non-convergence of the approximated model to a globally optimal solution*: The optimisation process considered to assign the eigenvalues of the approximated model can be sensible to the set of initial conditions provided by the algorithm. This set is generated randomly within upper and lower bounds defined by the nature of the frequency-domain data.
  - **Proposed fix 1**: Almost virtually every time this issue has appeared, it has been solved by increasing the number of initial conditions for the optimisation process, using the options structure (**Options.Optim.InitCond**).
- **Known issue 2** - *Slow convergence in automatic method*: The automatic method described in Section 2.2.1 can require of some time to compute a model if the number of frequencies selected β is large.
  - **Proposed fix 2**: we remind the user that, within this method, it is possible to pre-select a set of desired frequencies manually (size α). This usually helps the rate of convergence of the application, since the algorithm optimises for the remaining $\beta - \alpha$ interpolation points rather than the original number β.

Important!: Authors will highly appreciate any comments, suggestions and modifications that can make this application a useful identification tool for the marine research community. Feedback should be sent to coer@mu.ie