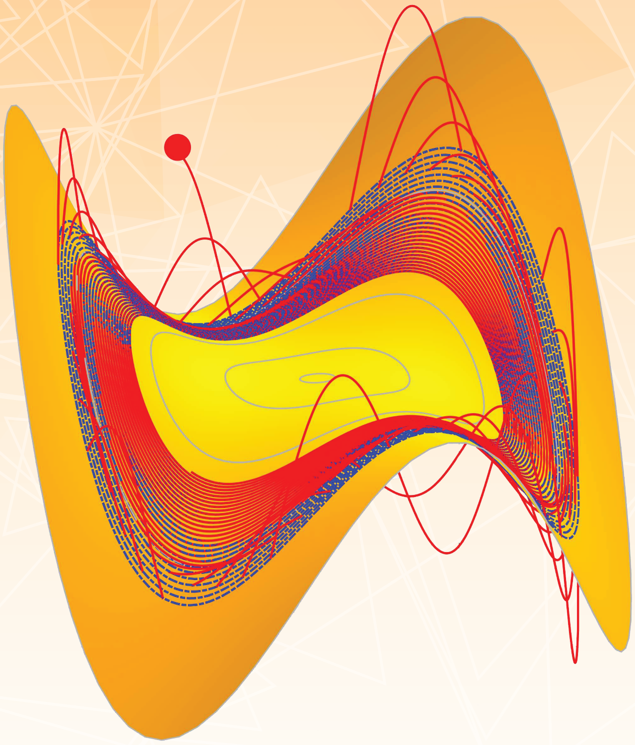
# SSMtool: A short guide



**S. Ponsioen & G. Haller**
Nonlinear Dynamics Group
Institute for Mechanical Systems
ETH zürich

# SSMtool: A short guide

**S. Ponsioen · G. Haller**

March 1, 2019

---

S. Ponsioen (✉) · G. Haller
Institute for Mechanical Systems, ETH Zürich, Leonhardstrasse 21, 8092, Zürich, Switzerland
E-mail: stenp@ethz.ch

## Contents

## 1 Introduction

SSMtool is a MATLAB-based computational tool for computing two-dimensional spectral submanifolds (SSMs) in nonlinear mechanical systems with arbitrary degrees of freedom. SSMtool is intended for researchers and students who are interested in extracting key-information, such as reduced-order models or backbone curves from a nonlinear, dissipative mechanical system with a potentially large number of degrees of freedom. The software achieves this without using any numerical integration or numerical continuation techniques, purely based on a reduction to SSMs.

SSMs where first introduced by Haller and Ponsioen [1] as the smoothest nonlinear continuations of modal subspaces of the linearized system, providing mathematically exact reduced models for the nonlinear system (cf. Ponsioen et al. [2, 3], Breunung and Haller [4] and Jain and Haller [5]). More specifically, let us consider $n$-degree-of-freedom, autonomous mechanical systems of the form

$$\mathbf{M}\ddot{\mathbf{y}} + \mathbf{C}\dot{\mathbf{y}} + \mathbf{K}\mathbf{y} + \mathbf{g}(\mathbf{y}, \dot{\mathbf{y}}) = \mathbf{0}, \quad \mathbf{g}(\mathbf{y}, \dot{\mathbf{y}}) = \mathcal{O}\left(|\mathbf{y}|^2, |\mathbf{y}|\,|\dot{\mathbf{y}}|, |\dot{\mathbf{y}}|^2\right), \tag{1.1}$$

where $\mathbf{y} \in \mathbb{R}^n$ is the generalized position vector; $\mathbf{M} = \mathbf{M}^T \in \mathbb{R}^{n\times n}$ is the positive definite mass matrix; $\mathbf{C} = \mathbf{C}^T \in \mathbb{R}^{n\times n}$ is the damping matrix; $\mathbf{K} = \mathbf{K}^T \in \mathbb{R}^{n\times n}$ is the stiffness matrix and $\mathbf{g}(\mathbf{y}, \dot{\mathbf{y}})$ denotes all the nonlinear terms in the system. These nonlinearities are assumed to be analytic for simplicity. Additionally, we require that the trivial fixed point of system (1.1) is asymptotically stable.

Under appropriate non-resonance conditions, we can construct a two-dimensional autonomous SSM, $\mathcal{W}(\mathcal{E})$, over a chosen spectral subspace $\mathcal{E}$ as an embedding of a two-dimensional open set $\mathcal{U}$ into the full phase space of the system via a mapping $\mathbf{W}(\mathbf{z})$. This mapping is approximated in a neighborhood of the origin using a Taylor expansion in the parameterization coordinates $\mathbf{z} \in \mathbb{C}^2$. Additionally, there exists a polynomial function $\mathbf{R}(\mathbf{z})$ representing the reduced dynamics on the SSM via

$$\dot{\mathbf{z}} = \mathbf{R}(\mathbf{z}). \tag{1.2}$$

By introducing a change to polar coordinates, eq. (1.2) can be rewritten as

$$\dot{\rho} = a(\rho), \quad \Omega = \dot{\theta} = b(\rho), \tag{1.3}$$

where we refer to section 5.2 of Ponsioen et al. [2] for a detailed explanation of this derivation.

In other words, using SSM theory we are able to reduce the $n$-degree-of-freedom system (1.1) to the two ordinary differential equations in (1.3). Additionally, as can be seen in eq. (1.3), the $\dot{\theta}$-equation provides a direct relationship between the amplitude $\rho$ and the instantaneous frequency $\Omega$. This relationship is generally called the backbone curve, corresponding to a given mode. We illustrate this concept in Figure 1.1.

SSMtool is designed to construct the Taylor approximations of the autonomous SSM $\mathbf{W}(\mathbf{z})$ and the reduced dynamics $\mathbf{R}(z)$ in an automated fashion. The graphical user interface (GUI) of SSMtool allows the user to import/define their nonlinear mechanical system and compute the SSM over an arbitrarily chosen spectral subspace under appropriate non-resonance conditions. The software then computes the reduced dynamics up to any predicted order of accuracy and determines the associated backbone curves from the reduced dynamics on the SSM without any further user input.

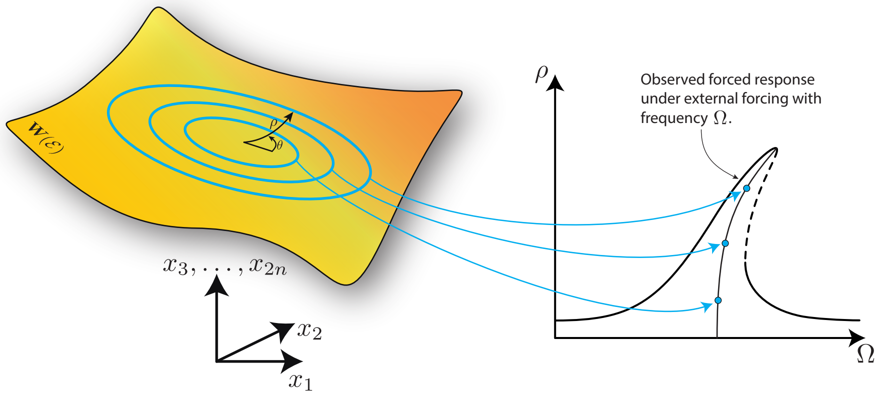


Fig. 1.1: Illustration of the backbone curve construction using the parameterized autonomous SSM. For each fixed radius $\rho_0$ on the SSM, we can identify an instantaneous frequency $\Omega(\rho_0)$ from the reduced system on the SSM, which, subsequently, will represent a point on the damped backbone curve.

### 1.1 Download SSMtool

This software is made public for research use only. It may be modified and redistributed under the terms of the GNU General Public License. SSMtool can be downloaded from the Haller Group software page at the ETH Zürich:

`http://www.georgehaller.com`

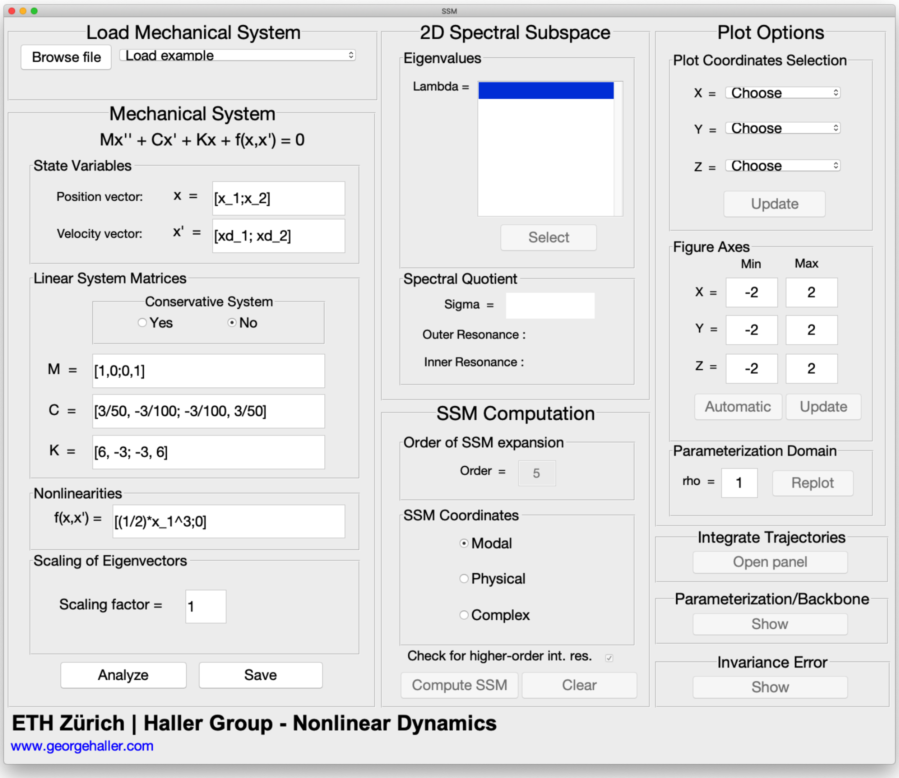
SSMtool requires MATLAB R2016b or newer. This code may be improved and is subject to change.

### 1.2 Installation

To install SSMtool, unzip the source code into a local directory. In MATLAB, change the current MATLAB folder to the SSMtool folder.

## 2 How to use ssMtool

After installing ssMtool, type “SSM” in the MATLAB command window to open the graphical user interface. The graphical user interface shown in Figure 2.1 is displayed.



Mx'' + Cx' + Kx + f(x,x') = 0

Position vector: x = [x_1;x_2]

Velocity vector: x' = [xd_1; xd_2]

Conservative System ○ Yes ● No

M = [1,0;0,1]

C = [3/50, -3/100; -3/100, 3/50]

K = [6, -3; -3, 6]

f(x,x') = [(1/2)*x_1^3;0]

Scaling factor = 1

Order = 5

SSM Coordinates ● Modal ○ Physical ○ Complex

X = Choose

Y = Choose

Z = Choose

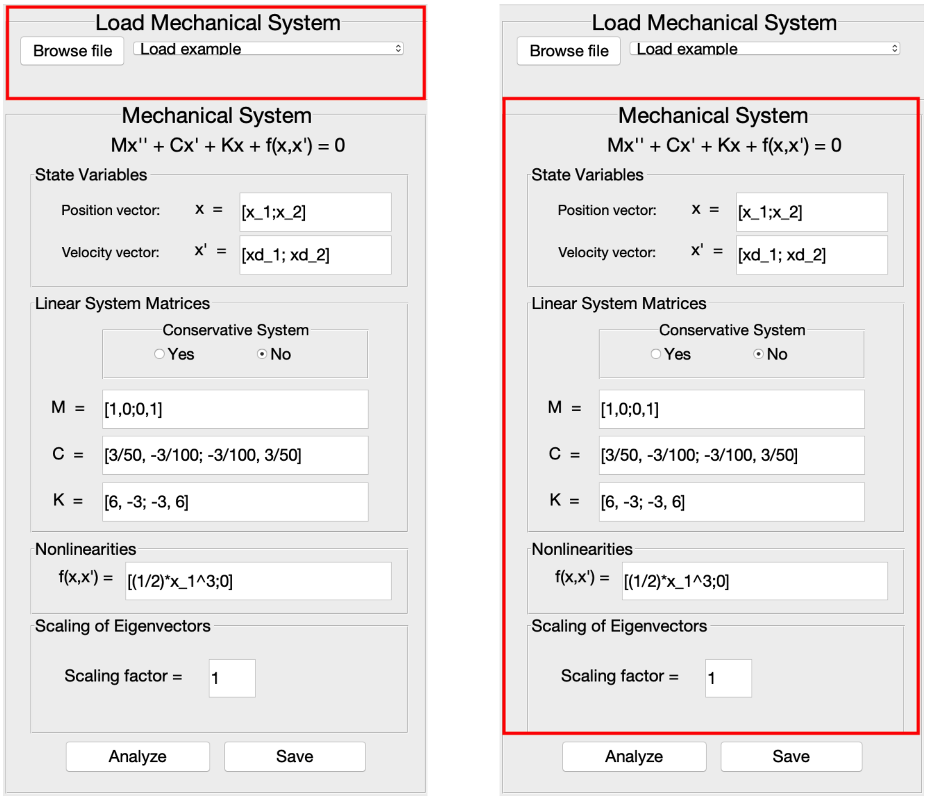
| | Min | Max |
|---|---|---|
| X = | -2 | 2 |
| Y = | -2 | 2 |
| Z = | -2 | 2 |

rho = 1

Fig. 2.1: Overview of the graphical user interface of ssMtool.

### 2.1 Loading a mechanical system

To load an autonomous mechanical system into ssMtool, there is a total of three options. The first option is to load a predefined mechanical system from the drop-down list in the top-left corner of the GUI, as shown in Figure 2.2 (a). The systems listed here have been analyzed, using ssMtool, in the work of Ponsioen et al. [2] and consist out of a two-degree-of-freedom nonlinear mass-spring-damper system with a near-inner resonance and a near-outer resonance respectively, and, additionally, a discretized clamped nonlinear Timoshenko beam having a 32-dimensional phase space. A graphical illustration of both systems is shown in Figure 2.3.



Position vector: x = [x_1;x_2]

Velocity vector: x' = [xd_1; xd_2]

Conservative System ○ Yes ● No

M = [1,0;0,1]

C = [3/50, -3/100; -3/100, 3/50]

K = [6, -3; -3, 6]

f(x,x') = [(1/2)*x_1^3;0]

Scaling factor = 1

(a) (b)

Fig. 2.2: (a) You can load a predefined mechanical system from the drop-down list or load a custom mechanical system from a *.mat* file. (b) Additionally, you can manually enter a mechanical system in the predefined input windows.

The second option is to define your mechanical system in the predefined input windows as shown in Figure 2.2 (b). Here, you are asked to specify the position and velocity vectors, then indicate if the system is conservative or non-conservative, specify the mass, damping and stiffness matrices, the nonlinear force vector and the scaling factor of the transformation matrix containing the eigenvectors of the linearized system at the trivial fixed point. The manually entered system can be saved to a *.mat* file by using the Save button and loaded back into SSMtool for future use by using the Browse file button.

The third and last option available is to load a custom mechanical system from a *.mat* file containing the variables listed in Table 2.1, where the "conservative" variable is either 0 or 1 to indicate if the system is non-conservative or conservative, respectively. The "scaling" variable, mentioned above, can be used to scale the eigenvectors of the system and can be any positive value. The *.mat* file can be loaded by using the Browse file button shown in Figure 2.2 (a).

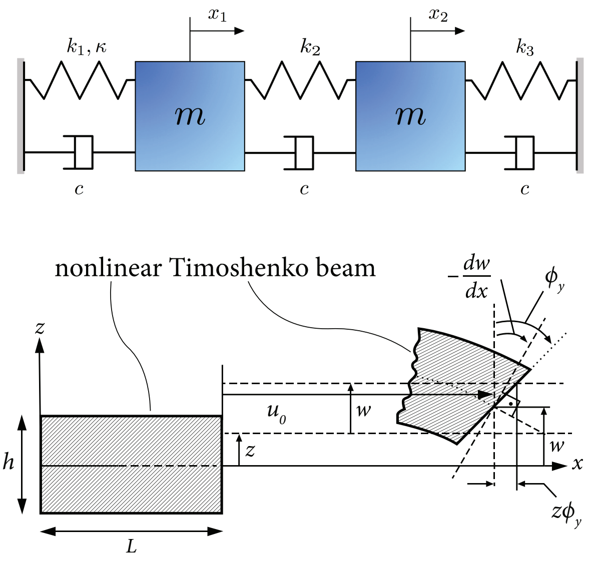


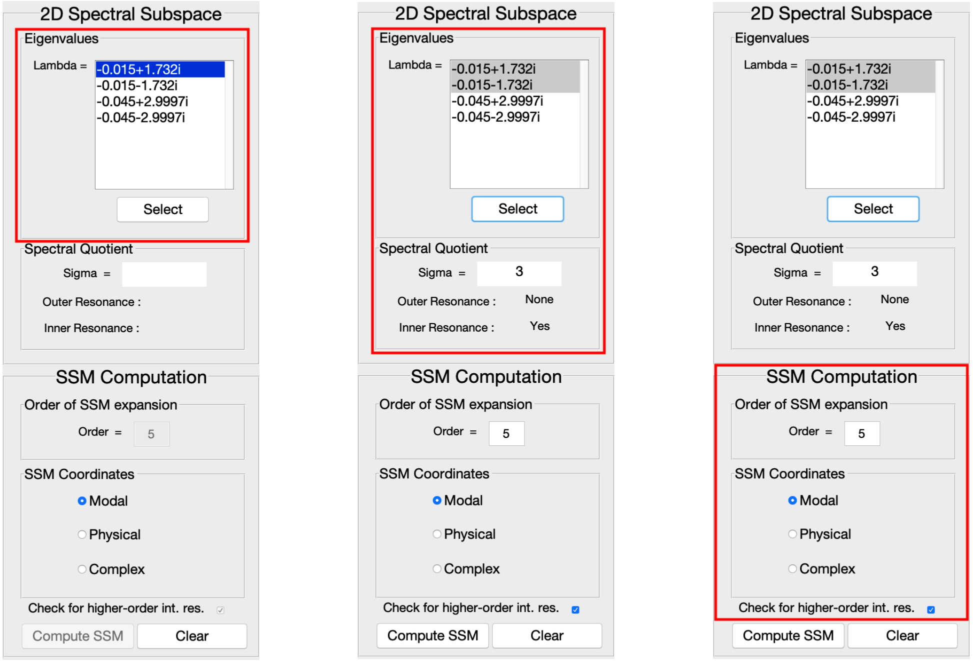
Fig. 2.3: Graphical illustration of the predefined mechanical systems in SSMtool. You can load the modified Shaw-Pierre example as explained in section 7.1 and 7.2 of Ponsioen et al. [2] or load the discretized clamped nonlinear Timoshenko beam from section 7.3 of [2].

Table 2.1: Required variables and variable names to load a *.mat* file into SSMtool.

| | Variable name | Class |
|---|---|---|
| Mass matrix | **M** | $n \times n$ double |
| Damping matrix | **C** | $n \times n$ double |
| Stiffness matrix | **K** | $n \times n$ double |
| Nonlinear force vector | **f** | $n \times 1$ sym |
| Position vector | **x** | $n \times 1$ sym |
| Velocity vector | **xd** | $n \times 1$ sym |
| Conservative constant | conservative | integer or double |
| Scaling constant | scaling | double |

### 2.2 Selecting the spectral subspace

After specifying an autonomous mechanical system, press Analyze to extract the eigenvalues of the system (see Figure 2.4 (a)). Select a two-dimensional spectral subspace $\mathcal{E}$, by selecting a pair of complex conjugate eigenvalues and pressing the Select button, as shown in Figure 2.4 (b). SSMtool will calculate the spectral quotient $\sigma(\mathcal{E})$, which indicates the minimum order of the Taylor expansion needed to be able to the capture the unique SSM. Additionally, SSMtool will check if the outer non-resonance conditions (equation (9) in Ponsioen et al. [2]) are satisfied in order to guarantee that the SSM of interest exists (see Figure 2.4 (b)). In case of a lightly damped spectral subspace $\mathcal{E}$, SSMtool will check for near inner-resonances (see section 5.2 in Ponsioen et al. [2]), up to the order dictated by the spectral quotient.



Lambda = -0.015+1.732i, -0.015-1.732i, -0.045+2.9997i, -0.045-2.9997i

(a) Sigma = ; Outer Resonance : ; Inner Resonance : ; Order = 5; ● Modal ○ Physical ○ Complex; Check for higher-order int. res. ☑

(b) Sigma = 3; Outer Resonance : None; Inner Resonance : Yes; Order = 5; ● Modal ○ Physical ○ Complex; Check for higher-order int. res. ☑

(c) Sigma = 3; Outer Resonance : None; Inner Resonance : Yes; Order = 5; ● Modal ○ Physical ○ Complex; Check for higher-order int. res. ☑

(a) (b) (c)

Fig. 2.4: (a) Analyzing the spectrum of the linearized system at the trivial fixed point. (b) Selecting a two-dimensional spectral subspace $\mathcal{E}$. (c) Specifying the order of the Taylor expansions and choosing the desired output coordinates.

### 2.3 SSM computation

You are now required to specify the order of the Taylor expansion and select if the output coordinates of the SSM should be in modal, physical or complex coordinates. The "Check for higher-order internal resonances" checkbox is there to specify if SSMtool should check for near internal-resonances up to the order of expansion instead of the order dictated by the spectral quotient (see Figure 2.4 (c)).

After pressing the Compute SSM button for the first time, SSMtool will try to start-up the local parallel MATLAB pool and identify the number of cores available, which will be indicated by the message window shown in Figure 2.5 (a). A small portion of the autonomous SSM computations can be run in parallel, however, it is not required to run the computations in parallel.

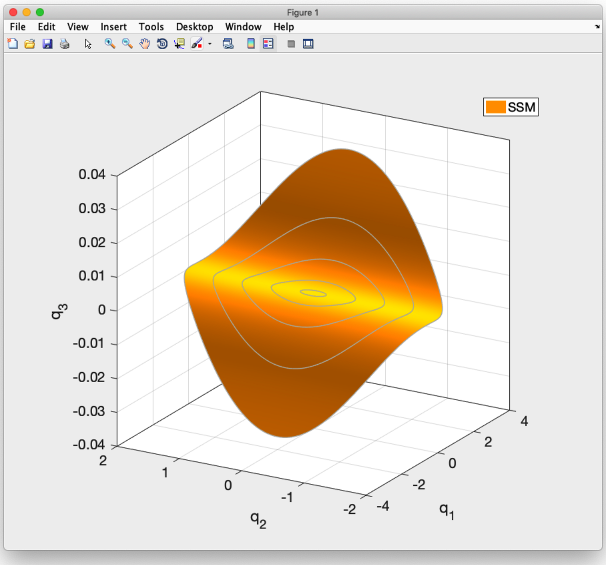
After this stage, SSMtool will proceed with constructing the SSM and the reduced dynamics. The progress is graphically displayed with the loading bar as shown in Figure 2.5 (b). When the SSM computation is complete, SSMtool will output a lower-dimensional projection of the SSM in modal or physical coordinates, depending on the coordinate selection (see Figure 2.6). If you selected the complex coordinates option, no figure is displayed.

Starting parallel pool for the first time and detecting number of available cores.

Computing Spectral Submanifold

(a) (b)

Fig. 2.5: (a) SSMtool will try to start the local parallel MATLAB pool during the first SSM computation and will automatically detect the available cores. If MATLAB fails to start the local pool, the SSM computation will be continued in series. (b) The progress of the SSM computation is shown in the loading bar.

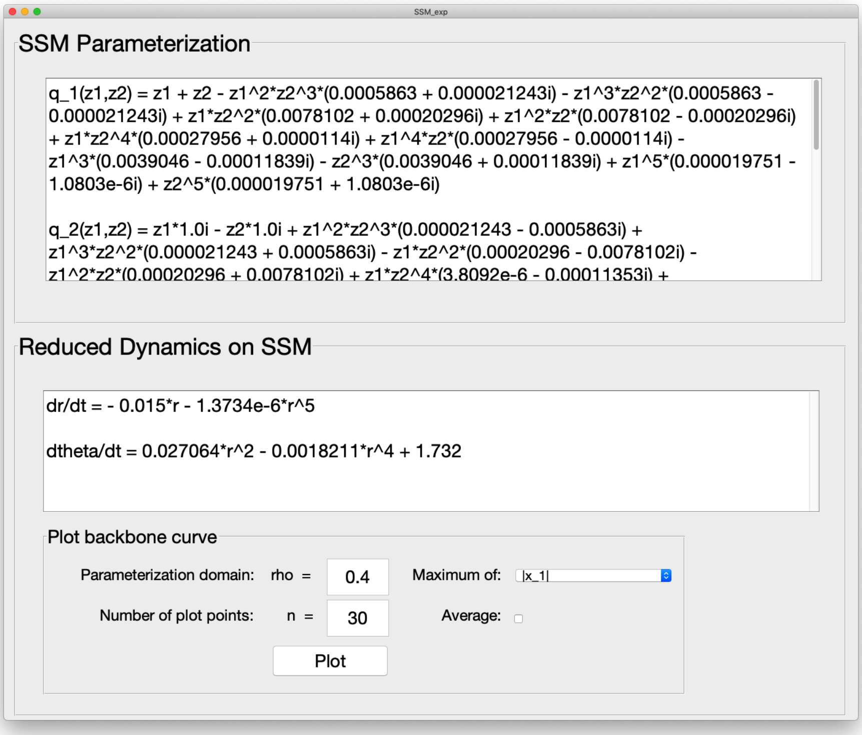


(a)

Fig. 2.6: Lower-dimensional projection of the computed SSM in modal or physical coordinates.

## 2.4 Parameterization/Backbone

The constructed SSM expressions and the reduced dynamics expressions can be inspected by pressing the Show button under the header "Parameterization/Backbone". A new window will be opened, as shown in Figure 2.7. The mapping $\mathbf{W}(\mathbf{z})$ and the reduced dynamics $\mathbf{R}(\mathbf{z})$ being displayed are both stored in a MATLAB function file under the names *SSM_function.m* and *R_function.m*.



```
q_1(z1,z2) = z1 + z2 - z1^2*z2^3*(0.0005863 + 0.000021243i) - z1^3*z2^2*(0.0005863 -
0.000021243i) + z1*z2^2*(0.0078102 + 0.00020296i) + z1^2*z2*(0.0078102 - 0.00020296i)
+ z1*z2^4*(0.00027956 + 0.0000114i) + z1^4*z2*(0.00027956 - 0.0000114i) -
z1^3*(0.0039046 - 0.00011839i) - z2^3*(0.0039046 + 0.00011839i) + z1^5*(0.000019751 -
1.0803e-6i) + z2^5*(0.000019751 + 1.0803e-6i)

q_2(z1,z2) = z1*1.0i - z2*1.0i + z1^2*z2^3*(0.000021243 - 0.0005863i) +
z1^3*z2^2*(0.000021243 + 0.0005863i) - z1*z2^2*(0.00020296 - 0.0078102i) -
z1^2*z2*(0.00020296 + 0.0078102i) + z1*z2^4*(3.8092e-6 - 0.00011353i) +
```

```
dr/dt = - 0.015*r - 1.3734e-6*r^5

dtheta/dt = 0.027064*r^2 - 0.0018211*r^4 + 1.732
```

Parameterization domain: rho = 0.4
Maximum of: |x_1|
Number of plot points: n = 30
Average: ☐

Fig. 2.7: The expressions for the autonomous SSM, $\mathbf{W}(\mathbf{z})$, and the reduced dynamics $\mathbf{R}(\mathbf{z})$ are displayed. The reduced dynamics has automatically been transformed to polar coordinates.

As explained in the introduction of this guide and in section 5.2 of Ponsioen et al. [2], the backbone curve can be extracted from the $\Omega(\rho) = \dot{\theta}(\rho)$ equation of the reduced system. By pressing the Plot button, SSMtool will compute the backbone curve in the physical coordinate system, see Figure 2.8. The output can be specified to be either the maximum absolute value of one of the physical coordinates of the system, or to be an average over all positional coordinates as defined in eq. (48) of [2].

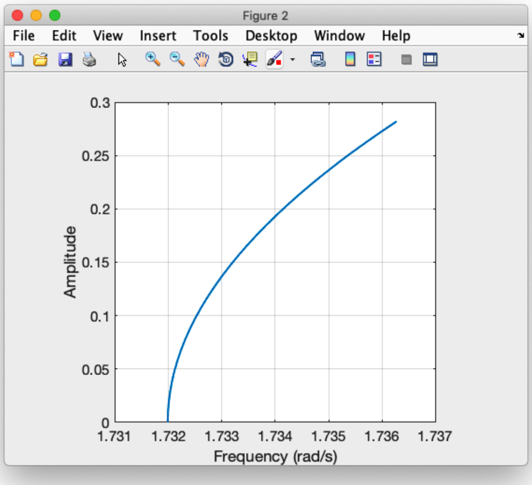
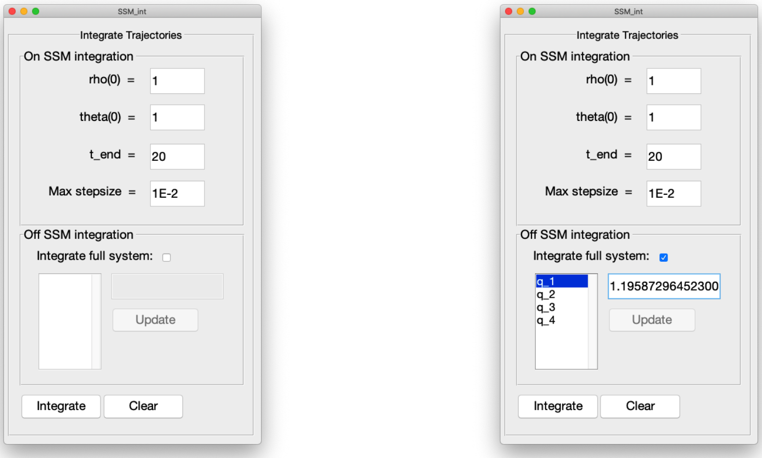


Fig. 2.8: Extracted backbone curve by SSMtool from the $\dot{\theta}(\rho)$ reduced equation.

## 2.5 Integrate trajectories

By pressing the Open Panel button under the "Integrate trajectories" header, a new window will open up, shown in Figure 2.9 (a). Here, you can integrate the reduced system for a given initial position $(\rho_0, \theta_0)$ and specify the total integration time $t_{\text{end}}$ and the maximum stepsize used by the MATLAB ordinary-differential-equation solver. The result is mapped back into the full phase space using the mapping $\mathbf{W}(\mathbf{z})$ as shown in Figure 2.10 (a). To compare the result of the reduced system with the full system, you can integrate a trajectory of the full system with the same initial condition of the reduced system on the SSM. This can be accomplished by checking the "Integrate full system" box and subsequently pressing the Integrate button. The results are shown in Figure 2.10 (b). As an extra feature, it is also possible to integrate the full system from an initial condition that is not necessarily on the SSM by simply modifying the initial position values as shown in Figure 2.9 (b).

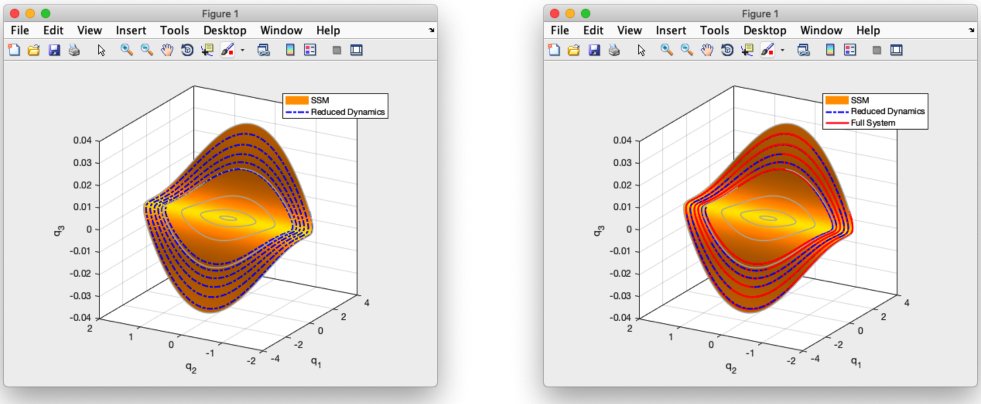


rho(0) = 1
theta(0) = 1
t_end = 20
Max stepsize = 1E-2
Integrate full system: ☐

rho(0) = 1
theta(0) = 1
t_end = 20
Max stepsize = 1E-2
Integrate full system: ☑
q_1 1.19587296452300

(a) (b)

Fig. 2.9: (a) The reduced dynamics can be integrated from 0 to $t_{\text{end}}$, for a given initial position $(\rho_0, \theta_0)$. The maximum integration stepsize can be specified for the MATLAB ordinary-differential-equation solver. (b) Additionally, the full system can be integrated for the same initial position on the SSM, in order to compare the results of the reduced system with the full system. Here it is also possible to integrate the full system from an initial condition that is not necessarily on the SSM.



(a) (b)

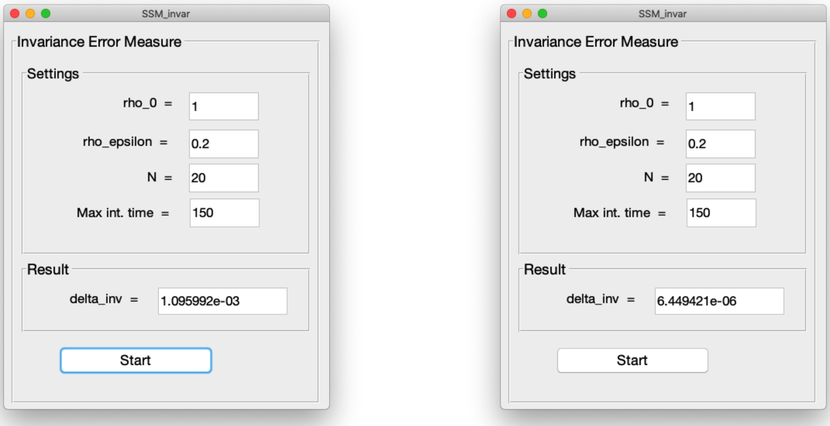
Fig. 2.10: (a) The reduced dynamics is integrated for a given initial position. The resulting reduced trajectory is mapped into the full phase space using the mapping $\mathbf{W}(\mathbf{z})$. (b) To compare the accuracy of the reduced model, the full system can be integrated for the same initial condition on the SSM, but mapped into the full phase space.

## 2.6 Invariance error

In order to quantify the accuracy of the computed invariant manifolds and the reduced dynamics on them, SSMtool can calculate an invariance error measure which compares trajectories of the full system $\mathbf{x}_i$, with trajectories of the reduced system $\tilde{\mathbf{x}}_i$. A total of $N$ trajectories from the full and reduced system are launched from a circle with fixed radius $\rho_0$ from the origin, and integrated until the reduced trajectories cross the inner circle of radius $\rho_\varepsilon < \rho_0$, therefore removing the time dependency. The error measure norm is mathematically formalized in eq. (2.1) (see section 6 of Ponsioen et al. [2]).

$$\delta_{\text{inv}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\text{dist}(i)}{\max\limits_{\theta\in S^1} \|\tilde{\mathbf{x}}(\rho_0,\theta)\|_2}, \qquad \text{dist}(i) = \max \left\| \mathbf{x}_i|_{\rho_0}^{\rho_\epsilon} - \tilde{\mathbf{x}}_i|_{\rho_0}^{\rho_\epsilon} \right\|_2 . \tag{2.1}$$

In Figure 2.11 (a) and (b) we show the computed invariance error measures for an $\mathcal{O}(5)$ and $\mathcal{O}(10)$ approximation of the SSM. As we increase the order of approximation, and therefore increase the accuracy of our approximation, the error measure decreases for a fixed radius $\rho_0$.



rho_0 = 1
rho_epsilon = 0.2
N = 20
Max int. time = 150
delta_inv = 1.095992e-03

rho_0 = 1
rho_epsilon = 0.2
N = 20
Max int. time = 150
delta_inv = 6.449421e-06

(a) (b)

Fig. 2.11: (a) Resulting invariance measure error for an $\mathcal{O}(5)$ approximation of the SSM and the reduced dynamics for a fixed radius $\rho_0 = 1$. (b) Resulting invariance measure error for an $\mathcal{O}(10)$ approximation of the SSM and the reduced dynamics for the same radius $\rho_0 = 1$. As expected the error decreases for an increasing order of approximation.

## 2.7 Plot options

You can change the plot coordinates of the lower-dimensional projection of the SSM by using the "Plot Options" as shown in Figure 2.12. Additionally, the axes limits and the parameterization domain can be modified as desired.

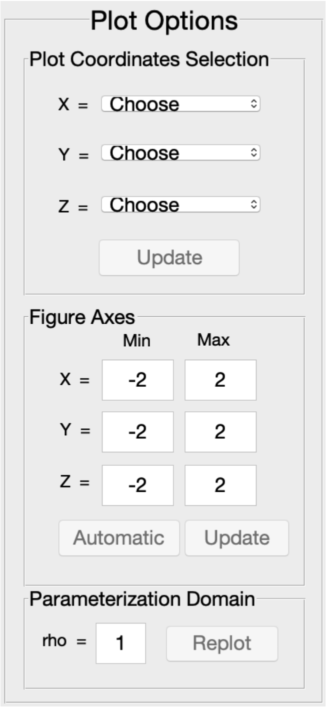


Fig. 2.12: Plot options that can be modified to change to visualization of the lower-dimensional projection of the SSM.

## 3 Final remarks

You are now able to fully take advantage of SSMtool. We would like to note that the future release of SSMtool will be able to handle the time-dependent periodic forcing case as explained in Ponsioen et al. [3]. This will allow you to extract forced response curves corresponding to vibration modes of interest for different forcing amplitudes, in an even more numerically efficient way. Additionally, and due to the use of SSM theory, it will be possible to detect isolated regions in the forced response curves.

Should you discover any bugs or malfunctions in the GUI, please send an email to stenp@ethz.ch.

## References

1. G. Haller and S. Ponsioen. Nonlinear normal modes and spectral submanifolds: existence, uniqueness and use in model reduction. *Nonlinear Dyn.*, 86(3):1493–1534, 2016.
2. S. Ponsioen, T. Pedergnana, and G. Haller. Automated computation of autonomous spectral submanifolds for nonlinear modal analysis. *Journal of Sound and Vibration*, 420:269–295, 2018.
3. S. Ponsioen, T. Pedergnana, and G. Haller. Analytic Prediction of Isolated Forced Response Curves from Spectral Submanifolds. *arXiv preprint arXiv:1812.06664*, 2018.
4. T. Breunung and G. Haller. Explicit backbone curves from spectral submanifolds of forced-damped nonlinear mechanical systems. *Proc. R. Soc. A*, 474(2213):20180083, 2018.
5. S. Jain, P. Tiso, and G. Haller. Exact nonlinear model reduction for a von Kármán beam: slow-fast decomposition and spectral submanifolds. *J. Sound Vib.*, 423:195–211, 2018.