The 5th Joint International Conference on Multibody System Dynamics
June 24 – 28, 2018, Lisboa, Portugal

Determination of Minimal Realizations in Multibody Systems

Bruce Minaker[1] and Francisco González[2]

[1]*Mechanical, Automotive, & Materials Engineering, University of Windsor, Canada, bminaker@uwindsor.ca*
[2]*Department of Mechanical Engineering, McGill University, Montréal, Canada, franglez@cim.mcgill.ca*

ABSTRACT — *In the ongoing search for mathematically efficient methods of predicting the motion of multibody dynamics systems, and presenting the associated results, one of the avenues of continued interest is the linearization of the equations of motion. While linearization can potentially result in reduced fidelity in the model, the benefits in computational speed often make it the pragmatic choice. When representing the governing equations of any linear system, one of the relevant problems is the determination of the mathematically equivalent formulation of the smallest size, and equivalently, the lowest order. This paper discusses the challenges associated with the relevant procedures, and explores the potential of an alternate approach based on modal identification, and the Jordan form of the system matrix. The paper includes example problems demonstrating the application of the proposed method.*

1 Introduction

In the ongoing search for mathematically efficient methods of predicting the motion of multibody dynamics systems, and presenting the associated results, one of the avenues of continued interest is the linearization of the equations of motion. While linearization can potentially result in reduced fidelity in the model, the benefits in computational speed often make it the pragmatic choice. When representing the governing equations of any linear system, one of the relevant problems is the determination of the mathematically equivalent formulation of the smallest size, and equivalently, the lowest order. This paper discusses the challenges associated with the relevant procedures.

This work relates specifically to the development of a multibody dynamics based vehicle motion simulation, based on the equations of motion generator code EoM, developed by the University of Windsor Vehicle Dynamics and Control research group [1], although the results would be equally applicable in any similar implementation. The EoM software is able to generate equations of motion for complex three dimensional multibody systems, but restricts the result to linear equations.

When generating the linearized equations of motion, many authors will choose to present them in the traditional linear second order form shown in Eqn. (1).

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{L}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f} \tag{1}$$

In this form, the matrices **M**, **L**, and **K** represent the mass, damping, and stiffness respectively, **x** is the vector of translational and rotational motions, and **f** is the vector of applied forces and moments. Another useful alternative is to prepare the equations in linear first order, or *state space* form, as shown in Eqn. (2).

$$\begin{Bmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \end{Bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{Bmatrix} \mathbf{x} \\ \mathbf{u} \end{Bmatrix} \tag{2}$$

where vectors **x**, **y**, and **u** represent the states of the system, the outputs, and the inputs, respectively. The state vector may be the translational and rotational displacements and velocities, but there are other possibilities. The

A, **B**, **C**, and **D** matrices are the system, input, output, and feed-through, respectively. The second order form can be easily reduced to state space form with standard mathematical manipulation, such as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{L} \end{bmatrix} \tag{3a}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix} \tag{3b}$$

The **C** and **D** matrices depend on the selection of outputs that one wishes to consider. One possibility is that the outputs would be the vector of position coordinates. In that case:

$$\mathbf{C} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \tag{4a}$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{0} \end{bmatrix} \tag{4b}$$

2 Background

One of the interesting properties of the first order form of the equations is the concept of the *minimal realization*. The input/output relationship between **u** and **y** is expressed using a non-unique set of matrices **A**, **B**, and **C**. An intriguing feature is that not all sets of **A**, **B**, and **C**, or *realizations*, need be of the same dimension. In fact, there is a theoretical lower limit, known as the *McMillan degree*, that denotes the minimum possible dimension of the square **A** matrix. A realization in which the dimension of the system matrix is matching the McMillan degree is known as a *minimal realization*, and the task of computing one, starting from an existing non-minimal realization is a well studied problem in linear systems analysis [2, 3, 4, 5].

2.1 Controllability and observability

In order to fully describe the minimization process, the concepts of *controllability* and *observability* must be introduced. If any of the modes that result from a modal analysis of the equations of motion lies perpendicular to the input forcing vector (a column of the input matrix), then that particular mode cannot be excited by the associated input. If any modes exist that cannot be excited by any of the inputs, then the system is said to be uncontrollable. Similarly, if a mode lies perpendicular to the measurement vector (a row of the output matrix), it cannot be detected. If any modes exist that cannot be detected by any of the outputs, the system is said to be unobservable. In a minimal system, no uncontrollable modes and no unobservable modes are present. The minimization procedure is based on the concept of eliminating any modes that are unobservable or uncontrollable. A system that is both controllable and observable is also minimal.

The controllability and observability of a linear system can be checked by computing the rank of two matrices, known as the *controllability Gramian* $\mathbf{W}_\mathrm{c}$ and the *observability Gramian* $\mathbf{W}_\mathrm{o}$, as defined in Eqns. (5a) and (5b), respectively.

$$\mathbf{W}_\mathrm{c} = \int_0^\infty \mathrm{e}^{\mathbf{A}t}\mathbf{B}\mathbf{B}^\mathrm{T}\mathrm{e}^{\mathbf{A}^\mathrm{T}t}\,\mathrm{d}t \tag{5a}$$

$$\mathbf{W}_\mathrm{o} = \int_0^\infty \mathrm{e}^{\mathbf{A}^\mathrm{T}t}\mathbf{C}^\mathrm{T}\mathbf{C}\mathrm{e}^{\mathbf{A}t}\,\mathrm{d}t \tag{5b}$$

If the controllability Gramian is full rank, then the system is controllable; the equivalent condition holds for the observability. The Gramian matrices are useful, as the rank of the matrix product of the Gramians is the McMillan degree, and a basis of this product is suitable to form a minimal realization. Often, however, the calculation of the Gramians can be numerically challenging. In the event that the system is stable (i.e., all the eigenvalues of the **A**

matrix lie in the left hand of the complex plane) then one popular technique to determine the Gramians is through a numerical solution of the Lyapunov equation, shown in Eqns. (6).

$$\mathbf{A}\mathbf{W}_\mathrm{c} + \mathbf{W}_\mathrm{c}\mathbf{A}^\mathrm{T} = -\mathbf{B}\mathbf{B}^\mathrm{T} \tag{6a}$$

$$\mathbf{A}^\mathrm{T}\mathbf{W}_\mathrm{o} + \mathbf{W}_\mathrm{o}\mathbf{A} = -\mathbf{C}^\mathrm{T}\mathbf{C} \tag{6b}$$

Unfortunately, many systems of interest are unstable, and these systems cannot be reduced using this approach, or must use a modified approach where the unstable modes are first separated from the stable modes [6]. This can be challenging in the event that there are modes that are very nearly neutrally stable. An alternative to solving the Lyapunov equation is to replace the Gramians with two matrices known to have identical range spaces. The matrices C and O are defined below.

$$C = \begin{bmatrix}\mathbf{B} & \mathbf{A}\mathbf{B} & \cdots & \mathbf{A}^{n-1}\mathbf{B}\end{bmatrix} \tag{7}$$

The rank of C is equal to the rank of $\mathbf{W}_\mathrm{c}$.

$$O = \begin{bmatrix}\mathbf{C}' & \mathbf{A}'\mathbf{C}' & \cdots & \mathbf{A}'^{n-1}\mathbf{C}'\end{bmatrix}' \tag{8}$$

The rank of O is equal to the rank of $\mathbf{W}_\mathrm{o}$. The McMillan degree then equals the rank of a Hankel matrix $\mathbf{H} = OC$ formed as shown in Eqn. (9), where n is the dimension of the system.

$$\mathbf{H} = \begin{bmatrix} \mathbf{C}\mathbf{B} & \mathbf{C}\mathbf{A}\mathbf{B} & \cdots & \mathbf{C}\mathbf{A}^{n-1}\mathbf{B} \\ \mathbf{C}\mathbf{A}\mathbf{B} & \mathbf{C}\mathbf{A}^2\mathbf{B} & \cdots & \\ \vdots & \vdots & \ddots & \\ \mathbf{C}\mathbf{A}^{n-1}\mathbf{B} & & & \mathbf{C}\mathbf{A}^{2n-2}\mathbf{B} \end{bmatrix} \tag{9}$$

One can see that in the event of a large system, the calculation involves raising a matrix to large powers, which can quickly lead to round-off errors, which in turn leads to errors in calculation of the rank. This round-off error can be avoided through the use of Arnoldi iteration, a process through which each additional multiplication by the $\mathbf{A}$ matrix is followed by an orthogonolization and renormalization, similar to the well known Gram-Schmidt procedure.

2.2 Eigensystem realization algorithm

The eigensystem realization (ER) algorithm is described by Juang and Pappa in [7] and is a commonly used method of producing a minimal realization. It begins with the singular value decomposition of a Hankel matrix similar to the one shown in Eqn. (9). One advantage of the ER algorithm is that the full Hankel matrix need not be evaluated; once the dimension is sufficiently large that the rank equals the McMillan degree, no further rows/columns are needed. Unfortunately, this limit may not be known in advance, so some user interaction may be required.

$$\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}' \tag{10}$$

Here $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, and Σ is a diagonal matrix. The matrix $\mathbf{H}$ is then reproduced by discarding the columns of $\mathbf{U}$ and the rows of $\mathbf{V}'$ that correspond to the zero singular values.

$$\mathbf{H} = \mathbf{U}_n\Sigma_n\mathbf{V}'_n \tag{11}$$

Next the diagonal matrix Σ_n is factored, so that

$$\mathbf{H} = [\mathbf{U}_n\Sigma_n^{1/2}][\Sigma_n^{1/2}\mathbf{V}'_n] \tag{12}$$

The reduced order system matrix is then formed from:

$$\tilde{\mathbf{A}} = \Sigma_n^{1/2} \mathbf{U}_n' O\mathbf{A}C\mathbf{V}_n \Sigma_n^{1/2} \tag{13}$$

while the reduced input and output matrices can be formed as

$$\tilde{\mathbf{B}} = \Sigma_n^{1/2} \mathbf{V}_n' \begin{bmatrix} \mathbf{I}_r \\ \mathbf{0} \end{bmatrix} \tag{14}$$

$$\tilde{\mathbf{C}} = \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \end{bmatrix} \mathbf{U}_n \Sigma_n^{1/2} \tag{15}$$

where $\mathbf{I}_r$ an $\mathbf{I}_m$ are identity matrices of size $r \times r$ and $m \times m$, where r is the number of inputs and m is the number of outputs.

2.3 Rosenbrock method

One of the most popular minimization methods is the Rosenbrock method [2]. It starts with the formation of the matrix $\mathbf{P}$.

$$\mathbf{P} = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{0} \end{array}\right] \tag{16}$$

This $\mathbf{P}$ matrix is then reduced using elementary row operations to the form shown in Eqn. (17).

$$\tilde{\mathbf{P}} = \left[\begin{array}{cc|c} \mathbf{A}_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{B}_2 \\ \hline \mathbf{C}_1 & \mathbf{C}_2 & \mathbf{0} \end{array}\right] = \left[\begin{array}{c|c} \tilde{\mathbf{A}} & \tilde{\mathbf{B}} \\ \hline \tilde{\mathbf{C}} & \mathbf{0} \end{array}\right] \tag{17}$$

Note that $\mathbf{C}_2\mathbf{A}_{22}^k\mathbf{B}_2 = \tilde{\mathbf{C}}\tilde{\mathbf{A}}^k\tilde{\mathbf{B}}$, or in other words, $\mathbf{A}_{22}$,$\mathbf{B}_2$,$\mathbf{C}_2$ is a smaller dimensional realization of $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, $\tilde{\mathbf{C}}$, that is achieved by removal of any uncontrollable states. Recognizing that the elementary row operations can be represented by a multiplication by a matrix $\mathbf{T}$, representing a coordinate transformation, gives:

$$\tilde{\mathbf{A}} = \mathbf{T}'\mathbf{A}\mathbf{T} \tag{18a}$$

$$\tilde{\mathbf{B}} = \mathbf{T}'\mathbf{B} \tag{18b}$$

$$\tilde{\mathbf{C}} = \mathbf{C}\mathbf{T} \tag{18c}$$

The controllable realization can be extracted from the appropriate submatrices of $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{C}}$. The next step in the algorithm is the creation of a second matrix as shown.

$$\mathbf{Q} = \left[\begin{array}{c|c} \mathbf{A}_{22}' & \mathbf{C}_2' \\ \hline \mathbf{B}_2' & \mathbf{0} \end{array}\right] \tag{19}$$

Another set of row operations are perfomed on $\mathbf{Q}$ to produce $\tilde{\mathbf{Q}}$ in a similar manner, which removes the unobservable states, resulting in another transformation matrix $\mathbf{U}$.

$$\mathbf{A}^* = \mathbf{U}'\mathbf{A}_{22}\mathbf{U} \tag{20a}$$

$$\mathbf{B}^* = \mathbf{U}'\mathbf{B}_2 \tag{20b}$$

$$\mathbf{C}^* = \mathbf{C}_2\mathbf{U} \tag{20c}$$

Finally, the minimal realization can be extracted from the appropriate submatrices of $\mathbf{A}^*$, $\mathbf{B}^*$, $\mathbf{C}^*$.

3 Proposed method

This work explores the potential of an alternate approach based on modal identification, and the Jordan form **J** of the system matrix.

3.1 Definition of Jordan form

The Jordan form of the system matrix, shown in Eqn. (21), is similar to the diagonal form that results from an eigen decomposition.

$$\mathbf{J} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T} = \begin{bmatrix} J_1 & & \\ & \ddots & \\ & & J_q \end{bmatrix} \text{ where } \mathbf{J}_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix} \tag{21}$$

The matrix **T** represents a coordinate transformation, and λ_i represents the i^{th} eigenvalue of **A**. The primary distinction of the Jordan form is in the case of repeated eigenvalues, and in particular the case where the eigenvectors fail to form a basis. The Jordan form utilizes the concept of *generalized eigenvectors* to complete the basis.

A diagonal system matrix is useful as it shows directly the contribution of each input to each mode, and the contribution of each mode to each output. Off-diagonal terms in the system matrix couple two state equations, and complicate the assessment of the the coupling of each input to each mode. The amount of off-diagonal coupling in the Jordan form of a general matrix varies, and is dependant on the number of repeated eigenvalues; it requires a distinction between the *algebraic multiplicity* and *geometric multiplicity* of the eigenvalues.

If an eigenvalue appears multiple times as a root of the characteristic equation, its algebraic multiplicity is simply the number of times that it appears. If the eigenvalues are unique, they all have an algebraic multiplicity of one. The geometric multiplicity of the eigenvalue is the maximum number of linearly independent eigenvectors associated with it. The maximum value of the geometric multiplicity of an eigenvalue is of course its algebraic multiplicity, but it may be less.

Note that Eqn. (21) does not describe the effect of complex eigenvalues on the Jordan form. If the eigenvalue is real, it is simply placed on the diagonal as shown. If any complex eigenvalues are found, they are used to form a 2×2 matrix, with the real parts of the eigenvalues on the main diagonal. The imaginary parts arranged such that they fill out the 2×2 block, with the positive imaginary part, as a real (i.e., the i is dropped), in the upper right corner, and the negative imaginary part, also as a real, in the lower left corner. In the case that the system equations represent the equations of motion of a multibody system, the form of the off-diagonal elements can be predicted, and used to simplify the determination of the contribution of each mode. Those modes that are shown to be non-contributing can be eliminated from the equations of motion.

3.2 Calculation of Jordan form

The process of computing the Jordan form begins with an eigenvalue/eigenvector decomposition of the system matrix, i.e., the solution of the vectors **X** such that

$$[\mathbf{A} - \lambda_i \mathbf{I}]\mathbf{X}_i = \mathbf{0} \tag{22}$$

which requires prior solution of the values λ_i such that

$$\det[\mathbf{A} - \lambda_i \mathbf{I}] = 0 \tag{23}$$

Note that the eigenvectors have the property that

$$\mathbf{X}\Lambda\mathbf{X}^{-1} = \mathbf{A} \tag{24}$$

where Λ is the matrix formed by placing the eigenvalues λ_i along the diagonal.

After the decomposition, the Jordan form can be found by analyzing each eigenvalue in turn, to identify the real and complex conjugate eigenvalues. This task is simplified by assuming the eigenvalues are returned with the conjugates in sequential locations in the eigenvalue list, as is typical.

The complex eigenvalues are broken apart and rewritten into the appropriate entries in the new system matrix. The resulting system matrix will be real and tridiagonal, with the first superdiagonal being the negative of the first subdiagonal. At the same time as the eigenvalues are placed in the system matrix, the eigenvectors are modified as well. The real eigenvectors remain unchanged, while the complex conjugate vectors are replaced with a pair of vectors, the first containing the real part of each entry, the second containing the imaginary part (again, as a real number). Consider the example shown in Eqns (25)– (27).

$$\mathbf{A} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -2 \end{bmatrix} \tag{25}$$

$$\Lambda = \begin{bmatrix} -1+\mathrm{i} & 0 & 0 \\ 0 & -1-\mathrm{i} & 0 \\ 0 & 0 & -2 \end{bmatrix} \Rightarrow \mathbf{J} = \begin{bmatrix} -1 & 1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix} \tag{26}$$

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 1 \\ -0.41-0.41\mathrm{i} & -0.41+0.41\mathrm{i} & 0 \\ 0.82 & 0.82 & 0 \end{bmatrix} \Rightarrow \mathbf{T} = \begin{bmatrix} 0 & 0 & 1 \\ -0.41 & -0.41 & 0 \\ 0.82 & 0 & 0 \end{bmatrix} \tag{27}$$

A quick calculation will confirm that $\mathbf{TJT}^{-1} = \mathbf{A}$. Once the complex eigenvalues have been replaced, the repeated roots must be considered. In the event that there is a repeated root, the geometric multiplicity of the root must be considered. The effect will be demonstrated with an example.

Consider a unit point mass, restricted to motion in the plane, with orthogonal coordinates x_1 and x_2. Suppose that the point mass is acted upon by two orthogonal actuators, aligned with the coordinate axes, and that both coordinates are measured. Two cases will be used to illustrate the effect of repeated roots. In the first case, the motion of the point mass will be resisted by a unit damping in both coordinate axes. In the second, no resistance forces act in the x_2 direction, but both a unit stiffness and twice a unit damping force act in the x_1 direction.

If the state vector is chosen as $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dot{x}_1 & \dot{x}_2 \end{bmatrix}'$, then in the first case, the system matrix becomes:

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1/\sqrt{2} & 0 \\ 0 & 1 & 0 & -1/\sqrt{2} \\ 0 & 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1/\sqrt{2} & 0 \\ 0 & 1 & 0 & -1/\sqrt{2} \\ 0 & 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1/\sqrt{2} \end{bmatrix}^{-1} \tag{28}$$

An eigen analysis will show a repeated zero root, and a second repeated root of negative one. Nevertheless, an examination of the eigenvectors will show that they have full rank (four), so in this case, the eigenvector decomposition is equivalent to the Jordan decomposition. In the second case, the system matrix becomes:

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{29}$$

An eigen analysis will show that the second case returns identical eigenvalues, a pair of zeros, and a pair of negative ones. However, in this case, the eigenvectors have a rank of only two.

$$\Lambda = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{30}$$

In this case, in order to create the Jordan form, the unit superdiagonal entries must be added, and the corresponding redundant eigenvectors are replaced with generalized eigenvectors $\mathbf{Y}_i$, defined as

$$[\mathbf{A} - \lambda_i \mathbf{I}]\mathbf{Y}_i = \mathbf{X}_i, \ \mathbf{Y}_i \perp \mathbf{X}_i \tag{31}$$

Evaluation of the expression will confirm the resulting decomposition.

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1/\sqrt{2} & 1/2\sqrt{2} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1/\sqrt{2} & 1/2\sqrt{2} \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1/\sqrt{2} & 1/2\sqrt{2} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1/\sqrt{2} & 1/2\sqrt{2} \\ 0 & 1 & 0 & 0 \end{bmatrix}^{-1} \tag{32}$$

Clearly, although the eigenvalues themselves are identical between the two cases, the multiplicity is not. In the first case, both the roots have both an algebraic and a geometric multiplicity of two. In the second case, both roots retain their algebraic multiplicity, but both fall to a geometric multiplicity of one. As a result, the Jordan form requires the calculation of generalized eigenvectors, and the unit off-diagonal entries to be placed in the system matrix.

In the general case, the size of the individual Jordan blocks depends on the specifics of the repeated root. For example, if a root appears four times, the Jordan form may contain four blocks of size one, or two blocks of size two, or a block of size three and a block of size one, or a block of size two and two blocks of size one. However, in the equations of motion, the maximum size of any Jordan block can be two by two. This is stated without mathematical proof, but relies on the requirement that the system contains only stiffness, damping, and mass terms, i.e., that the equations can be expressed in a traditional linear second order form. This excludes systems that require higher order differential equations, e.g., those with active PID feedback control in place. The only way that a repeated root can appear in the equations of motion is either by symmetry of the physical system itself, or coincidence between two arbitrary modes, or through an exactly critically damped mode. The most common case of repeated roots is anticipated to be repeated zeros when one or more rigid body modes are present in the system.

3.3 Minimization using the Jordan form

Through a careful examination of the off-diagonals of the Jordan form, the amount of inter-modal coupling can be established, and used to determine the potential for each mode to be excited, either by its own input, or through the neighbouring mode in a complex conjugate or a repeated root. Consider the case when the transformed system matrix has the following Jordan form:

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{Bmatrix} = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & -a_{34} & a_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \{u_1\} \tag{33a}$$

$$\{y_1\} = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} \tag{33b}$$

In this example, the free response of the first state x_1 can only be self-excited, i.e., there is no coupling to x_2, x_3, or x_4. The contribution of x_1 to the forced response y_1 due to input u_1 depends on the size of the product b_1c_1. Similarly, x_2 is decoupled, and its contribution to y_1 is only due to the product b_2c_2. If, for example, the magnitude of b_1c_1 was zero, or at least very much smaller than b_2c_2, the the state x_1 can be safely eliminated from the system of equations. The case for state x_3 is more complicated. Its contribution certainly depends on the product b_3c_3, but it can also be excited by state x_4, so the magnitude of the product b_4c_3 should also be considered. Even in the event that c_3 is zero, state x_3 can in turn excite state x_4, which can contribute to the output, so the product b_3c_4 should also be considered before the state x_3 could be judged as non-contributing. In the event that the system matrix contained block diagonal elements larger than 2×2, then the determination of the contribution would become more challenging still. Fortunately, that situation cannot occur in Jordan forms that result from the reduction of second order systems, like those that occur in the dynamics of multibody systems.

Once non-contributing modes are identified and removed from the system, then the final step is to search for repeated modes in which the geometric multiplicity is equal to the algebraic multiplicity. These would typically occur as a result of symmetry in the physical system. The repeated modes can be compacted. Consider the system in Eqn (34).

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ -a_{12} & a_{22} & 0 & 0 \\ 0 & 0 & a_{11} & a_{12} \\ 0 & 0 & -a_{12} & a_{22} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \{u_1\} \tag{34a}$$

$$\{y_1\} = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} \tag{34b}$$

This system is equivalent to the system:

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ -a_{12} & a_{22} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{bmatrix} b_5 \\ b_6 \end{bmatrix} \{u_1\} \tag{35a}$$

$$\{y_1\} = \begin{bmatrix} c_5 & c_6 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} \tag{35b}$$

where:

$$\begin{bmatrix} b_5 \\ b_6 \end{bmatrix} \begin{bmatrix} c_5 & c_6 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \begin{bmatrix} c_1 & c_2 \end{bmatrix} + \begin{bmatrix} b_3 \\ b_4 \end{bmatrix} \begin{bmatrix} c_3 & c_4 \end{bmatrix} \tag{36}$$

Multiple solutions exit for b_5, b_6, c_5, and c_6. One can be isolated by adding the requirement that $|\mathbf{B}| = |\tilde{\mathbf{B}}|$. It is important to note that collapsing repeated roots in systems with multiple inputs or multiple outputs pose more challenge than those with single inputs and single outputs, and has additional requirements. The affected rows of the **B** matrix must be co-linear, as must the columns of the **C** matrix.

3.4 EoM implementation

The equations of motion are prepared in the EoM sofware in an atypical form. The algorithm begins by preparing the equations in first order form, as shown in Eqn. (37). In this form, the constraint forces are ignored, as they will later be eliminated.

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} & -\mathbf{G} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{w}} \\ \dot{\mathbf{u}} \end{Bmatrix} + \begin{bmatrix} \mathbf{V} & -\mathbf{I} & \mathbf{0} \\ \mathbf{K} & \mathbf{L} & -\mathbf{F} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \mathbf{p} \\ \mathbf{w} \\ \mathbf{u} \end{Bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{I} \end{bmatrix} \{\mathbf{u}\} \tag{37}$$

The state vector consists of six global position coordinates for each body gathered into vector **p**, and six body fixed velocity coordinates for each body gathered into vector **w**. The input vector **u** is appended to the state. The first row of the system are the linearized kinematic differential equations. The matrix **V** is determined only by the velocity around which the linearization occurs; if all bodies in the system are at rest at the point of linearization, it goes to zero. The second row of the system are the linearized Newton Euler equations. The matrix **M** represents the mass and inertia terms, the matrix **G** allows the inclusion of systems that have dependency on the rate of input. The matrix **K** is the stiffness matrix, and includes terms that depend on the physical stiffness and the geometry. Additionally, it includes a series of terms known as the tangent stiffness matrix, which depend on the static preload forces carried in the connections. The weights of each of the bodies in the system, plus any additional external forces, are used to compute the preloads in all the constraint and elastic elements, when the system is in equilibrium. These preloads are then used to compute the tangent stiffness terms. The matrix **L** is the damping matrix, and depends primarily on the physical damping coefficients, and the geometry of the system. It also depends on the velocity of the bodies in the equilibrium point around which the equations are linearized, in order to include the centripetal and gyroscopic terms. The matrix **F** represents the sensitivity to the various inputs. The bottom row appends the inputs to the state vector. The system is subject to constraints, as represented in Eqn. (38).

$$\begin{bmatrix} \mathbf{J}_\mathrm{h} & \mathbf{0} & \mathbf{0} \\ -\mathbf{J}_\mathrm{h}\mathbf{V} & \mathbf{J}_\mathrm{h} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_\mathrm{nh} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}} & \mathbf{p} \\ \dot{\mathbf{w}} & \mathbf{w} \\ \dot{\mathbf{u}} & \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{38}$$

The first row of the system are the holonomic constraints, applied to position and rate of change of position. The $\mathbf{J}_\mathrm{h}$ matrix is the constraint Jacobian. The second row of the system are also the holonomic constraints, applied to velocity and rate of change of velocity. Because the equations are represented in unconstrained first order form, the position and velocity are initially independent states. As such, the constraints must be applied to both. In this formulation, there is redundancy between the first entry in the first row and the second entry in the second row, but it does not cause issues in the solution process. The third row of the system are the nonholonomic constraints, applied to velocity and rate of change of velocity. A null space of the constraint matrix is found, and used to reduce the system to a smaller set of coordinates. The result is combined with a set of output equations, also transformed to minimal coordinates, to give the descriptor state space form of the equations.

$$\begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \end{Bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{Bmatrix} \mathbf{x} \\ \mathbf{u} \end{Bmatrix} \tag{39}$$

The vector **x** is the state vector; its contents do not have a direct physical interpretation, but are related to the positions and velocities of the bodies in the system by a coordinate transformation that ensures the constraints are satisfied. The input vector **u** consists of applied forces and torques, and the output vector **y** consists of displacements, velocities, and accelerations of interest. The system matrix **A** contains the damping and stiffness terms, the input matrix **B** relates the applied external forces to the states, the output matrix **C** relates the states to outputs, and the feedthough matrix **D** bypasses the transient effects, and relates the outputs directly to the inputs. The matrix **E** contains the mass terms, and if singular, indicates that the equations are linear differential algebraic equations (DAEs), as opposed to the typical nonsingular case, where they become ordinary differential equations (ODEs). In the event that the behaviour is described by a set of DAEs, they can be reduced to a lower dimensional ODE system in standard state space form, while maintaining an equivalent input/output relationship. The Jordan form algorithm has been implemented in the EoM software tool. However, the calculation of the Jordan form requires first that descriptor state space equations produced by the EoM tool are first reduced to standard state space form.

3.5 Descriptor state space conversion

The first step in the process is the singular value decomposition of the **E** matrix.

$$\mathbf{E} = \mathbf{U}\Sigma\mathbf{V}' \tag{40}$$

where $\mathbf{V}$ and $\mathbf{U}$ are orthonormal, and Σ is diagonal, with the individual entries σ_i recognized as the singular values. Consider the system equation.

$$\mathbf{E}\dot{\mathbf{x}} = \mathbf{U}\Sigma\mathbf{V}'\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{41}$$

Defining $\mathbf{z} = \mathbf{V}'\mathbf{x}$, and multiplying by $\mathbf{Q}'$ gives:

$$\Sigma\dot{\mathbf{z}} = \mathbf{Q}'\mathbf{A}\mathbf{P}\mathbf{z} + \mathbf{Q}'\mathbf{B}\mathbf{u} \tag{42}$$

Defining $\tilde{\mathbf{A}} = \mathbf{Q}'\mathbf{A}\mathbf{P}$ and $\tilde{\mathbf{B}} = \mathbf{Q}'\mathbf{B}$ gives:

$$\begin{bmatrix} \Sigma_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{z}}_1 \\ \dot{\mathbf{z}}_2 \end{Bmatrix} = \begin{bmatrix} \tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{12} \\ \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22} \end{bmatrix} \begin{Bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{Bmatrix} + \begin{bmatrix} \tilde{\mathbf{B}}_1 \\ \tilde{\mathbf{B}}_2 \end{bmatrix} \{\mathbf{u}\} \tag{43}$$

where Σ_n represents the diagonal matrix where the diagonal entries $\sigma_i > 0$. The bottom equation can be separated and solved for $\mathbf{z}_2$.

$$\mathbf{z}_2 = -\tilde{\mathbf{A}}_{22}^{-1}\tilde{\mathbf{A}}_{21}\mathbf{z}_1 - \tilde{\mathbf{A}}_{22}^{-1}\tilde{\mathbf{B}}_2\mathbf{u} \tag{44}$$

This expression for $\mathbf{z}_2$ can then be substituted back into the original system of equations, and eliminated.

$$\Sigma_n\dot{\mathbf{z}}_1 = [\tilde{\mathbf{A}}_{11} - \tilde{\mathbf{A}}_{12}\tilde{\mathbf{A}}_{22}^{-1}\tilde{\mathbf{A}}_{21}]\mathbf{z}_1 + [\tilde{\mathbf{B}}_1 - \tilde{\mathbf{A}}_{12}\tilde{\mathbf{A}}_{22}^{-1}\tilde{\mathbf{B}}_2]\mathbf{u} \tag{45}$$

The result is that all infinite eigenvalues that existed in the original ($\mathbf{E}$,$\mathbf{A}$) pair are removed, and the system can then be written in standard state space form. The leading Σ_n is by definition diagonal with non-zero entries, and so can be easily inverted and pre-multiplied to complete the transformation.

$$\dot{\mathbf{z}}_1 = \Sigma_n^{-1}[\tilde{\mathbf{A}}_{11} - \tilde{\mathbf{A}}_{12}\tilde{\mathbf{A}}_{22}^{-1}\tilde{\mathbf{A}}_{21}]\mathbf{z}_1 + \Sigma_n^{-1}[\tilde{\mathbf{B}}_1 - \tilde{\mathbf{A}}_{12}\tilde{\mathbf{A}}_{22}^{-1}\tilde{\mathbf{B}}_2]\mathbf{u} \tag{46}$$

The output equation must also be modified to accomodate the new state vector.

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} = \mathbf{C}\mathbf{P}\mathbf{z} + \mathbf{D}\mathbf{u} = \tilde{\mathbf{C}}\mathbf{z} + \mathbf{D}\mathbf{u} \tag{47}$$

Once the equations have been cast into standard first order form, a minimal realization can be found, to remove any states that do not contribute to the input output

4 Examples

The Jordan form minimization method is evaluated using three example problems. The resulting minimal realizations are generated and compared against those produced by the Rosenbrock method, and the ER algorithm. The example problems are the 10-DOF example system from Rahrovani et al. [8], the rigid rider bicycle model described by Meijaard et al. [9], and a multibody quarter car suspension model.

4.1 10-DOF sliding masses

A schematic of the 10-DOF sliding mass example is shown in Figure 1. As the name implies, it consists of ten sliding point masses that are separated into two groups of five, and connected with a series of identical springs. The sliders are taken to have unit mass, and the springs to have unit stiffness. The two groups are coupled by an actuator that pushes against slider number 4 and number 7, and the output of the system is a measurement of the distance between slider number 2 and number 9. The eigenvalues of the equation of motion of the sliding mass system are given in Table 1, and the frequency response is shown in Figure 2.

One interesting feature of the model is that despite the 10 DOF, and corresponding 20×20 system matrix, the McMillan degree is only 8. Also noteworthy is that there are two distinct reasons for elimination of some states from the model. An examination of the eigenvalues shows many repeated roots, as one might expect from the

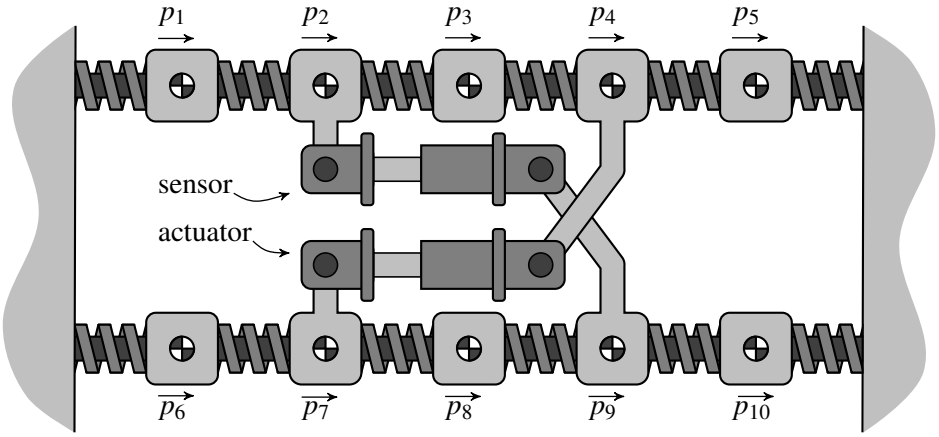

Fig. 1: Example 10 degree of freedom sliding mass system. Despite the 10 degrees of freedom, the McMillan degree of the system is only 8.

Tab. 1: Eigenvalues of the sample 10-dof system. Note that all real parts are zero, and that imaginary parts appear as both positive and negative. The negative imaginary parts are omitted for brevity.

No.	Imaginary [rad/s]	Frequency (ω_n) [Hz]
1	$1.9318516526 \times 10^{0}$	$3.0746373983 \times 10^{-1}$
2	$1.7320508076 \times 10^{0}$	$2.7566444771 \times 10^{-1}$
3	$1.4142135624 \times 10^{0}$	$2.2507907904 \times 10^{-1}$
4	$1.0000000000 \times 10^{0}$	$1.5915494309 \times 10^{-1}$
5	$5.1763809021 \times 10^{-1}$	$8.2384660790 \times 10^{-2}$
6	$1.9318516526 \times 10^{0}$	$3.0746373983 \times 10^{-1}$
7	$5.1763809021 \times 10^{-1}$	$8.2384660790 \times 10^{-2}$
8	$1.4142135624 \times 10^{0}$	$2.2507907904 \times 10^{-1}$
9	$1.7320508076 \times 10^{0}$	$2.7566444771 \times 10^{-1}$
10	$1.0000000000 \times 10^{0}$	$1.5915494309 \times 10^{-1}$

symmetry of the problem. An examination of the mode shapes shows that four modes ($\lambda = \pm 1.4142135624$i) have nodes at sliders 2 and 4, and at 7 and 9. As a result, none of the those four modes contribute to the output of the system, or are excited by the input to the system. All four can be safely eliminated from the system of equations. These modes can be identified using the Jordan form algorithm as detailed above. However, this is sufficient to reduce the system only to 16×16. The resulting system contains eight conjugate pairs, but again, symmetry requires that there are only four distinct pairs. After eliminating the first four modes, the algorithm searches for those remaining modes that have identical eigenvalues, and combines them.

All three algorithms listed in the paper are capable of successfully reducing the 20×20 system to 8×8. Some distinct patterns are notable. The Jordan form algorithm gives a sparse predictable form to the system matrix; of the 64 entries, only 4 unique values appear, and their negatives, along with 56 values of zero. The disparity in the magnitudes of the individual entries depends only on the eigenvalues of the system. The Jordan minimal realization is shown in Eqn. (48). The ER algorithm results in a system matrix that is relatively sparse, but the pattern is not predictable. However, the ER algorithm gives a 'balanced' realization, where the input and output matrices show a clear pattern. Each state is equally controllable and observable. The ER minimal realization is shown in Eqn. (49). The Rosenbrock method also results in a rather sparse system matrix, with only 5 unique entries, but with each appearing multiple times. Again, no obvious pattern exists. The algorithm appears to separate the states into those that are acted upon by the inputs, and those that are measured by the outputs. As a consequence, there are many zero entries that appear in the input and output matrices. The Rosenbrock minimal realization is shown in Eqn. (50).

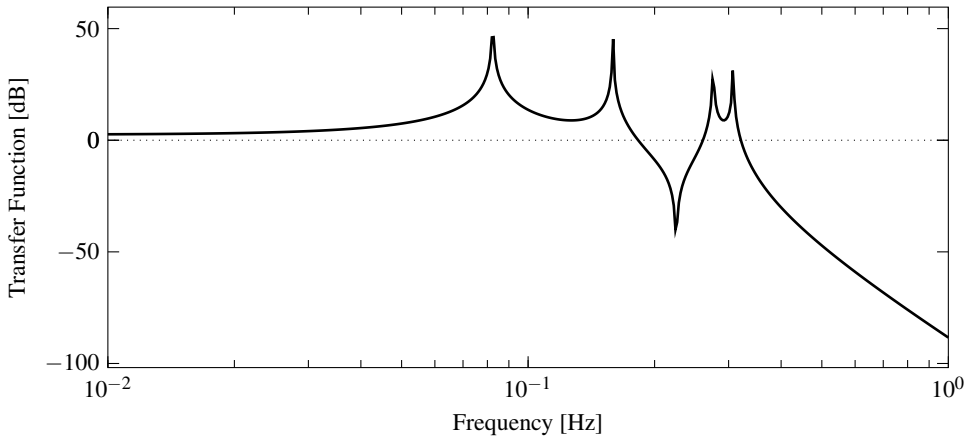

Fig. 2: Frequency response of the 10 DOF slider system. Note that the mode shape at the natural frequency $\omega_n = 0.225$ Hz results in no extension of the sensor, so it appears as a zero in the frequency reponse. The four remaining frequencies (0.307 Hz, 0.276 Hz, 0.159 Hz, 0.0823 Hz) can be clearly identified.

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \left[\begin{array}{cccccccc|c} 0.000\times10^{0} & 1.932\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 7.962\times10^{-1} \\ -1.932\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} \\ 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 5.176\times10^{-1} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} \\ 0.000\times10^{0} & 0.000\times10^{0} & -5.176\times10^{-1} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 1.538\times10^{0} \\ 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 1.732\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 8.165\times10^{-1} \\ 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & -1.732\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} \\ 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 1.000\times10^{0} & 1.000\times10^{0} \\ 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & -1.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} \\ \hline 0.000\times10^{0} & 3.251\times10^{-1} & -6.280\times10^{-1} & 0.000\times10^{0} & 0.000\times10^{0} & -3.536\times10^{-1} & 0.000\times10^{0} & -5.000\times10^{-1} & 0.000\times10^{0} \end{array}\right] \tag{48}$$

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \left[\begin{array}{cccccccc|c} 0.000\times10^{0} & 1.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} \\ -4.018\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & -5.145\times10^{-1} & -3.497\times10^{-2} & -3.818\times10^{-2} & 1.801\times10^{-1} & 2.486\times10^{-2} & -3.743\times10^{-1} \\ -5.145\times10^{-1} & 0.000\times10^{0} & 0.000\times10^{0} & 2.343\times10^{0} & 1.094\times10^{-1} & 1.195\times10^{-1} & -6.791\times10^{-1} & -9.374\times10^{-2} & 5.708\times10^{-1} \\ 0.000\times10^{0} & 0.000\times10^{0} & -1.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} \\ -3.818\times10^{-2} & 0.000\times10^{0} & 0.000\times10^{0} & 1.195\times10^{-1} & 6.913\times10^{-1} & -2.452\times10^{-1} & -7.644\times10^{-1} & -1.055\times10^{-1} & 3.186\times10^{-1} \\ 3.497\times10^{-2} & 0.000\times10^{0} & 0.000\times10^{0} & -1.094\times10^{-1} & 3.668\times10^{-1} & -6.913\times10^{-1} & 7.001\times10^{-1} & 9.664\times10^{-2} & -2.918\times10^{-1} \\ 2.486\times10^{-2} & 0.000\times10^{0} & 0.000\times10^{0} & -9.374\times10^{-2} & -9.664\times10^{-2} & -1.055\times10^{-1} & 1.391\times10^{-1} & 1.019\times10^{0} & 4.246\times10^{-3} \\ -1.801\times10^{-1} & 0.000\times10^{0} & 0.000\times10^{0} & 6.791\times10^{-1} & 7.001\times10^{-1} & 7.644\times10^{-1} & -2.008\times10^{0} & -1.391\times10^{-1} & -3.076\times10^{-2} \\ \hline 3.743\times10^{-1} & 0.000\times10^{0} & 0.000\times10^{0} & -5.708\times10^{-1} & -2.918\times10^{-1} & -3.186\times10^{-1} & -3.076\times10^{-2} & -4.246\times10^{-3} & 0.000\times10^{0} \end{array}\right] \tag{49}$$

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \left[\begin{array}{cccccccc|c} 0.000\times10^{0} & -2.000\times10^{0} & 0.000\times10^{0} & 1.225\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} \\ 1.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} \\ 0.000\times10^{0} & 1.225\times10^{0} & 0.000\times10^{0} & -2.000\times10^{0} & 0.000\times10^{0} & 7.071\times10^{-1} & 0.000\times10^{0} & 0.000\times10^{0} & -1.414\times10^{0} \\ 0.000\times10^{0} & 0.000\times10^{0} & 1.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} \\ 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 7.071\times10^{-1} & 0.000\times10^{0} & -2.000\times10^{0} & 0.000\times10^{0} & 1.414\times10^{0} & 0.000\times10^{0} \\ 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 1.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} \\ 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 1.414\times10^{0} & 0.000\times10^{0} & -2.000\times10^{0} & 0.000\times10^{0} \\ 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 1.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} \\ \hline 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 1.414\times10^{0} & 0.000\times10^{0} \end{array}\right] \tag{50}$$

4.2 Rigid rider bicycle

The bicycle benchmark is based on the paper by Meijaard et al. [9]. The bicycle is modeled as four rigid bodies: the frame and rider treated as one body, the handlebar and fork assembly, the front wheel, and the rear wheel. The fork assembly is attached to the frame with a revolute joint representing the steering head bearing. The location of the steering head bearing is not given, but it must lie on the steer axis, which is defined by the axis tilt, and the trail (the distance from the front wheel ground contact to the point where the steer axis intersects the ground plane, with positive trail indicating the intersection lies in front of the wheel contact). Each wheel is also attached with a revolute joint at its centre, the front wheel to the fork assembly, and the rear wheel to the frame. Friction is neglected in all three joints. The wheels are assumed to be perfectly round, uniform, and 'knife edged'. The bottom of each wheel contacts the ground on a horizontal plane. The contact is treated as a rolling constraint, i.e., the lowest point on the wheel has zero velocity, and must be in the ground plane. The vertical and longitudinal motion of the contact point is treated as holonomic, i.e., the constraints are written in terms of position and orientation. The lateral motion of the tire contact point is treated as nonholonomic, i.e., the constraint is written only in terms of velocities. The contact point may displace laterally, but only when the wheel steer history allows it. As a result, the bicycle has two degrees of freedom, in steer angle δ and lean angle ϕ, that have dynamic responses. In addition, it has neutrally stable modes in longitudinal, lateral and yaw motions. The equations of motion are linearized around a fixed forward speed in the range from 0 to 10 m/s. The properties of the system are given in Table 2. Note that the sign convention used is the SAE standard, with positive z pointing downward, and that the rear wheel contact point is taken at the origin. The eigenvalues are plotted as a function of forward speed in Figure 3.

The EoM implementation of the model includes a single input, a torque applied to the handlebar, and reacted against the frame. There are two outputs measured, the roll angle (y_1) and the steer angle (y_2). These are the coordinates chosen by Meijaard et al, and they express the equations of motion in second order form, using only these two coordinates. The corresponding first order system is 4×4, and so one may correctly surmise that the McMillan degree is 4.

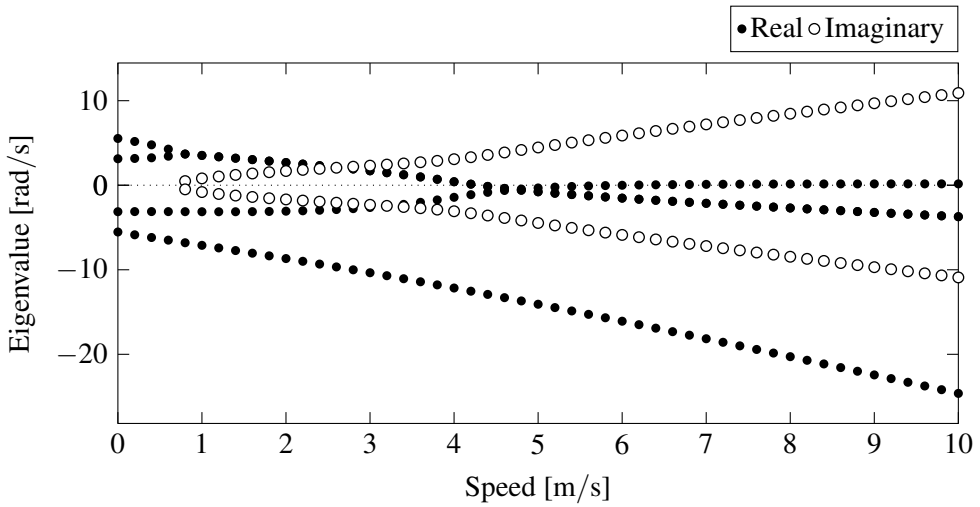

Fig. 3: Eigenvalues vs speed of the rigid rider bicycle model. Note that there are four distinct regions of varying behaviour. At low speed the behaviour is exponentially unstable. The instability transitions to oscillatory above 1 m/s. Above 4.5 m/s, the bicycle becomes self stable, though still oscillatory. Above 5.5 m/s a slow exponential instability appears.

However, the EoM software generates the equations as a 7×7 descriptor state space system. It automatically includes the input as a state in the formation of the descriptor state space, in order that systems that respond to both the input and the rate of change of input can be accommodated. This input can be eliminated can be eliminated from the state vector, using the method outlined in Section 3.5. The resulting system is 6×6, which is still not minimal. All three algorithms (Jordan, ER, Rosenbrock) can correctly further reduce the system to 4×4, i.e., a minimal realization. The algorithms were evaluated using forward speeds of both $u = 2$ m/s and $u = 8$ m/s. The

Tab. 2: Benchmark bicycle properties

Parameter	Symbol	Value for benchmark
Wheelbase	w	1.02 m
Trail	c	0.08 m
Steer axis tilt ($\pi/2$-head angle)	λ	$\pi/10$ rad (90°-72°)
Front wheel		
Radius	r_F	0.35 m
Mass	m_F	3 kg
Moments of inertia	$(I_{\mathrm{F}xx}, I_{\mathrm{F}yy})$	$(0.1405, 0.28)$ kg m^2
Rear wheel		
Radius	r_R	0.3 m
Mass	m_R	2 kg
Moments of inertia	$(I_{\mathrm{R}xx}, I_{\mathrm{R}yy})$	$(0.0603, 0.12)$ kg m^2
Rear frame assembly		
Centre of mass	$(x_\mathrm{B}, z_\mathrm{B})$	$(0.3, -0.9)$ m
Mass	m_B	85 kg
Moments of inertia	$\begin{bmatrix} I_{\mathrm{B}xx} & 0 & I_{\mathrm{B}xz} \\ 0 & I_{\mathrm{B}yy} & 0 \\ I_{\mathrm{B}xz} & 0 & I_{\mathrm{B}zz} \end{bmatrix}$	$\begin{bmatrix} 9.2 & 0 & 2.4 \\ 0 & 11 & 0 \\ 2.4 & 0 & 2.8 \end{bmatrix}$ kg m^2
Handlebar and fork assembly		
Centre of mass	$(x_\mathrm{H}, z_\mathrm{H})$	$(0.9, -0.7)$ m
Mass	m_H	4 kg
Moments of inertia	$\begin{bmatrix} I_{\mathrm{H}xx} & 0 & I_{\mathrm{H}xz} \\ 0 & I_{\mathrm{H}yy} & 0 \\ I_{\mathrm{H}xz} & 0 & I_{\mathrm{H}zz} \end{bmatrix}$	$\begin{bmatrix} 0.05892 & 0 & -0.00756 \\ 0 & 0.06 & 0 \\ -0.00756 & 0 & 0.00708 \end{bmatrix}$ kg m^2

bicycle is unstable at both speeds, but the resulting minimal realizations showed no significant effects, other than the positive diagonal entries expected in the Jordan form. The results for $u = 2$ m/s are shown below, the results for $u = 8$ m/s are omitted for brevity.

Again, the Jordan form algorithm gives a sparse predictable form to the system matrix; of the 16 entries, only 4 unique values appear, with one repeated, and the negative of another also appearing. One notable feature is the difference in magnitudes in the input and output matrices, with the output matrix entries being generally smaller in magnitude. The Jordan minimal realization is shown in Eqn. (51). The sparsity of the ER algorithm is lost in the second example system; it now generates a system matrix that is full and unpredictable. Also, the balancing property is not apparent due to the multiple outputs. The ER minimal realization is shown in Eqn. (52). The Rosenbrock algorithm also results in a full system matrix with no obvious pattern, but it appears to retain the separation of the states into those that are acted upon by the inputs, and those that are measured by the outputs. The Rosenbrock minimal realization is shown in Eqn. (53).

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \left[\begin{array}{cccc|c} -8.674\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & -4.245\times10^{0} \\ 0.000\times10^{0} & -3.072\times10^{0} & 0.000\times10^{0} & 0.000\times10^{0} & -2.004\times10^{-1} \\ 0.000\times10^{0} & 0.000\times10^{0} & 2.682\times10^{0} & 1.681\times10^{0} & -3.147\times10^{0} \\ 0.000\times10^{0} & 0.000\times10^{0} & -1.681\times10^{0} & 2.682\times10^{0} & 1.994\times10^{0} \\ \hline 2.255\times10^{-3} & 1.256\times10^{-1} & -4.016\times10^{-2} & -4.596\times10^{-2} & 0.000\times10^{0} \\ 9.025\times10^{-2} & 4.196\times10^{-2} & -9.516\times10^{-2} & 4.616\times10^{-2} & 0.000\times10^{0} \end{array}\right] \tag{51}$$

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \left[\begin{array}{cccc|c} -8.651\times10^{0} & 5.316\times10^{-1} & -2.334\times10^{-2} & -7.460\times10^{-3} & -6.116\times10^{-1} \\ 4.098\times10^{-1} & 2.458\times10^{0} & 5.192\times10^{-1} & -4.269\times10^{-1} & 3.191\times10^{-1} \\ -7.248\times10^{-1} & -3.833\times10^{0} & 2.409\times10^{0} & -1.684\times10^{0} & 1.420\times10^{0} \\ 5.798\times10^{-1} & 3.051\times10^{0} & -1.810\times10^{0} & -2.597\times10^{0} & -8.644\times10^{-1} \\ \hline 2.316\times10^{-4} & -3.313\times10^{-1} & 9.912\times10^{-2} & 4.033\times10^{-2} & 0.000\times10^{0} \\ 6.118\times10^{-1} & 1.054\times10^{-1} & 2.059\times10^{-1} & -5.584\times10^{-2} & 0.000\times10^{0} \end{array}\right] \tag{52}$$

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \left[\begin{array}{ccccc} 7.685\times10^{-1} & 2.386\times10^{0} & 2.690\times10^{0} & 1.228\times10^{1} & 1.048\times10^{0} \\ -4.699\times10^{0} & -5.332\times10^{0} & -1.756\times10^{1} & -4.920\times10^{0} & -6.457\times10^{0} \\ 1.789\times10^{-1} & -1.858\times10^{0} & -1.748\times10^{0} & 3.799\times10^{-1} & 0.000\times10^{0} \\ 8.593\times10^{-1} & 3.867\times10^{-1} & 3.216\times10^{-1} & -6.989\times10^{-2} & 0.000\times10^{0} \\ \hline 0.000\times10^{0} & 0.000\times10^{0} & 5.338\times10^{-2} & 4.850\times10^{-1} & 0.000\times10^{0} \\ 0.000\times10^{0} & 0.000\times10^{0} & 3.451\times10^{-1} & -7.501\times10^{-2} & 0.000\times10^{0} \end{array}\right] \tag{53}$$

4.3 Multibody quarter car model

For the third example, a 'quarter car' type model is developed. The model is not based on the traditional approach with two translating lumped masses representing the sprung and unsprung masses, but instead includes a full A-arm-type suspension, with a push-rod and bell-crank mechanism to actuate the suspension spring, as often found in racing applications. The suspension is not based on a particular vehicle; rather, typical values are used for the properties.

The model consists of eight rigid bodies: the chassis, the upper and lower A-arm suspension links, the steering tie-rod, the suspension push-rod, the bell-crank, the upright, and a lumped wheel and hub. The chassis is constrained to allow vertical translation only. The upper and lower A-arms are each fixed to the chassis with two joints. One 'branch' of the A-arm uses a spherical joint, while the other branch uses a sliding spherical joint (as two spherical joints would over constrain the system; to prevent this, one of the joints restricts translation in only two directions). Each of the A-arms is also attached to the upright via a spherical joint.

The bell-crank is connected to the chassis with a revolute joint, allowing rotation around an axis in the $y-z$ plane. The push-rod is connected to the the lower control arm at the lower end and to the bell-crank at the upper end using spherical joints. Additionally, one rotational constraint aligned with the axis of the push-rod is added at the lower end to prevent the rotation of the push-rod around its own axis. The tie-rod is connected in the same fashion, to the chassis at one end, and to the upright at the other. The wheel+hub is connected to the upright with a revolute joint to represent the wheel bearing.

Tables 3 and 4 present the locations of the centres of mass of the suspension components, and their inertia properties, respectively. The cross products of inertia are ignored for all of the components, except the push-rod and tie-rod; these are treated as slender rods, and their inertia properties are calculated by coordinate transformation, based on the locations of the end points. Note that the chassis inertia values are included by default, but have no influence on the resulting equations of motion.

Tab. 3: Body CG Locations and Mass

No.	Name	r_x r_y r_z [m]	Mass [kg]
1	Chassis	0.500, 0.000, 0.300	250.000
2	Upright	0.500, 0.800, 0.300	5.000
3	Wheel+hub	0.500, 0.900, 0.300	10.000
4	Lower A-arm	0.500, 0.600, 0.150	5.000
5	Upper A-arm	0.500, 0.600, 0.400	5.000
6	Bell-crank	0.450, 0.350, 0.450	1.000
7	Damper mount	0.000, 0.300, 0.500	0.000
8	Push-rod	0.500, 0.550, 0.300	1.000
9	Tie-rod	0.340, 0.550, 0.280	1.000
10	Contact patch	0.500, 0.900, 0.000	0.000

The suspension spring and damper are linear. The spring is connected between the bell-crank and the chassis, and is oriented along the x-axis. The damper runs parallel to the spring, and is connected between the bell-crank and a massless dummy body. The dummy body is constrained to the chassis to allow only a single translation, in the direction of the damper. The dummy body is also connected to the chassis by a bushing. The dummy body effectively allows a spring and damper to be connected in series, better approximating the properties of the mount bushing of a physical damper.

Tab. 4: Quarter car model body inertia properties. Inertias are defined as the positive integral over the body, e.g., $I_{xy} = +\int r_x r_y \, dm$.

No.	Name	I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{yz}	I_{zx} [kg m^2]
1	Chassis	1.0000	1.0000	1.0000	0.0000	0.0000	0.0000
2	Upright	0.1000	0.1000	0.1000	0.0000	0.0000	0.0000
3	Wheel+hub	2.0000	4.0000	2.0000	0.0000	0.0000	0.0000
4	Lower A-arm	1.0000	1.0000	2.0000	0.0000	0.0000	0.0000
5	Upper A-arm	1.0000	1.0000	2.0000	0.0000	0.0000	0.0000
6	Bell-crank	0.0500	0.0500	0.0500	0.0000	0.0000	0.0000
7	Damper mount	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
8	Push-rod	0.0108	0.0033	0.0075	0.0000	−0.0050	0.0000
9	Tie-rod	0.0210	0.0028	0.0235	0.0075	0.0017	0.0006
10	Contact patch	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

The flexible properties of the tire are modelled as a linear bushing between the wheel and the ground, with a unidirectional vertical stiffness and damping. The vertical damping in the tire requires that the model responds to both the road vertical displacements, and vertical velocity. Tire damping is often neglected in the typical quarter car model, as it is in practice relatively small, and also because it adds significantly to the complexity of the resulting system of equations.

The tire model also includes a massless dummy body. The dummy body is constrained to the wheel allowing only a single lateral translation, at the contact point between the tire and the ground. The dummy body is restricted by a bushing representing the lateral stiffness of the tire, and is also connected to the ground by a bidirectional horizontal damping. When modelled in this fashion, the tire is equivalent to a traditional linear slip model, with a transient response in the lateral direction to include the relaxation length effect.

Additionally, the tire is treated as a rolling contact, rotating about the lateral y-axis, and the equilibrium normal force at the contact point is used to generate the related tangent stiffness matrix. As a result, the location of the vertical reaction moves along the x-axis with the tire as it rolls, but not laterally as it tips (cambers). This causes a moment around the x-axis in response to camber, but not around the y-axis. The connection information is presented in Tables 5 and 6.

There are three outputs included in the quarter car model, namely, the chassis motion, the relative motion between the sprung and unsprung mass (i.e., the suspension travel), and the relative motion between the unsprung mass and the ground (i.e., the tire compression).

The model is implemented using one input actuator representing the motion of the ground under the tire. As previously mentioned, the presence of tire damping implies that both the input and rate of input are included in the forcing function. To accommodate systems of this type, the EoM algorithm always appends the input vector to the state vector. Doing so allows that the equations can then act on the rates of change of the input as well as the value. When the input rate is not a factor, the dimension of the descriptor system is larger than that of the standard state space by the number of inputs.

The eigenvalues of the quarter car model are shown in Table 7. The corresponding natural frequencies, damping ratios, time constants, and wavelengths of the model are shown in Table 8. The model shows a pair of stable oscillatory modes, with a lower frequency at approximately 1.2 Hz that consists of mostly sprung mass motion, and a higher frequency at approximately 12 Hz that is mostly unsprung mass motion (wheel hop). There are additionally three exponential decay motions corresponding to the first order responses of the damper mount bushing, the lateral deflection of the tire contact patch, and the rotational speed of the wheel around its axis. Finally, there is a zero eigenvalue (a 'rigid body' mode) corresponding to the angular position of the wheel; any angular position can be selected, and there is no tendency of the model to react. This rigid body mode is not observable by any of the three outputs, and can safely be eliminated. The frequency response of the quarter car model can be seen in Figure 4.

The dimension of the descriptor state space generated by EoM in this case is 11×11. There are three infinite eigenvalues, two resulting from the massless dummy bodies in the system, and one from the input added as a state.

Tab. 5: Connection location and directionality, by unit vector

No.	Name	r_x r_y r_z [m]	u_x u_y u_z
1	Damper mount slider	0.000, 0.300, 0.500	1.000, 0.000, 0.000
2	Wheel bearing	0.500, 0.900, 0.300	0.000, 1.000, 0.000
3	Lower ball joint	0.500, 0.800, 0.150	– , – , –
4	Lower A-arm pivot, rear	0.300, 0.100, 0.150	– , – , –
5	Lower A-arm pivot, front	0.700, 0.100, 0.100	1.000, 0.000, 0.000
6	Bell-crank pivot	0.400, 0.400, 0.400	0.000, 0.707, 0.707
7	Upper A-arm pivot, front	0.700, 0.300, 0.350	1.000, 0.000, 0.000
8	Upper A-arm pivot, rear	0.300, 0.300, 0.350	– , – , –
9	Upper ball joint	0.500, 0.750, 0.450	– , – , –
10	Lower push-rod end	0.500, 0.700, 0.200	0.000, −0.832, 0.555
11	Upper push-rod end	0.500, 0.400, 0.400	– , – , –
12	Inner tie-rod end	0.250, 0.300, 0.260	– , – , –
13	Outer tie-rod end	0.430, 0.800, 0.300	0.338, 0.938, 0.075
14	Chassis slider	0.500, 0.000, 0.500	0.000, 0.000, 1.000
15	Contact patch constraint	0.500, 0.900, 0.000	0.000, 1.000, 0.000
16	Damper mount bushing	0.000, 0.300, 0.500	1.000, 0.000, 0.000
17	Tire, vertical	0.500, 0.900, 0.000	0.000, 0.000, 1.000
18	Tire, sidewall	0.500, 0.900, 0.000	0.000, 1.000, 0.000
19	Tire, horizontal	0.500, 0.900, 0.000	0.000, 0.000, 1.000
20	Rear spring/damper end	0.000, 0.300, 0.500	– , – , –
21	Front spring/damper end	0.450, 0.300, 0.500	– , – , –

Tab. 6: Properties of the connecting elements in the quarter car model

No.	Name	Stiffness [N/m]	Damping [Ns/m]
1	Damper mount bushing	20,000	0
2	Tire, vertical	150,000	100
3	Tire, sidewall	135,000	0
4	Tire, horizontal	0	3,000
5	Suspension spring	20,000	0
6	Suspension damper	0	1,000

The conversion from descriptor form to standard state space form eliminates these three states to give an 8×8 system. However, the McMillan degree in this case is only 7. The Jordan and Rosenbrock algorithms were both successful in reducing the system to 7×7. However, the ER algorithm suffered round-off error such that it could not succeed.

One of the challenges of the ER algorithm is to decide on the size of the Hankel matrix. It must be sufficiently larger that its rank can reach the McMillan degree. However, the McMillan degree is not known in advance, and so repeated attempts may be necessary. However, as the Hankel matrix is increased in size, the effect of the exponent in the individual terms becomes more dominant, and round off error can occur as a result. In the author's implementation, once the maximum achievable rank was 5 before round-off error took over.

As in the previous examples, the Jordan form algorithm gives a sparse predictable form to the system matrix; of the 49 entries, only 7 unique values appear, with 2 repeated, and the negatives of 2 others also appearing. The difference in magnitudes in the input and output matrices is even more apparent than in the second example, with the output matrix entries being much smaller in magnitude. The Jordan minimal realization is shown in Eqn. (54). The Rosenbrock algorithm again results in a nearly full system matrix with no obvious pattern, but with a block of zero entries (values smaller in magnitude than 1×10^{-9} are rounded to zero). However, the state matrix contains entries over a span of disparate magnitudes, resulting in uncertainty about the precision required to record it without errors. The realization, shown in Eqn. (55), also appears to retain a number of zero entries in the outputs.

Tab. 7: Eigenvalues of the quarter car model

No.	Real [rad/s]	Imaginary [rad/s]
1	$-2.7124597711 \times 10^{0}$	$7.8900865886 \times 10^{1}$
2	$-2.7124597711 \times 10^{0}$	$-7.8900865886 \times 10^{1}$
3	$-6.7987473020 \times 10^{1}$	$0.0000000000 \times 10^{0}$
4	$-1.7032008218 \times 10^{1}$	$0.0000000000 \times 10^{0}$
5	$-8.7165056701 \times 10^{-1}$	$7.7487824748 \times 10^{0}$
6	$-8.7165056701 \times 10^{-1}$	$-7.7487824748 \times 10^{0}$
7	$-4.4916900938 \times 10^{1}$	$0.0000000000 \times 10^{0}$
8	$0.0000000000 \times 10^{0}$	$0.0000000000 \times 10^{0}$

Tab. 8: Natural frequencies, damping ratios, time constants, and wavelengths of the quarter car model. Note the presence of two oscillatory modes, three exponential decay modes, and a single rigid body mode.

No.	Frequency (ω_n) [Hz]	Damping Ratio (ζ)	Time Constant (τ) [s]	Wavelength (λ) [s]
1	$1.2564881160 \times 10^{1}$	$3.4357776649 \times 10^{-2}$	$3.6866906218 \times 10^{-1}$	$7.9633920827 \times 10^{-2}$
2	$1.2564881160 \times 10^{1}$	$3.4357776649 \times 10^{-2}$	$3.6866906218 \times 10^{-1}$	$7.9633920827 \times 10^{-2}$
3	–	–	$1.4708591974 \times 10^{-2}$	–
4	–	–	$5.8712982475 \times 10^{-2}$	–
5	$1.2410351444 \times 10^{0}$	$1.1178369687 \times 10^{-1}$	$1.1472487231 \times 10^{0}$	$8.1086097430 \times 10^{-1}$
6	$1.2410351444 \times 10^{0}$	$1.1178369687 \times 10^{-1}$	$1.1472487231 \times 10^{0}$	$8.1086097430 \times 10^{-1}$
7	–	–	$2.2263334716 \times 10^{-2}$	–
8	–	–	–	–

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \left[\begin{array}{cccccccc|c} -2.712 \times 10^{0} & 7.890 \times 10^{1} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & -1.579 \times 10^{3} \\ -7.890 \times 10^{1} & -2.712 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & -1.654 \times 10^{3} \\ 0.000 \times 10^{0} & 0.000 \times 10^{0} & -6.799 \times 10^{1} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & -9.613 \times 10^{2} \\ 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & -4.492 \times 10^{1} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 3.721 \times 10^{0} \\ 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & -1.703 \times 10^{1} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 8.293 \times 10^{1} \\ 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & -8.717 \times 10^{-1} & 7.749 \times 10^{0} & 2.042 \times 10^{2} \\ 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & -7.749 \times 10^{0} & -8.717 \times 10^{-1} & -6.275 \times 10^{1} \\ \hline -1.147 \times 10^{-4} & 1.606 \times 10^{-4} & -3.943 \times 10^{-5} & -3.353 \times 10^{-3} & -1.737 \times 10^{-2} & -4.325 \times 10^{-3} & -3.925 \times 10^{-2} & 0.000 \times 10^{0} \\ 2.197 \times 10^{-2} & -1.932 \times 10^{-2} & -2.283 \times 10^{-4} & -1.242 \times 10^{-2} & -2.569 \times 10^{-2} & -4.783 \times 10^{-3} & -3.515 \times 10^{-2} & 0.000 \times 10^{0} \\ -2.209 \times 10^{-2} & 1.948 \times 10^{-2} & 1.889 \times 10^{-4} & 9.069 \times 10^{-3} & 8.320 \times 10^{-3} & 4.580 \times 10^{-4} & -4.105 \times 10^{-3} & -1.000 \times 10^{0} \end{array}\right] \tag{54}$$

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \left[\begin{array}{cccccccc|c} -6.748 \times 10^{1} & 8.061 \times 10^{-1} & -6.080 \times 10^{-1} & -1.194 \times 10^{0} & 1.090 \times 10^{2} & 1.761 \times 10^{-2} & 5.881 \times 10^{-2} & -1.087 \times 10^{2} \\ 8.061 \times 10^{-1} & -4.082 \times 10^{1} & -9.333 \times 10^{0} & 2.207 \times 10^{-2} & -2.064 \times 10^{0} & -3.177 \times 10^{-1} & -1.040 \times 10^{-2} & 2.057 \times 10^{0} \\ -4.251 \times 10^{-1} & -9.338 \times 10^{0} & -2.420 \times 10^{1} & 2.359 \times 10^{-2} & -2.044 \times 10^{0} & -1.424 \times 10^{-1} & -3.694 \times 10^{-3} & 2.043 \times 10^{0} \\ 0.000 \times 10^{0} & 0.000 \times 10^{0} & -2.010 \times 10^{2} & -1.270 \times 10^{-2} & 1.158 \times 10^{0} & -2.935 \times 10^{1} & -8.275 \times 10^{-1} & -1.230 \times 10^{0} \\ 0.000 \times 10^{0} & 0.000 \times 10^{0} & 1.807 \times 10^{2} & -2.755 \times 10^{-1} & -4.591 \times 10^{0} & 1.418 \times 10^{2} & -4.702 \times 10^{2} & -1.494 \times 10^{3} \\ 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 2.366 \times 10^{-1} & -2.315 \times 10^{1} & 3.027 \times 10^{-6} & 8.295 \times 10^{-8} & 2.313 \times 10^{1} \\ 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 9.718 \times 10^{-1} & 5.635 \times 10^{0} & 8.296 \times 10^{-8} & 2.272 \times 10^{-9} & -5.630 \times 10^{0} \\ \hline 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & -7.403 \times 10^{-2} & -3.184 \times 10^{-1} & 0.000 \times 10^{0} \\ 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & -1.510 \times 10^{-1} & -4.139 \times 10^{-3} & 0.000 \times 10^{0} \\ 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 0.000 \times 10^{0} & 7.699 \times 10^{-2} & -3.142 \times 10^{-1} & -1.000 \times 10^{0} \end{array}\right] \tag{55}$$

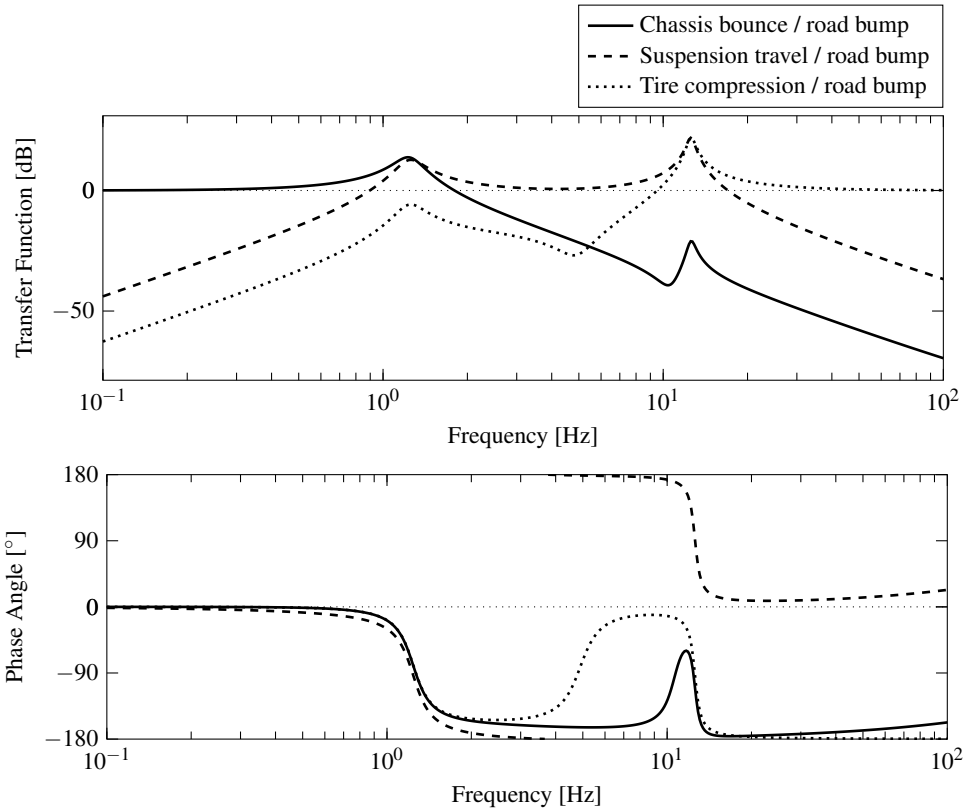

Fig. 4: Frequency response of the quarter car model, showing chassis motion, suspension travel, and tire compression in response to bumps under the wheel

5 Conclusions

The paper demonstrates a method for producing minimal realizations for the equations of motion, based on the Jordan form of the system matrix, and compares the resulting matrices with those produced by two other popular methods. The Jordan form based algorithm was capable of generating minimal realizations that were comparable with the other methods. The sparse system matrix that is a feature of the method may provide some significant memory savings if used in large systems. It also appears the method may have some advantages in precision by avoiding entries with a large range of magnitudes.

The method relies on the fact that the maximum size of the individual Jordan blocks have a maximum size of two by two, as is believed to be the case for systems that represent the equations of motion of a multibody system. This may not be case for generic state space systems.

Although not directly explored in paper, computing efficiency may be a factor for very large systems. Computation of the Jordan form begins with a complete eigen system calculation. This is computationally expensive. However, in the study of many linear systems, the eigenvalues are generally of interest anyway, as they directly describe the system stability. The advantage of immediately reading the stability from the system matrix is an advantage, so the additional computational burden may not be so severe.

Secondly, one of the most common analyses done on any linear system is the calculation of its frequency response function, to determine how the system reacts to sinusoidal inputs over a range of frequencies. The expression is:

$$\mathbf{G}(s) = \mathbf{C}[\mathbf{I}s - \mathbf{A}]^{-1}]\mathbf{B} + \mathbf{D} \tag{56}$$

where $s = \mathrm{i}\omega$, with ω being varied over the range of interest. For each new value of ω, the inverse term (or

equivalent solution) must be computed. This is computationally expensive. If the system matrix is written in the Jordan form, the diagonal or block two by two nature of the system matrix should allow a direct calculation of the inverse matrix for each mode, resulting in a significant reduction in computational effort. This may be significant for systems that require a particularly fine resolution in the frequency range.

Finally, it should be noted that the particular case of the repeated zero root poses a challenge for the Jordan algorithm. It is not uncommon when building large multibody systems to find a result that includes a rigid body mode. These are particularly common if the system happens to be a vehicle. Unfortunately, when finding the eigenvalues of such a system, the expected repeated zero can often appear as a complex conjugate pair, with very small real and imaginary parts, due to round-off error. Unless precautions are taken against it, the Jordan algorithm will detect this as any other conjugate pair, and attempt to separate the real and imaginary components of the eigenvectors. The vector resulting from the real parts will be fine, but the one resulting from the imaginary parts will contain all entries as nearly zero. The inversion of the transformation matrix containing this vector will result in extremely large entries being present. Then, when the contribution of each mode is being computed, these artificially huge entries 'trick' the ranking algorithm into ranking them as very important.

References

[1] B. P. Minaker and R. J. Rieveley, "Automatic generation of the non-holonomic equations of motion for vehicle stability analysis," *Vehicle System Dynamics*, vol. 48, no. 9, pp. 1043–1063, 2010.

[2] H. Rosenbrock, *State-space and multivariable theory*. Studies in Dynamical Systems Series, Wiley Interscience Division, 1970.

[3] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 17–32, 1981.

[4] P. Van Dooren, "The generalized eigenstructure problem in linear system theory," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 111–129, 1981.

[5] B. De Schutter, "Minimal state-space realization in linear system theory: an overview," *Journal of computational and applied mathematics*, vol. 121, no. 1-2, pp. 331–354, 2000.

[6] N. Mirnateghi and E. Mirnateghi, "Model reduction of unstable systems using balanced truncation," in *2013 IEEE 3rd International Conference on System Engineering and Technology*, Aug. 19-20, Shah Alam, Malaysia, 2013.

[7] J.-N. Juang and R. S. Pappa, "An eigensystem realization algorithm for modal parameter identification and model reduction," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 620–627, 1985.

[8] S. Rahrovani, M. K. Vakilzadeh, and T. Abrahamsson, *Topics in Modal Analysis*, vol. 7, ch. On Gramian-Based Techniques for Minimal Realization of Large-Scale Mechanical Systems, pp. 797–805. Springer New York, 2013.

[9] J. Meijaard, J. M. Papadopoulos, A. Ruina, and A. Schwab, "Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 463, no. 2084, pp. 1955–1982, 2007.