

JORDAN FORMS IN VEHICLE DYNAMICS

BP Minaker PhD PEng
 Department of Mechanical, Automotive, & Materials Engineering
 University of Windsor
 Windsor, Ontario, Canada
 bminaker@uwindsor.ca

Abstract— In the ongoing search for mathematically efficient methods of predicting the motion of vehicle and other multi-body 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 solving this system is the most mathematically efficient manner. This paper discusses and explores the potential of an approach based on modal identification, and the Jordan form of the system matrix. The paper includes an example where the method is applied to the classic linear bicycle model.

Keywords- vehicle dynamics; multibody dynamics; equations of motion; eigenvalues; Jordan form

I. INTRODUCTION

In the ongoing search for mathematically efficient methods of predicting the motion of vehicle and other multibody 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 solving this system is the most mathematically efficient manner.

This work relates to the ongoing development of a multi-body 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 as ordinary differential equation (ODEs), many authors will choose to present them in the traditional linear second order form shown in Eqn. (1).

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{L}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f} \quad (1)$$

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

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

where vectors \mathbf{x} , \mathbf{y} , and \mathbf{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 \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{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:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{L} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix} \quad (3a,b)$$

The \mathbf{C} and \mathbf{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:

$$\mathbf{C} = [\mathbf{I} \quad \mathbf{0}], \mathbf{D} = [\mathbf{0}] \quad (4a,b)$$

In fact, the previously mentioned EoM code generates the equations in the *descriptor state space* form, where the state

equation is slightly different, as shown in Eqn. (5). Nevertheless, they can be converted to the standard state space form. The process is somewhat complicated if the \mathbf{E} matrix is singular, as in the case of differential algebraic equations (DAEs), but the result is the same.

$$\begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \end{Bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{Bmatrix} \mathbf{x} \\ \mathbf{u} \end{Bmatrix} \quad (5)$$

II. JORDAN FORMS

This work explores the potential of a solution approach based on modal identification, and the Jordan form \mathbf{J} of the system matrix. The existence of the Jordan form is well known, but it is still the subject of active study, in hopes of utilizing its properties for efficient numerical solutions[2]. The Jordan form has potential to speed up the calculation of the time history solution of an ODE, as it gives many zero entries in the system matrix, allowing the use of sparse matrix algebra.

To illustrate the advantages of sparse matrix computing using the Jordan form, a simple experiment was conducted. A discrete time solution to a set of linear differential equations was computed using a matrix exponential approach. The solution was implemented in the Julia programming language. Two systems were simulated: a small 4×4 system, and a larger 14×14 system. In the smaller system, 9 of 16 entries in the state transition matrix were zero, while in the larger system, 168 of 196 entries were zero. In both cases, the simulations were conducted using standard matrix multiplication, and repeated using sparse arrays.

One might expect that as the system size grows larger, the additional overhead in using sparse matrices would be proportionally less, and as the sparsity of the matrix increased, the advantage of using sparse matrix algebra would increase. Nevertheless, the results showed that the sparse matrix implementation took approximately half the time for both systems. While certainly not exhaustive, this simple experiment illustrates the potential of the Jordan form.

A. Definition of Jordan form

The Jordan form of the system matrix, shown in Eqn. (6), is similar to the diagonal form that results from an eigen decomposition.

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & & \\ & \ddots & \\ & & \mathbf{J}_q \end{bmatrix} \text{ where } \mathbf{J}_i = \begin{bmatrix} s_i & 1 & & \\ & s_i & \ddots & \\ & & \ddots & 1 \\ & & & s_i \end{bmatrix} \quad (6)$$

where:

$$\mathbf{J} = \mathbf{T}^{-1} \mathbf{A} \mathbf{T} \quad (7)$$

The matrix \mathbf{T} represents a coordinate transformation, and s_i represents the i^{th} eigenvalue of \mathbf{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. (6) 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.

B. Calculation of Jordan form

The process of computing the Jordan form begins with an eigenvalue/eigenvector decomposition of the system matrix, using a process such as that given in [3], i.e., the solution of the matrix \mathbf{X} such that:

$$\mathbf{X} \mathbf{A} \mathbf{X}^{-1} = \mathbf{A} \quad (8)$$

where \mathbf{A} is the matrix formed by placing the eigenvalues s_i along the diagonal, and \mathbf{X} is the matrix formed by using the eigenvectors \mathbf{X}_i as columns. The eigenvectors are found by solving Eqn. (9),

$$[\mathbf{I}s - \mathbf{A}] \mathbf{X}_i = \mathbf{0} \quad (9)$$

which requires prior solution of the values s_i from Eqn. (10).

$$\det[\mathbf{I}s - \mathbf{A}] = 0 \quad (10)$$

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 (11)– (14).

$$\mathbf{A} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -2 \end{bmatrix} \quad (11)$$

$$\Lambda = \begin{bmatrix} -1+i & 0 & 0 \\ 0 & -1-i & 0 \\ 0 & 0 & -2 \end{bmatrix} \Rightarrow \mathbf{J} = \begin{bmatrix} -1 & 1 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & -2 \end{bmatrix} \quad (12)$$

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 1 \\ -0.41 - 0.41i & -0.41 + 0.41i & 0 \\ 0.82 & 0.82 & 0 \end{bmatrix} \quad (13)$$

$$\mathbf{T} = \begin{bmatrix} 0 & 0 & 1 \\ -0.41 & -0.41 & 0 \\ 0.82 & 0 & 0 \end{bmatrix} \quad (14)$$

A quick calculation will confirm that $\mathbf{T}\mathbf{J}\mathbf{T}^{-1} = \mathbf{A}$. 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 $\mathbf{x} = [x_1 \ x_2 \ \dot{x}_1 \ \dot{x}_2]'$, then in the first case, the system matrix becomes:

$$\mathbf{A}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad (15)$$

Reduction to Jordan form gives:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & -1/\sqrt{2} & 0 \\ 0 & 1 & 0 & -1/\sqrt{2} \\ 0 & 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1/\sqrt{2} \end{bmatrix} \quad (16)$$

$$\mathbf{J} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad (17)$$

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:

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (18)$$

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.

$$\Lambda = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad (19)$$

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (20)$$

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:

$$[\mathbf{I}s_i - \mathbf{A}]\mathbf{Y}_i = \mathbf{X}_i \quad (21)$$

Evaluation of the expression will confirm the resulting decomposition.

$$\mathbf{T} = \begin{bmatrix} 0 & 0 & 1/\sqrt{2} & 1/2\sqrt{2} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1/\sqrt{2} & 1/2\sqrt{2} \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (22)$$

$$\mathbf{J} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad (23)$$

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 most cases where state space model represents the equations of motion of a multibody system, the maximum size of any Jordan block will be 2×2 . 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.

Often in the case of vehicle dynamics, the equations of motion will be augmented with a set of *kinematic differential equations*, relating the velocities to the rate of change of positions. The relationship may not be a simple differentiation, in the event that multiple coordinate systems are used to describe the motion. In the case that the state space model contains these additional kinematic relationships, the 2×2 limit on the size of the Jordan block will not hold, as will be shown.

III. EXAMPLE

A. Yaw plane model

To explore the applicability of the Jordan form to vehicle dynamics, a simple vehicle model is considered. Also widely known as the *bicycle model*, the yaw plane model has been used extensively for vehicle handling studies since its introduction, and has appeared in many variations in the literature. The nickname is applied because the effect of the width of the vehicle is considered unimportant during a certain aspect of the model development. When the vehicle is pictured in a view from above with the width neglected, its appearance is similar to a bicycle. The model itself has no connection to

bicycle dynamics. Of course, the vehicle width influences the lateral weight transfer experienced while cornering, which in turn affects tire performance, but this is a secondary effect, and is ignored in the yaw plane model.

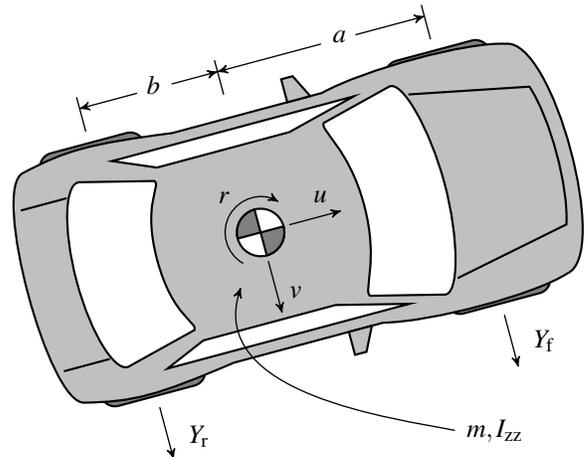


Figure 1. The yaw plane model treats the vehicle as a single rigid body, with motion constrained to the horizontal plane. It is commonly referred to as the *bicycle model* due the assumption about width effects. Figure reproduced from Minaker[4]

The degrees of freedom of the yaw plane model are the lateral velocity v , and the yaw velocity r of the vehicle. The forward speed u is assumed to be under driver control, and held constant, and as a result is treated as a parameter of the model rather than a variable. The motion is assumed to take place on a flat and level road, so all other motions, e.g., roll or heave, are ignored. The constant forward speed is more precisely called a *nonholonomic* constraint; this implies that despite having only two degrees of freedom, three position coordinates are required to fully specify the state of the vehicle: the (x, y) location of the centre of mass, and the heading angle ψ . Despite using only two degrees of freedom, the model provides a great deal of insight into vehicle handling.

The mass m and yaw moment of inertia I_{zz} are the relevant inertial properties of the vehicle, while the distances from the centre of mass to the front and rear axles, denoted a and b respectively, provide the necessary geometric information. The *cornering stiffnesses* of the front and rear tires (lumped) are denoted c_f and c_r , respectively. The lateral forces acting at each of the front and rear axles are Y_f and Y_r , respectively. The steering angle of the front tires, assumed to be the same on the left and right side, is δ_f . A schematic diagram is shown in Figure 1.

The equations of motion of the yaw plane model can be found by summing lateral forces, and moments around a vertical axis. The complete model, including the kinematic differential equations, is presented in Eqn. (24); for a full development, see Minaker[4].

$$\begin{Bmatrix} \dot{y} \\ \dot{\psi} \\ \dot{v} \\ \dot{r} \end{Bmatrix} = \mathbf{A} \begin{Bmatrix} y \\ \psi \\ v \\ r \end{Bmatrix} + \mathbf{B} \{\delta_f\} \quad (24)$$

where:

$$\mathbf{A} = \begin{bmatrix} 0 & u & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -(c_f + c_r)/mu & -(ac_f - bc_r)/mu - u \\ 0 & 0 & -(ac_f - bc_r)/I_{zz}u & -(a^2c_f + b^2c_r)/I_{zz}u \end{bmatrix} \quad (25)$$

and:

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ c_f/m \\ ac_f/I_{zz} \end{bmatrix} \quad (26)$$

B. Result

An eigenvalue analysis of the model above reveals two possibilities. A pair of repeated zeros, and a pair distinct real roots, or a pair of repeated zeros, and a complex conjugate pair. The repeated zeros correspond to the rigid body modes associated with lateral displacement, and yaw. If the vehicle's initial conditions are a simple lateral shift, or an initial yaw angle, both of these conditions will persist indefinitely unless corrected by the driver. The non-zero roots correspond to the well-known ability of the model to predict *understeer*, or *oversteer* behaviour. A simplistic explanation of these terms is that in the case of understeer, as forward speed increases, the vehicle tends toward an increasing cornering radius for a fixed steer angle, where for oversteer, the opposite is true. An understeering vehicle will remain stable over the entire speed range, i.e., the non-zero roots will have negative real parts, where an oversteering vehicle will not display any oscillatory behaviour, but may have unstable response, indicated by a positive real root.

While exploring the dynamics of the yaw plane model, a special case was identified: where $c_f = 0$, and $b = 0$, i.e., a vehicle with only one wheel or axle, located precisely at the centre of mass. While arguably physically impractical, it is notable that a single wheel skateboard style vehicle called the Onewheel is marketed by Future Motion Inc. While the motion of any skateboard style vehicle is highly influenced by the rider, and not well predicted by the bicycle model, considering the single wheel case is mathematically interesting, as it results in three repeated zero roots, and a single distinct non-zero real root. In this case, the system matrix collapses to:

$$\mathbf{A} = \begin{bmatrix} 0 & u & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -c_r/mu & -u \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (27)$$

In this case, only two eigenvectors are identified, as shown in Eqn. (28).

$$\mathbf{X}_1 = \begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix}, \mathbf{X}_2 = \begin{Bmatrix} mu/c_r \\ 0 \\ 1 \\ 0 \end{Bmatrix} \quad (28)$$

The first vector (\mathbf{X}_1) corresponds to the zero root (i.e., $s = 0$), and the second (\mathbf{X}_2) to the root:

$$s = -\frac{c_r}{mu} \quad (29)$$

A solution of the generalized eigenvector using \mathbf{X}_1 , as in Eqn. (21), gives:

$$\mathbf{Y}_1 = \begin{Bmatrix} x_1 \\ x_2 \\ 0 \\ 0 \end{Bmatrix} \quad (30)$$

where x_1 and x_2 are arbitrary. However, this is insufficient to give an independent set of vectors, so a further iteration of the generalized eigenvector is given by:

$$[\mathbf{I}s_i - \mathbf{A}]\mathbf{Z}_i = \mathbf{Y}_i \quad (31)$$

In this case:

$$\mathbf{Z}_1 = \begin{Bmatrix} x_3 \\ x_4 \\ -x_5 mu^2/c_r \\ x_5 \end{Bmatrix} \quad (32)$$

where x_3 , x_4 , and x_5 are arbitrary. The generalized eigenvectors are chosen as:

$$\mathbf{Y}_1 = \begin{Bmatrix} 0 \\ 1/u \\ 0 \\ 0 \end{Bmatrix}, \mathbf{Z}_1 = \begin{Bmatrix} 0 \\ m/c_r \\ mu/c_r \\ -1/u \end{Bmatrix} \quad (33)$$

This gives:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & mu/c_r \\ 0 & 1/u & m/c_r & 0 \\ 0 & 0 & mu/c_r & 1 \\ 0 & 0 & -1/u & 0 \end{bmatrix} \quad (34)$$

$$\mathbf{J} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -c_r/mu \end{bmatrix} \quad (35)$$

Again, a calculation will confirm that $\mathbf{T}\mathbf{J}\mathbf{T}^{-1} = \mathbf{A}$.

The repeated zero root implies that the transient solution will contain terms of the form e^{0t} , te^{0t} , and t^2e^{0t} , or equivalently, constant, linear, and quadratic in time, in addition to the exponential decay term. The solution can be written in terms of the columns of the \mathbf{T} matrix, as given in Eqn. (36), where y_0 , ψ_0 , v_0 , and r_0 are arbitrary constants determined by the initial states, and $s = -c_r/mu$. The result can be confirmed by differentiation and substitution into the state equations.

$$\begin{aligned} \begin{Bmatrix} y \\ \psi \\ v \\ r \end{Bmatrix} &= v_0 \begin{Bmatrix} -1/s \\ 0 \\ 1 \\ 0 \end{Bmatrix} e^{st} + y_0 \begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix} e^{0t} + \frac{r_0 u}{2} \begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix} t^2 e^{0t} \\ &+ \left(\psi_0 - \frac{r_0}{s} \right) u \left\{ \begin{Bmatrix} 0 \\ 1/u \\ 0 \\ 0 \end{Bmatrix} e^{0t} + \begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix} t e^{0t} \right\} \\ &- r_0 u \left\{ \begin{Bmatrix} 0 \\ -1/su \\ -1/s \\ -1/u \end{Bmatrix} e^{0t} + \begin{Bmatrix} 0 \\ 1/u \\ 0 \\ 0 \end{Bmatrix} t e^{0t} \right\} \end{aligned} \quad (36)$$

Alternatively, the solution can be simplified, and broken into individual expressions, as in Eqn. (37). In this form, the expressions become more relatable to the physical system; for example, if there is only one external force acting at the centre of mass, clearly the yaw rate must remain constant, and thus the heading angle must grow linearly with time. (Note that the equations are derived using the SAE standard coordinate system, where the body fixed z-axis points downward, so the heading angle decreases with a positive yaw rate.)

$$y(t) = y_0 + \left(\psi_0 - \frac{r_0}{s} \right) ut - \frac{r_0}{2} ut^2 - \frac{v_0}{s} e^{st} \quad (37a)$$

$$\psi(t) = \psi_0 - r_0 t \quad (37b)$$

$$v(t) = \frac{r_0 u}{s} + v_0 e^{st} \quad (37c)$$

$$r(t) = r_0 \quad (37d)$$

Similarly, if the forward speed and yaw rate are both held constant, there will be a steady component to lateral acceleration.

The force generated due to the slip in the tire should produce this force in the steady state. Finally, the lateral offset shows all three of the terms due to the zero root: a constant initial offset, a linearly increasing term due to heading angle, and a parabolic increase due to steady yaw rate.

IV. CONCLUSIONS

The paper explores the application of the Jordan forms to vehicle dynamics problems. It is shown by example that augmenting the equations of motion with the kinematic differential equations, as would be common in a vehicle dynamics context, results in a system with a Jordan block larger than the 2×2 size that would normally be expected in a multibody vibration problem.

This result is somewhat unfortunate, as the algorithm for computing the Jordan form in the general case could be simplified significantly with foreknowledge that the largest block expected was 2×2 . Also, when computing minimal realizations (the smallest equivalent system matrix), one potential savings is the removal of repeated roots. The calculation of the equivalent system is quite straightforward if the blocks are only 2×2 . Accounting for the possibility of blocks of size three (or potentially even larger?) imposes an extra layer of complexity.

The result leaves many unanswered questions, as to the most effective way to compute the Jordan form, as clearly the algorithm must be capable of accommodating triply repeating roots. Nevertheless, the sparse nature of the Jordan form suggests that it remains a promising candidate for efficient numerical solution of the equations of motion, and further research into the area is warranted.

REFERENCES

- [1] B. Minaker and R. Rieveley, "Automatic generation of the nonholonomic equations of motion for vehicle stability analysis", *Vehicle Systems Dynamics*, vol. 48 (9), 2010, pp. 1043–1063
- [2] K.N. Pirbazari and M. Azari, "The order of minimal realization of Jordan canonical form systems", *Boletim da Sociedade Paranaense de Matemática* 36, no. 3, 2018, pp. 81-88
- [3] E. Davison, W. Gesing, and S. Wang, "An algorithm for obtaining the minimal realization of a linear time-invariant system and determining if a system is stabilizable-detectable", *IEEE Transactions on Automatic Control* 23, no. 6, 1978, pp. 1048–1054
- [4] B. Minaker, "Fundamentals of Vehicle Dynamics and Modelling: A Textbook for Engineers with Illustrations and Examples", Wiley, 2019