MODELLING, IMPLEMENTATION AND ANALYSIS OF THE CARVALLO-WHIPPLE BICYCLE MODEL IN MSC ADAMS

[1]THIHA TUN, [2]LUTZ ROTHENBUSCH, [3]PHILIPP INGENLATH, [4]ALEXANDER BREZING, [5]BURKHARD CORVES

[1,4]The Sirindhorn International Thai-German Graduate School of Engineering, King Mongkut's University of Technology North Bangkok, Thailand
[2,3,5]IGMR RWTH Aachen University
E-mail: [1]thiha.t-mesd2016@tggs.kmutnb.ac.th, [2]rothenbusch@igmr.rwth-aachen.de

Abstract - Despite its age, the Carvallo-Whipple bicycle model is still the benchmark model for bicycle dynamics. To study the effects different bicycle designs have on drivability or stability, it is beneficial to build a computer simulation of the Carvallo-Whipple bicycle model. Thus, this paper presents the implementation of the model in the general purpose simulation software Matlab and the multibody simulation suite MSC Adams. To compare both models the frequency domain behavior and as an extension of the state of the art, the time domain behavior of the MSC Adams model is evaluated. Since it cannot be directly computed by MSC Adams, the frequency domain behavior is computed by time integration and subsequent usage of Matlab's system identification toolbox. Within a certain range of velocities the multibody Adams model is able to reproduce the time domain behavior of the Matlab model.

Index Terms - Bicycle Dynamics, Bicycle Stability, Multibody Modelling

I. INTRODUCTION

Developing and evaluating bicycle models to analyze the bicycle's behavior has been a field of continuous study for more than a century[3]. In 1899, the formal proof that bicycles can demonstrate self-stability has been delivered by Carvallo and Whipple who developed the eponymous bicycle model. Their work laid the foundation for the field of bicycle dynamics, meaning the study of the motion of a bicycle with forces acting on it [1]. The study of bicycle dynamics is directly concerned with drivability and maneuverability of bicycles[3]. For the drivability of a bicycle self-stability plays an important role[5]. Irrespective of the bicycle moving straight or turning, a bicycle's self-stability will help the driver to keep upright[8]. To explore the influence of the different parameters of the bicycle, it is advantageous to implement the Carvallo-Whipple model in simulation software. There is a plethora of choices for such software, and different choices enable the exploration of different aspects of the dynamic behavior. In this paper, two of those aspects of the Carvallo-Whipple model are explored, the behavior in the time domain and the behavior in the frequency domain.

The two simulation suites utilized in this paper are Matlab and the multi-body simulation software MSC Adams. They offer two different perspectives and avenues of exploration on the Carvallo-Whipple model: The mathematical, abstract view of Matlab reduces the model into a set of equations with inputs and outputs and is well suited for direct analysis of the system in both time and frequency domain. Conversely, MSC Adams provides a much more intuitive, direct modelling approach. Also, extending the model e.g. by parameters such as cornering stiffness and the contact properties of the tires, as has been shown in[2], or even flexible body systems is a much less daunting task in MSC Adams than it is in Matlab. On the other hand, in MSC Adams the frequency domain behavior cannot be directly computed. Instead, a detour is necessary: The time domain behavior of the bicycle can be simulated over a certain timeframe and for a given speed, and then the frequency domain behavior can be ascertained via system identification in Matlab. Thus, this paper shows how the time domain behavior of the Carvallo-Whipple model in both Matlab and MSC Adams is computed. Based on these results, the frequency domain behavior is derived for the MSC Adams model and compared to the Matlab results, which can be directly computed.

II. THE CARVALLO-WHIPPLE MODEL

The Carvallo-Whipple bicycle model encompasses four connected rigid bodies: (a) the front frame containing the handle and the front fork, (b) the rear main frame, (c) the front wheel and (d) the rear wheel (cf. Figure 1). The front and the rear frames are connected with a revolute joint that allows the front frame to rotate about the steering axis relative to the rear frame by the steering angle (δ). The rider's mass properties are included in the rear main frame. Therefore, the rider is rigidly connected to the bicycle's rear frame and is riding the bike with hands off the handlebar.

As an additional condition, the wheels are considered to be rolling without slipping[5]. The lateral motion of the main frame is described with two degrees of freedom, the rolling angle (φ) and the yaw angle (ψ) at the rear main frame. Because of the two contact points between the bicycle and the ground it is an inherently unstable system, akin to an inverted pendulum. Just like an inverted pendulum, the bicycle

can be stabilized by moving the support point to the side it is falling towards[8].

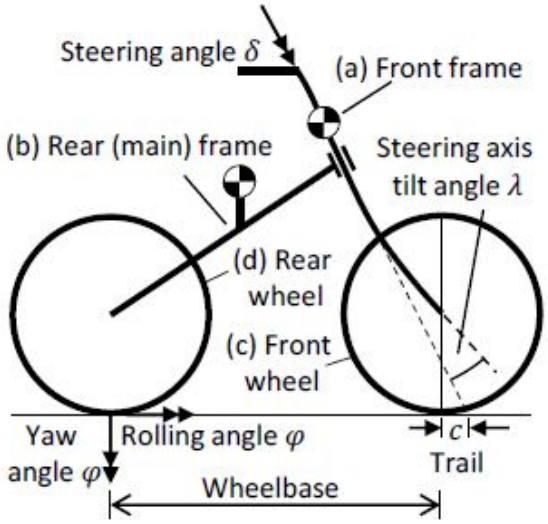


Figure 1: The Carvallo-Whipple Model

In the Carvallo-Whipple model, the rider does not have any facilities for such a correcting action. In a certain range of speeds, the moving bicycle nevertheless exerts a stabilizing corrective action, leading to a stable system. A dynamic system that shows this behavior is said to be self-stable. The exact mechanism for the self-stability of bicycles has been a somewhat contentious[4]. Despite this, bicycle stability can be evaluated mathematically: If and where a dynamic system is stable can be determined by calculating the eigenvalues of the system. For the Carvallo-Whipple model, the eigenvalues belong to three distinct eigenmodes, namely capsize, weave and castering modes. The effect that the moving bicycle will start to lean onto a spiraling path until it ultimately keels over is called capsize mode. The weave mode is an oscillation around the heading of the bicycle, while castering describes a motion around the steering axis. According to conventional wisdom only capsize and weave are significant for bicycle stability [9]. The two models are verified by comparing the eigenvalues of the weave mode at the same range of velocities for the two models with the results presented in the aforementioned benchmark. Similar work has been shown in[2]. As an extension, here the models are also compared in the time domain by evaluating the reaction of the bicycle to a steering impulse.

III. THE BICYCLE MODEL IN MATLAB

The Matlab implementation is based on the exhaustive review and benchmark of prior implementations of the Carvallo-Whipple model in [6]. The (linearized) equation describing the model can be written in the form

$$M\ddot{q} + vC_1\dot{q} + qgK_0 + v^2K_2q = f \quad (1)$$

Here, M is a symmetric mass matrix of the bicycle model, C_1 represents the damping terms of due to the steering rate and the stiffness term consists of two parts: a symmetric matrix (K_0) proportional to the gravitational acceleration and a non-symmetric matrix (K_2) due to the gyroscopic and centrifugal effects. The term q is a vector consisting of the rolling angle ϕ and the steering angle δ and forcing term f includes the steering torque depending on the steering angle $M\delta$ and the lateral leaning torque M [8]. The Matlab model as it is implemented is taken from[6], so the same qualifications apply, namely that the model is linearized around the upright standing position, the restriction to knife-edge rolling point contact and the general limitations of the Carvallo-Whipple model, e.g. that the rider is integrated into the rear frame assembly and not modelled separately. The resulting model has 25 design parameters, but as presented in [6], the number can be reduced to 17: By integrating the inertial properties of the front in rear wheels into the front and rear frame, respectively, the number of parameters can be reduced to 21. The wheel radii can be eliminated by further introducing gyrostat constants, yielding 19 design parameters since the radii are irrelevant for lean and steer geometry in linear modelling. Finally, the number of design parameters can be reduced to 17, since the polar moments of inertia of the two frames are irrelevant in the linearized equations of motion. In this model, the speed of the bicycle is taken as an input into the system. The results shown in this paper are based on the parameters in the appendix.

IV. THE BICYCLE MODEL IN MSC ADAMS

In MSC Adams the bicycle model mirrors the model as it is described in section II: There are four rigid bodes, namely front and rear frames and front and rear wheels. The rear frame is linked to the front frame at the steering head by the revolute joint so that it can rotate around the steering axis. Additionally, an expression for the ground contact is necessary. The contact between the edge of the rigid wheels and the ground is assumed to be ideal with no lateral slip at the tires. This yields an expression for the tire-ground contact [7, p. 62]:

$$V_x - v_x + r\left(\cos\phi\sin\psi\,\dot{\phi} + \sin\phi\cos\psi\,\dot{\psi} + \cos\psi\,\dot{\theta}\right) \quad (2)$$

and

$$V_y = v_y \quad r\left(\cos\phi\cos\psi\,\dot{\phi} + \sin\phi\sin\psi\,\dot{\psi} + \cos\psi\,\dot{\theta}\right) \quad (3)$$

Here, Vx and Vy are the relative linear velocities of the bicycle's wheel at the ground contact point with respect to the X and Y coordinate, $vx$and vy the same quantities for the center of gravity of the bicycle wheel, r is the radius of the bicycle's wheels,ϕ is the rolling angle,ψ is the yaw angle, ψ the corresponding yaw velocity, and θ the angular velocity with which the wheel rotates.

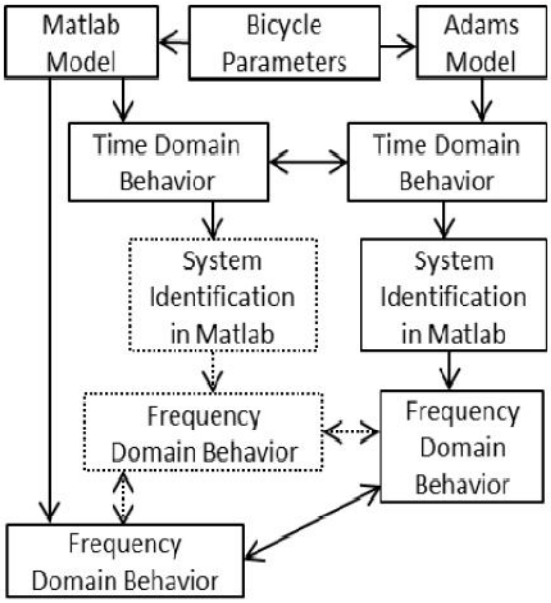


Figure 2: Model Comparison Workflow

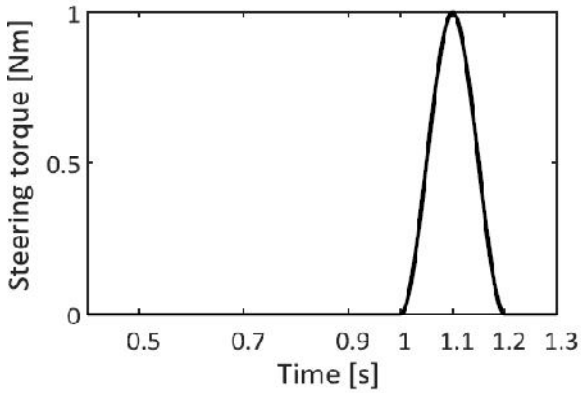


Figure 3: Application of Steering Torque

V. MODEL COMPARISON

For the comparison of the models two different approaches are investigated, described in Figure 2.Firstly, the models are compared in the time domain, i.e. the bicycle models are set upright, given a starting velocity and then the trajectories are integrated for a certain amount of time. As a second approach, the models are compared in the frequency domain. For the Matlab model, the eigenvalues are calculated directly as the eigenvalues of the matrices set up for the linear model. For the Adams Model this is not directly possible. Instead, starting with the bicycle in an upright position having an initial velocity ranging from 4 to 15 m/s, the motion of the bicycle is disturbed by applying a steering torque impulse with a magnitude of 0.1 Nm after one second of simulation time has passed. The system response gathered from this time integration is then used as input to Matlab's system identification toolbox. The frequency domain results are then the eigenvalues of the system matrices that are the output of the system identification process.

Function	Parameter	Value
τ_1		
	$h_{0,1}$	0 Nm(τ_{max})
	$h_{1,1}$	0.1 Nm(τ_{min})
	$t_{0,1}$	1 s
	$t_{1,1}$	1.1 s
τ_2		
	$h_{0,2}$	0.1 Nm(τ_{max})
	$h_{1,2}$	0 Nm(τ_{min})
	$t_{0,2}$	1.1 s

Table I: Parameters for step functions $\tau 1 \tau 2$

It is advantageous to compare eigenvalues between different models or different implementations of the same model, since eigenvalues are "independent of all of these differences. So, for example, a computer-based study of a bicycle based on any formulation can be checked for correctness and accuracy by comparing with the benchmark eigenvalues here"[6].

The steering torque in the Adams model is applied gradually starting from $t = 1\,s$ to the peak value of $\tau_{max} = 0.1$ Nmat $t = 1.1$ s and then unloading steadily to $\tau_{min} = 0$ Nm at $t = 1.2$ s.

Mathematically, this perturbation of the steering torque is implemented via two Heaviside step functions τ_1, τ_2 that are each approximated by a cubic polynomial. This approximation in general is

$$\tau_i(t) = \begin{cases} h_{0,i} & t \le t_{0,i} \\ h_{0,i} - a\Delta^2(3 - 2\Delta) & t_{0,i} < t < t_{1,i} \\ h_{1,i} & t > t_{1,i} \end{cases} \quad (4)$$

With $a = h_{0,i} - h_{1,i}$ and $\Delta = (t - t_{0,i})/(t_{1,i} - t_{0,i})$.

The torque over time is then computed by subtracting the two step functions, i.e. $\tau(t) = \tau_1(t) - \tau_2(t)$, with the parameters $h_{i,j}$ and $t_{i,j}$ as given in table I.The resulting torque is shown for in Figure 3 for points in time between $t = 0.4\,s$ and $t = 1.3\,s$.

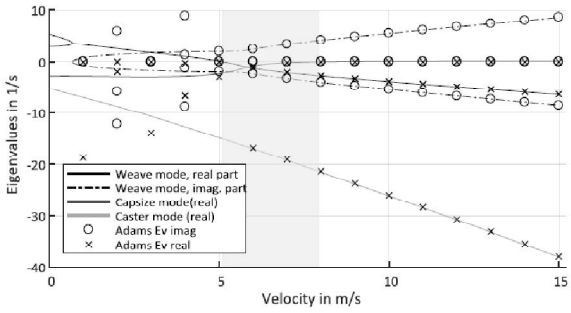


Figure 4: Behavior in the Frequency Domain

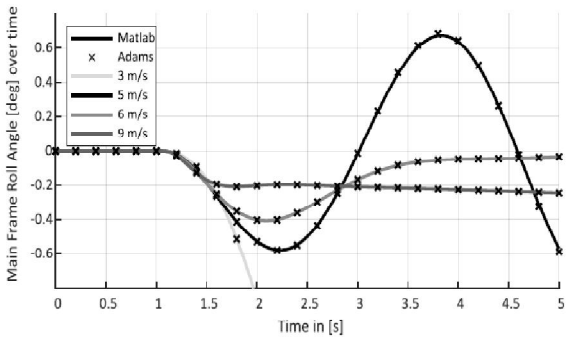


Figure 5: Behavior in the time domain

RESULTS AND CONCLUSION

The results for the frequency domain are shown in Figure 4. Here, the Matlab results for the real parts of the eigenvalues are depicted as solid lines, and the imaginary part of the weave mode as a dashed line. The results of the system identification on the Adams model are shown as circles for the imaginary parts and crosses for the real parts.

The frequency domain can phenomenologically be divided into four regions. The first region is situated at speeds from standing to about v = 0.7 m/s and is characterized by the bicycle falling over due to dominant weave mode. Since the bicycle is too slow for self-stabilizing behavior to emerge, it behaves akin to an unstabilized inverted pendulum. In the figure, this is signified by the bifurcated wave mode. In the second region, at speeds between 0.7 m/s and 5.1 m/s, the bicycle is somewhat less instable, as signified by the decreasing real part of the weave mode eigenvalue. As shown in the time domain trajectory for v = 3 m/s in light grey in Figure 5, the bicycle will exhibit overcorrective behavior after the perturbation of the steering torque at t = 1 s and thus ultimately fall over. For speeds closer to the stable region, such as the second trajectory for v = 5 m/s shown in black in Figure 5, the steering perturbation leads to an escalating weave oscillation.

Conversely, there is an area of self-stability in the third region, located between about 5.1 m/s and 7.9 m/s and

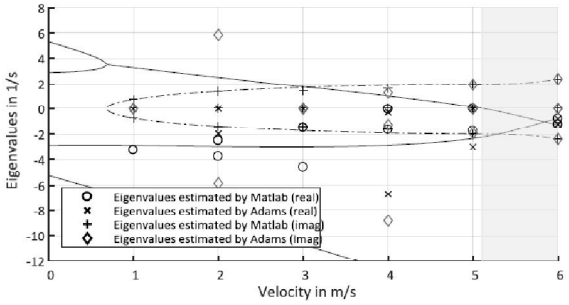


Figure 6: Behavior of identified system below stable speed

indicated in grey in the plot. Mathematically, this is expressed by the fact that all eigenvalues have negative real parts (solid lines in Figure 4). Here, the corrective action that arises from the inherent stability of the bicycle at that speed is large enough to prevent the bicycle from just falling over, but at the same time small enough to not exhibit overcorrection. Such a stable trajectory is shown in Figure 5 in grey. For even higher speeds, the bicycle loses self-stability again. Unlike at lower speeds, the instability in this region stems from the capsize mode, i.e. the trajectory follows a tightening spiral where lean and steer both increase until the bicycle falls over. As signified by the very small, but positive real parts of the eigenvalues, this behavior only has a gentle influence on the system and needs considerable time to develop until the bicycle finally keels over. The dark grey trajectory shown in Figure 5 thus does not have enough time to develop to conclusion. In the time domain, the simulation results in Matlab and Adams are in good agreement, both in conforming with each other and mirroring the expected phenomenological behavior as discussed before and as it is exhibited in the benchmark[6]. In the frequency domain, the Matlab results are agree with the benchmark, but the Adams model shows some discrepancies. One discrepancy is that the Adams model does not generate good frequency domain results in the region below stable speeds. To verify that this does not indicate a deficiency in the implementation, the comparison path indicated in dashed lines in Figure 2is taken. Here, the same system identification that is necessary for the frequency behavior of the Adams model is applied to the time domain results of the Matlab model. The results are shown in Figure 6 and are similarly bad for both paths, signifying that this deficiency is not an error in the Adams model, but rather a result of the system identification. This reflects the mathematical difficulty of identifying the system behavior from the very limited amount of data that can be provided for a system that is in a evaluable state only for a very short time (from the start of the simulation until it has fallen over). Above the stable speed, the Matlab results correspond well with the results from Adams, apart from one discrepancy: As laid out before, the stable region ends when real parts of the eigenvalues of the capsize mode cross the abscissa and become positive. For the eigenvalues identified from result of the Adams simulation, this does not hold true. Instead, the real parts of the eigenvalues are very close to zero, signifying a system behavior that borders stability. Supposedly, this happens because again the simulated timeframes do not allow capsizing behavior to emerge and subsequently this behavior cannot be identified from the simulation results.

One distinctive result of the system identification is how well the castering mode was fitted. Since the castering mode starts with negative real parts of the eigenvalues and continues to diminish throughout the simulated speed spectrum, conventional wisdom is that the castering mode is insignificant, at least for the stability of the system [6]. Because of the comparatively good fit of the castering mode, it is at least possible to say that it has a significant influence on the system behavior; otherwise it would not have been identified so well.

BIBLIOGRAPHY

[1] Bombach, E. Traktionskonzept für ein Lastenfahrrad, Untersuchung von Fahrverhalten und Antriebsstrang Wiesbaden: Springer Vieweg, 2016, ISBN 978-3-658-16408-9.

[2] Bulsink, V. E.; Doria, A.; van de Belt, D.; Koopman, B. The effect of tyre and rider properties on the stability of a bicycle In: Advances in Mechanical Engineering, 7 (2015) 12, 168781401562259, DOI 10.1177/1687814015622596.

[3] Cerone, V.; Andreo, D.; Larsson, M.; Regruto, D. Stabilization of a Riderless Bicycle [Applications of Control In: IEEE Control Systems Magazine, 30 (2010) 5, pp. 23–32, DOI 10.1109/MCS.2010.937745.

[4] Kooijman, J. D. G.; Meijaard, J. P.; Papadopoulos, J. M.; Ruina, A.; Schwab, A. L. A Bicycle Can Be Self-Stable Without Gyroscopic or Caster Effects In: Science, 332 (2011) 6027, pp. 339–342, DOI 10.1126/science.1201959.

[5] Meijaard, J. P. et al. History of thoughts about bicycle self-stability: Supplementary Material to A Bicycle Can Be Self-Stable Without Gyroscopic or Caster Effects https://ecommons.cornell.edu/handle/1813/22497 (accessed on 13.06.2018).

[6] Meijaard, J. P.; Papadopoulos, J. M.; Ruina, A.; Schwab, A. L. Linearized dynamics equations for the balance and steer of a bicycle, A benchmark and review In: Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 463 (2007) 2084, pp. 1955–1982, DOI 10.1098/rspa.2007.1857.

[7] Rill, G.; Schaeffer, T. Grundlagen und Methodik der Mehrkörpersimulation, Vertieft in Matlab-Beispielen, Übungen und Anwendungen Wiesbaden: Springer Vieweg, 2., überarb. u. erw. Aufl., 2014, ISBN 978-3-658-06083-1.

[8] Schwab, A. L.; Meijaard, J. P. A review on bicycle dynamics and rider control In: Vehicle System Dynamics, 51 (2013) 7, pp. 1059–1090, DOI 10.1080/00423114.2013.793365.

[9] Schwab, A. L.; Meijaard, J. P.; Kooijman, J. D.G. Lateral dynamics of a bicycle with a passive rider 6 model: stability and controllability In: Vehicle System Dynamics, 50 (2012) 8, pp. 1209–1224, DOI 10.1080/00423114.2011.610898.

APPENDIX

Parameter	Value	Unit
Wheelbase	1.055	m
Trail	0.0695	m
Steering axis tilt angle	19.366	deg
Rear Wheel radius	0.3365	m
Rear wheel mass	2.67	kg
Rear wheel moment of inertia I_{xx}	0.07455	kgm^2
Rear wheel moment of inertia I_{yy}	0.115	kgm^2
Front Wheel radius	0.3365	m
Front wheel mass	2.525	kg
Front wheel moment of inertia I_{xx}	0.07535	kgm^2
Front wheel moment of inertia I_{yy}	0.136	kgm^2
Main frame CG in x direction	0.3169	m
Main frame CG in y direction	0	m
Main frame CG in z direction	-1.0791	m
Main frame mass	81.67	kg
Main frame moment of inertia M_{xx}	11.2537	kgm^2
Main frame moment of inertia M_{yy}	11.8201	kgm^2
Main frame moment of inertia M_{zz}	2.9535	kgm^2
Main frame moment of inertia M_{xy}	0	kgm^2
Main frame moment of inertia M_{xz}	-1.9189	kgm^2
Main frame moment of inertia M_{yz}	0	kgm^2
Front frame CG in x direction	0.8869	m
Front frame CG in y direction	0	m
Front frame CG in z direction	-0.7582	m
Front frame mass	3.42	kg
Front frame moment of inertia M_{xx}	0.2529	kgm^2
Front frame moment of inertia M_{yy}	0.2315	kgm^2
Front frame moment of inertia M_{zz}	0.09288	kgm^2
Front frame moment of inertia M_{xy}	0	kgm^2
Front frame moment of inertia M_{xz}	-0.0044	kgm^2
Front frame moment of inertia M_{yz}	0	kgm^2

★★★