A non-linear rider model for motorcycles

2 authors:

Roberto Lot
University of Southampton
110 PUBLICATIONS 1,919 CITATIONS

SEE PROFILE

Vittore Cossalter
University of Padova
97 PUBLICATIONS 1,535 CITATIONS

SEE PROFILE

F2006V075

A NON LINEAR RIDER MODEL FOR MOTORCYCLES

Lot Roberto, Vittore Cossalter
Department of Mechanical Engineering - University ofPadova, Italy

KEYWORDS – rider, motorcycle, control, multibody, dynamics

ABSTRACT – The aim of the work presented is to develop a rider model for controlling a motorcycle during multibody simulations. In particular, it had to simulate human behavior during standard handling tests such as lane change, U-turn and slalom.

The paper illustrates three main ideas: an algorithm for vehicle tracking and look-ahead, an architecture for the virtual rider, and a procedure for the optimization of control gains.

The vehicle tracking algorithm is based on the description of the road and the vehicle position using Curvilinear coordinates: the advantage of this method is that the space covered along the road and the lateral deviation from the road center are included in the model variables, hence no additional tracking algorithms are needed. This technique is applied to a look-ahead point too.

The input of the rider model is the target path and roll angle. The output is the steering torque, whereas any torso movements of the rider are neglected. The proposed steering control is based on the classical PD architecture and consists in a look-ahead strategy on the lateral deviation from the desired path and the roll angle. Two additional terms, proportional respectively to the yaw and steering rates, are used to improve vehicle stability. The controller is completed with a low-pass filter whose aim is to replicate the limited ability of human riders in performing very fast maneuvers. All control parameters are adjustable as the speed, the acceleration and the roll angle varies. The rider model includes also a speed control, which is quite simple and is not described in the paper.

Since the tuning of control parameters is not easy, an automatic optimization procedure has been introduced. For any given set of control parameters, the performance index is defined as the rms error between the target and simulated motion. Unilateral constraints are also considered, such as vehicle stability, roll angle overshoot, maximum steering torque, maximum deviation from the desired path, etc. The best control gains are found by optimizing the performance index using a non-derivative algorithm. Since the exact evaluation of the performance index is computationally onerous, the performance is estimated by basing it on the linearized model of the vehicle. Time histories are obtained by using Laplace’s transform techniques, which make it possible to estimate the vehicle motion without integrating the equations of motion. The advantages of this approach are various: the (in)stability of the system is immediately recognizable, the method is much faster than time integration and is insensitive to the step-size.

This rider model has been implemented in a Fortran code and has been coupled to a complex, non-linear multibody model of the motorcycle. Examples of control optimization and non-linear simulation will be illustrated for a lane change, a slalom and a severe cornering maneuvers. The proposed virtual rider reproduces the behavior of human riders very well. The extension of this method to more general maneuvers (e.g. the replication of a track lap) is under development.

INTRODUCTION

Multibody simulation represents an important instrument for the design of a motorcycle, especially in the early stages of the process. For simulating the motorcycle dynamics it is necessary not only to develop a motorcycle model, but also to define a virtual rider. Indeed, two-wheel vehicles are intrinsically unstable and it is not possible to simulate open-loop maneuvers like for cars.
Riding a vehicle from one place to another may be considered as the combination of three tasks: plan the trajectory, follow the planned trajectory and keep the vehicle in stable condition. Each of these tasks involves different rider skill level**s**, in particular the vehicle stabilization and path following requires more reactivity than trajectory planning, i.e. they are higher frequency activities. This characteristic offers the opportunity of using an appropriate technique for automatic trajectory planning and a different technique for vehicle stability and riding. In the next sections a smart algorithm for the motorcycle tracking, an architecture for the virtual rider and a procedure for optimizing the control gains will be proposed.
Results will be discussed for a lane change, a slalom, and a severe cornering maneuver.
The trajectory planning problem will be not discussed in this paper, whereas it is assumed that the target motion is assessed using the optimal maneuver method (1-3).

VEHICLE TRACKING AND LOOK AHEAD ALGORITHM

Track Description using Curvilinear Coordinates

Real roads are similar to strings: they are long and narrow. Hence, a very effective road description is obtained by means of its curvature Θ (i.e. the reciprocal of the curvature radius r) as a function of the road length s, as shown in Figure 1.

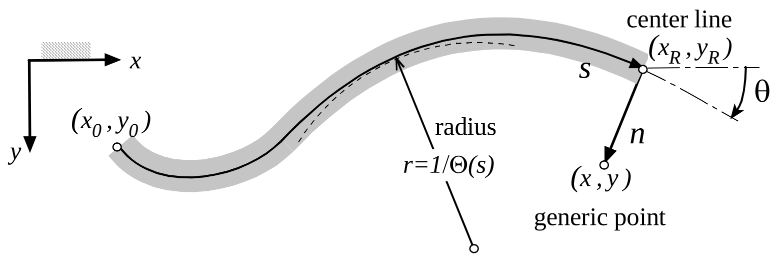

Figure 1 – Track description using curvilinear coordinates (top view)

The Cartesian coordinates of the center line (x_R,y_R) and its orientation θ with respect the ground may be obtained by integrating the following equations:

$$
\begin{aligned}
\frac{\partial}{\partial s}\theta &= \Theta(s) \\
\frac{\partial}{\partial s}x_R &= \cos(\theta) \\
\frac{\partial}{\partial s}y_R &= \sin(\theta)
\end{aligned}
\qquad (1)
$$

To describe the position of a generic point P which do not belong to the center line is then necessary to introduce the additional coordinate *n*, i.e. its distance from the center line. Coordinates *(s,n)* are very effective for path following purposes because the path deviation is immediately available as a state variable. On the contrary, the utilization of Cartesian coordinates (x_R,y_R) will make it difficult to describe a string-shaped road and will require an additional algorithm for estimating the distance of a generic point from the road center-line.

Vehicle Tracking and Look-Ahead

Vehicle dynamics is frequently described by using quasi-coordinates (u,v,ψ_{dot}), i.e. the forward and lateral speed plus the yaw rate. For vehicle tracking, it is very effective to couple quasi-coordinates with curvilinear coordinates, i.e. the vehicle position along the target path s_V and its lateral deviation n_V plus the vehicle orientation χ_V relative to the path as shown in Figure 2.

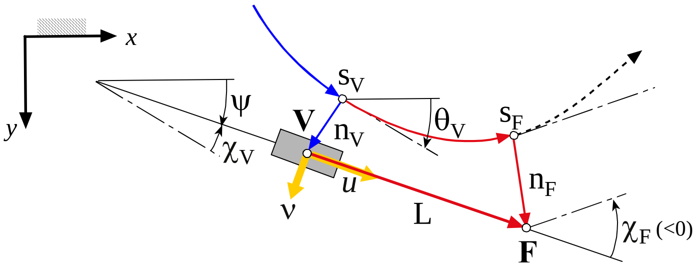

Figure 2 – Vehicle tracking using curvilinear coordinates

The additional curvilinear coordinates (s_V, s_V, χ_V) are correlated to the quasi-coordinates by means of the following differential equations:

$$\begin{aligned} \Theta(s_V)\left(\frac{\partial}{\partial t}s_V\right) &= \psi_{dot} - \left(\frac{\partial}{\partial t}\chi_V\right) \\ (1 - n_V\,\Theta(s_V))\left(\frac{\partial}{\partial t}s_V\right) &= u\cos(\chi_V) - v\sin(\chi_V) \\ \frac{\partial}{\partial t}n_V &= v\cos(\chi_V) + u\sin(\chi_V) \end{aligned} \tag{2}$$

It is straightforward that this representation *is* an algorithm for vehicle tracking, and it is simple, concise and numerically efficient compared to other methods (4-5).
It is well known that human riders look at the road ahead for estimating its future position and a similar action is introduced in this rider model by defining a look-ahead point **F**, which is located at a distance L in front of the vehicle. The tracking estimation simply consists in the replication of the curvilinear equations (2) for this new point, as follows:

$$\begin{aligned} \Theta(s_F)\left(\frac{\partial}{\partial t}s_F\right) &= \psi_{dot} - \left(\frac{\partial}{\partial t}\chi_F\right) \\ (1 - n_F\,\Theta(s_F))\left(\frac{\partial}{\partial t}s_F\right) &= \left(u + \left(\frac{\partial}{\partial t}L\right)\right)\cos(\chi_F) - (\psi_{dot}L + v)\sin(\chi_F) \\ \frac{\partial}{\partial t}n_F &= (\psi_{dot}L + v)\cos(\chi_F) + \left(u + \left(\frac{\partial}{\partial t}L\right)\right)\sin(\chi_F) \end{aligned} \tag{3}$$

The look-ahead distance L is one of the control parameters, and it should be changed as the speed changes (the greater the speed, the greater the distance) and as the track curvature changes too (greater the curvature, i.e. the smaller the curvature radius, the smaller the distance):

$$L = \mathrm{L}(u, \Theta) \tag{4}$$

THE RIDER MODEL

Human riders control the motorcycle by steering with the handlebar, by moving the body on the saddle, and by acting on the brake level and/or pedal and throttle. It is very difficult to transpose human skill into a virtual rider, and the proposed rider is much simpler. The main simplified assumption is that the motorcycle lateral dynamics is controlled only by means of the handlebar, whereas the rider body movements are neglected. Indeed, the steering torque represents the main input of the motorcycle, whereas the body leaning action is more limited both in amplitude and in frequency. Moreover, brakes and throttle are used for speed control only, since only expert riders are able to influence lateral dynamics by applying the brakes or the accelerator while cornering.
The rider model architecture is depicted in the Figure 3: the motion error is the input of a proportional-derivative block, which output is filtered and gives the steering torque which should be applied to the handlebar

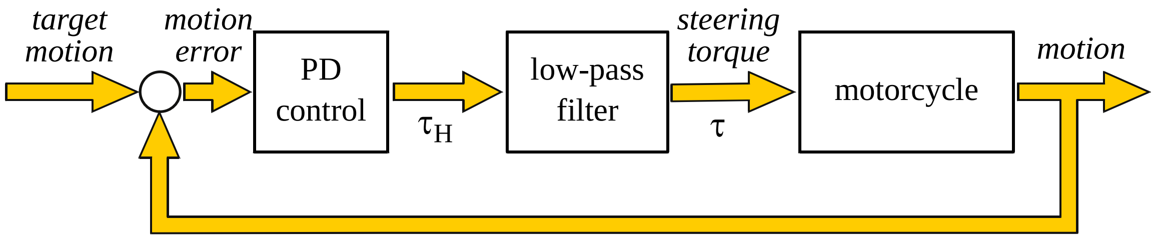

Figure 3 – vehicle tracking using curvilinear coordinates

The unfiltered steering torque contains six terms, as follows:

$$\begin{aligned}\tau_H = {} & KP_\phi\,(\phi_T(s_F) - \phi) - KD_\phi\,\phi_{dot} \\ & + KP_n\,(n_T(s_F) - n_F) - KD_n\left(\frac{\partial}{\partial t}n_F\right) \\ & + KD_\psi\,(\psi_{dot} - u\,\Theta(s_V)) + KD_\delta\,\delta_{dot}\end{aligned} \tag{5}$$

The first row corresponds to a proportional-derivative action on the roll angle and has the primary mission of stabilizing the motorcycle capsize; the second row corresponds to a proportional-derivative action on the lateral deviation of the look-ahead point and its function is to keep the vehicle on the target trajectory; finally the third row contains two derivative terms on the yaw and steering angle which are useful for stabilizing the weave and wobble modes.
Some authors propose the utilization of a proportional term on the steering angle: this choice is effective for car control because for cornering to the right it is necessary to rotate the steering-wheel considerably in the same way. On the contrary this choice does not appear attractive for motorcycle control since the steering angle has a small amplitude, which also depends on the vehicle camber. Moreover, motorcycle riding includes several situations of counter-steering, i.e. the handlebar is rotated to the left when the yaw rate on the right and vice-versa.

Human riders may control the handlebar only with a limited band-pass, indeed riders cannot stabilize wobble (typically 7-9 Hz) and it is also difficult to control the weave mode (typically 2-3 Hz). In order to reproduce this human behavior, the control action (5) is filtered using a 2nd order low-pass filter, defined as follows:

$$\left(\frac{\partial}{\partial t}\tau_{dot}\right) + 2\,\zeta\,\omega\,\tau_{dot} + \omega^2\,\tau = \omega^2\,\tau_H$$
$$\frac{\partial}{\partial t}\tau = \tau_{dot} \qquad (6)$$

It has been found that appropriate values are f=ω/2π = 8-10 Hz for the cut frequency and $\zeta \simeq 0.7$ for the damping ratio.

THE CONTROL OPTIMIZATION ALGORITHM

The maximization of the control performance consists in the determination of the gains value which minimize the difference between the target and vehicle motion, i.e. which minimizes the following penalty function:

$$P = \int_0^T (n_V(s_V) - n_T(s_V))^2\, dt \qquad (7)$$

where s_V is the vehicle position along the track, n_T and n_V are respectively the target and vehicle deviation from the road center. Moreover, it is convenient to add to the penalty (7) some terms which may prevent vehicle instability, in particular additional terms on the yaw rate ψ_{dot} and steering rate δ_{dot} may reduce weave and wobble. So, the penalty function is updated as follows:

$$P = \int_0^T w_n\,(n_V(s_V) - n_T(s_V))^2 + w_\psi\,(\psi_{dot} - u\,\Theta(s_V))^2 + w_\delta\,\delta_{dot}^{\;2}\, dt \qquad (8)$$

where weights w_n, w_ψ and w_δ are used for properly balancing the different terms.
A realistic rider model should also take into account that:

- the lateral deviation n_V from the target path cannot exceed the lane width;
- the motorcycle roll angle φ cannot exceed a maximum, which depends on tires;
- the steering torque τ cannot exceed a maximum, which depends on the rider physiological capability.

The above conditions yield to the following inequality constraints, which must be taken into account during the penalty minimization:

$$|n_V| < \mathrm{W}_{lane}$$
$$|\tau| < \tau_{max} \qquad (9)$$
$$|\phi| < \phi_{max}$$

There are several control optimization techniques available in literature, however most of them are applicable to linear systems only and cannot include inequality constraints. A potential alternative is to find the penalty minimum by using a non-derivative algorithm (commonly available), however it has the inconvenience of requiring a lot of evaluations of the penalty function with a different set of control gains. If the penalty function was computed by integrating the motorcycle equation of motion together with the control equations (2), (3), (5), the computation time would be so great that this solution would have no practical interest. For the stated reasons, a different technique which drastically reduce the penalty computation time has been developed. This technique is based on the linearization of the model equations about the conditions of target motion and on the evaluation of the time history using Laplace's transform techniques. Laplace's transform has two main advantages: first, it is possible to immediately recognize the system stability, secondly, the time history may be computed as an explicit function of the time, as shown in the Appendix. Since the computation time is proportional to the number of evaluated points, this method is much faster than time integration and it is also insensitive to the step-size. The price paid for it is that the technique is applicable to linear systems only, hence the model equation must be linearized and the results will be approximated. However, it has been found that the errors may be kept reasonably small by linearizing the equations about the target motion condition.

SIMULATION EXAMPLES

The control system and the optimization algorithm described above have been translated into a Fortran program and coupled with the full non-linear motorcycle model, as described in reference (6-7).

The first example consists in the control optimization and non-linear time simulation of a lane change maneuver: the lane width is 4m and the maneuver is completed in a distance of 21 m. The equations of motion have been linearized around the straight motion condition at a speed of 18 m/s. The control parameters that have been optimized are, respectively, the look-ahead distance L, the roll gains KP_{φ}, KD_{φ} and the lateral displacement gains KP_n, KD_n. First guess values, upper and lower boundary, and optimized values are collected in Table 1.

The target motion, the inequality constraints as well as the non-linear simulation of the maneuver are depicted in Figure 4, which also shows data recording during a real maneuver. Figure 4a shows that the motorcycle path is very similar to the target path and in particular, it is completely inside the allowable area (the acceptable deviation is 1.6 m side to side). Figure 4b shows that the actual steering torque slightly exceed the admissible torque: this is a consequence of the linear approximation during control optimization. The same figure shows the steering torque measured on a real maneuver: it may be seen that the virtual rider action is

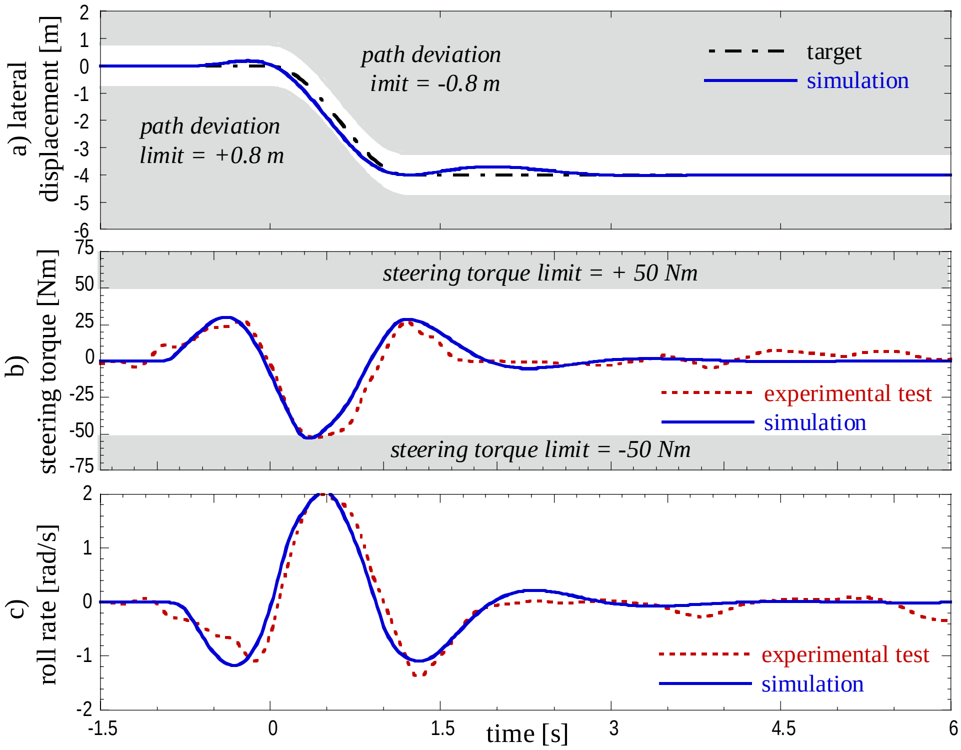

Figure 4 – Lane change maneuver (speed = 18 m/s)

gain	first guess	lower boundary	upper boundary	optimal value
L	10	2	35	18
KP_{φ},	-140	-350	00	-210.5
KD_{φ}	-35	-250	00	-53.0
KP_n	-25	-150	00	-53.8
KD_n	-15	-70	00	-54.5

Table 1 – Variation of the control parameters for the lane change optimization (optimization in 400 iterations, time < 15s)

gain	first guess	lower boundary	upper boundary	optimal value
L	10	2	30	7.4
KP_{φ},	-40	-250	00	-147.6
KD_{φ}	-35	-140	00	-64.9
KP_n	-25	-50	00	-22.6
KD_n	-15	-100	00	-38.6

Table 2 – Variation of the control parameters for the slalom optimization (optimization in 300 iterations, time < 10s)

very similar to the human one. Finally, Figure 4c shows the roll rate both for the simulated and experimental maneuver: once again there is a very good agreement, which has been repeatedly confirmed for different speeds and lane change lengths, as reported in reference (8-9).

The second example regards a slalom simulation at a speed of 21.3 m/s and a distance between cones of 21 m. The optimized control parameters are the same as of lane change: first guess values, upper and lower boundary, and optimized values are collected in Table 2. Results are collected in Figure 5, in particular Figure 5a) shows the target and the simulated

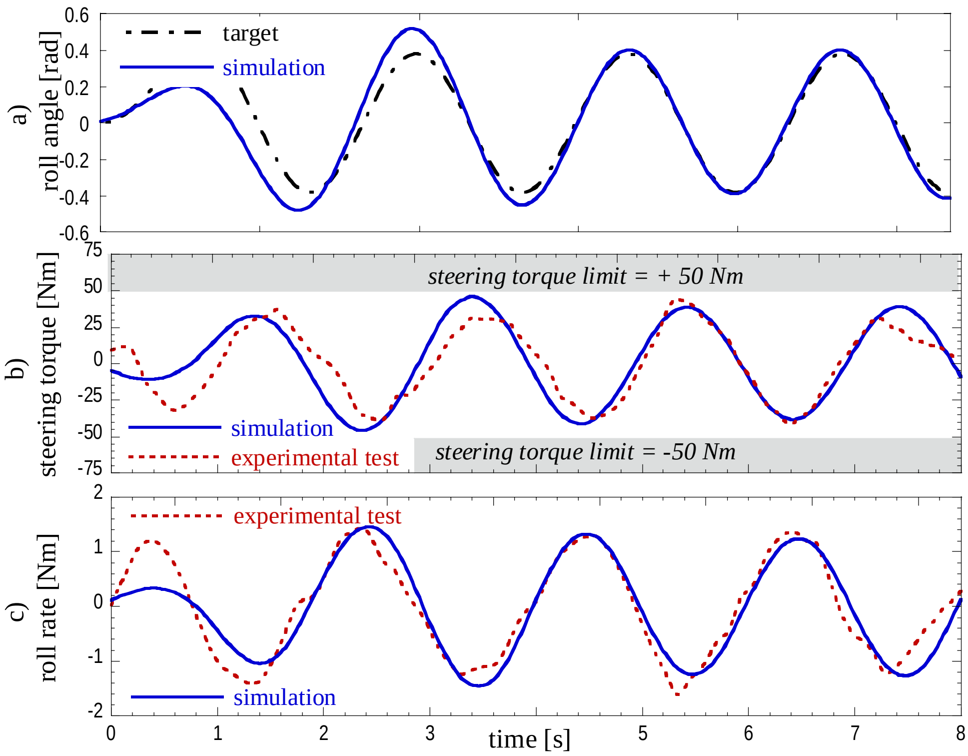

Figure 5 – Slalom maneuver (speed = 21.3 m/s, cone distance = 21 m)

roll angle: there is an initial discrepancy due to the transient from straight running to slalom, while in the second part the target and roll angles obtained are coincident. Figure 5b) and Figure 5c) compare the steering torque and roll rate obtained in the simulation with the experimental measurements made in a real slalom: there is very good agreement both in amplitude and phase.

The third example consists in the control optimization and non-linear time simulation of a cornering maneuver at a speed of 22 m/s and cornering radius equal to 50m. This maneuver is particularly severe because the steady state value of the lateral acceleration is equal to 10.6 m/s^2 and corresponds to a roll angle of 0.9 rad. Moreover, the transition between straight running and cornering condition in a time of 1.2s is fast. The equations of motion have been linearized in the neighborhood of final cornering conditions, the control parameters that have been optimized are**,** respectively**,** the look-ahead distance L, the roll proportional, derivative, and integral gains KP_φ, KD_φ KI_φ, as depicted in Table 3. The results are shown in Figure 6, in terms of roll angle, steering torque, and yaw rate. Figure 6a shows that the roll angle varies regularly and rapidly (the roll rate peak is about 1.13 rad/s = 65°/s). For this maneuver it was essential to restrain the roll angle because any overshoot would be unrealistic and could cause the motorcycle fall down. Figure 6b shows that the steering torque is about 50 Nm, i.e. inside the normal riding range. Finally, Figure 6c shows some oscillations in the yaw rate, which testify that the vehicle is approaching unstable conditions.

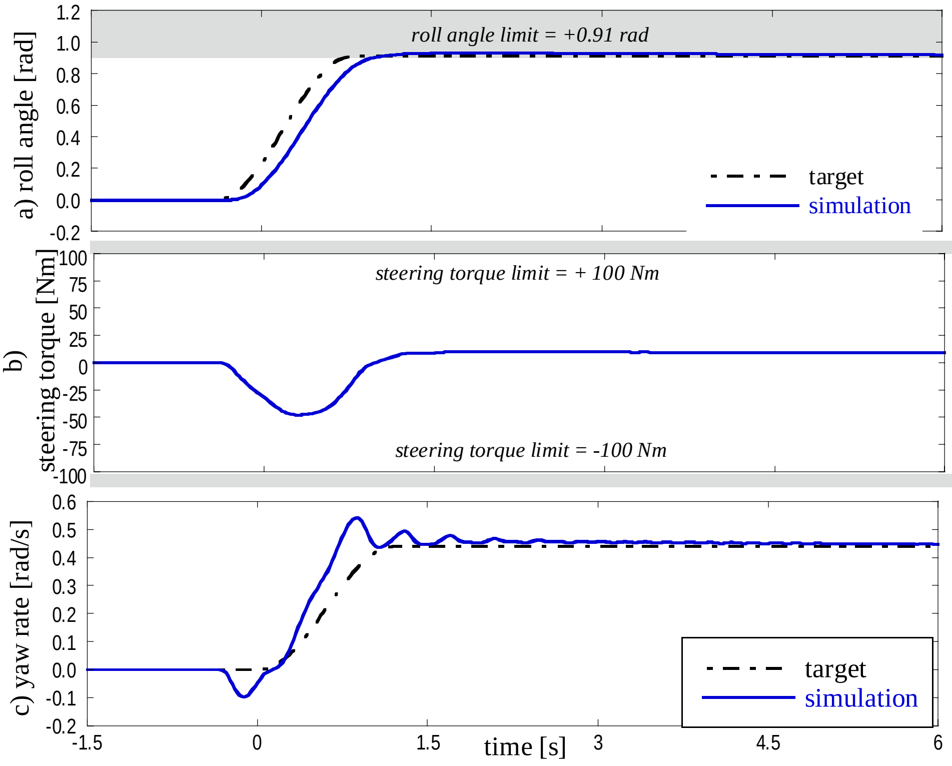

Figure 6 – Cornering maneuver (speed = 22 m/s)

gain	first guess	lower boundary	upper boundary	**optimal value**
L	20	9.2	40	**9.2**
KP_φ,	-240	-400	0	**-386.3**
KD_φ	-135	-200	0	**-29.6**
KI_φ	-20	-50	0	**-45.5**

Table 3 – Variation of the control parameters for the cornering optimization (optimization in 600 iterations, time < 30s)

CONCLUSION AND FORTHCOMING DEVELOPMENT

In this paper an algorithm for vehicle tracking, a virtual rider model, and a control gain optimization algorithm have been proposed.
The vehicle tracking algorithm consists in the appropriate utilization of Curvilinear coordinates for describing the position of the vehicle with respect to the center line. This method is applicable not only to motorcycles, but also to any other terrain vehicle. Curvilinear coordinates are also used for tracking the look-ahead point, i.e. a point located in front of the vehicle at an adjustable distance.
The rider model consists in a revised proportional-derivative control with the purpose of stabilizing the motorcycle and following a given trajectory. The rider controls the vehicle by means of the steering torque, whereas body movements are neglected. Steering torque is computed using a PD controller whose inputs are the deviations from the road center, the roll angle. A low-pass filter is also included in the control for reproducing the limited band-pass of human riders. The speed control is simple and has not been described in the paper. This rider model does not include any algorithm for trajectory planning; however the utilization of the “optimal maneuver method” described in reference (1-3) is proposed.
Since tuning control gains is difficult, a control optimization method has been proposed. This method is based on the minimization of the deviation from the target path, under inequality constraints such as lane width, maximum roll, etc. For fast computation, the time history is estimated using Laplace’s transform techniques, which require the linearization of model equations near the target motion condition.
The ideas proposed have been used to simulate maneuvers which are commonly used for rating motorcycle handling, i.e. a lane change, a slalom and a cornering maneuver. The discussions of results and the comparison with experimental data has demonstrated that the virtual rider is able to perform the required maneuver, moreover the simulated maneuvers are similar to human rider maneuvers.
The application of the techniques presented for motorcycle riding in more complex situations, e.g. the replication of a track lap, is still under progress. The Main difficulties are related to the necessity of linearizing the model to control optimization (not for simulation) and to the requirement of/for updating the control if the variation in the speed and roll angles are large.

ACKNOWLEDGEMENTS

This research was partially supported by funds from the Italian Ministry for Universities and Scientific and Technological Research (MURST ex40% funds).

REFERENCES

(1) F. Biral and M. Da Lio: Modelling drivers with the optimal manoeuvre method, ATA Conference 2001, Florence (Italy)

(2) E. Bertolazzi, F. Biral, M. Da Lio: Symbolic-Numeric Indirect Method for Solving Optimal Control Problems for Large Multibody Systems: The Time-Optimal Racing Vehicle Example, Multibody Dynamics Conference, Lisbon, Portugal, July 1-4 2003

(3) E. Bertolazzi, F. Biral, M. Da Lio; Motion Planning Algorithms based on Optimal Control for Motorcycle-Rider System, FISITA 2006

(4) Y.H. Cho and J. Kim: An improved Handling Model of a Driver/Vehicle System following an Arbitrary Path, Active Control of Noise Vibration, Vol. 38, pp. 311-317.

(5) P. Antos and J. Ambrosio: A Control Strategy for Vehicle Trajectory Tracking using Multibody Models.

(6) V. Cossalter, R. Lot: A motorcycle multi-body model for real time simulations based on the natural coordinates approach, Vehicle System Dynamics: Vol., 37, n°6 pp. 423-447

(7) V. Cossalter, R. Lot, and F. Maggio: A Multibody Code for Motorcycle Handling and Stability Analysis with Validation and Examples of Application, SAE Paper 2003-32-0035

(8) V. Cossalter: Motorcycle Dynamics, 2002, ISBN 0972051406

(9) V. Cossalter and J. Sadauckas: Elaboration and quantitative assessment of manoeuvrability for motorcycle lane change, Vehicle System Dynamics, in printing.

APPENDIX 1: TIME SIMULATION USING LAPLACE'S TRANSFORM

The equation of motion of a linear system may be written in the well known standard formulation as follows:

$$\begin{aligned}\dot{\mathbf{z}} &= \mathbf{A}_z\mathbf{z} + \mathbf{B}_z\mathbf{u}\\ \mathbf{y} &= \mathbf{C}_z\mathbf{z} + \mathbf{D}\,\mathbf{u}\end{aligned} \tag{A.1}$$

where $\mathbf{z}$ is the *n* independent state variables vector, $\mathbf{u}$ the system inputs and $\mathbf{y}$ the output variables. The computation of complex eigenvalues $\mathbf{e}$ and eigenvectors $\mathbf{T}_z$ of the state matrix $\mathbf{A}_z$ makes it possible to transform the equations (A.1) into the principal coordinates:

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{\Lambda}\,\mathbf{q} + \mathbf{B}_q\mathbf{u}\\ \mathbf{y} &= \mathbf{C}_q\mathbf{q} + \mathbf{D}\,\mathbf{u}\end{aligned} \tag{A.2}$$

where $\mathbf{q}$ is principal coordinates vector, $\mathbf{\Lambda}$ is a diagonal matrix the elements of which are the eigenvalues $\mathbf{e}$, while the remaining state matrices may be easily computed as $\mathbf{B}_q = \mathbf{T}_z^{-1}\mathbf{B}_z$ and $\mathbf{C}_q = \mathbf{C}_z\mathbf{T}_z$.
The formulation (A.2) is very effective because the system modes are uncoupled, even if the new state variables $\mathbf{q}$ are complex and so they lose the original physical meaning.
Applying Laplace's Transform to (A.2) one obtains:

$$s\mathbf{Q}(s) = \mathbf{\Lambda}\mathbf{Q}(s) + \mathbf{B}_q\mathbf{U}(s) \quad \Rightarrow \quad \mathbf{Q}(s) = \left(\mathbf{I}s - \mathbf{\Lambda}\right)^{-1}\mathbf{B}_q\mathbf{U}(s)$$

while output variables may be computed as follows

$$\mathbf{Y}(s) = \mathbf{C}_q\mathbf{Q}(s) + \mathbf{D}\,\mathbf{U}(s) = \left[\mathbf{C}_q\left(\mathbf{I}s - \mathbf{\Lambda}\right)^{-1}\mathbf{B}_q + \mathbf{D}\right]\mathbf{U}(s) \tag{A.3}$$

Since matrix $\mathbf{I}s - \mathbf{\Lambda}$ is diagonal, it is very easy to compute the inverse, and equation (A.3) may be re-written as follows:

$$\mathbf{Y}(s) = \left(\sum_{i=1,n}\mathbf{c}_i\frac{1}{s-e_i}\mathbf{b}_i + \mathbf{D}\right)\mathbf{U}(s) \tag{A.4}$$

where $\mathbf{c}_i$ are the columns of the matrix $\mathbf{C}_q$ and $\mathbf{b}_i$ are the rows of the matrix $\mathbf{B}_q$.
The application of the inverse Laplace transform makes it possible to move back into time domain as follows:

$$\mathbf{y}(t) = \sum_{i=1,n}\mathbf{c}_i\mathbf{r}_i\,\mathbf{F}(\mathbf{u}(t),\lambda_i) + \mathbf{D}\,\mathbf{u}(t) \tag{A.5}$$

where the only difficulty is the evaluation of $\mathbf{F}(\mathbf{u}(t), e_i)$ as the inverse transform of $\mathbf{U}(s)/\left(s-e_i\right)$ because in the presence of a generic input $\mathbf{u}(t)$ the computation of $\mathbf{F}$ is numerically onerous. However, when the system input may be described as the linear composition of standard functions $f_k(t)$ (e.g. step, ramp, sine and cosine, etc.)

$$u_i(t) = \sum_k f_k(t)$$

the computation of $F\left(f_k, e\right)$ may be done analytically. In conclusion, the implementation of a moderate number of shape functions f_k and $F\left(f_k, e\right)$ makes it possible to describe system inputs and compute system outputs in a very easy and computationally efficient way.