MULTIBODY DYNAMICS 2007, ECCOMAS Thematic Conference
C.L. Bottasso, P. Masarati, L. Trainelli (eds.)
Milano, Italy, 25–28 June 2007

APPLICATION OF LAPLACE TRANSFORM TECHNIQUES TO NON-LINEAR CONTROL OPTIMIZATION

Matteo Massaro, Roberto Lot

Department of Mechanical Engineering
University of Padova, Via Venezia 1, 35131 Padova, Italy
e-mails: matteo.massaro@unipd.it, roberto.lot@unipd.it

Keywords: optimization, non-linear, control, Laplace

Abstract. *This paper illustrates an original procedure for the non-linear optimization of a control system, including als unilateral constraints. The proposed procedure is articulated in two phases: first the control optimization problem is formulated as the minimization of the difference between the target and the actual evolution of the controlled system, then the constrained minimization is converted into an equivalent unconstrained problem, which solution is found by using a genetic algorithm. This technique requires the evaluation of the system response for a remarkable number of different control gains, therefore it is essential to adopt an efficient method. For this reason the following algorithm has been developed: 1) the controlled system is linearized and the model equations are re-formulate according to the modal decomposition technique, 2) the target state of the system is decomposed into basic tiles (e.g. combination of steps, ramps, etc.), 3) the response of the system is evaluated by means of analytical Inverse Laplace Transformation.*

The case-study of the control of a motorcycle is presented, which is particularly interesting because the dynamics is highly non-linear. The motorcycle is modelled taking into account lateral, yaw, roll, and steer motion; the vehicle input is the handlebar torque, whereas rider body movements are neglected. It has been implemented a virtual rider which includes a PID controller, a look-ahead strategy and a frequency low-pass filter for the steering torque in order to reproduce the behaviour of human riders.

1 INTRODUCTION

This paper illustrates an original procedure for the non-linear optimization of a control system including unilateral constraints. The work is organized as follows: section 2 presents the virtual rider architecture, section 3 illustrates the formulation of the optimization problem in presence of unilateral constraints, section 4 explains the technique for the efficient computation of the system behaviour using Laplace's technique, finally section 5 shows some results of control optimization and demonstrates that simulated manoeuvres of lane change, slalom and cornering are very similar to the actual manoeuvres performed by human riders.

2 VIRTUAL RIDER ARCHITECTURE

It is very difficult to transpose human skills into a virtual rider: even if there is a large number of works in literature (Ref. [2], [3], [4] among many others), this problem has not satisfactory solved yet. The virtual rider has been developed with the aim of reproducing the behaviour of human riders, i.e. the steering torque is limited both in amplitude and in frequency.

The architecture of the virtual rider is depicted in Figure 1: the control has separate loops for the speed and trajectory control. The main simplified assumption is that the lateral dynamics of the motorcycle is controlled only by means of the handlebar, whereas any rider movements is neglected. This assumption is not too limitative, because the steering torque really represents the main input of the motorcycle, whereas rider leaning, as well as other motions, is limited both in amplitude and in frequency. Moreover, brakes and throttle are used only for controlling the speed, since only expert riders are able to influence lateral dynamics by braking or accelerating while cornering. The speed control architecture is straightforward and takes the following form:

$$S_{eq} = M_{eq}(K_P \frac{V_T(t+t_{LD})-V(t)}{t_{LD}} + K_I \int (V - V_T)dt) + k_D V^2$$

where M_{eq} is the equivalent mass (which includes all translating and rotating inertia), S_{eq} is the equivalent longitudinal force (which is defined as the ratio between the power necessary to accelerate the vehicle and the speed), K_P and K_I are respectively the proportional and integral gains, V and V_T are respectively the actual and target speed, k_D is the drag coefficient, t_{LD} is the lead time of the traction control (usually about 0.1 s). Afterwards, the equivalent force S_{eq} is split among engine torque, rear and front brake torques, according to user specifications.

The trajectory control is essentially a PID control, which takes benefit of the use of curvi-

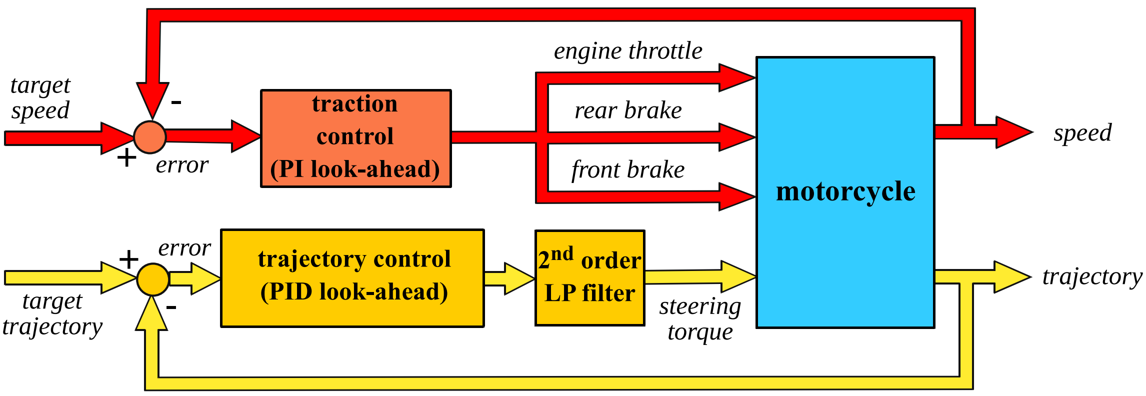

Figure 1: Architecture of the Virtual Rider.

linear coordinates for tracking the actual position of the vehicle with respect the road centre line. Indeed, curvilinear coordinates are defined as the vehicle position along the road s_V and its deviation from the centre line n_V, as shown in Figure 2. Curvilinear coordinates are used also for tracking the look-ahead point (s_F, n_F,) located in front of the vehicle. As human riders do, the look-ahead distance is changed as the speed changes (the greater the speed, the greater the distance) and as the track curvature changes too (the greater the curvature, i.e. the smaller the curvature radius, the smaller the distance). Basing on this smart technique, a simple PID control is adequate for riding the motorcycle:

$$\tau = \tau_{LP} + \tau_0$$

where τ_{LP} is the filtered steering torque of PID controller and τ_0 is the steady state torque.

The PID control consists of six terms:

$$\begin{aligned}\tau_{PID} = & KP_\varphi\left(\varphi_T[s_F] - \varphi_V\right) - K_{D\varphi}\dot{\varphi}_V - K_{I,\varphi}\int \varphi_V\, dt \\ & + KP_n\left(n_T[s_F] - n_F\right) - KD_n\dot{n}_V \\ & + KD_\psi\left(\dot{\psi}_V - u_V\,\Theta\right) \\ & + KD_\delta\dot{\delta}_V\end{aligned} \quad (1)$$

where suffixes V, F and T refer respectively to the vehicle, the look-ahead point and target variables, ϕ, ψ, δ are respectively the camber, yaw and steering angles, u is the motorcycle speed, finally Θ is the road curvature. The first row of the control law above corresponds to a PID action on the roll angle and has the primary function 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; the third row contains a derivative term on the yaw angle which is useful for stabilizing the weave mode, the fourth line is a derivative term on and the steering angle useful to stabilize the wobble mode.

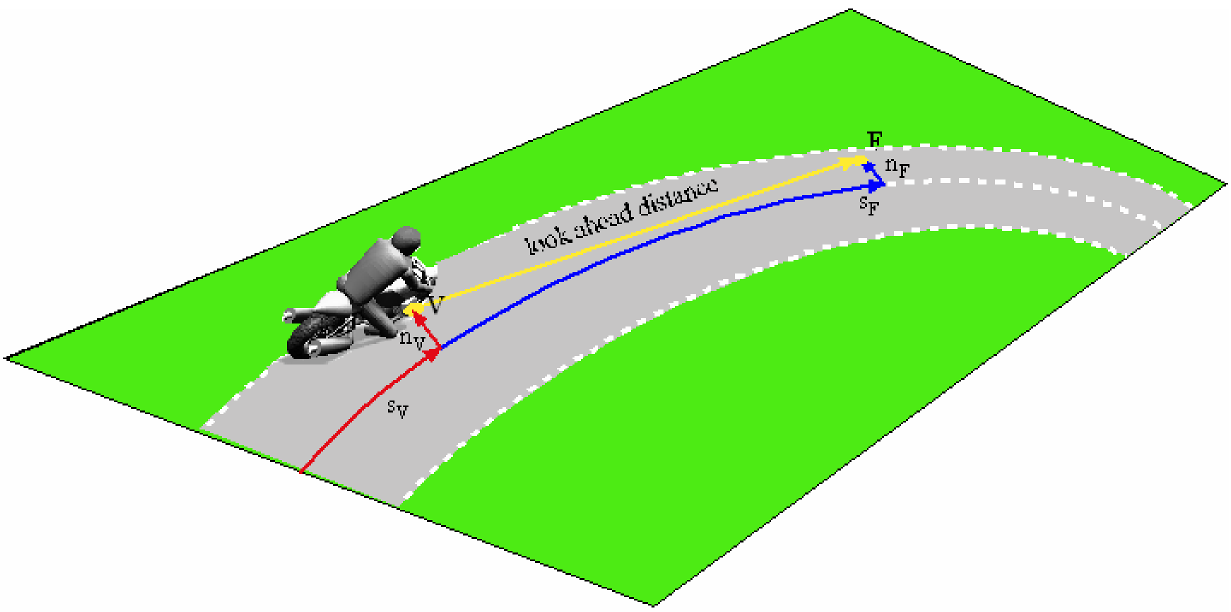

Figure 2: Position and Look-Ahead Point tracking using curvilinear coordinates

The steering torque (1) is then filtered with a 2nd order low-pass in order to replicate the limited human control capability (i.e. riders cannot stabilize wobble at a frequency of 7-9 Hz and they find also difficult to control weave mode at a frequency of 2-3 Hz).

3 FORMULATION OF THE OPTIMIZATION PROBLEM

After the definition of the control architecture (section 2), it is necessary to find the control gains, which both stabilize the vehicle and minimize the error between the target and actual trajectory.

From a mathematical point of view the vehicle stability requirement may be expressed as an unilateral constraint since all eigenvalues must have negative real part

$$\mathrm{Re}(\lambda_i) < 0$$

while the trajectory may be expressed by means of the following penalty function

$$J_{trajectory} = w_n \int (n_V - n_T)^2 dt + w_\varphi \int (\varphi_V - \varphi_T)^2 dt \qquad (2)$$

where n_V and n_T are respectively the actual and target vehicle deviation from the road centre, φ_V and φ_T are the actual and target vehicle roll angle, w_n and w_φ are weights. It should be noted that the target trajectory can be defined as a combination of n_T and φ_T, only by n_T, only by φ_T.

Moreover, it must be considered that the trajectory cannot exceed the lane width, the steering torque cannot exceed the rider physiological capability and roll angle overshoot must be avoided. This conditions lead to the following additional unilateral constraints:

$$|n_V| < n_{\max}$$
$$|\tau| < \tau_{\max}$$
$$|\varphi| < \varphi_{\max}$$

Since algorithms for the constrained minimization of the penalty are not really efficient, the above problem is transformed into an equivalent unconstrained minimization by substituting unilateral constraints with additional costs in the penalty (2) as shown in Figure 3 for the lane width.

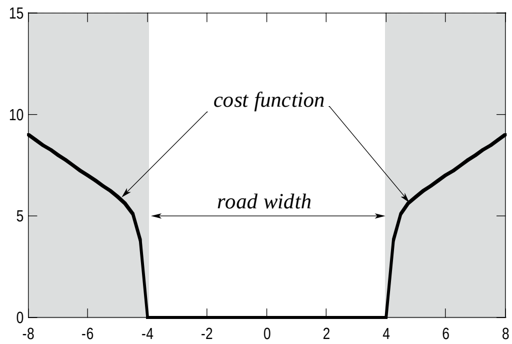

Figure 3: Unilateral cost function for the lane width.

As a consequence, the total cost of the system becomes

$$J = J_{stability} + J_{unilateral} + J_{trajectory}$$

In addition, it has been found that the optimization performance increases if a cost for the yaw vibrations is included, so the following term has been added to $J_{trajectory}$.

$$w_{\psi} \int \left(\dot{\psi}_V - \dot{\psi}_T \right)^2 dt$$

The unconstrained minimization problem is then solved using a genetic algorithm.

4 EVALUATION OF THE SYSTEM TIME RESPONSE

The exact evaluation of the performance index is computationally onerous because it would require the integration of the non-linear system equations. A different approach is proposed: the system behaviour is estimated from the linearized model and time histories are obtained by using analytical Laplace's transform techniques instead of time integration.

Since it is necessary to evaluate the system response for a lot of different gains sets, it is convenient to linearize separately the motorcycle equations (it may be done only once) and the control equations (it must be done for each set of control gains), and finally to assembly the state-space matrices of the controlled vehicle.

4.1 Linearization of the Motorcycle Equations

First of all the non-linear equations of motion (Ref. [5]) are linearized around a proper position depending on the target manoeuvre and the state-space matrices are computed. For example, for a lane-change manoeuvre it is sufficient to linearized around the straight running configuration, whereas for a cornering manoeuvre it is recommended to linearized around the steady cornering configuration. The linearized dependent coordinates formulation takes the following form:

$$\mathbf{A}_{x1} \dot{\mathbf{x}} = \mathbf{A}_{x0} \mathbf{x} + \mathbf{B}_x \mathbf{u}_C$$

where $\mathbf{x}$ is the vector of the *nx* dependent variables, $\mathbf{u}_C$ is the input vector (which contains the steer torque), $\mathbf{A}_{x1}, \mathbf{A}_{x0}, \mathbf{B}_x$ are the state matrices.

By taking into account the *nc* holonomic constraints

$$\mathbf{\Phi}_x \mathbf{x} = \mathbf{0} \qquad (3)$$

the system is converted into the independent formulation

$$\mathbf{A}_{z1} \dot{\mathbf{z}} = \mathbf{A}_{z0} \mathbf{z} + \mathbf{B}_z \mathbf{u}_C \qquad (4)$$

where $\mathbf{z}$ is the vector of the *nz* independent variables, $\mathbf{A}_{z1}, \mathbf{A}_{z0}, \mathbf{B}_z$ are the state matrices. The following relationship holds between dependent and independent variables and matrices:

$$\mathbf{x} = \mathbf{R}_z \mathbf{z} \qquad (5)$$

$$\mathbf{A}_{z1} = \mathbf{R}_z^T \mathbf{A}_{x1} \mathbf{R}_z$$

$$\mathbf{A}_{z0} = \mathbf{R}_z^T \mathbf{A}_{x0} \mathbf{R}_z$$

$$\mathbf{B}_z = \mathbf{R}_z^t \mathbf{B}_x$$

where $\mathbf{R}_z$ is the velocity projection matrix. One way to compute the projection matrix is the use of the SVD decomposition of the constraint Jacobian $\mathbf{\Phi}_x$

$$\mathbf{U}^T \mathbf{\Phi}_x \mathbf{V} = \begin{bmatrix} \mathbf{D} & \mathbf{0} \end{bmatrix}$$

in particular it can be shown (see appendix) that

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_L & \mathbf{R}_z \end{bmatrix} \quad (6)$$

i.e. the last *nz* columns of the right singular vector is the projection matrix $\mathbf{R}_Z$ of size *nx* x *nz*. $\mathbf{U}$ and $\mathbf{V}$ are the singular vectors, $\mathbf{D}$ is a diagonal square matrix containing the singular values of $\mathbf{\Phi}_x$. As an example, this motorcycle model uses 115 dependents state variables which are reduced to 28 after the conversion to the independent formulation.

4.2 Linearization of the Control Equations

It has been explained above how the virtual rider model uses curvilinear coordinates for tracking the vehicle trajectory and for the look-ahead point, uses low-pass filter as well as the integral error of the roll angle. Therefore additional state variables $\mathbf{r}$ and corresponding state equations are introduced.

The linearization of the PID equations (1) which transform the target motion $\mathbf{u}_K$ into the motorcycle inputs $\mathbf{u}_C$ leads to the following equation:

$$\mathbf{u}_C = \mathbf{E}_x \mathbf{x} + \mathbf{E}_K \mathbf{u}_K + \mathbf{E}_r \mathbf{r} \quad (7)$$

where the target motion vector $\mathbf{u}_K$ contains the curvature of the road path Θ, the target lateral position n_T, the target roll angle ϕ_T.

The linearization of the control state equations yields to:

$$\mathbf{A}^*_{x1}\dot{\mathbf{x}} + \mathbf{A}^*_{r1}\dot{\mathbf{r}} = \mathbf{A}^*_{x0}\mathbf{x} + \mathbf{A}^*_{r0}\mathbf{r} + \mathbf{B}^*\mathbf{u}_K$$

where matrix $\mathbf{A}^*_{x1}, \mathbf{A}^*_{x0}$ are related to the dependent coordinates formulation and must be transformed into the independent one as follows:

$$\mathbf{A}^*_{z1} = \mathbf{A}^*_{x1}\mathbf{R}_z$$

$$\mathbf{A}^*_{z0} = \mathbf{A}^*_{x0}\mathbf{R}_z$$

It is now possible to write the control state equations in the independent formulation:

$$\mathbf{A}^*_{z1}\dot{\mathbf{z}} + \mathbf{A}^*_{r1}\dot{\mathbf{r}} = \mathbf{A}^*_{z0}\mathbf{z} + \mathbf{A}^*_{r0}\mathbf{r} + \mathbf{B}^*\mathbf{u}_K$$

Finally PID equations (7) may be transformed in the independent formulation

$$\begin{aligned} \mathbf{u}_C &= \left(\mathbf{E}_x \mathbf{R}_z\right)\mathbf{z} + \mathbf{E}_r \mathbf{r} + \mathbf{E}_K \mathbf{u}_K \\ &= \mathbf{E}_z \mathbf{z} + \mathbf{E}_r \mathbf{r} + \mathbf{E}_K \mathbf{u}_K \end{aligned}$$

4.3 Assembly of the Controlled Motorcycle State-Space Matrices

After having linearized the motorcycle equations (just once) and the control equations (for each set of control gains) it is necessary to assembly the state-space matrices of the controlled motorcycle.

By substituting in the motorcycle equations the vehicle inputs $\mathbf{u}_C$ with the target motion $\mathbf{u}_K$ one obtains

$$\begin{aligned} \mathbf{A}_{z1}\dot{\mathbf{z}} &= \mathbf{A}_{z0}\mathbf{z} + \mathbf{B}_z\left(\mathbf{E}_z\mathbf{z} + \mathbf{E}_r\mathbf{r} + \mathbf{E}_K\mathbf{u}_K\right) \\ &= \left(\mathbf{A}_{z0} + \mathbf{B}_z\mathbf{E}_z\right)\mathbf{z} + \left(\mathbf{B}_z\mathbf{E}_r\right)\mathbf{r} + \left(\mathbf{B}_z\mathbf{E}_K\right)\mathbf{u}_K \end{aligned}$$

By assembling both motorcycle and control equations one obtains

$$\begin{bmatrix} \mathbf{A}_{z1} & \mathbf{0} \\ \mathbf{A}^*_{z1} & \mathbf{A}^*_{r1} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{z}} \\ \dot{\mathbf{r}} \end{Bmatrix} = \begin{bmatrix} \mathbf{A}_{z0} + \mathbf{B}_z \boxed{\mathbf{E}_z} & \mathbf{B}_z \boxed{\mathbf{E}_r} \\ \mathbf{A}^*_{z0} & \mathbf{A}^*_{r0} \end{bmatrix} \begin{Bmatrix} \mathbf{z} \\ \mathbf{r} \end{Bmatrix} + \begin{bmatrix} \mathbf{B}_z \boxed{\mathbf{E}_K} \\ \mathbf{B}^* \end{bmatrix} \begin{Bmatrix} \mathbf{u}_K \end{Bmatrix} \quad (8)$$

The only matrices which depend on the control gains and must be computer each time are $\mathbf{E}_z, \mathbf{E}_r, \mathbf{E}_K$, whereas other matrices are constant. However, if the look-ahead distance or the steering torque filter characteristics are changed during optimization all control matrices $\mathbf{A}^*, \mathbf{B}^*$ must be recomputed.

As well as the output is concerned the following equations hold

$$\begin{aligned} \{\mathbf{y}\} &= [\mathbf{C}_x\ \mathbf{C}_r] \begin{Bmatrix} \mathbf{x} \\ \mathbf{r} \end{Bmatrix} + [\mathbf{D}_K]\{\mathbf{u}_K\} = \\ &= [\mathbf{C}_z\ \mathbf{C}_r] \begin{Bmatrix} \mathbf{z} \\ \mathbf{r} \end{Bmatrix} + [\mathbf{D}_K]\{\mathbf{u}_K\} \end{aligned} \quad (9)$$

where

$$\mathbf{C}_z = \mathbf{C}_x \mathbf{R}_z$$

Once again matrices may be computed just once because the do not depend on the control gains.

4.4 Evaluation of Time Response using Analytical Inverse Laplace Transform

Once the system has been linearized around a proper position, the time histories can be obtained by using Laplace's transform techniques, which make it possible to estimate the vehicle motion without integrating the equations of motion.

The computation of complex eigenvalues $\boldsymbol{\lambda}$ and eigenvectors $\mathbf{T}_z$ of the state matrix $\mathbf{A}_z$ makes it possible to transform the system (8),(9) into the following principal coordinates formulation:

$$\begin{aligned} \dot{\mathbf{q}} &= \boldsymbol{\Lambda}\,\mathbf{q} + \mathbf{B}_q \mathbf{u}_K \\ \mathbf{y} &= \mathbf{C}_q \mathbf{q} + \mathbf{D}\,\mathbf{u}_K \end{aligned} \quad (10)$$

where $\mathbf{q} = \mathbf{T}_z^{-1}\mathbf{z}$ is the principal coordinates vector, $\boldsymbol{\Lambda}$ is a diagonal matrix whose elements are the eigenvalues $\boldsymbol{\lambda}$, 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 is very effective because the system modes are uncoupled, even if the new state variables $\mathbf{q}$ are complex and so they loose the original physical meaning. Applying Laplace's Transform to (10) one obtains:

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

while output variables may be computed as follows

$$\mathbf{Y}(s) = \mathbf{C}_q\mathbf{Q}(s) + \mathbf{D}\,\mathbf{U}(s) = \left[\mathbf{C}_q\left(\mathbf{I}s - \boldsymbol{\Lambda}\right)^{-1}\mathbf{B}_q + \mathbf{D}\right]\mathbf{U}(s) \quad (11)$$

Since matrix $\mathbf{I}s-\mathbf{\Lambda}$ is diagonal, it is very easy to compute the inverse, and equation (11) may be re-written as follows (modal decomposition)

$$\mathbf{Y}(s)=\mathbf{C}_q\begin{bmatrix}\ddots & 0 & 0\\ 0 & \frac{1}{s-\lambda_i} & 0\\ 0 & 0 & \ddots\end{bmatrix}\mathbf{B}_q\mathbf{U}(s)+\mathbf{D}\mathbf{U}(s)$$

The application of the inverse Laplace transform makes it possible to move back into time domain where the only difficult is the computation of the inverse transform of $\mathbf{U}(s)/\left(s-\lambda_i\right)$ because in the presence of a generic input $\mathbf{u}(t)$ the computation 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.), i.e.

$$u_i(t)=\sum_k f_k(t) \quad U_i(s)=\sum_k F_k(s)$$

the computation can be done analytically. In conclusion, the implementation of a moderate number of shape functions f_k and $L^{-1}(F_k/(s-\lambda))$ makes it possible to describe system inputs and compute system outputs in a very easy and computationally efficient way.

5 EXAMPLE OF APPLICATION

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 described in Ref. [5]. The first example consists in the control optimization and non-linear time simulation of a lane change manoeuvre: the lane width is 4m and the manoeuvre 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. The target motion, the inequality constraints as well as the non-linear simulation of the manoeuvre are depicted in Figure 4, which also shows data recording during a real manoeuvre.

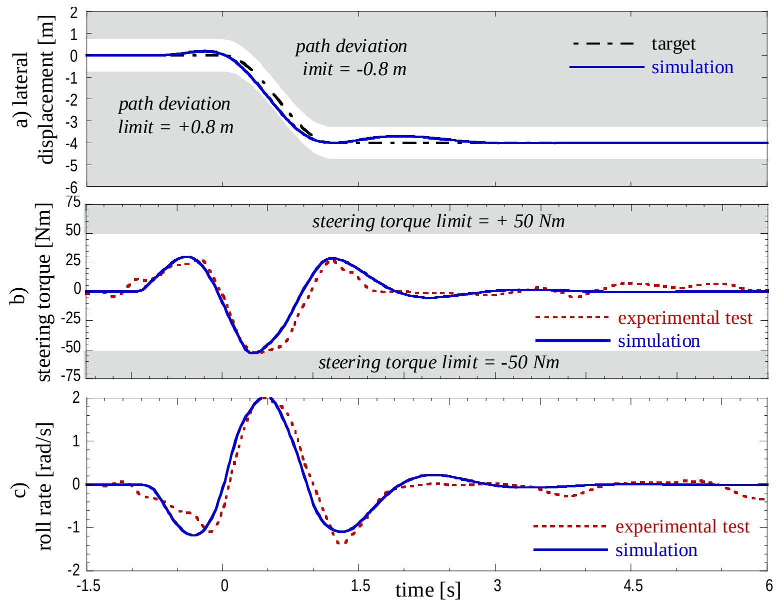

Figure 4: Lane change maneuver (speed = 18 m/s)

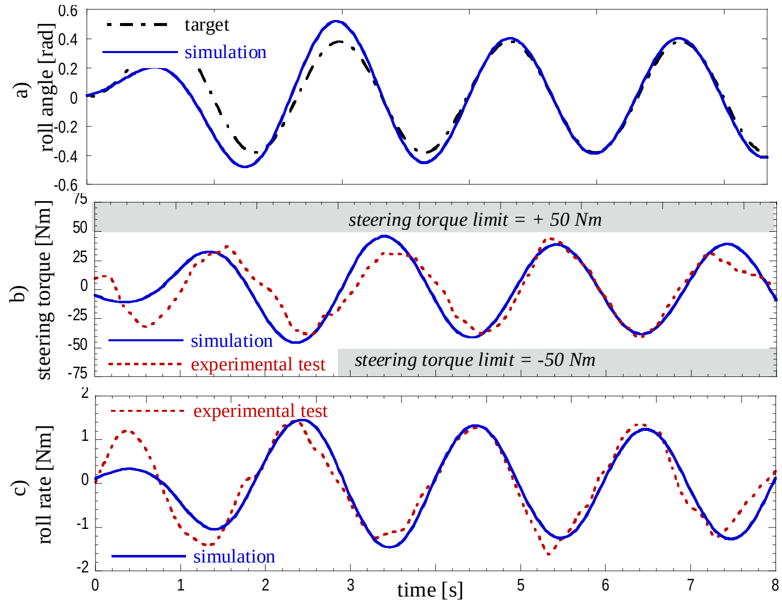

Figure 5: Slalom maneuver (speed = 21.3 m/s, cone distance = 21 m)

The second example (Figure 5) is about a slalom simulation at a speed of 21.3 m/s with a distance between cones of 21 m. The optimized control parameters are the same as of lane change.

The third example consists in the control optimization and non-linear time simulation of a cornering manoeuvre at a speed of 22 m/s and cornering radius equal to 50m (Figure 6). This manoeuvre 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 (~51°). Moreover, the transition between straight running and cornering condition in a time of 1.2s is fast. The equations of

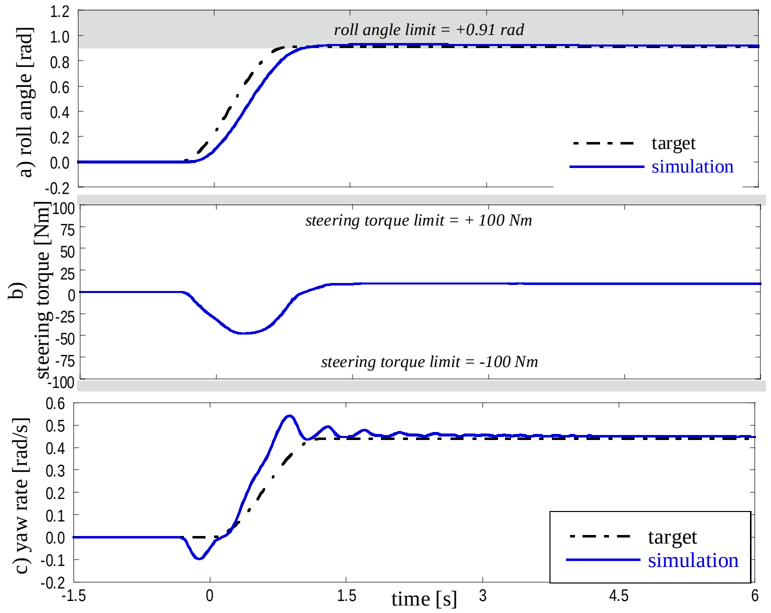

Figure 6: Cornering maneuver (speed = 22 m/s)

motion have been linearized in the neighbourhood 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}. For this manoeuvre it was essential to restrain the roll angle because any overshoot would be unrealistic and could cause the motorcycle fall down

6 CONCLUSIONS

An automatic way for the non-linear control optimization in presence of unilateral constraints has been presented. The application to the control of a non-linear motorcycle model is discussed and a comparison between experimental test and simulations is included.

The block diagram of the algorithm is depicted in Figure 7: first the system (vehicle plus control) is linearized around a proper equilibrium position (*Vehicle & Control*) and the target motion is defined (*Target Manoeuvre*). Secondly the *Modal Decomposition* is carried out and the target manoeuvre is decomposed into basic tiles having an analytic inverse Laplace transformation (*Maneuver Decomposition into basic tiles*). At the end the time response is evaluated by computing the inverse Laplace transform (*Evaluation of the Time Response* of Figure 7).

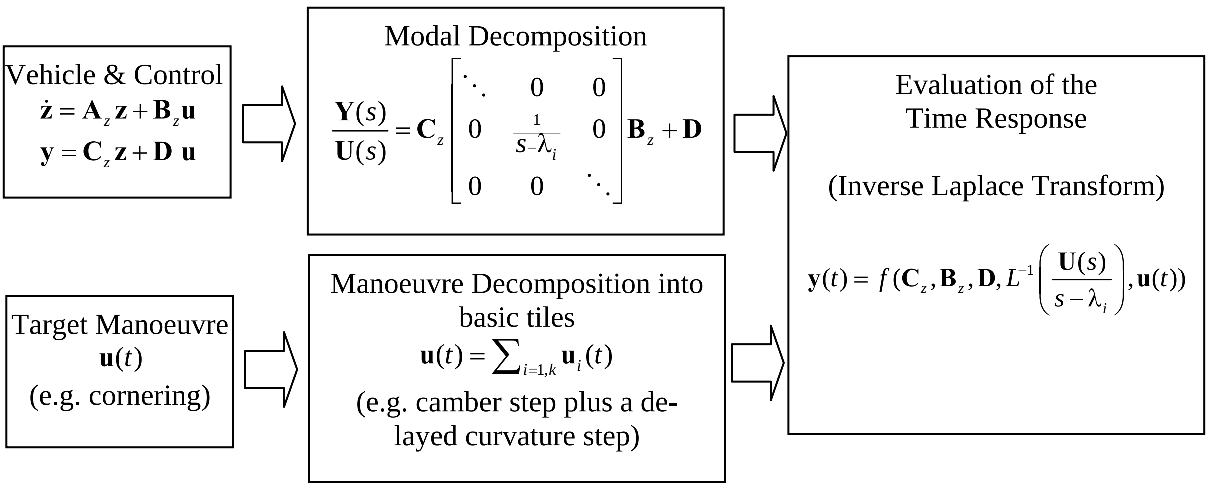

Figure 7: Time Response Evaluation

Finally, it can be observed that:

- for a proper control of the motorcycle it is essential to use the look-ahead values of the target parameters;
- the use of the Laplace technique allows an efficient gains optimization;
- to speed up the optimization algorithm it is necessary a smart approach to linearization, i.e. only parts of the state matrices should be computed at each step of the optimization;
- the comparison with experimental test shows that the presented algorithm is effective and reproduces human behavior.

REFERENCES

[1] V.Cossalter. *Motorcycle Dynamics*. Lulu.com, 2006, ISBN: 978-1-4303-0861-4

[2] P. Antos and Ambrosio. A Control Strategy for Vehicle Trajectory Tracking using Multibody Models, *Multibody System Dynamics*, 11, 365-394, 2004.

[3] F. Biral and M. Da Lio. Modelling drivers with the optimal manoeuvre method, Proceedings of the ATA Conference, Florence, Italy, 2001.

[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*, 38, 311-317, 1992.

[5] V. Cossalter, R. Lot. A motorcycle multi-body model for real time simulations based on the natural coordinates approach. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, Vol., 37, 423-447, 2002.

[6] García de Jalón and Bayo. *Kinematic and Dynamic Simulation of Multibody Systems. The Real-Time Challenge*. ISBN 0-387-94096-0, Springer-Verlag, New-York, 1994.

7 APPENDIX: COMPUTATION OF THE PROJECTION MATRICES

This appendix explains how to convert the dependent coordinates state-space equations

$$\mathbf{A}_{x1}\dot{\mathbf{x}} = \mathbf{A}_{x0}\mathbf{x} + \mathbf{B}_x\mathbf{u} \quad \text{(a 1)}$$

subjected to the constraints equations

$$\mathbf{\Phi}_x\mathbf{x} + \mathbf{\Phi}_u\mathbf{u}_C = \mathbf{0} \quad \text{(a 2)}$$

into the following independent coordinates formulation

$$\mathbf{A}_{z1}\dot{\mathbf{z}} = \mathbf{A}_{z0}\mathbf{z} + \mathbf{B}_z\mathbf{u} \quad \text{(a 3)}$$

using projection matrices

$$\mathbf{x} = \mathbf{R}_z\mathbf{z} + \mathbf{R}_u\mathbf{u}_C \quad \text{(a 4)}$$

By substituting (a 4) into (a 2) one obtains

$$\mathbf{\Phi}_x\mathbf{R}_z\mathbf{z} + (\mathbf{\Phi}_x\mathbf{R}_u + \mathbf{\Phi}_u)\mathbf{u}_C = \mathbf{0}$$

Since the equation above must be satisfied for any value of $\mathbf{z}$ and $\mathbf{u}_C$ one obtains

$$\begin{cases} \mathbf{\Phi}_x\mathbf{R}_z = \mathbf{0} & \text{(a 5)} \\ \mathbf{\Phi}_x\mathbf{R}_u + \mathbf{\Phi}_u = \mathbf{0} & \text{(a 6)} \end{cases}$$

The SVD decomposition of $\mathbf{\Phi}_x$ leads to

$$\mathbf{U}^T\mathbf{\Phi}_x\mathbf{V} = \begin{bmatrix}\mathbf{D} & \mathbf{0}\end{bmatrix},\ \mathbf{V} = \begin{bmatrix}\mathbf{V}_L & \mathbf{V}_R\end{bmatrix}$$

Assuming $\mathbf{R}_z = \mathbf{V}_R$ and introducing in (a 5) one obtains $\mathbf{\Phi}_x\mathbf{V}_R = \mathbf{0}$, i.e.

$$\mathbf{U}\begin{bmatrix}\mathbf{D} & \mathbf{0}\end{bmatrix}\mathbf{V}^T\mathbf{V}_R = \mathbf{U}\begin{bmatrix}\mathbf{D} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{V}_L^T \\ \mathbf{V}_R^T\end{bmatrix}\mathbf{V}_R = \mathbf{U}\mathbf{D}\mathbf{V}_L^T\mathbf{V}_R = \mathbf{0}$$

since $\mathbf{V}_L^T\mathbf{V}_R = \mathbf{0}$ (the columns of $\mathbf{V}$ are orthogonal each other). It has been proved that $\mathbf{R}_z = \mathbf{V}_R$

In similar manner, by introducing $\mathbf{R}_u = \mathbf{V}\mathbf{S}$ (where $\mathbf{S}$ is unknown) into (a 6) one obtains:

$$\mathbf{\Phi}_x\mathbf{V}\mathbf{S} + \mathbf{\Phi}_u = \mathbf{0}$$

$$\mathbf{U}\begin{bmatrix}\mathbf{D} & \mathbf{0}\end{bmatrix}\mathbf{V}^T\mathbf{V}\mathbf{S} + \mathbf{\Phi}_u = \mathbf{0}$$

considering that $\mathbf{V}$ is an orthogonal matrix, i.e. $\mathbf{V}^T\mathbf{V} = \mathbf{1}$

$$\mathbf{U}\begin{bmatrix}\mathbf{D} & \mathbf{0}\end{bmatrix}\mathbf{S} + \mathbf{\Phi}_u = \mathbf{0}$$

$$\begin{bmatrix}\mathbf{D} & \mathbf{0}\end{bmatrix}\mathbf{S} = \begin{bmatrix}\mathbf{D} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{S}_U \\ \mathbf{S}_L\end{bmatrix} = \mathbf{D}\mathbf{S}_U = -\mathbf{U}^T\mathbf{\Phi}_u \quad \textbf{(a 7)}$$

$$\mathbf{S}_U = -\mathbf{D}^{-1}\mathbf{U}^T\mathbf{\Phi}_u$$

whereas $\mathbf{S}_L$ can be everything (since it is pre-multiply by a zero matrix), in particular we can assume $\mathbf{S}_L = \mathbf{0}$, and therefore

$$\mathbf{R}_u = \begin{bmatrix}\mathbf{V}_L & \mathbf{V}_R\end{bmatrix}\begin{bmatrix}\mathbf{S}_U \\ \mathbf{S}_L\end{bmatrix} = \begin{bmatrix}\mathbf{V}_L & \mathbf{V}_R\end{bmatrix}\begin{bmatrix}\mathbf{S}_U \\ \mathbf{0}\end{bmatrix} = \mathbf{V}_L\mathbf{S}_U \quad \text{(a 8)}$$

If we insert (a 7) into (a 8) we finally obtain $\mathbf{R}_u = -\mathbf{V}_L\mathbf{D}^{-1}\mathbf{U}^T\mathbf{\Phi}_u$.