2004.08.02

A Model for Motorcycle Rider Operation Based on Genetic Algorithms

Shigeru Fujii Sergey A. Panfilov Sergey V. Ulyanov
●Advanced Technology Research Div. Research & Development Operations / Yamaha Motor Europe N.V.

Abstract

In order to program computer simulations for a running motorcycle it is necessary to create a model of the motorcycle body plant and also a model for the rider's operational actions. In this research project we developed a motorcycle rider operation model using genetic algorithms to replace the operational actions of the rider and make the motorcycle follow a specified course. We created the motorcycle body model using the SimMechanics, mechanical modeling tool of the MATLAB software. This model is made to represent a scooter and consists of seven rigid bodies and 12 degrees of freedom. The rider operation model adopts a control method in which the machine roll angle is kept coinciding with the target roll angle, which is calculated so as to reach the prescribed course point at the forward reference length. Furthermore, the gain parameters used in the control method were optimized by genetic algorithms used in performing simulations at set velocities and roll angles. The running simulations using this rider operation model were performed over a velocity range of 1-15 m/s and were successful in having the machine run the designated course with good stability. Furthermore, a good level of agreement was achieved between the results of actual test runs and the simulation runs.

1 INTRODUCTION

In recent years, several types of motorcycle operation models have been proposed[1-4]. These include examples where the data from rider operational actions are recreated into a model and examples of models in which various methods are used to achieve a balance of roll angle stability and good course following ability. All of these are informative and useful. However, the fact that much time is required to create control parameters that can achieve stable running for each different type of running condition, is probably the reason why practical use of simulations is not made in motorcycle development work. Here we introduce the case of a motorcycle rider operation model we developed using genetic algorithms (GA) to enable the running of simulations in which the velocity and course can be set freely when applied to a scooter body model (**Figures 1-3**) created using the SimMechanics option of MATLAB/Simlink.

Figure 1. Actual Scooter

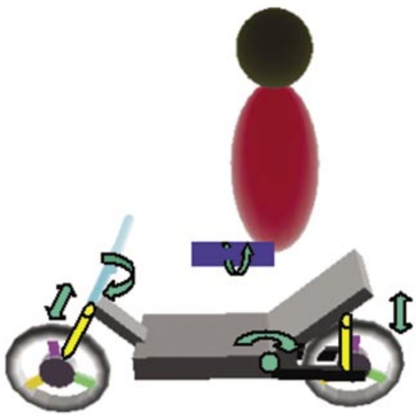


Figure 2. Motorcycle Plant Model

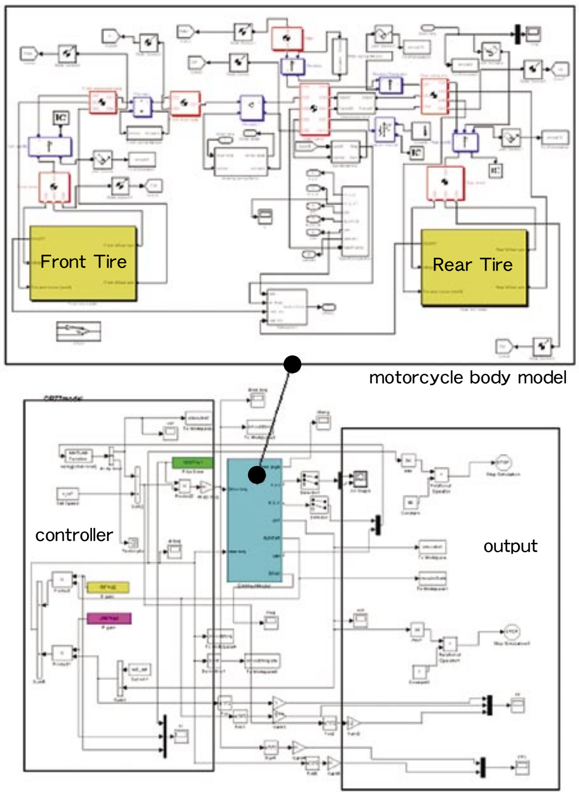


Figure 3. Motorcycle Body Model Circuit Diagram

2 THE MOTORCYCLE BODY MODEL

The motorcycle body model was created using SimMechanics. In concept, it consists of seven rigid bodies (main frame, rider, steering assembly, lower part of the front suspension, rear arm, front and rear wheels), five revolution joints (steering, rear arm pivot, rider lean, front and rear wheels), one extension/contraction joint (front suspension) and four spring dampers (front and rear suspensions, rider lean, steering). The model involves 12 degrees of freedom, six related to the standard points of the main frame (front-rear, right-left, up-down, roll angle, pitch angle and yaw angle) and six more related to the revolutions of the front and rear wheels, the compression and rebound of the front suspension, the angle of rotation of the steering column, rider roll angle and angle of rotation of the rear arm. The external forces included were gravity, air resistance, lift and tire load.

The input data for each rigid body included the coordinates of the center of gravity and the connection points, weight and inertial moment. Constant values for the springs, the dampers and the tires were also inputted. For the tire model, a linear form tire model with first-order lag was created using Simlink. This tire model produces calculations for longitudinal and lateral forces, self-aligning torque and overturning moment at the point of contact between the tire and the road surface and the relaxation lengths. The calculations are based on parameter values for each tire and the information about the relationship between the tire and road surface (height, camber angle, lateral slip angle and longitudinal slip rate). Because this model is based on linear premises, it cannot function when the values representing lateral slip or roll angle become so large that they exceed the formula bounds, but it is applicable within the bounds of normal scooter riding conditions. The parameters for the tire model were based mainly on data measured by JARI(Japan Automobile Research Institute) In creating our model, we referred to the SimMechanics demo model created by The Mathworks, Inc. based on reference material[5].

3 THE RIDER OPERATION MODEL

As shown in **Figure 3**, our riding simulations were conducted in a format in which the rider operation model created using Simlink controls the motorcycle body model described in the previous section 2. The rider operation model calculates drive torque and steering torque according to equations (1), (2). In other words, this is a model in which the effect of the steering damper is added to the P (D) control applied to bring the machine velocity and roll angle closer to the target velocity and roll angle values. Although as an item steering damper effect is usually included as a function in the motorcycle body model, because the friction force produced in the steering assembly is usually so small and is actually the force applied by the rider, we included it in equation (2).

For simulations in which the motorcycle follows a prescribed course, the target roll angle is calculated with equations (3)-(6). In other words, this target roll angle is calculated so as to reach the prescribed course at the forward reference length through a circular course(**Figure 4**).

$$\tau_d = -K_{p1}(v-v_{ref}) \quad (1)$$

$$\tau_s = -K_{p2}(\phi - \phi_{ref})-K_{d2}\phi'-K_{d3}\alpha' \quad (2)$$

$$\omega = v/R \quad (3)$$

$$a = v^*\omega \quad (4)$$

From the geometric relationships of **Figure 4**:

$$R = (y^2+LR^2)/2y \quad (5)$$

$$\phi_{ref} = \tan^{-1}(a/g) = \tan^{-1}\left(\frac{2y^*v^2}{g^*(y^2+LR^2)}\right) \quad (6)$$

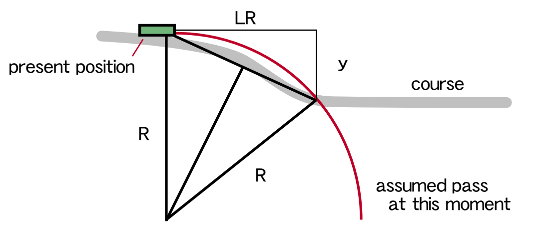


Figure 4. Assumed Pass Route While Following a Defined Course

In the following, we will discuss the method we used to design the fuzzy controller for setting our model's gain parameters, in other words the $Kp1$、$Kp2$、$Kd2$、$Kd3$ values of equations (1), (2), at values corresponding to different conditions. As shown in **Figure 5**, the method used here employs a first stage in which the ideal gain is sought for a number of target velocity and target roll angle values and a second stage in which a fuzzy controller is created to connect the gaps between the scattered ideal values smoothly. This model is a motorcycle-use adaptation of the SCoptimizer design software that uses GA to produce an optimum fuzzy controller[6].

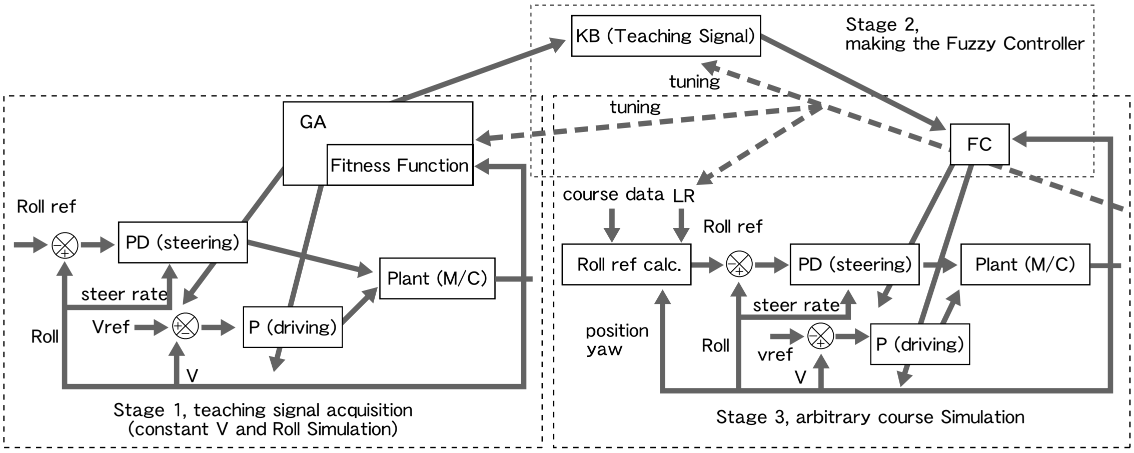


Figure 5. Structure of Fuzzy Controller and its Development

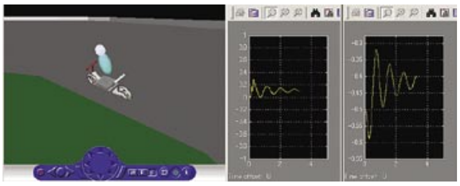


Figure 6. The optimizing process immediately after the start of optimizing

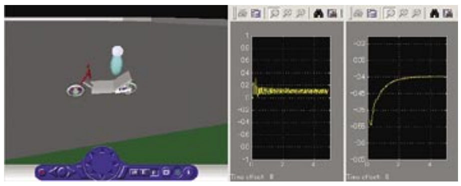


Figure 7. The optimizing process after a given amount of optimizing

3.1 Simulation at Set Velocity and Set Roll Angle

Motorcycle rider operation involves following the desired course while maintaining a stable roll angle. And, believing that the basic element in creating a rider operation model is to find a control method capable of maintaining a stable roll angle as the machine runs, our first-stage aim was to seek optimum gain values for each target value while running simulations with the velocity and roll angle set at single values. Using SCoptimizer, values were set for target velocity, target roll angle, GA parameters, fitness function, etc., and optimum gain values sought by running simulations(**Figure 6** and **7**) with the running model shown in **Figure 3**. Since the optimization of the GA depends on the setting of the fitness function, it is necessary to use a process of trial and error to adjust the weighting factors of the fitness function based on the results of the simulation runs. **Figure 8** shows gain sets produced when uniform weighting factors of the fitness function were applied in all cases without a process of trial and error.

Target velocities = 1,2,3,5,7,10,15 m/s

Target roll angle = 0°,2°,10°,20°,30°

Optimized gains $K_{p1},K_{p2},K_{d2},K_{d3}$

GA parameters

Population size = 200,

Number of generations = 5

Mutation rate = 0.01, Crossover rate = 0.64,

Delete rate = 0.8

Fitness function

$$FF = (v-v_{ref})^2 W_v + (\phi - \phi_{ref})^2 W_\phi + \phi'^2 W_{d\phi} + \tau_d^2 W_{\tau d} + \tau_s^2 W_{\tau s} \quad (7)$$

$$W_v = 0.2, W_\phi = 1, W_{d\phi} = 1, W_{\tau d} = 0.0001, W_{\tau s} = 0.0005$$

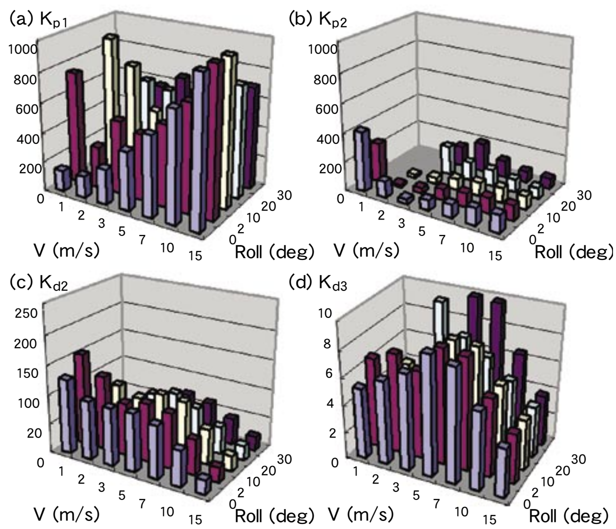


Figure 8. Example of Gain Sets Produced by Uniform Weighting Factor for Fitness Function

3.2 Creating the Fuzzy Controller

Using as teaching signals the gain sets obtained by the method described in the previous section, we used SCoptimizer software to produce a fuzzy controller capable of selecting the suitable gain values for the conditions at each moment. The input data for SCoptimizer consisted of gain sets of the type shown in **Figure 8** and the conditions (velocity, roll angle) were converted into fuzzy input values by setting membership functions as shown in **Figure 9**. Then, using GA, fuzzy rules were generated to produce smooth parameter modulations in accordance with the teaching signals of the gain sets.

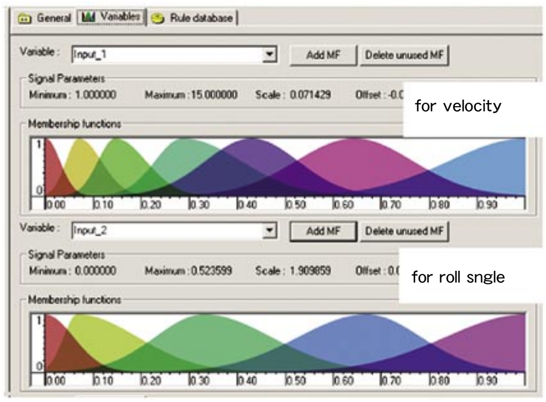
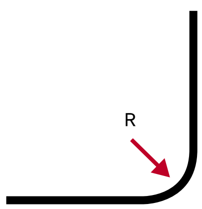


Figure 9. Membership Functions



v (m/s)	R (m)
1	2
2	4
3	6
5	10
7	14
10	20
15	40

Figure 10. Left Turning Course

3.3 Comparison of Course Follow Simulations and Actual Test Run Data

Using the fuzzy controller created on the basis of the gain sets in **Figure 8**, simulations were run for a left turning course of the type shown in **Figure 10** which were created by setting the turning radius (R) for

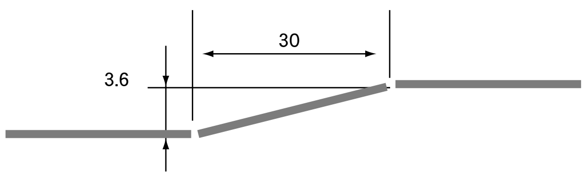


Figure 11. Lane Change Course

each velocity, a lane change course as shown in **Figure 11** and a circuit course as shown in **Figure 12**. The results showed a tendency for course following capability to be deficient in the 1-7 m/s velocity range, while in the case of a 15 m/s velocity the steering torque showed a tendency for extreme fluctuation. Also, simulations run for following the circuit course at velocities in the range of 4-6 m/s produced lines that went far off the course, as seen in Track A of **Figure 13**.

To remedy these types of incongruities, we began GA optimization by adjusting the weighting values for the fitness function used in the simulations run at set velocity and roll angle. This corrected the results in the 1-3 m/s range, but not in the 4-7m/s range. However, gain values with a good balance of course following ability and stability performance were achieved for the simulations run at set velocity and roll angle when the steering gain K_{d3} was not included in the optimizing process and given a set value while the other three parameters were subjected to optimizing. Also we tried treating both K_{d3} and LR separately and determined values for them based on simulations run for the left turn course and the lane change course, etc. These improvements produced gains of the type shown in **Figure 14**. When a fuzzy controller based on these gains was employed, it produced the result shown as Track B in **Figure 13**. This track aligns with the Defined Course almost exactly.

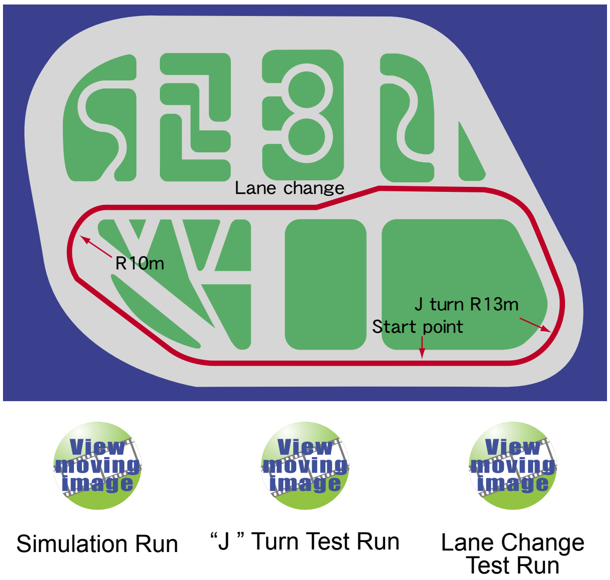


Figure 12. Circuit Course

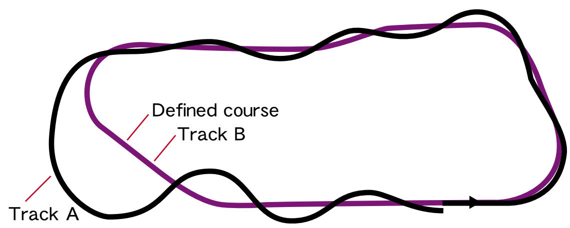


Figure 13. Tracks Run of Circuit Course

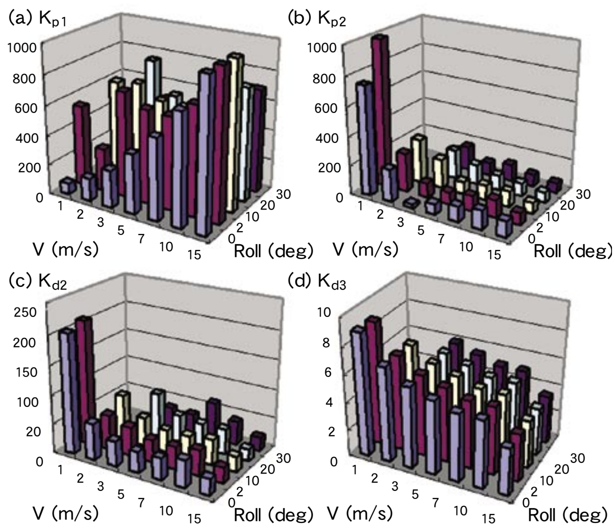


Figure 14. Optimized Gain Sets

Figure 15(a)-(e) shows a comparison of simulation results and actual test run data produced by a rider running the same type of course. The velocity results naturally match closely because the smoothed out results of the rider test runs were used for the simulation target velocity, but the simulation results for the other parameters also matched the test results closely. **Figure 16** shows the changes in gain during the simulation.

Also, shown in video form is the simulation run for the course shown in **Figure 17** involving large fluctuations in velocity. As this video shows, the controllers we created were able to handle operation simulations run at a wide range of velocities.

4 CONCLUSION

We created a fuzzy controller for operation in a simulation model for motorcycle running that used GA in the optimization method. This fuzzy controller was able to run with stability at a wide range of velocities over a large variety of courses. Also the simulation results matched well with actual test run data.

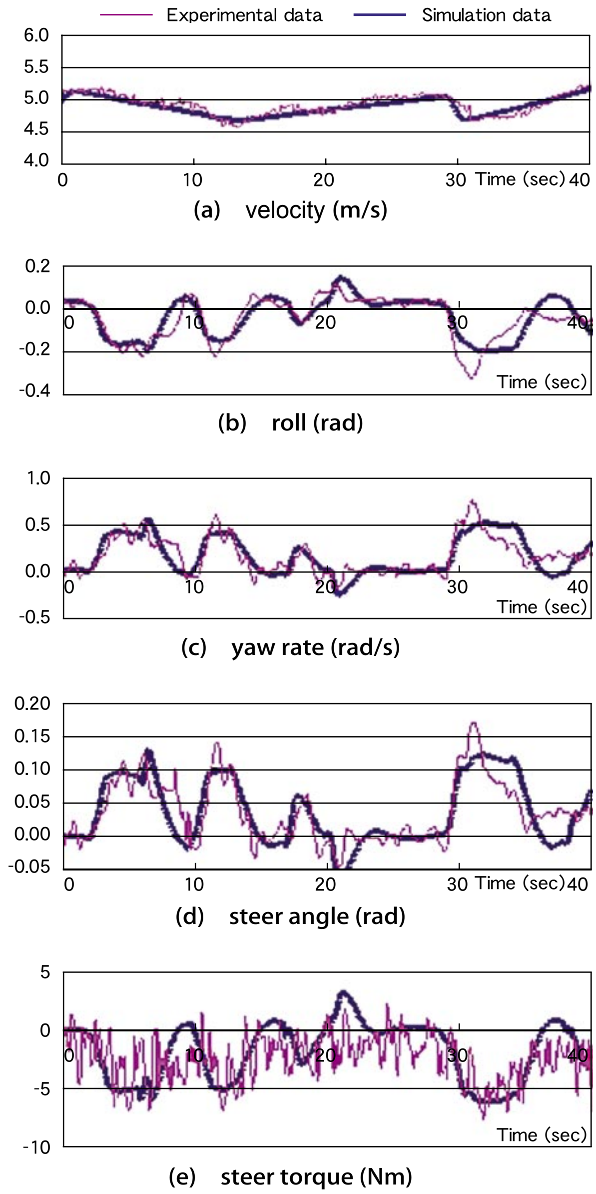


Figure 15. Comparison of Simulation with Test Data for Circuit Course Run

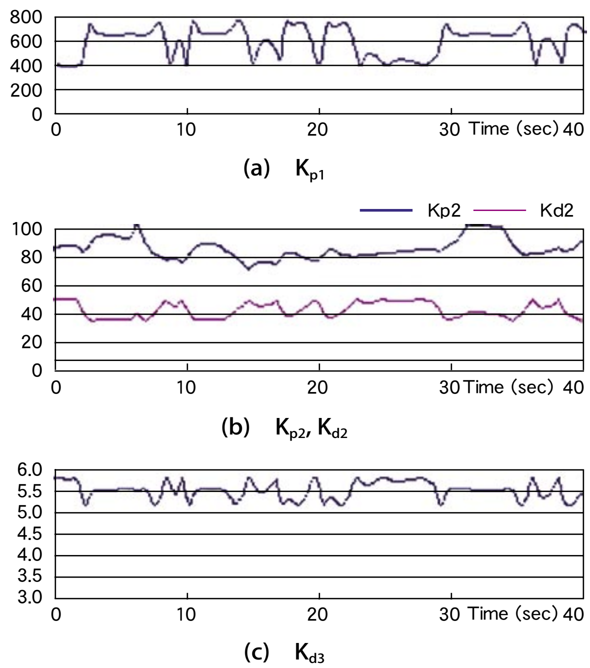
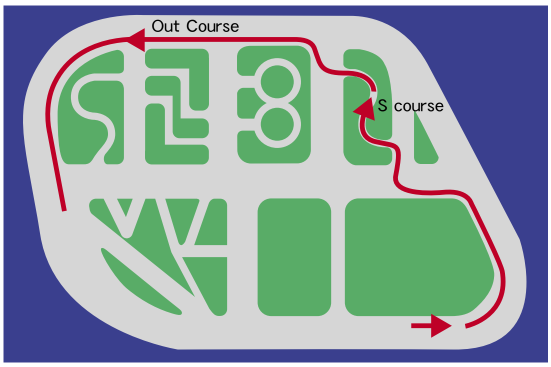


Figure 16. Change in Gain During Simulation



View moving image

Figure 17. S Course

NOTATION

τ_d : drive torque(Nm)
K_{p1},K_{p2},K_{d2},K_{d3}: gain
v_{ref} : reference speed(m/s)
ϕ_{ref} : refernce roll angle(rad)
α : ateering angle(rad)
R : course radius(m)
LR : length of reference(m)
g : gravity acceleration(m/s^2)
W_v,W_ϕ,$W_{d\phi}$,$W_{\tau d}$,$W_{\tau s}$: weight factor
τ_s : steering torque(Nm)
v : forward speed(m/s)
ϕ : roll angle(rad)
ϕ' : roll rate(rad/s)
ω : yaw rate(rad/s)
a : required lateral acceleration(m/s^2)
y : deviation at LR(m)
FF : fitness function

REFERENCES

[1] H.Imaizumi, T.Fujioka: Motorcycle Motion Analysis using Mechanical System Simulation Language (in Japanese with English summary), Proceedings of JSAE No. 942, pp. 21-24 (1994)

[2] Y.Ohwada, I.Kageyama: A Study of a Rider Model for Two Wheeled Vehicle Handling (in Japanese with English summary), Proceedings of JSAE No. 942, pp. 41-44 (1994)

[3] T.Breuer, D.A.Pruckner: Advanced Dynamic Motorbike Analysis and Driver Simulation, Studienarbeit am Institut fur Kraftfahrwesen Aachen (1999)

[4] H.Watanabe: Motorcycle Running Simulation Using a DADS/Plant (in Japanese), Matlab Automotive user's meeting (2000)

[5] D.J.N. Limebeer, et al.: The stability of motorcycles under acceleration and braking. Proceedings Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science, (2001), vol. 215, no. 9, pp. 1095-1110

[6] S.A.Panfilov, et al.: Soft Computing Optimizer: Structure and SW for Simulation of Universal Fuzzy Approximators, Second International Conference on Soft Computing and Computing with Words in System Analysis, Decision and Control (ICSCCW'03), pp. 112-122 (2003)

AUTHOR

Sergey A. Panfilov, Sergey V. Ulyanov, Shigeru Fujii