



This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.


C O M M O N S D E E D

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:



Attribution. You must attribute the work in the manner specified by the author or licensor.



Noncommercial. You may not use this work for commercial purposes.



No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

On the Synthesis of Driver Inputs for the Simulation of Closed-Loop Handling Manoeuvres

T.J.Gordon⁽¹⁾ and M.C.Best⁽²⁾

(1) University of Michigan Transportation Research Institute and Dept. of Mechanical Engineering
2901 Baxter Road, Ann Arbor, MI 48109, USA

E-mail tjgordon@umich.edu

(2) Department of Aeronautical and Automotive Engineering
Loughborough University, Loughborough, Leics., LE11 3TU, England

ABSTRACT

This paper concerns a new ‘Dual Model’ methodology for the synthesis of steering, throttle and braking inputs for the closed-loop simulation of linear or non-linear vehicle handling dynamics. The method provides near-optimal driver control inputs that are both insensitive to driver model assumptions, and feasible for use with complex non-linear vehicle handling models. The paper describes the Dual Model technique, and evaluates its effectiveness, in the context of a low-order non-linear handling model, via comparison with independently derived optimal control inputs. A test case of an obstacle avoidance manoeuvre is considered. The methodology is particularly applicable to the design and development of future chassis control systems.

Keywords: Vehicle Dynamics, Optimization, Vehicle Control, Handling, Driver Modelling, Vehicle Design

Biographical Notes:

Tim Gordon is based at the University of Michigan, USA, working as a Research Professor in the Transportation Research Institute (UMTRI) and as Professor in the Department of Mechanical Engineering. He obtained his Bachelors in Mathematics and Doctorate in Theoretical Physics at Cambridge University, England, and worked previously as lecturer in Engineering Mathematics, Senior Lecturer in Dynamics and Control, and Professor of Automotive Engineering at Loughborough University, England.

Matthew Best is Lecturer in Dynamics and Control in the Department of Aeronautical and Automotive Engineering, Loughborough University, England. He obtained a Bachelors degree in Mathematical Engineering and his doctoral thesis was centred on the design and implementation of Kalman filters for semi-active suspension systems. His current research is in control, system identification and state estimation for automotive chassis and powertrain systems.

1. INTRODUCTION

This paper concerns the synthesis of steering, throttle and braking inputs for the simulation of vehicle handling dynamics, using a new reduced form of nonlinear control optimisation. There are many situations that require such control inputs, particularly in simulation studies aimed at improving vehicle design for dynamic handling performance. Traditionally, vehicle handling dynamics has focussed on open-loop dynamic responses, in the form of ‘objective handling tests’ (Gillespie. Here the control inputs are prescribed as functions of time ahead of the test (or simulation), for example as random, sinusoidal or step-like inputs. With the development of driver models, it has become increasingly common to evaluate vehicle, and especially chassis system, performance using closed-loop driver control, where some initial conditions and road geometry variables define the manoeuvre, and some control synthesis procedure leads to the dynamic control inputs for steering, throttle and brakes (e.g. Modjtahedzadeh and Hess 1993, Horiuchi and Yuhara 2000). One way to achieve this is to pre-define a driver model as a dynamic input-output system (e.g. Sharp et al 2000, Prokop 2001) but a problem with this approach is that any improvement in closed-loop performance cannot be uniquely attributed to the vehicle (or chassis system), and may result from an interaction with the specific driver model. The use of an optimisation technique to directly prescribe the driver controls can remove this confounding issue, because the controls again do not depend on any aspect of the driver. On the other hand, formal optimisation depends critically on the internal states and functions that define the vehicle dynamic model (as well as on initial conditions and the geometric definition of the manoeuvre) and the computational burdens tend to become overwhelming as the complexity of the vehicle model increases.

The objective of this research is to develop an intermediate technique which approximates to optimal driver control behaviour, but without relying on the detailed formulation of the vehicle dynamics model. A Dual Model approach is developed, in which a sub-optimal set of driver controls is synthesised by iteration between two levels of model. An early version of this method was presented previously (Gordon and Best 2002) though in a different context and with a general application to lap optimization rather than obstacle avoidance. The underlying concept for this is now briefly outlined, and taken up in detail in subsequent sections.

Initially, a set of optimal controls is determined for a simple nonlinear particle model of the vehicle, one that incorporates mass-centre acceleration constraints, but no other limitations, transients or delays. These results provide a reference input to a more realistic simulation that consists of a vehicle handling model (of essentially arbitrary complexity) coupled to a generic driver model. The particle-based reference is defined as a distribution of desired mass-centre velocities across the two-dimensional domain of the relevant road surface. From deviations between particle and driver-vehicle simulations, transient delays in the latter are estimated, as are effective friction limits, and the latter provides a modified set of conditions to revisit the reference optimisation. This entire process is continued until, after a small number of iterations, no significant changes in controls are observed, and an approximately optimal input is obtained. It is proposed that the resulting control input shares the strengths of both approaches - coping with model complexity in the same way as using a fixed driver model, plus an inherent robustness to driver model assumptions, derived from the compensation mechanisms implemented within the iterative process.

It is worth emphasising the central importance of using a 'spatial velocity reference'. This appears to be essential, to accommodate path deviations that arise between the two models, in a way that a time-based reference cannot. Path deviations result from the inherent differences between the simple and 'realistic' models, and in particular via the transient dynamics of roll, yaw, load transfer, etc. that are not present in the reference particle model. To illustrate this point, suppose that in an obstacle avoidance scenario, the vehicle model path starts to intersect with an obstacle in a way that the preceding particle trajectory did not; it makes no sense for the vehicle model to continue to track the original particle motion in variables such as acceleration, speed and yaw rate – the change in position means there has to be an appropriate change in the target (or reference) motion. This is most naturally achieved by mapping the particle reference as a spatial distribution, or reference velocity field. And while this results in an additional computational burden, it is only in the context of the particle model, and the resulting optimisation problem turns out to be perfectly tractable. This approach is also much preferred to directly tracking a single particle trajectory, as this places a much greater burden on the driver control model, and increases its direct influence on the final results.

In this paper, a relatively low-order nonlinear vehicle model is used, which then permits a comparison with an independent optimisation of the handling control problem. A single limit handling manoeuvre is considered, and compared to a second approach using direct optimisation of the driver control inputs. The comparison technique is referred to as *Generalised Optimal Control* (GOC); it has been applied previously by the authors in a number of non-linear optimal control studies (e.g. Gordon and Sharp 1998, Best and Gordon 2002) and involves the solution of a well-known two point boundary value problem over a fixed time interval, directly obtained from Pontryagin's Maximum Principle. The method has some advantages over more commonly applied optimal control techniques, in that it can be applied to any (smoothly) nonlinear plant, and the cost functions are not restricted to be linear or quadratic. A practical limitation of GOC is that model complexity should be limited, for otherwise the optimisation times become prohibitive - it requires the explicit evaluation of all derivatives of all functions in the model differential equations, as well as multiple simulations to eventually converge towards an optimal control sequence.

For this reason a relatively simple non-linear four degree of freedom handling model is employed, to allow direct comparison between the optimised results. This model is described in Section 2 below, together with the generic driver control model. Section 3 describes the method for optimisation of the reference vector field, and correspondingly Section 4 outlines the GOC method. Section 5 is then devoted to the description of the overall DM technique, in the context of a simple closed-loop limit handling manoeuvre. Results are presented and compared in Section 6, and Section 7 discusses the wider significance of the results obtained and presents some final conclusions.

2. VEHICLE AND DRIVER CONTROL MODELS

The vehicle model is based on the well known three degree of freedom - yaw, roll, sideslip - model, with a fixed inclined roll axis, and using a load dependent, combined-slip Pacejka tyre model. A fourth, longitudinal, degree of freedom is also included, as are additional dynamics for wheel-spin, and first-order lags for brake actuation and drive torque delivery. The intention is to include the major dynamic processes relevant to non-linear transient dynamics, without making the model particularly complex, or requiring an excessive number of defining parameters.

The equations of motion are as follows

$$\text{longitudinal:} \quad M\dot{U} = \sum_{i=1,4} F_{xi} + MrV + Mhrp \quad (1)$$

$$\text{lateral:} \quad M\dot{V} + Mh\dot{p} = \sum_{i=1,4} F_{yi} - MUr \quad (2)$$

$$\text{yaw:} \quad I_{zz}\dot{r} + (\varepsilon I_{zz} - I_{xz})\dot{p} = a \sum_{i=1,2} F_{yi} - b \sum_{i=3,4} F_{yi} \quad (3)$$

$$\text{roll:} \quad -I_{xz}\dot{r} + Mh\dot{V} + (I_{xx} - \varepsilon I_{xz})\dot{p} = -MhUr - (B_f + B_r)p + (Mgh - K_f - K_r)\phi \\ + (h_f - h_0) \sum_{i=1,2} F_{yi} + (h_r - h_0) \sum_{i=3,4} F_{yi} \quad (4)$$

$$\text{roll kinematics:} \quad \dot{\phi} = p \quad (5)$$

Here standard SAE axes are used (Gillespie 1992) fixed relative to the vehicle wheelbase; the wheels are labelled (1-4) in ascending order as (front-left, front-right, rear-left, rear-right). The principal notation and parameter values are given in Table 1.

** INSERT TABLE 1 **

The tyre forces (F_{xi}, F_{yi}) controlling the vehicle motion allow for large steer angles

$$\begin{aligned} F_{x1,2} &= F_{tx1,2} \cos \delta - F_{ty1,2} \sin \delta, & F_{x3,4} &= F_{tx3,4} \\ F_{y1,2} &= F_{ty1,2} \cos \delta - F_{tx1,2} \sin \delta, & F_{y3,4} &= F_{ty3,4} \end{aligned} \quad (6)$$

where the tyre forces (F_{txi}, F_{tyi}) are modelled according to the Pacejka magic formula

$$P(\alpha) = P(\alpha; B, C, D, E) \equiv D \sin\left(C \tan^{-1}\left(B\alpha - E\left(B\alpha - \tan^{-1} B\alpha\right)\right)\right) \quad (7)$$

using normalized slip and isotropic similarity scaling (Milliken and Milliken 1995, Pacejka 2002). In more detail, the normalized slip vector is

$$\mathbf{k} = \begin{pmatrix} k_x \\ k_y \end{pmatrix} = \frac{C_\alpha}{F_p} \begin{pmatrix} s \\ \tan \alpha \end{pmatrix} \quad (8)$$

where s is the longitudinal slip ratio, and α is the slip angle. The friction circle at each tyre contact patch is defined by the following simple analytic function of vertical load w

$$\sqrt{F_x^2 + F_y^2} \leq F_p(w) = \frac{w}{1 + (3w/2Mg)^3} \quad (9)$$

and the load-dependent cornering/longitudinal stiffness for each tyre is of the form

$$C_\alpha(w) = c_1 \left(1 - e^{-w/c_2}\right) \quad (10)$$

(see Table 1 for values) and the resulting tyre force vector is

$$\begin{pmatrix} F_x \\ F_y \end{pmatrix} = P(|\mathbf{k}|) \frac{F_p}{|\mathbf{k}|} \begin{pmatrix} k_x \\ k_y \end{pmatrix} \quad (11)$$

The above tyre model is only broadly representative of real tyre behaviour, but is thought to incorporate sufficiently realistic aspects of force saturation and load dependence to properly test the Dual Model optimization process. Wheel rotational dynamics are modelled as

$$\dot{\omega}_i = I_w^{-1} (T_i - r_r F_{txi}) \quad (12)$$

where T_i is the drive torque (positive) or brake torque (negative) torque, and $I_w = 5 \text{ kg m}^2$ is the nominal wheel inertia, incorporating tyre, engine and driveline components. The wheel torques τ_i are directly commanded as a driver input, apportioned equally between left and right wheels, and in the case of brake torque, apportioned in the ratio 60:40 between front and rear axles. Drive torque is apportioned entirely to the front (FWD). Torque delivery assumes a first order delay of the form

$$\dot{T} = \tau^{-1}(T_d - T) \quad (13)$$

where T is the delivered wheel torque and T_d is the (apportioned) demand torque. The time constant τ was set to 0.15 seconds for drive and 0.05 seconds for braking.

Vehicle position and orientation in global coordinates is then calculated using yaw angle, such that

$$\begin{aligned} \dot{\theta} &= r \\ \dot{x}_G &= U \cos \theta - V \sin \theta \\ \dot{y}_G &= U \sin \theta + V \cos \theta \end{aligned} \quad (14)$$

As noted above, the DM method requires a basic driver model for the simulation stage, and this in turn is based on an optimised velocity field reference $\mathbf{w}(\mathbf{x})$. A suitable model was developed previously by the authors (Gordon et al, 2002).

The controller comprises parallel single-loop feedback loops for steering and forward acceleration. It is designed to be simple but effective near the limits of adhesion, and in the context of this work there is no intention that it be ‘optimal’ in any particular sense; it is only important that it be capable of providing reasonable tracking performance for the reference signals derived from the vector field \mathbf{w} . These are defined in the form of magnitude $\bar{v} = |\mathbf{w}|$ and direction $\bar{\phi} = w_y / w_x$, supplemented by the tangential and normal components of the associated flow acceleration:

$$\bar{a}_t = \bar{\mathbf{e}}_t \cdot (\mathbf{w} \cdot \nabla) \mathbf{w}, \quad \bar{a}_n = \bar{\mathbf{e}}_n \cdot (\mathbf{w} \cdot \nabla) \mathbf{w} \quad (15)$$

Here $\bar{\mathbf{e}}_t$ and $\bar{\mathbf{e}}_n$ are unit vectors tangential and normal to the reference field respectively, and \mathbf{e}_t and \mathbf{e}_n are defined as the corresponding vectors for the vehicle mass centre velocity:

$$\mathbf{v}_G = v_G \mathbf{e}_t \quad (16)$$

$$\mathbf{a}_G = \dot{v}_G \mathbf{e}_t + v_G \dot{\phi} \mathbf{e}_n \quad (17)$$

Longitudinal control is executed via an acceleration command signal $u_1(t)$ which is then converted to a commanded drive force or braking torque. The control is determined via a forward acceleration demand

$$a_0(t) = \bar{a}_t + (\bar{v} - v_G) / \tau_1 \quad (18)$$

and feedback of longitudinal vehicle acceleration

$$\dot{u}_1 = (a_0 - \dot{U}) / \tau_2 \quad (19)$$

The demand signal (18) is a simple sum of reference plus feedback, and τ_1 is a control parameter in the form of an assumed time constant. Equation (19) is essentially an integral control, and while \dot{U} is not strictly the forward vehicle acceleration, the error is small in normal circumstances when the lateral velocity V is small compared to U .

For lateral control, the steering angle $u_2(t)$ is computed using a similar integrator approach, though (angular) displacement errors also included in the feedback. The vehicle yaw angle θ and slip angle

$$\beta = \tan^{-1}(V/U) \quad (20)$$

together determine the direction ϕ of \mathbf{v}_G

$$\phi = \theta + \beta \quad (21)$$

The reference angular velocity of \mathbf{v}_G is given by

$$\bar{\omega} = v_G \bar{a}_n / \bar{v}^2 \quad (22)$$

This is then combined with angular error to provide an angular velocity demand

$$\omega_0(t) = \bar{\omega} - (\phi - \bar{\phi}) / \tau_3 \quad (23)$$

which is translated into steering control via

$$\dot{u}_2 = k_1(\omega_0 - r) + k_2\sigma(\beta\dot{\beta})\dot{\beta} \quad (24)$$

where $r = \dot{\theta}$ is the yaw rate, and the first term is for tracking the desired velocity direction; the second is to compensate for any large sideslip conditions, incorporating a switching term $\sigma(\cdot)$:

$$\sigma(u) = \begin{cases} 1 & \text{if } u > 0 \\ 0 & \text{if } u \leq 0 \end{cases} \quad (25)$$

Finally, to avoid excessive integrator windup, equations (19), (24) are supplemented by the condition

$$\dot{u}_i \rightarrow 0 \quad \text{when } u_i \dot{u}_i > 0 \quad \text{and} \quad |u_i| > u_i^{\max} \quad (26)$$

Although the above model is essentially the same as presented in (Gordon et al, 2002), additional adaptive phase compensation will be applied to the model, in an attempt to improve reference tracking and to reduce the sensitivity of the overall optimised performance to many aspects of the driver model's parametric definition – see Section 5 below. Also, whereas the previous application of the model required a specific look-ahead strategy to prescribe $\mathbf{w}(\mathbf{x})$, the reference field is now in a general format of velocity data on a spatial grid. The definition, interpolation and optimisation of this field now considered.

3. REFERENCE VECTOR FIELD

While in the previous section $\mathbf{w}(\mathbf{x})$ and its derivatives are proposed as a spatial reference for the driver-vehicle model, here it is considered further in the context of a simple particle model of the vehicle. The particle model has four states representing the position $\mathbf{r}(t)$ and velocity $\mathbf{v}(t)$ of the vehicle mass centre, and is controlled directly by an acceleration demand $\mathbf{u}(t)$

$$\begin{aligned}\dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{u}\end{aligned}\tag{27}$$

This model will be subjected to a fixed acceleration constraint, but no other transient dynamic behaviour:

$$\|\mathbf{u}(t)\| = \sqrt{u_x^2 + u_y^2} \leq \mu_0\tag{28}$$

This model can be thought to represent an ‘optimistic’ version of what the vehicle (and driver) might achieve, and in this context μ_0 should be set to a ‘high but realistic’ value, and in the present study a value $\mu_0 = 8 \text{ ms}^{-2}$ has been chosen. To track the reference field without error requires $\mathbf{u}(t) = \mathbf{a}(\mathbf{r}(t))$, where $\mathbf{a}(\mathbf{x})$ is the required (flow) acceleration of the reference field

$$\mathbf{a}(\mathbf{x}) = (\mathbf{w} \cdot \nabla)\mathbf{w} = \left(w_x \frac{\partial}{\partial x} + w_y \frac{\partial}{\partial y} \right) \mathbf{w}\tag{29}$$

Typically we expect somewhat tighter constraints on the reference accelerations

$$\|\mathbf{a}(\mathbf{x})\| \equiv \sqrt{a_x^2 + a_y^2} \leq \mu(\mathbf{x}) < \mu_0\tag{30}$$

where $\mu(\mathbf{x})$ is to be estimated separately from the driver-vehicle model limits. This notation is suggestive of a position-dependent vehicle ‘friction’ limit, though it should be emphasised that the physical source of these limits refer as much to context-dependent handling transients associated with the vehicle and driver, as to the tyre-road friction. (In this study the tyre-road friction is assumed constant, though clearly the formalism would very naturally accommodate surface friction variations).

As will be seen below, there may be occasions when the reference acceleration constraint (30) is violated, in which case only approximate tracking is possible. To represent the resulting acceleration-limited

tracking behaviour, the particle motion equations (27) are augmented by a feedback control, based on the velocity error variable

$$\mathbf{e}(t) = \mathbf{v}(t) - \mathbf{w}(\mathbf{x}(t)) \quad (31)$$

in the form

$$\mathbf{u}(t) = \mathbf{a}(\mathbf{r}(t)) - k\mathbf{e}(t) \quad (32)$$

In this case the error dynamics are given by

$$\begin{aligned} \dot{\mathbf{e}} &= \dot{\mathbf{v}} - (\mathbf{v} \cdot \nabla)\mathbf{w} \\ &= \mathbf{a} - k\mathbf{e} - (\mathbf{v} \cdot \nabla)\mathbf{w} \end{aligned} \quad (33)$$

or using equations (29) and (31)

$$\begin{aligned} \dot{\mathbf{e}} + k\mathbf{e} &= (\mathbf{w} \cdot \nabla)\mathbf{w} - (\mathbf{v} \cdot \nabla)\mathbf{w} \\ &= -(\mathbf{e} \cdot \nabla)\mathbf{w} \end{aligned} \quad (34)$$

Taking the scalar product of this equation with \mathbf{e} , and defining $e = \|\mathbf{e}(t)\|$ as the error magnitude, it is then quite easy to re-write this equation in the simplified scalar form

$$e\dot{e} = -ke^2 - \mathbf{e}^T H \mathbf{e} \quad (35)$$

where H is the 2×2 symmetric matrix with components

$$H_{ij} = \frac{1}{2} \left(\frac{\partial w_i}{\partial x_j} + \frac{\partial w_j}{\partial x_i} \right) \quad (36)$$

Then, in terms of the eigenvalues of H

$$h = \max\{eig(-H), 0\} \quad (37)$$

we clearly have $h \geq 0$, and from (35)

$$e\dot{e} \leq -ke^2 + he^2 \quad (38)$$

or

$$\dot{e} \leq -(k-h)e \quad (39)$$

(see Gordon et al 2002, for further discussion of the relevance of this eigenvalue). This equation allow a suitable choice to be made for k , dependent on the particular reference field. For the obstacle avoidance geometry considered below, it turns out that $\overline{\max}(h) \approx 5$, and hence $k = 10$ provides suitable dominance in equation (39) to ensure $e(t) \rightarrow 0$, provided of course the friction limit μ_0 is sufficient to deliver the feedback control (33). In general the feedback will take the modified form

$$\mathbf{u}(t) = \begin{cases} \mathbf{a} - k\mathbf{e} & \text{if } \|\mathbf{a} - k\mathbf{e}\| \leq \mu_0 \\ \mu_0 \frac{\mathbf{a} - k\mathbf{e}}{\|\mathbf{a} - k\mathbf{e}\|} & \text{otherwise} \end{cases} \quad (40)$$

Clearly, if the reference accelerations are globally bounded as in equation (30), and any initial velocity error $e(0)$ is small, the $k\mathbf{e}$ term must remain small, and this is the same as the full control (33).

Turning now to the synthesis of the reference field, we note that previously (Gordon et al 2002) this was defined by simple geometric construction relative to a desired target path, namely the track centreline C . However here, for a collision avoidance scenario, there is no such pre-defined reference trajectory, and $\mathbf{w}(\mathbf{x})$ will be obtained directly by optimality considerations. Figure 1 shows part of a typical rectangular grid construction, appropriate to a collision avoidance manoeuvre. Here $\Omega \subset \mathfrak{R}^2$ is the track region available to the vehicle, and its boundary $\partial\Omega$ consists of track and obstacle edges ($\partial\Omega|_{edge}$) together with entry ($\partial\Omega|_{in}$) and exit ($\partial\Omega|_{out}$) points.

** INSERT FIGURE 1 **

Reference velocities will be defined at each vertex or node N and interpolated to fill the space within each rectangle or ‘tile’ T . Nodes will be referenced in the two alternative formats: $\mathbf{V}_N \leftrightarrow \mathbf{V}_{(i,j)}$, where i corresponds to an increment in the longitudinal (x) direction, and j to the lateral (y) direction – see Figure 2. The N -form simplifies some notation, while (i, j) is more specific to the grid coordinates. Similarly, tiles are referenced according to the lowest indexed node (upper left in Figure 2) using square brackets;

for example, flow accelerations will be evaluated as a ‘tile mean’ (see below), so in this case we may write

$$\bar{\mathbf{a}}_N \equiv \bar{\mathbf{a}}_{[i,j]}.$$

** INSERT FIGURE 2 **

Within any particular tile, $\mathbf{w}(\mathbf{x})$ is defined via linear interpolation of the nodal values:

$$\mathbf{w}(\mathbf{x}) = (1-\xi)(1-\eta)\mathbf{V}_{(i,j)} + \xi(1-\eta)\mathbf{V}_{(i+1,j)} + (1-\xi)\eta\mathbf{V}_{(i,j+1)} + \xi\eta\mathbf{V}_{(i+1,j+1)} \quad (41)$$

where ξ and η are normalized local coordinates

$$\xi = \frac{x - x_{(i,j)}}{d_x}, \quad \eta = \frac{y - y_{(i,j)}}{d_y} \quad (42)$$

Clearly $\mathbf{w}(\mathbf{x})$ so defined is continuous across tile boundaries, though in general it will suffer gradient discontinuities. The ‘flow optimisation’ of the nodal velocity components will be to essentially maximise the mean flow speed while keeping the flow acceleration within the given bounds. Formally this is specified as follows:

$$\text{maximise } P = \frac{1}{2} \iint_{\Omega} \beta(\mathbf{x}) \|\mathbf{w}(\mathbf{x})\|^2 dx dy \text{ subject to } \|\mathbf{a}(\mathbf{x})\| \leq \mu(\mathbf{x}) \quad (43)$$

where $\beta(\mathbf{x})$ may in principle be chosen to emphasise certain desired aspects of the flow, though in this study $\beta(\mathbf{x}) \equiv 1$ will be assumed. Further constraints arise from boundary conditions: if \mathbf{n} is a unit normal to $\partial\Omega$, pointing into the ‘active region’ Ω on $\partial\Omega|_{in}$ and $\partial\Omega|_{edge}$, and out of the active region on $\partial\Omega|_{out}$, then

$$\mathbf{w} \cdot \mathbf{n} \geq 0 \text{ on } \partial\Omega \quad (44)$$

Note that some additional constraints are required to avoid additional ‘defects’ in the flow, as will be described in Section 5. However, at this point, the above defines the optimisation problem sufficiently well to describe how the optimisation is to be approximated within the discretised nodal form of the flow.

The flow acceleration is obtained from the linear interpolation formula (41) on tile T . We obtain

$$a_k = (\mathbf{w} \cdot \nabla) w_k = \frac{w_1}{d_x} \frac{\partial w_k}{\partial \xi} + \frac{w_2}{d_y} \frac{\partial w_k}{\partial \eta} \quad (k = 1 \text{ or } 2) \quad (45)$$

After evaluating the derivatives, the result is greatly simplified as a node is approached ($\xi \rightarrow 0$ or 1 , $\eta \rightarrow 0$ or 1) to obtain the following set of flow accelerations on tile $[i, j]$:

$$\begin{aligned} a_{k(i,j)} &= d_x^{-1} V_{1(i,j)} (V_{k(i+1,j)} - V_{k(i,j)}) + d_y^{-1} V_{2(i,j)} (V_{k(i,j+1)} - V_{k(i,j)}) \\ a_{k(i+1,j)} &= d_x^{-1} V_{1(i+1,j)} (V_{k(i+1,j)} - V_{k(i,j)}) + d_y^{-1} V_{2(i+1,j)} (V_{k(i+1,j+1)} - V_{k(i+1,j)}) \\ a_{k(i,j+1)} &= d_x^{-1} V_{1(i,j+1)} (V_{k(i+1,j+1)} - V_{k(i,j+1)}) + d_y^{-1} V_{2(i,j+1)} (V_{k(i,j+1)} - V_{k(i,j)}) \\ a_{k(i+1,j+1)} &= d_x^{-1} V_{1(i+1,j+1)} (V_{k(i+1,j+1)} - V_{k(i,j+1)}) + d_y^{-1} V_{2(i+1,j+1)} (V_{k(i+1,j+1)} - V_{k(i+1,j)}) \end{aligned} \quad (46)$$

It is then natural to define the ‘tile acceleration’ as the mean of these expressions

$$\bar{\mathbf{a}}_{[i,j]} = \frac{1}{4} (\mathbf{a}_{(i,j)} + \mathbf{a}_{(i+1,j)} + \mathbf{a}_{(i,j+1)} + \mathbf{a}_{(i+1,j+1)}) \quad (47)$$

The optimisation problem (43) can then be re-written

$$\begin{aligned} \text{maximise : } P &= \frac{1}{2} \sum_N \beta_N \|\mathbf{V}_N\|^2 \\ \text{subject to : } \|\mathbf{a}_T\| &\leq \mu(\mathbf{x}) \text{ on } \Omega \\ &: \mathbf{V}_N \cdot \mathbf{n}_N \geq 0 \text{ on } \partial\Omega \end{aligned} \quad (48)$$

The last of these conditions needs a little care when the boundary $\partial\Omega$ includes corners, since then there exist two vectors \mathbf{n}_N , one for each intersecting boundary. In this case, for convex boundary corners we require $\mathbf{V}_N \cdot \mathbf{n}_N \geq 0$ for *at least one* \mathbf{n}_N , while for concave corners the condition is imposed relative to *both* normal vectors.

According to the standard Kuhn-Tucker conditions (e.g. Liu et al 2003) the above optimisation problem may be re-stated by introducing appropriate Lagrange multipliers α_T and γ_N :

$$\begin{aligned} \text{minimize : } J &= \frac{1}{2} \sum_T \alpha_T \|\bar{\mathbf{a}}_T\|^2 - \frac{1}{2} \sum_N \beta_N \|\mathbf{V}_N\|^2 - \sum_{N \in \partial\Omega} \gamma_N (\mathbf{V}_N \cdot \mathbf{n}_N) \\ \text{subject to : } \alpha_T &\left[\|\bar{\mathbf{a}}_T\|^2 - \mu_T^2 \right] = 0 \\ &: \gamma_N (\mathbf{V}_N \cdot \mathbf{n}_N) = 0 \\ &: \alpha_T \geq 0, \quad \gamma_N \geq 0 \end{aligned} \quad (49)$$

Thus J can be minimized as an unconstrained function, with the Lagrange multipliers separately adapted to satisfy the additional conditions. For the present investigation, a simple steepest descent method was sufficient to solve this problem. $\alpha_T > 0$ was intermittently adapted to achieve the first of the supplementary conditions, while the $\mathbf{V}_N \cdot \mathbf{n}_N \geq 0$ condition is more simply achieved by imposing it directly on the relevant components of \mathbf{V}_N , directly within the gradient descent. Since the objective function is explicitly a quartic function of the \mathbf{V}_N components, it would be quite possible to use a Hessian-based optimisation, such as Sequential Quadratic Programming (Liu et al, 2003), potentially making the optimisation more efficient. On the other hand, for a small scale problem of the type considered in Section 5, steepest descents proved more than adequate.

In the above we have imposed an upper limit on the ‘tile mean’ flow acceleration. This still allows the possibility of higher accelerations occurring locally within any tile, particularly at one vertex. This can then cause excessive requirements on either model (particle or vehicle) during simulation, essentially due to the limited resolution of the tiles. This feature has motivated giving the particle model an inherent acceleration limit, essentially to test the validity of the synthesized reference (to check for example that it doesn’t hit barriers) prior to the use of the full vehicle model. Of course, if the particle does track the reference accurately, the reference is *fully equivalent* to the dynamic particle model, and it is in this sense that the optimised reference field is equivalent to the simultaneous multiple optimisation of the particle model, using all possible initial positions in Ω , again subject to grid resolution.

4. GENERALISED OPTIMAL CONTROL

Generalized Optimal Control (GOC) is an independent approach to providing driver inputs for the ‘optimal’ obstacle avoidance problem, and is used here to provide a baseline for performance comparisons. The method uses a gradient descent implementation of Pontryagin’s Maximum Principle (see Bryson and Ho 1975 for the basic method, and Marsh 1992 for the application in automotive systems). GOC utilises a dynamic cost function L plus a residual cost associated with final vehicle states, L_T

$$J = L_T[\mathbf{x}(T)] + \int_0^T L[\mathbf{x}(t), \mathbf{u}(t)] dt \quad (50)$$

Adding constraint equations via a vector of Lagrange multiplier functions, $\mathbf{p}(t)$, yields

$$J = L_T[\mathbf{x}(T)] + \int_0^T \left\{ L[\mathbf{x}(t), \mathbf{u}(t)] + \mathbf{p}^T(t) [g[\mathbf{x}(t), \mathbf{u}(t)] - \dot{\mathbf{x}}(t)] \right\} dt \quad (51)$$

where the function g is defined via the nonlinear system equations

$$\dot{\mathbf{x}} = g[\mathbf{x}(t), \mathbf{u}(t)] \quad (52)$$

Lagrange multipliers (*costates*) $\mathbf{p}(t)$ are introduced and a Hamiltonian function is defined:

$$H = L[\mathbf{x}(t), \mathbf{u}(t)] + \mathbf{p}^T(t) g[\mathbf{x}(t), \mathbf{u}(t)] \quad (53)$$

The costates satisfy the following differential equations

$$\dot{\mathbf{p}}^T(t) = -\frac{\partial H}{\partial \mathbf{x}} = -\frac{\partial L}{\partial \mathbf{x}} - \mathbf{p}^T \frac{\partial g}{\partial \mathbf{x}}, \quad \mathbf{p}^T(T) = \frac{\partial L_T}{\partial \mathbf{x}} \quad (54)$$

and the optimal controls are found from the Hamiltonian via

$$\frac{\partial H}{\partial \mathbf{u}} = 0 \quad (55)$$

Equations (52)-(55) are solved over the time interval $[0, T]$ assuming fixed initial states $\mathbf{x}(0) = \mathbf{x}_0$, and since the costate ‘initial’ conditions are set at $t = T$, the system is a two-point boundary value problem. Following the methods used in (Marsh 1992, Best and Gordon 2002), an approximate solution is found via a discrete sequence of controls, each held constant for a small time Δt . Within the time period for each control, the cost gradient is obtained directly from the Hamiltonian:

$$\frac{\partial J}{\partial u_i} = \int_{t_{i-1}}^{t_i} \frac{\partial H}{\partial u_i} dt \quad (56)$$

which then enables a gradient-based iteration optimisation to determine the optimal control sequence.

** INSERT FIGURE 3 **

Figure 3 provides a summary of the algorithm used to conduct the GOC optimisation:-

- 1 Using the current discrete control sequence, integrate the state-space system from $\mathbf{x}(0)$ and evaluate $J_{[0,T]}$
- 2 Evaluate the residual cost L_T and hence $\mathbf{p}(T)$ from equation (54)
- 3 Integrate the costate system and $\partial H / \partial \mathbf{u}$ in reverse-time from the initial condition $\mathbf{p}(I)$. Calculate cost gradients from equation (56)
- 4 Update the control sequence by a line search optimisation along the steepest descent direction to minimise J (evaluated by repeating Stages 1 & 2).

Stages 1-4 are repeated until suitable convergence of cost and controls, and reduction of cost gradients is achieved.

The costate system of equations (54) for the vehicle model is prohibitively complex to establish by hand, so three techniques are employed to create accurate, yet efficient simulation code. Firstly, the equations are manipulated using an analytical math processor – the Matlab Symbolic toolbox (Mathworks 2004).

Direct evaluation of the partial derivatives is then possible, but the resultant formulae are long and inefficient (for example $\partial H / \partial u$ leads to an equation comprising over 100,000 characters). These direct formulae are thus only used to validate the final code, which is generated by first breaking each partial derivative into its component parts, eg

$$\frac{\partial H}{\partial u} = \frac{\partial H'}{\partial u} + \sum_i \frac{\partial H'}{\partial f_i} \frac{\partial f_i}{\partial u} \quad (57)$$

where H' is the Hamiltonian written in terms of the tyre forces f_i , and $\partial f_i / \partial u$ is further broken down in to component derivatives of the Pacejka formulae. The resulting derivatives are then converted into lines of computer code by an iterative extraction of common terms, to produce the shortest possible function.

To increase the efficiency of time integration of the states and costates, a discrete-time integration algorithm is employed; this is the Cash/Karp 5th/6th order algorithm (Press et al, 1992). The time-step is kept constant within each control (u_i) time interval, and to ensure accuracy the code is written to monitor state errors and adjust the time-step duration accordingly. One further modification was made to improve optimisation of the handling model; although the (u_i) controls remain functions of time, they are held constant for a specific *distance* along the vehicle trajectory, improving the speed of convergence when steer and torque inputs are optimised simultaneously. Finally, the integration and derivative codes are compiled to achieve the fastest possible simulation execution time. This level of computation is in marked contrast to that of the DM method, which requires no such efficiency savings.

For application to the obstacle avoidance manoeuvre, under limit handling conditions, the target has been to achieve maximum distance in the x direction, whilst maintaining adequate lateral control of the vehicle. A residual, final cost function is thus set as :

$$L_T = (X_G - X(T))^2 \quad (58)$$

with X_G set at some large, unattainable distance (in this case 50 m in $T = 3$ seconds). . The integral cost term varying is dominated by a boundary avoidance terms of the form

$$L_{track} = \lambda \chi^2 \quad (59)$$

where χ is the perpendicular distance of the vehicle from a nominal track centre (defined using straight-line and arc segments). λ is chosen, to provide the minimum track following cost whilst ensuring that the obstacles are avoided ($\lambda = 100$). Additionally, in the present study, the initial vehicle velocity vector is left free, and this is simply accommodated by setting the corresponding initial costates to zero, $p_i(0) = 0$.

5. DUAL MODEL METHOD APPLIED TO AN OBSTACLE AVOIDANCE MANOEUVRE

The Dual Model (DM) method was applied to the following problem. The vehicle enters a track section, nominally 35m long and 10 m wide, as shown in Figure 4, with the initial entry point indicated by the arrow on the left boundary. For the case presented – a double lane-change – the entry speed and direction are to be freely chosen within the optimisation procedure. Other vehicle states are set as simply as possible (wheel speeds correspond to a free rolling condition, but other states zero – yaw velocity, slip angle, roll angle, roll velocity, steer angle and drive/brake torque). As described above, the first stage of the optimisation process is used to generate the reference field, also shown in Figure 4, and a uniform acceleration limit is chosen, $\mu(\mathbf{x}) = 6 \text{ ms}^{-2}$ (c.f. equation (48)).

** INSERT FIGURE 4 **

As mentioned above, additional constraints were imposed to avoid certain ‘defects’ in the flow. These can occur at lateral ‘source’ and ‘sink’ boundaries. The source boundaries, shown at distance $x = 5\text{m}$ and 20m , are lateral boundaries where the reference field ‘flows’ out of the obstacle. A limiting exit speed needs to be defined in such cases, for otherwise the flow speed can increase without bound, especially near the exit. Strictly this does not influence the vehicle path, since of course the vehicle should not really exit from either obstruction! But the potential for unbounded velocities is highly undesirable for the numerical procedures, and may cause difficulties in early vehicle simulations when the reference following may be subject to sizeable errors. In the present case, the (longitudinal components of the) exit velocities were limited to 5 m/s .

Where sink boundaries ($x = 10\text{m}$ and 25m) meet the lateral track boundaries (at a concave corner), it is possible to create a topological defect in the flow. This is where, during the optimization from some randomised initial conditions, the velocities vectors tend to point towards the corner, and essentially become ‘trapped’ – and the optimization settles into a local minimum corresponding to ‘braking to a stop’

at the corner. To avoid this a minimum lateral velocity (2.5 m/s) away from the track edge is set on the nodes of the sink boundary. One final velocity constraint is to regulate the exit flow to be purely longitudinal, a feature that can be easily seen at $x = 35\text{m}$. We are now in a position to describe the overall Dual Model technique.

** INSERT FIGURE 5 **

Figure 5 shows a comparison between flow model and vehicle case, both starting from the same initial location fixed for the test. While the vehicle simulation is unsatisfactory, in the sense that the vehicle hits the second barrier, it does not go unstable or generate large path errors; the driver is doing a ‘reasonable job’ in following the reference, but would expect to ‘do better next time’. The plot gives a strong clue to the source of the problem – the direction of the vehicle path is consistently lagged behind that of the reference particle. This is shown more clearly in Figure 6, where the vehicle path angle is plotted against the angle of the collocated (interpolated) reference field vector (left plot); there appears to be a consistent time delay, or phase lag, and of course this should be expected – the driver model uses instantaneously available reference velocity information, and there are dynamic lags associate with all aspects of the vehicle and drivers actions. A similar, though more complex lag behaviour is apparent in the right hand plot, where the vehicle acceleration lags behind that inferred from the collocated reference field.

** INSERT FIGURE 6 **

** INSERT FIGURE 7 **

To compensate for this effect the driver should clearly introduce an element of ‘anticipation’, which is quite simply achieved as follows. Referring to Figure 7, for each $t \in [0, T]$, an estimate of the time delay is derived in a non-parametric fashion. The path angle $\theta(t)$ is determined, and this is ‘inverted’ from the reference angles; thus (within the resolution of the simulated trajectory) all times t' for which $\theta_0(t') = \theta(t)$ within a small tolerance ε are formed into a set T' - this set is illustrated by the marking on

the time axis in Figure 7. Of course some of these points are completely erroneous, but since the time delay is likely to be small, the time delay $\hat{t}_d(t)$ is determined by the nearest member of T' to t . Formally,

$$\hat{t}_d(t) = t - \operatorname{argmin}\{t - t' \mid t' \in T'\} \quad (60)$$

and here $\hat{t}_d(t)$ is positive for the case of a lagged response to the reference. If T' turns out to be empty, no result is available, and all non-trivial estimates are used in a regression-based estimate, taken here to be a quadratic function:

$$\hat{t}_d(t) \approx c_0 + c_1 t + c_2 t^2 \quad (61)$$

This ensures that the phase compensation is smoothly defined, and relatively insensitive to uncertainties that typically occur at or near regions of zero slope in $\theta(t)$.

A precisely similar phase estimation was also carried out for the longitudinal acceleration and brake control. In both cases, the delay compensation for $\hat{t}_d(t)$ is easily implemented in a repeat simulation, using a translated reference:

$$\hat{\mathbf{w}}(\mathbf{r}(t)) = \mathbf{w}(\mathbf{r}(t) + \mathbf{v}(t)\hat{t}_d(t)) \quad (62)$$

Note that a different $\hat{\mathbf{w}}$ is used for the lateral and longitudinal control aspects of the driver model of Section 2, corresponding to the fact that the lateral and longitudinal dynamics of the vehicle (and driver) have different transient characteristics. One further important aspect of the phase compensation is that there exists some interaction between the control modes, and for this reason the estimated $\hat{t}_d(t)$ was truncated at a suitably small value ($|\hat{t}_d(t)| < 0.1$) and the entire process was run iteratively until no significant additional phase change resulted.

The final component of the iterative DM optimisation is adaptation of the ‘friction’ distribution $\mu(\mathbf{x})$. This should be based on the ‘level of saturation’ of the driver-vehicle system – where increased mass-centre accelerations can be delivered whilst maintaining control of the vehicle, then $\mu(\mathbf{x})$ can be increased of such points. Where control or stability is compromised, for example where tyres experience excessive

slip, or the vehicle slip angle is large, then $\mu(\mathbf{x})$ should be reduced in such regions. For simplicity, the adaptation criterion adopted here is based purely on tyre force saturation. For any location $\mathbf{r}(t)$ on the vehicle path, we can associate a tyre force utilization parameter:

$$\varepsilon(t) = \max_i \begin{cases} \frac{F(t)}{F_p(t)} & \text{if } k < k_p \\ \frac{F_p(t)}{F(t)} & \text{if } k > k_p \end{cases} \quad (63)$$

where $F(t)$ is the tyre force magnitude (at wheel i) and $F_p(t)$ is the peak force available. In the tyre adhesion region, with the normalized combined slip parameter k below its value at the force peak (c.f. equation (8)) the utilization is less than one. Above this critical slip parameter the tyre is ‘over-utilized’ and the ratio is reversed. A revised acceleration limit $\hat{\mu}(\mathbf{x})$ may then be defined using a Gaussian neighbourhood function; let $d = \min_t |\mathbf{x} - \mathbf{r}(t)|$ be the distance from \mathbf{x} to the vehicle trajectory, and let $\sigma > 0$ be a characteristic length ($\sigma = 1\text{m}$ in this study). The revised acceleration limit is then given by

$$\hat{\mu}(\mathbf{x}) = \frac{\mu(\mathbf{x})}{1 + (\varepsilon - 1)e^{-d^2/2\sigma^2}} \quad (64)$$

Clearly if $\varepsilon = 1$ there is no change in $\hat{\mu}(\mathbf{x})$, but where the tyres are over-utilized ($\varepsilon > 1$) $\hat{\mu}(\mathbf{x})$ will be reduced in the neighbourhood of the vehicle path, and where the tyres are under-utilized ($\varepsilon < 1$) $\hat{\mu}(\mathbf{x})$ will be increased. It is then clear that, if a significant change in $\hat{\mu}(\mathbf{x})$ is implied by equation (64), the underlying field optimisation needs to be refined and the vehicle simulation repeated. The DM process then continues until the change in $\hat{\mu}(\mathbf{x})$ is insignificant – meaning that the reference field reoptimisation causes insignificant changes in the driver-vehicle model response.

The above procedure may seem computationally burdensome, so it is perhaps worth making some informal comments about the computational efficiency of the DM optimisation process. The flow optimisation was implemented in Matlab, and the result presented in Figure 4 took approximately 20 minutes, on a standard 1600 MHz Pentium PC, starting from a highly random initial velocity distribution. Flow refinement, required for any modest changes in acceleration limit, are much quicker – typically taking around 5

minutes. The phase compensation stages of the DM process required less than 20 simulations of the combined driver-vehicle model, and including the various numerical routines for the phase compensation itself, this took less than 10 minutes to run. Overall, the whole analysis takes 30-60 minutes on a standard PC, depending on the number of acceleration limit revisions required. Of course these numbers will increase if finer discretisations of the track are used, or if more lengthy manoeuvres are optimised, but overall the method appears to be inherently much quicker than the full GOC optimisation. In the next section we consider its effectiveness in this test problem.

6. OPTIMIZATION RESULTS

We first consider how the DM iterations develop, and then make comparisons with GOC. The phase compensation converged in 18 iterations of the process described above, and the result is shown in Figure 8.

**** INSERT FIGURE 8 ****

Compared to Figure 6, the directional phase lag is almost entirely removed. There is also clear improvement in the phasing of the longitudinal acceleration, though clearly the picture remains more complex. The result might be improved further, for example by introducing additional basis functions into equation (61) for the regression fit. On the other hand this is only a very simple aspect of the relatively complex driving task being modelled, and excessive focus on ‘improving the phase result’ would certainly bias the overall performance of the driver-vehicle system. For example, the delayed braking at the start is a direct effect of the transient lags in the driver-vehicle system, and no further compensation is possible. Again, the delay in braking at around $t = 1.5$ seconds is due to the fact that the vehicle has a slower speed than the reference assumes, this in turn being due to the sharp spikes in the reference acceleration, which cannot be achieved by the vehicle model (note that such spikes are a feature of the discretisation mentioned earlier – even though the ‘tile mean’ demand remains at approximately 6 ms^{-2} , locally the peaks can be higher; fortunately this does not seem to significantly degrade the overall performance).

We now turn to the adaptation of acceleration limits, equation (64). It turns out that the tyre utilization – see Figure 9 – was very close to unity ($\varepsilon = 1$) even in the first iteration. This is a somewhat surprising outcome, resulting perhaps from a ‘lucky guess’ ($\mu(\mathbf{x}) = 6 \text{ ms}^{-2}$) for the initial acceleration limit. This issue will be discussed further in Section 7, but for now the conclusion is that further iterations of the DM sequence are pointless – according to Figure 9, $\varepsilon \approx 1$, and the application of equation (64) makes no significant difference to the distribution of $\mu(\mathbf{x})$, and hence no difference to the optimisation results.

**** INSERT FIGURE 9 ****

Figure 10 shows the revised path of the DM vehicle (solid line), and clearly the phase compensation has been successful in improving the path of the vehicle, which now has a slight bias towards the ‘safe side’ of the reference vehicle path in avoiding the obstacles. The GOC - optimal path is also shown, and this clearly has a much stronger tendency to avoid the barriers. This is simply a feature of the cost function parameter (λ in equation (59)) and is not a particular indicator of one solution being ‘more optimal’ than another. One other feature Figure 10 is that the GOC path is slightly offset from the intended start position; this was the result of starting the GOC optimisation a little ‘early’ (at $x = -5$ m) in order to remove some initial transients. In any case we do not expect high levels correlation between time-histories, and the advantage that GOC enjoys here is compensated for by its greater tendency to avoid the obstacles.

**** INSERT FIGURE 10 ****

More objective performance metrics are presented in Table 2. The first row of data relates to the acceleration-limited particle model, which takes 2.76 seconds to complete the 35 m section. The initial DM vehicle (prior to phase compensation) is 0.16 seconds slower, but of course it collides with a barrier. The effect of phase compensation is to reduce this figure to 2.81 seconds (0.5 seconds slower) which is close to the 2.78 seconds achieved by GOC. Another notable feature from Table 2 are the very slight differences between maximum speeds, as compared to the exit speeds, which are much more variable, and with GOC exiting the fastest. Note that in the last two columns, ‘angle error’ is not meaningful for the GOC result (as there is no relevant reference $\mathbf{w}(\mathbf{x})$ for comparison), and ‘path error’ (maximum deviation from the particle model trajectory) is only meaningful for the DM vehicle simulations. It is finally worth noting how the phase compensation greatly reduces the peak angle error, and that the maximum path deviations are a little misleading, since the 0.39 m in row 2 is not only reduced to 0.25 m in row 3, but the deviation moves from the ‘wrong side’ of the reference to the ‘right side’.

** INSERT TABLE 2 **

** INSERT FIGURE 11 **

Figure 11 shows how the vehicle speeds differ between the (phase compensated) DM case and the GOC case. The DM speed is initially higher and there is a very simple interpretation of the curve – deceleration towards the corner of the first obstacle, acceleration until the second obstacle corner is approached, brief deceleration and then acceleration towards the exit. The speed in the GOC case is harder to interpret, but clearly the advantage of a slower start is to avoid sharp decelerations and to start the exit acceleration earlier. In a sense therefore, the GOC case appears to operate more smoothly in terms of longitudinal control. On the other hand, the vehicle slip angles are considerably higher for the GOC case, so one might anticipate that the ‘GOC driver’ is more aggressive with the steering control, and this is clearly confirmed by reference to Figure 12, the driver control inputs. The left plot shows the steering inputs (degrees of steer at the front wheels) with the GOC driver applying sharp initial right steer, followed by even sharper counter-steer at around 0.5 seconds. Hence the GOC driver is prepared to exercise the vehicle controls at the limits far more than the DM driver. This is also apparent in the right hand plot. The solid line shows a fairly simple torque demand (a nominal control input, equivalent to Mra_{dem} , where M is vehicle mass, r is the rolling radius of the road wheels, and a_{dem} is a demanded vehicle acceleration) from the DM driver, which is almost a ‘bang-bang’ control sequence – hard braking on approaching the obstacle corners, with hard acceleration in between. On the other hand, the GOC driver has independent control of the drive and brake torques, and at one point – around 0.5 seconds – they are applied sharply and simultaneously. This is a way that ‘rear-only’ braking can be applied (recall the vehicle is FWD) inducing temporary oversteer, to assist with the counter-steering action in the left plot. This is a highly sophisticated and coordinated use of the three control inputs, typical of a racing driver, and while it reinforces the validity of the GOC control optimization method, it pushes the optimisation into a specialist domain that the DM method is unlikely to go, not at least without a very much more elaborate control model.

** INSERT FIGURE 12 **

7. DISCUSSION AND CONCLUSIONS

In the foregoing analysis, a modular ‘Dual Model’ driver control optimisation has been defined and applied to the simple case of an obstacle avoidance manoeuvre. The DM method combines the use of a simple particle model to determine a velocity reference for the actual vehicle model. The use of the two-dimensional track coordinates as independent variables for the velocity reference is a key and distinctive feature of the approach. The method is superficially analogous to asking a skilled human driver to repeatedly execute a given manoeuvre, with performance gradually improving until it is considered ‘optimal’. However, a key step in this iteration is a global optimisation of the reference ‘flow field’, something that is presumably outside of the capability of a human driver. The other notionally important aspect of the iterative DM optimisation – adaptation of friction limits – has not been exercised in this study, because the tyre utilization limit was achieved in the first iteration. One possible reason for this is that the particular vehicle model has very uniform acceleration limits, even under transient conditions, and these are very close to the limits chosen; so, for a less ‘well behaved’ vehicle the adaptation would be much more significant. A second possibility, is that the driver model has a natural tendency to use all remaining friction under longitudinal control, provided the lateral tracking requirements are met. This is unlikely however, since the optimized reference pays equal attention to speed and lateral control. It is also possible that the ‘worst tyre’ utilization condition is too simple, and can be replaced by a more appropriate metric in the future. This issue, although interesting, is of reduced concern since the overall performance in the obstacle avoidance is very close to that of GOC, (as well as the reference particle model).

The use of an intermediate flow optimisation not only provides an essential input for the driver model, it also allows an independent assessment of the validity of the driver-vehicle performance, via the angle and path errors. Compared to the GOC benchmark, it turns out that the fine details of the driver control input time histories are indeed quite different, but that the overall performance is very similar indeed. The conclusion is that the DM result is ‘close to optimal’ in a space of solutions where many different driving styles are also ‘close to optimal’.

One important issue must be considered. It is conceivable that results equivalent to those achieved above could have been obtained directly via a carefully designed driver model. But in a future application, perhaps with a different vehicle model, we would clearly have little confidence in the optimality of the results without again needing some independent benchmark. So what is there about the DM technique that should be different. One reason has already been mentioned: there are intrinsic benchmarks within the methodology. A second reason would be that the final results are insensitive to the particular assumptions of the driver model. A full study of this phenomenon is outside of the scope of this paper, but for interest a repeat of the avoidance manoeuvre was undertaken, with the yaw control made less aggressive and the slip angle control entirely removed (k_1 reduced from 1 to 0, k_2 set to zero in equation (24)). The relevant vehicle paths are shown in Figure 13. The initial path is considerably worse than before, confirming that the control has indeed changed, but after 17 iterations of phase compensation relative to the fixed reference field, the transfer time is reduced to 2.82 seconds (compared to 2.81 previously), the maximum path error is 0.20 m, and the peak angle error is 1.5 degrees.

** INSERT FIGURE 13 **

Of course, the critical link between the simulation and the confirmation of optimality is that the particle model is representative of the driver plus vehicle. If this is true, satisfactory reference field tracking is sufficient to infer the (near) optimality of the vehicle results, and from the results presented this appears to be the case, provided the driver model has some ‘minimum level of competence’, and a time-dependent phase compensation is applied. It is interesting to note (MacAdam 2003, Weir and McRuer 1970) that similarly simple models of coupled driver-vehicle dynamics have been inferred even in the case of the human driver performance. Clearly further work is needed to firmly establish this technique as a practical design tool, but the indications from this research is that the Dual Model technique has fundamental and durable significance.

REFERENCES

1. Gillespie, T.D. (1992), "Fundamentals of Vehicle Dynamics", SAE International.
2. Modjtahedzadeh, A. and Hess, R.A., (1993) "A Model of Driver Steering Control Behavior for Use in Assessing Vehicle Handling Qualities", *J. of Dynamic Systems, Measurement and Control*, Trans. of the ASME, Vol. 115, pp 456-464.
3. Horiuchi, S. and Yuhara, N. (2000) "An Analytical Approach to the Prediction of Handling Qualities of Vehicles With Advanced Steering Control System Using Multi-Input Driver Model", *J. of Dynamic Systems, Measurement and Control*, Trans. of the ASME, Vol. 122, pp 490-497.
4. Sharp, R.S., Casanova, D. and Symonds, P. (2000) "A Mathematical Model for Driver Steering Control, with Design, Tuning and Performance Results", *Vehicle System Dynamics*, Vol. 33, pp 289-326.
5. Prokop, G. (2001) "Modeling Human Vehicle Driving by Model Predictive Online Optimization", *Vehicle System Dynamics*, Vol. 35, pp 19-53.
6. Gordon, T.J. and Best, M.C. (2002) "A Sequential Dual Model Approach to Lap Optimisation", *Proceedings of the 6th International Symposium on Advanced Vehicle Control (AVEC)*, Hiroshima, Japan, pp 99-104.
7. Gordon T.J. and Sharp R.S. (1998) "On Improving the Performance of Automotive Semi-Active Suspension Systems Through Road Preview" *Journal of Sound and Vibration*, Vol 217(1), pp 163-182.
8. Best, M.C. and Gordon, T.J. (2002) "Simultaneous Optimisation of Vehicle Parameter and Control Action to Examine the Validity of Handling Control Assumptions", *Proceedings of the 6th International Symposium on Advanced Vehicle Control (AVEC)*, Hiroshima, Japan, pp 87-92.
9. Milliken, D.L. and Milliken, W.F. (1995), "Race Car Vehicle Dynamics", SAE International.
10. Pacejka, H.B. (2002) "Tyre and Vehicle Dynamics", Butterworth Heinemann.
11. Gordon, T.J., Best, M.C. and Dixon, P.J. (2002), "An Automated Driver Based on Convergent Vector Fields", *Proc Inst Mech Engrs*, vol 216, part D, pp329-347.
12. Liu, G.P., Yang, J.B. and Whidborne, J.F. (2003), "Multiobjective Optimisation and Control", Research Studies Press Ltd., England.
13. Bryson A.E. and Ho, Y.C. (1975) "Applied Optimal Control: Optimisation, Estimation and Control," Hemisphere, New York.
14. Marsh C., (1992) "A Nonlinear Control Design Methodology for Computer-controlled Vehicle Suspension Systems," *PhD Thesis*, Loughborough University.
15. The MathWorks Inc (2004), Matlab and Simulink, Release 14.
16. Press, W.H. Teukolsky, S.A. Vetterling W.T. and Flannery, B.P. (1992), "Numerical Recipes : The Art of Scientific Computing," Cambridge University Press, Cambridge.
17. MacAdam, C.C. (2003), "Understanding and Modeling the Human Driver", *Vehicle System Dynamics*, Vol. 40, Nos. 1-3, pp 101-134.
18. Weir, D.H. and McRuer, D.T. (1970) "Dynamics of Driver Vehicle Steering Control", *Automatica*, 1970, Vol. 6, pp 87-98.

TABLES

Table 1. Vehicle Model Notation and Parameter Values

States and Dynamic Variables (<i>units</i>)	
U	vehicle forward velocity ($m s^{-1}$)
V	sideslip velocity ($m s^{-1}$)
p	roll angular velocity ($rad s^{-1}$)
r	yaw angular velocity ($rad s^{-1}$)
ϕ	roll angle (rad)
θ	yaw angle (rad)
$\mathbf{r}_G = (x_G, y_G)$	vehicle position in global coordinates (m)
T_i	driveline or brake torque at wheel i
ω	wheel angular velocity ($rad s^{-1}$)
w	magnitude of tyre vertical load (N)
Parameters (<i>value, units</i>)	
I_{xx}	body roll moment of inertia ($200 kgm^2$)
I_{zz}	yaw moment of inertia ($2500 kgm^2$)
I_{xz}	roll/yaw product of inertia ($0 kgm^2$)
I_w	wheel (plus associated driveline) moment of inertia ($5 kgm^2$)
M	vehicle mass ($1400 kg$)
a	longitudinal Distance of C of G to front axle ($1.16 m$)
b	longitudinal Distance of C of G to rear axle ($1.54 m$)
h	C of G height above roll axis ($0.33 m$)
h_0	ground plane to roll axis distance below C of G ($0.27 m$)
h_f	front suspension roll centre height above ground ($0.1 m$)
h_r	rear suspension roll centre height above ground ($0.5 m$)
ε	roll axis inclination angle - downwards to front of vehicle ($8.4 deg$)
t_f	front track ($1.5 m$)
t_r	rear track ($1.5 m$)
r_r	tyre rolling radius ($0.3 m$)
K_f	front roll stiffness ($37 kNm/rad$)
K_r	rear roll stiffness ($16 kNm/rad$)
B_f	front roll damping ($790 Nms/rad$)
B_r	rear roll damping ($860 Nms/rad$)
B, C, D, E	Pacejka formula coefficients ($0.709, 1.41, 1.0, 0.0$ - dimensionless)
c_1, c_2	cornering stiffness parameters ($69 kN/rad, 1.4 kN$)
τ_1, τ_2	time-constants for driver longitudinal control ($0.25 sec, 0.25 sec$)
k_1, k_2, k_3	driver lateral control gains ($1, 1, 1$)

Table 2. Performance Summary for the Obstacle Avoidance Manoeuvre

Simulation Model	total time (sec)	max speed (ms⁻¹)	min speed (ms⁻¹)	exit speed (ms⁻¹)	peak angle error (deg)	max path deviation (m)
Reference particle model	2.76	15.4	10.4	15.2	0.7	-
Initial DM vehicle	2.92	15.4	9.8	13.5	2.8	0.39
Phase compensated DM vehicle	2.81	15.4	10.4	14.6	1.3	0.25
GOC optimised vehicle	2.78	15.5	12.0	15.5	-	-

FIGURE CAPTIONS

Figure 1. Illustration of Track Geometry

Figure 2. Discretised Velocity Reference

Figure 3. Summary of GOC Algorithm

Figure 4. Track Boundaries and Optimised Velocity Reference

Figure 5. Reference Path from Particle Model and Vehicle Path from Combined Driver-Vehicle Model

Figure 6. Phase Delay in Vehicle Path Angle and Longitudinal Acceleration

Figure 7. Algorithm for Phase Delay Estimation

Figure 8. Delay after Phase Compensation in Vehicle Path Angle and Longitudinal Acceleration

Figure 9. Time History of the Tyre Utilization Parameter (ε)

Figure 10. Comparison of Vehicle Paths

Figure 11. Comparison of Vehicle Speeds and Slip Angles

Figure 12. Comparison of Vehicle Control Inputs

Figure 13. Vehicle Paths for a Modified Driver

FIGURES

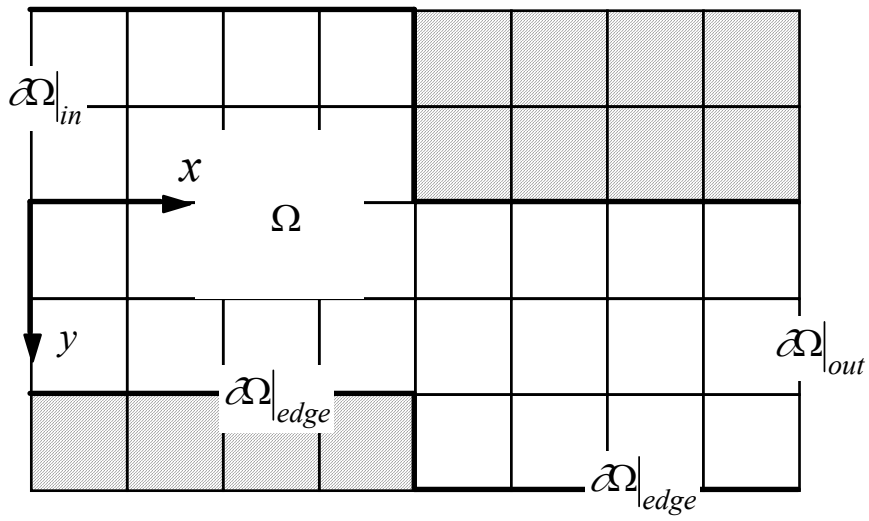


Figure 1. Illustration of Track Geometry

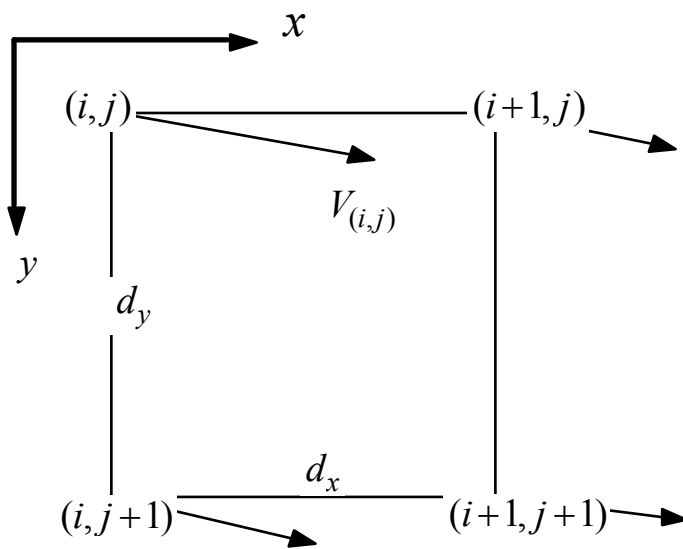


Figure 2. Discretised Velocity Reference

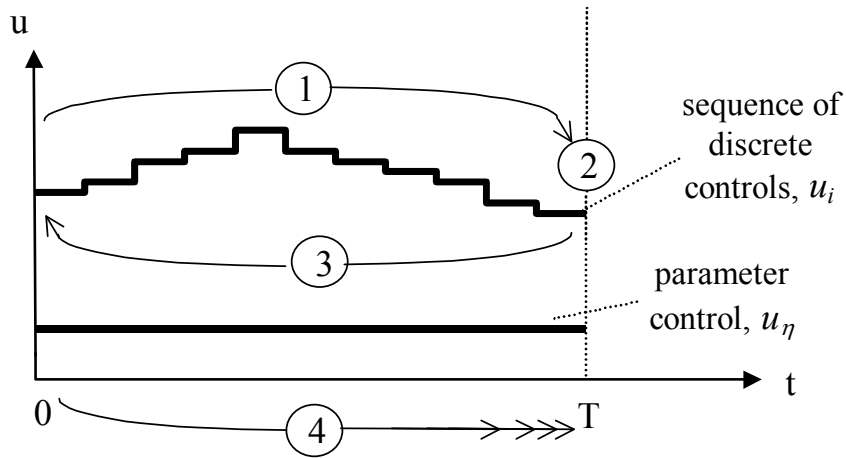


Figure 3. Summary of GOC Algorithm

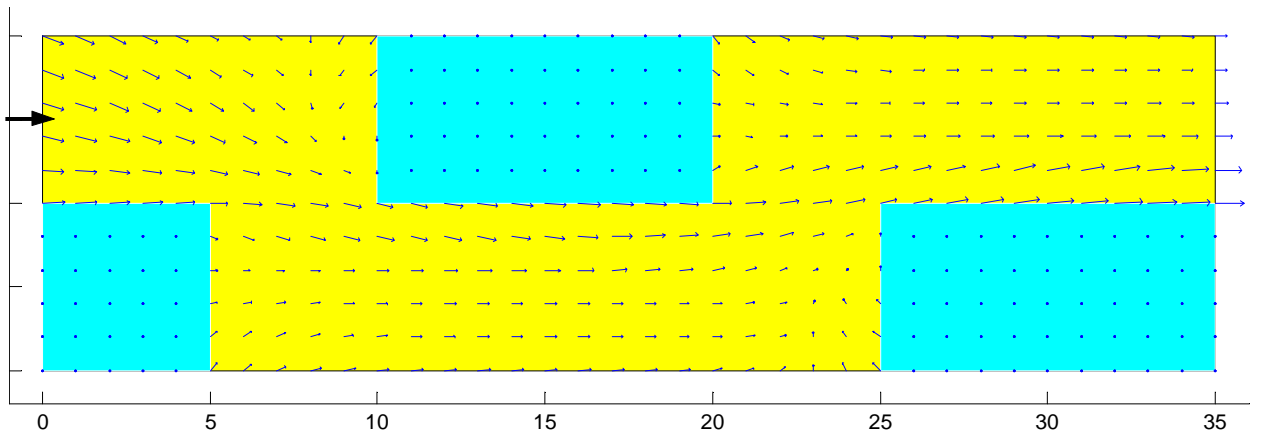


Figure 4. Track Boundaries and Optimised Velocity Reference

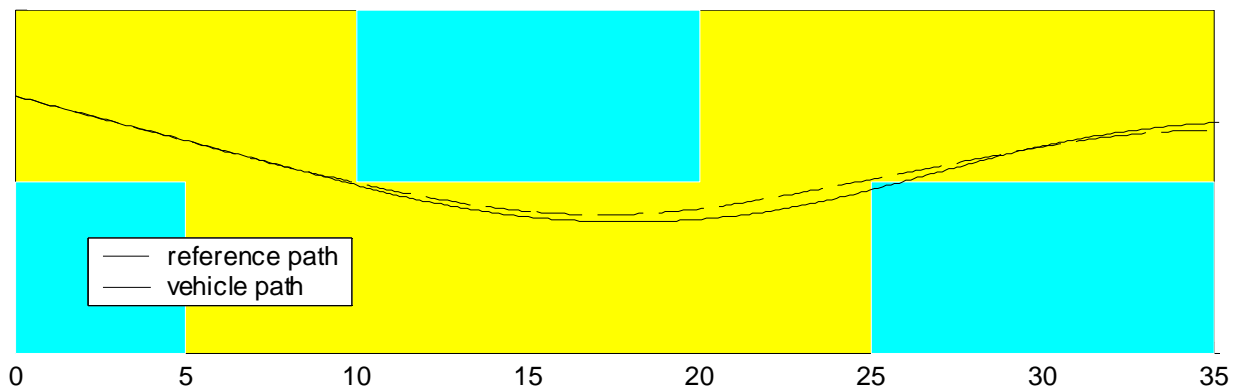


Figure 5. Reference Path from Particle Model and Vehicle Path from Combined Driver-Vehicle Model

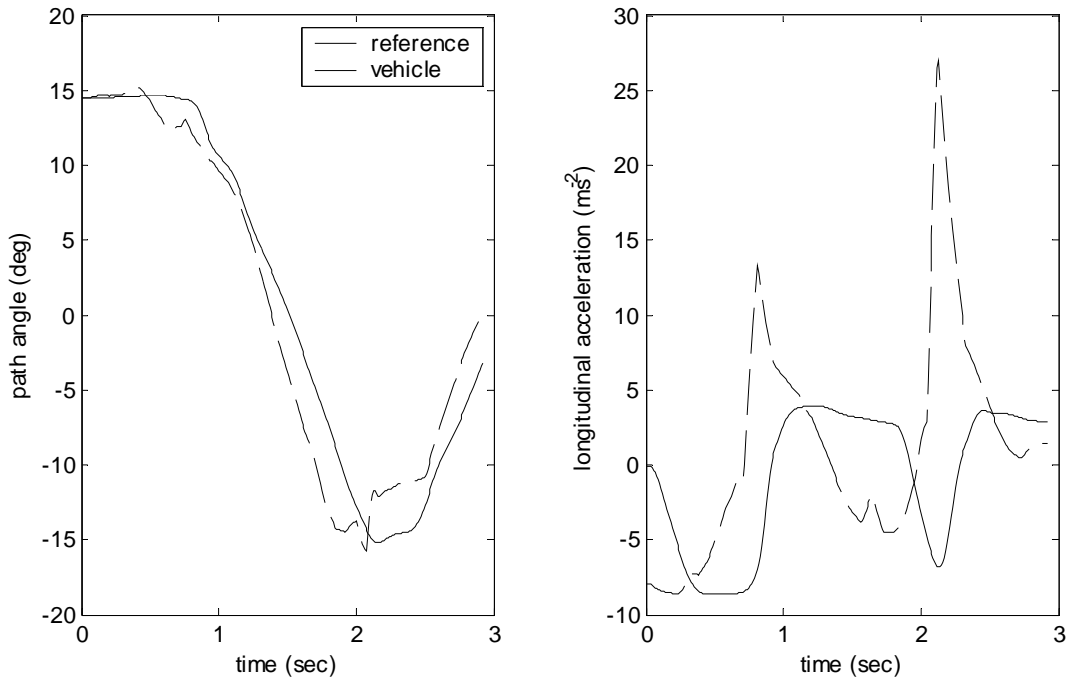


Figure 6. Phase Delay in Vehicle Path Angle and Longitudinal Acceleration

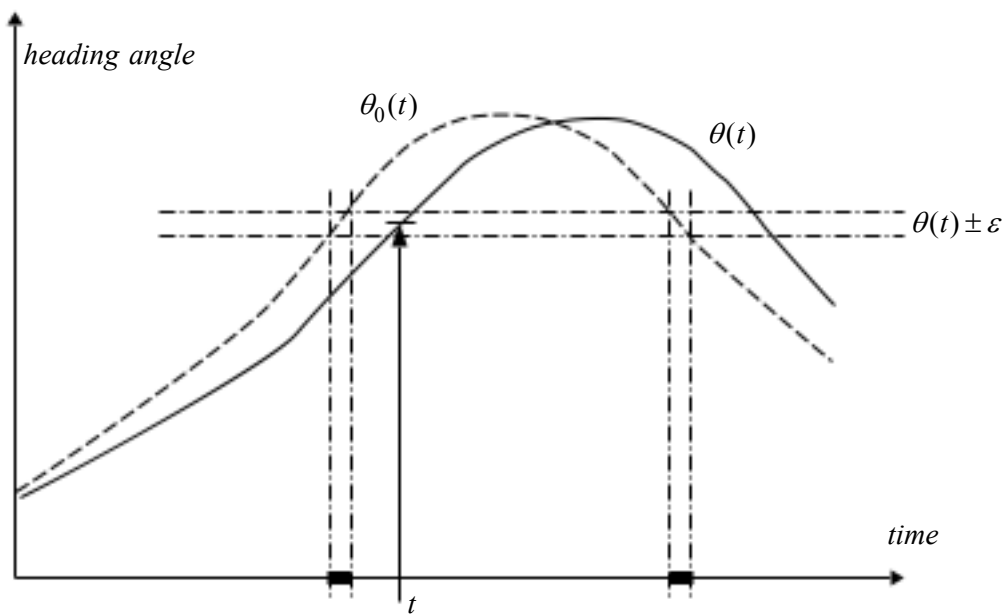


Figure 7. Algorithm for Phase Delay Estimation

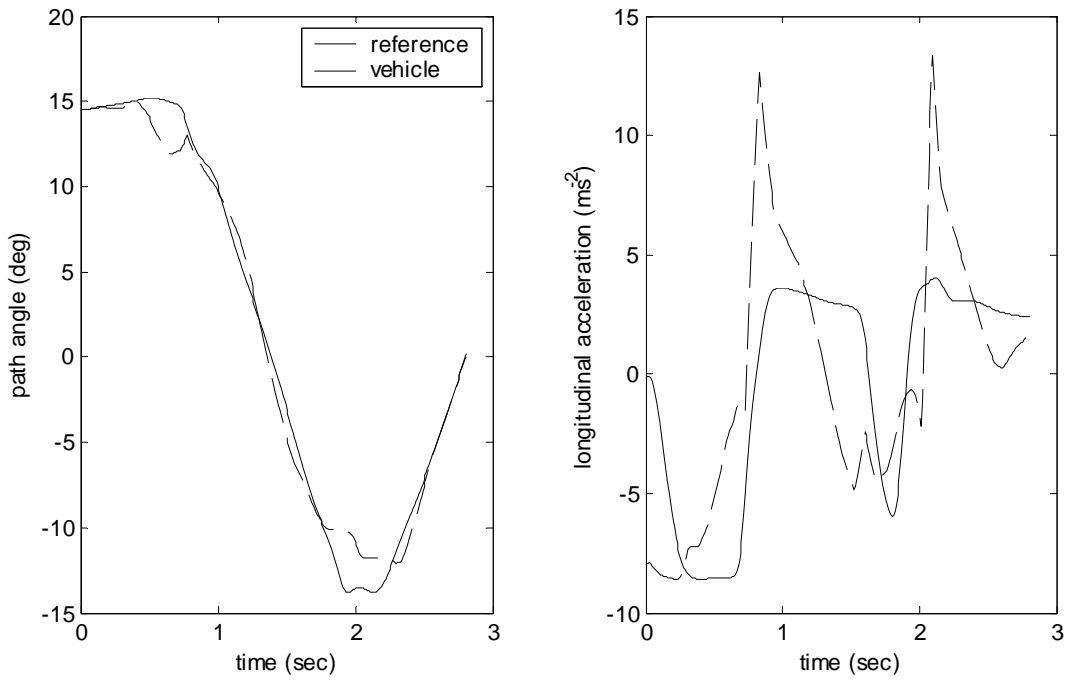


Figure 8. Delay after Phase Compensation in Vehicle Path Angle and Longitudinal Acceleration

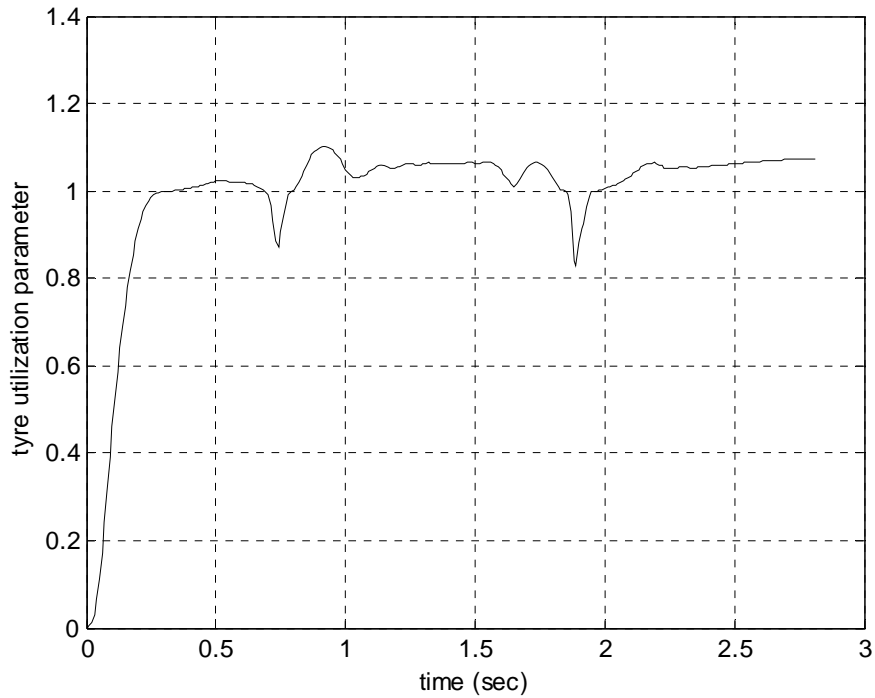


Figure 9. Time History of the Tyre Utilization Parameter (ϵ)

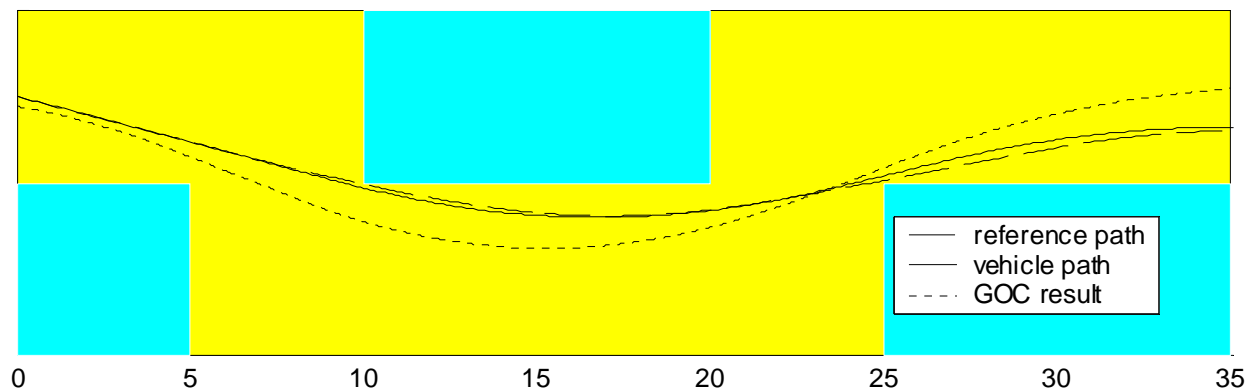


Figure 10. Comparison of Vehicle Paths

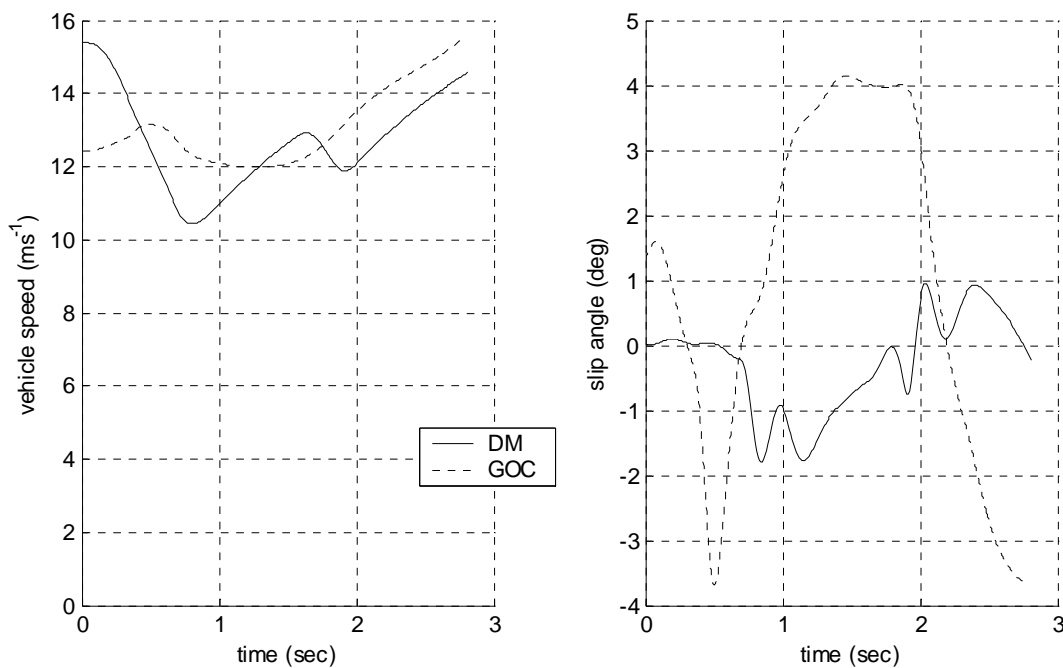


Figure 11. Comparison of Vehicle Speeds and Slip Angles

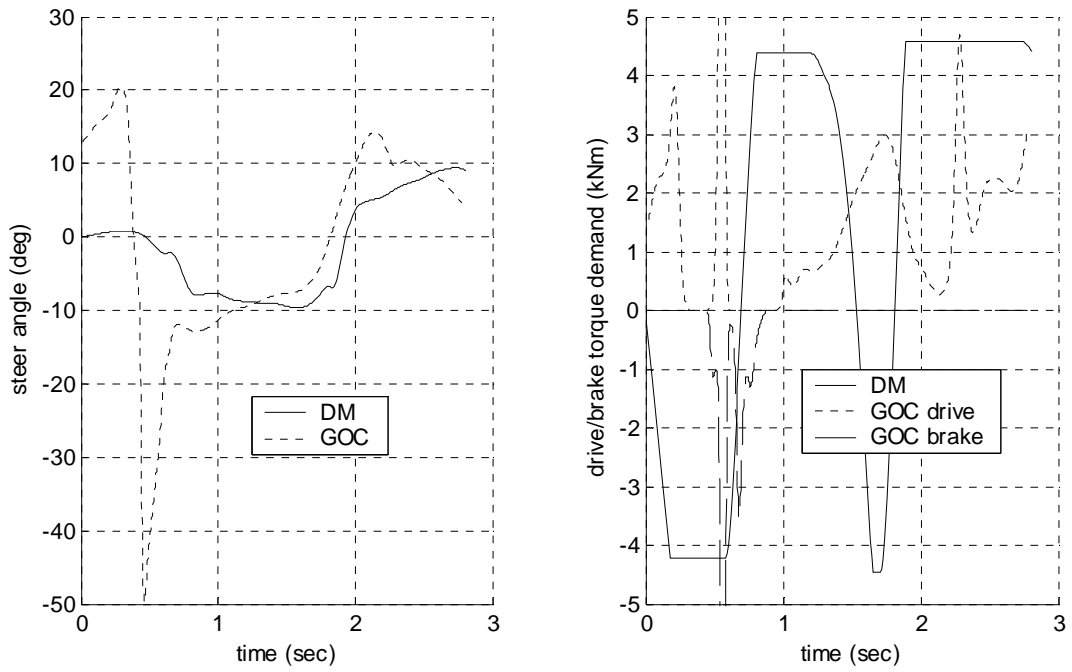


Figure 12. Comparison of Vehicle Control Inputs

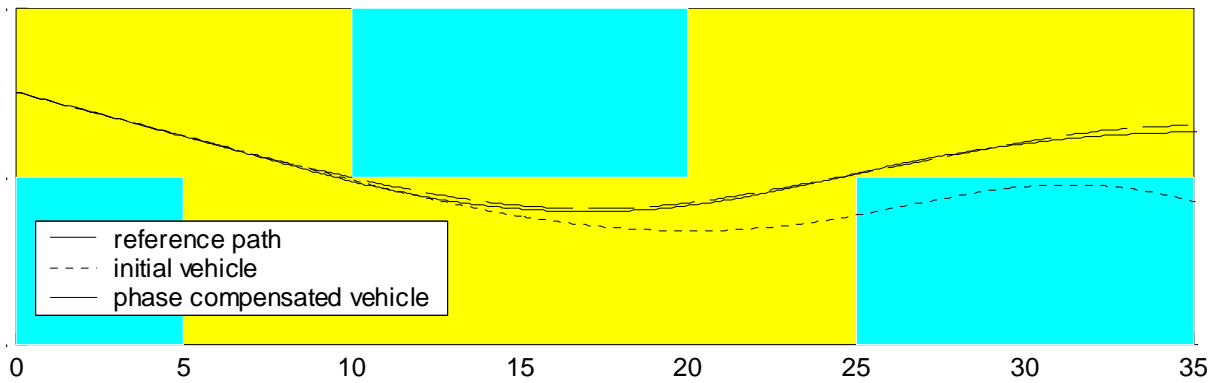


Figure 13. Vehicle Paths for a Modified Driver