Simultaneous optimisation of vehicle parameter and control action to examine the validity of handling control assumptions

This item was submitted to Loughborough University's Institutional Repository by the/an author.

**Citation:** BEST, M.C. and GORDON, T.J., 2002. Simultaneous optimisation of vehicle parameter and control action to examine the validity of handling control assumptions. IN: Proceedings of the 6th International Symposium on Advanced Vehicle Control (AVEC), Hiroshima, Japan, September 2002, pp. 87-92.

**Additional Information:**

- This is a conference paper.

**Metadata Record:** [https://dspace.lboro.ac.uk/2134/8326](https://dspace.lboro.ac.uk/2134/8326)

**Version:** Accepted for publication

**Publisher:** © Society of Automotive Engineers of Japan (JSAE)

Please cite the published version.
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.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
Simultaneous Optimisation of Vehicle Parameter and Control Action to Examine the Validity of Handling Control Assumptions

Matthew C Best and Timothy J Gordon
Loughborough University, UK

In this paper a general method is presented for optimising system parameters and inputs. The Generalised Optimal Control technique involves iterative resimulation of system states, but is applicable to any (smoothly) nonlinear system, and can be operated using non-quadratic cost functions. Here it is applied to find optimal steer and torque inputs for a 2DOF vehicle handling model with a (combined slip) nonlinear tyre model. System parameters for centre of gravity and yaw inertia are simultaneously optimised, and hence the validity of some handling control assumptions – particularly the benefits of zero sideslip – is examined. The results are satisfactory, and they are mainly in keeping with expectation. The method is proven to be effective, though computationally rather expensive!

Keywords / Vehicle dynamics and control, Vehicle Handling, Optimisation

1. INTRODUCTION
Researchers who consider algorithms for controlling race-car handling, must make assumptions about what behaviour is desirable in the vehicle. For example, they frequently consider that zero sideslip velocity is desirable during steady-state cornering, and so design the vehicle to exhibit neutral steer behaviour and / or prescribe control algorithms about this reference (examples are [1] and [2]). Another assumption is that a particular ‘magic’ ratio between mass and yaw inertia increases control of yaw transients, and hence yields competitive advantage. However, the validity of these assumptions can be strongly influenced by combination effects between driver or automated control inputs, vehicle setup and driver dynamics.

In this paper we consider a way to combine the choice of control sequence and selected vehicle parameters in an optimisation technique which can test these assumptions, within a fairly simple model of driver capability. A Generalised Optimal Control (GOC) technique is applied; the process operates iteratively, solving a two point boundary value problem over a fixed time interval, using Pontryagin’s Minimum Principle. The method has advantages over standard optimal control techniques in that it can be applied to any (smoothly) nonlinear plant, and the cost model is not restricted to quadratic functions. The only limitation is that model / simulation complexity must be kept at a suitably simple level, as the method uses multiple simulations to converge on optimal behaviour.

Here the technique has been adapted and extended from earlier work (eg in [3] and [4]), so that model parameters can be selected as additional control inputs, which remain constant over time; in this way the final solution can provide vehicle setup as well as control information. The revised GOC method is presented in Section 2.

The simulations use a two degree-of-freedom yaw / sideslip handling model with a stiff suspension model which allows meaningful load transfer to four independent tyre models; the tyres employ combined slip Pacejka ‘magic’ formulae to impose realistic friction limits. The time sequence of acceleration / braking and steering inputs are optimised along with the vehicle parameter for centre of gravity position or yaw inertia, with a cost function set to reflect the simple objective of minimum time taken over a simulated section of race track.

The model, implementation detail and cost functions are described in Section 3, and a series of tests are conducted in Section 4 which illustrate the capability of the method, and present some basic findings in combined optimisation of vehicle set-up and control.
2. GENERALISED OPTIMAL CONTROL

The control optimisation is a nonlinear formulation of LQR; controls are sought to minimise a Hamiltonian which is prescribed in terms of a (nonlinear) system of costate equations over a fixed time period. Given a cost function of time, \( L \) and a residual cost associated with final states, \( L_T \):

\[
J = L_T \left[ x(T) \right] + \int_0^T L \left[ x(t), u(t) \right] dt
\]  

(1)

Adding constraint equations to this with a vector of Lagrange multiplier functions, \( p(t) \):

\[
J = L_T \left[ x(T) \right] + \int_0^T \left[ L \left[ x(t), u(t) \right] + p^T(t) \left[ g \left[ x(t), u(t) \right] - x(t) \right] \right] dt
\]

(2)

where \( g \) is given by the system equations, \( x = g \left[ x(t), u(t) \right] \). The lagrange multipliers can be formed as a so-called costate system, and the Hamiltonian function can then be defined (see for example [5]) as

\[
H = L \left[ x(t), u(t) \right] + p^T(t) g \left[ x(t), u(t) \right]
\]

(3)

Eqn. 2 can now be integrated by parts to give,

\[
J = L_T \left[ x(T) \right] + p^T(0)x(0) - p^T(T)x(T) + \int_0^T \left[ H + p^T(t)x(t) \right] dt
\]

(4)

Considering small changes \( \delta J \) in the dynamic cost caused by small changes in the controls \( \delta u(t) \) and in the states \( \delta x(t) \):

\[
\delta J = \left[ \frac{\partial L_T}{\partial x} - p^T(T) \right] \delta x(t) + p^T(0) \delta x(0) + \int_0^T \left[ \frac{\partial H}{\partial x} + p^T(t) \right] \delta x(t) + \frac{\partial H}{\partial u} \delta u(t) \right] dt
\]

(5)

and costates can be chosen such that \( \delta J \) depends only on changes in the controls by imposing the following conditions:

\[
p^T(t) = -\frac{\partial H}{\partial x} = \frac{\partial L}{\partial x} - p^T \frac{\partial g}{\partial x}, \quad p^T(T) = \frac{\partial L_T}{\partial x}
\]

(6)

hence,

\[
\delta J = p^T(0) \delta x(0) + \int_0^T \left[ \frac{\partial H}{\partial u} \delta u(t) \right] dt
\]

(7)

As we seek an open loop series of controls to minimise the dynamic cost \( J \) for constant conditions, \( \delta x(0) = 0 \), and the minimum cost must therefore exist where

\[
\frac{\partial H}{\partial u} = 0, \quad \forall t
\]

(8)

In [6] an approximation to the continuous solution is found using a discrete sequence of controls, each held constant for a small time \( dt \). Within the time period for each control, the cost gradient can then be identified as

\[
\frac{\partial J}{\partial u_i} = \int_{t_i}^{t_{i+1}} \frac{\partial H}{\partial u_i} dt
\]

(9)

So it is feasible to establish a gradient based iteration optimisation of a sequence of discrete controls spanning the required time frame (Fig 1).

Note that, provided the control remains constant for its discretisation period, the method is valid irrespective of the duration. Also, independent controls can take different discretisations. Coupling this with the fact that in the nonlinear model, any variable can be designated a control, it is straightforward to include model parameters within the optimisation. These are defined simply as controls which remain constant over the entire simulation period, and whose gradients are thus computed as

\[
\frac{\partial J}{\partial \eta} = \int_0^T \frac{\partial H}{\partial \eta} dt
\]

(10)

Figure 1 provides a summary of the algorithm which can be used to conduct the GOC optimisation.

---

**Figure 1 : Summary of GOC algorithm**

1. Using the current discrete control sequence, integrate the state-space system from \( x(0) \) and evaluate \( J_{[0,T]} \).
2. Evaluate the residual cost \( L_T \) and hence \( p(T) \) from Eqn. 6.
3. Integrate the costate system and \( \partial H/\partial u \) in reverse-time from the initial condition \( p(T) \). Calculate cost gradients from Eqn 9.
4. Update the control sequence by a line search optimisation along the steepest descent or successively conjugate gradients to minimise \( J \) (evaluated by repeating Stages 1 & 2).

Repeat Stages 1-4 until suitable convergence of cost and controls, and reduction of cost gradients is achieved.
3. SIMULATION

The GOC algorithm has significant benefits in flexibility, but it can be computationally expensive; the need to calculate partial derivatives of the Hamiltonian with respect to each state, coupled with iterative simulations should caution the user to favour relatively simple, low order models. (Prior papers [3,4] have used two degree of freedom ride and torsional vibration models.) However, by astute use of compiled code and automated code generation, in this paper we stretch the method’s capability to a full vehicle handling model, with four independent combined slip tyre force models. This allows investigation of independent steer and torque, whilst ensuring acceptable accuracy.

3.1 Vehicle Handling Model

The well known normalised combined slip Pacejka model (eg [7]) is used to determine longitudinal and lateral tyre forces (F_x and F_y respectively in the SAE vehicle convention):

$$ F_{xi} = h(u_i, v_i, r_i, \delta_{yi}, Z_i) $$

(11)

Tyre vertical loads, Z are determined using a 'stiff suspension' model which imposes equilibrium conditions on (unmodelled) roll, pitch and bounce degrees of freedom (see also [8]). Assuming a ratio $$ \eta $$ between the front and rear suspension roll moments, the effect of both pitch and roll load transfer is accommodated via:

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & Z_1 \\
-a & -a & b & a & Z_2 \\
c & c & c & -c & Z_3 \\
1 & -1 & -\lambda & \lambda & Z_4
\end{pmatrix} =
\begin{pmatrix}
Mg \\
h \sum f_{xi} \\
h \sum f_{yi} \\
0
\end{pmatrix}
$$

(12)

and the friction coefficient is modified with respect to load, according to the simple expression:

$$ \mu(Z) = \frac{\mu_0}{1 + (2Z / Mg)} $$

(13)

The system equations are then, for the rigid vehicle body:

$$
\dot{u} = \frac{1}{M} \sum f_{x_i} + vr \\
\dot{v} = \frac{1}{I_z} \sum f_{y_i} - ur \\
\dot{r} = \frac{1}{I_z} \left[ c(f_{x_1} - f_{x_2} + f_{x_3} - f_{x_4}) + a(f_{y_1} + f_{y_2} - (b - a)(f_{y_1} + f_{y_2})) \right] \\
\dot{\theta} = r \\
\dot{X} = u \cos \theta - v \sin \theta \\
\dot{Y} = u \sin \theta + v \cos \theta \\
\dot{\omega}_i = \frac{1}{I_w} \left[ \tau_i - r_i f_{x_i} \right]
$$

(14)

and first order lag functions are employed to simulate tyre force generation and impose a simple driver / vehicle bandwidth limitation on torque and steer inputs at the wheels:

$$
\dot{f}_{xi/yi} = \rho_i \left( f_{xi/yi} - \hat{f}_{xi/yi} \right) \\
\dot{\delta}_i = \rho_\delta (\delta_{ui} - \delta_i), \quad \tau_i = \rho_\tau (\tau_{ui} - \tau_i)
$$

(15)

Control inputs \( \delta_i \) and \( \tau_i \) are applied equally to both wheels at either the front and/or rear axle.

### Parameters, \( \eta \) (default values)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>mass</td>
<td>1400 kg</td>
</tr>
<tr>
<td>I_{xz}</td>
<td>yaw moment of inertia</td>
<td>2300 kgm²</td>
</tr>
<tr>
<td>I_{xw}</td>
<td>wheel moment of inertia</td>
<td>0.8 kgm²</td>
</tr>
<tr>
<td>a</td>
<td>longitudinal Distance of CoG to front axle</td>
<td>1.2 m</td>
</tr>
<tr>
<td>b</td>
<td>wheelbase</td>
<td>2.7 m</td>
</tr>
<tr>
<td>c</td>
<td>half track</td>
<td>0.7 m</td>
</tr>
<tr>
<td>h</td>
<td>C of G height above roll axis</td>
<td>0.4 m</td>
</tr>
<tr>
<td>r_i</td>
<td>wheel rolling radius</td>
<td>0.3 m</td>
</tr>
<tr>
<td>\lambda</td>
<td>roll moment distribution factor</td>
<td>0.5</td>
</tr>
<tr>
<td>\mu_0</td>
<td>tyre friction coefficient</td>
<td>0.9</td>
</tr>
<tr>
<td>\rho_0</td>
<td>tyre delay time constant</td>
<td>100</td>
</tr>
<tr>
<td>\rho_\delta</td>
<td>steer input delay time constant</td>
<td>30</td>
</tr>
<tr>
<td>\rho_c</td>
<td>torque input delay time constant</td>
<td>30</td>
</tr>
<tr>
<td>Paceika tyrec model shape coefficients</td>
<td>(0.714, 1.4, 1.0, -0.2)</td>
<td></td>
</tr>
<tr>
<td>zero lateral slip cornering stiffness</td>
<td>50 kN</td>
<td></td>
</tr>
<tr>
<td>zero longitudinal tyre slip rate</td>
<td>60 kN</td>
<td></td>
</tr>
</tbody>
</table>

### Table 1: Model nomenclature & parameters

3.2 Implementing the Optimisation

The costate system (Eqn 6) 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. Direct evaluation of the partial derivatives is then possible, but the resultant formulae are long and inefficient (eg \( \partial H/\partial u \) leads to an equation comprising 134,386 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

$$
\frac{\partial H}{\partial u} = \frac{\partial H'}{\partial u} + \sum \frac{\partial H'}{\partial f_i} \frac{\partial f_i}{\partial u}
$$

(16)

where H’ is the Hamiltonian written in terms of the tyre forces \( f_i \) and \( \partial f_i/\partial u \) is further broken down into 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 (described in [9]). The timestep is
kept constant within each control \((u_t)\) time interval, and
to ensure accuracy the code is written to monitor state
errors and adjust the timestep duration accordingly.
Finally, the integration and derivative codes are
compiled to achieve the fastest possible simulation
execution time.

One further modification is made to improve
optimisation of the handling model; although the \((u_t)\)
controls remain functions of time, they are held
constant for a specific distance \((\delta\ell)\) which the vehicle
travels – this improves the speed of convergence when
steer and torque inputs are to be optimised
simultaneously.

### 3.3 Cost Functions

In all these simulations the target is to gain
maximum distance in a given direction, whilst
maintaining yaw (velocity and angle) control of the
vehicle. The residual, final cost function is set as:

\[
L_T = 0.001(X_G - X(T))^2 + 100r^2(T) + 100\delta^2(T) \tag{17}
\]

with \(X_G\) set at some large, unattainable distance (in this
case 200m). The coefficients are chosen to ensure that
the maximum distance term dominates by a factor of
approximately ten.

The time varying cost is dominated by track
following terms:

\[
\begin{align*}
L_{\text{track}} &= \chi^2, \quad \text{if} |\chi| < \ell \\
L_{\text{track}} &= \chi^2 + 0.01(|\chi| - \ell)^4, \quad \text{if} |\chi| \geq \ell
\end{align*} \tag{18}
\]

where \(\chi\) is the perpendicular distance of the vehicle to
the track centre, which is defined using straight-line
segments. \(\ell\) is the half track width, set \(\ell = 2.5\). An
additional term is also set, at a relatively low level, to
guard against excessive wheel spin, such that

\[
L = L_{\text{track}} + 10\sum_{i} \left(\frac{w_i r_i - u_i}{\delta^2}\right)^2 \tag{19}
\]

### 4. EXPERIMENTS

#### 4.1 Steer only

First consider a simple 90° turn; the vehicle sets
off at 20m/s in the positive X direction \((\theta(0) = 0)\), and
the distance objective is to maximise Y (with no
‘track’). Figures 2 compares two conditions. In the
first, front steer angle is optimised, with rear steer and
all torque inputs set to zero. The parameter control
seeks the best fore / aft centre of gravity position, \(u_{\eta} = a\). The experiment is then repeated with both front and
rear steer \((\delta\ell)\) optimised.

In the front steer case, a high steer angle (7°) is first
applied to drive the vehicle into a limit friction yaw
rate (the normalised slip plot shows the front steer
case). This is recovered using a small opposite steer
action around 5 sec. The centre of gravity optimises
well ahead of the centre of the vehicle \((\delta a_{\text{opt}} = 1.07)\),
though this provides good balance given that the roll
moment distribution is set to induce oversteer \((\lambda = 0.5)\).

The dual steer vehicle adopts less dramatic
steering, achieving a greater distance to the right (see
vehicle path) with less vehicle sideslip. The centre of
gravity is slightly further back here, which is
intuitively sensible as the rear steer prevents terminal
oversteering, allowing a more even weight distribution.

#### 4.2 Combined steer and torque – lanechange

Here we introduce a torque input, at the rear; this
should alter the optimal fore/aft balance – we might
expect a larger value of $a$ to increase acceleration capacity. Figure 3 shows how the model tackles a track-imposed lane-change manoeuvre over four seconds, with the initial speed again set to 20m/s. The results are again presented with and without the inclusion of rear steer.

Here the dual steer vehicle achieves greater distance due to a slightly increased torque at the rear wheels, but note that the transient vehicle sideslip trace is very similar to the front steer car. Interestingly, with the exception of the inner rear, the tyres remain well within their friction limits through both turns. This is true for both cases, though again only the front steer vehicle is shown here. As expected the centre of gravity position has converged further back – almost at the centre of the vehicle in both cases.

### 4.3 Yaw inertia experiment

If a realistic driver model were included in the system to be optimised, it might be possible to examine the performance value of tuning the ratio of yaw inertia to mass (the dynamic index). A ratio of 0.92 is thought by many to offer a good compromise between rapid vehicle yaw response and driver controllability.

Here we have rather too basic a driver model to make a judgement, but it is interesting to note the results of optimisation on $I_{zz}$. Figure 4 shows how the parameter and cost converges against iteration number for a test which is again executed on the lane-change track, though in this instance with only front wheel drive and steer.

Note how $I_{zz}$ changes within only around 200 iterations, remaining almost unchanged during the other iterations; this occurs (although to a lesser extent) with all the optimisations – the time varying controls converge almost independently of $\alpha$ until they become ‘aligned’ in such a way that cost gain can be achieved by change in all controls.

The bottom plot of Figure 4 shows how lower $I_{zz}$ has allowed more rapid change in control, with the first
steer event delayed, allowing for a longer period of acceleration initially, and hence lower cost. This might be a pathological case for the method, as the controlled parameter has a relatively small effect on cost. It does serve to caution the user to diligence in the monitoring of convergence however; this should principally be based on successive iterations yielding consistently low control gradients.

5. CONCLUDING REMARKS
These results show the scope of this adapted GOC method for examining optimal behaviour, though the relatively small set of scenarios do not provide strong enough insight into specific vehicle set-up or sideslip requirements. The drawback with the method (in this guise) is in its very slow convergence.

For these simulations each iteration took approximately 30 seconds on a modern (1.5GHz) PC, so some of the results needed more than one day to converge. The convergence issue is also exacerbated when the tyres are close to, or beyond their limit of friction; this makes it inefficient to carry out ‘follow-on’ optimisations – for example using the front steer results as a basis on which to optimise the dual steer case. As a result it was necessary to re-optimise all the cases presented above, from very simple initial conditions.

The method is effective if not efficient however, so further research is planned to reduce computational load in handling optimisations. Simplification of the tyre model, perhaps by spline fitting the magic formula, is one possibility. Another is to further reduce the control’s dependence on time, by restating the two point boundary value problem in terms of distance travelled. If each control action is posed as a function of \( \delta x \) rather than \( \delta t \), the torque and steer controls will be more successfully decoupled.

REFERENCES