CHALMERS | GÖTEBORG UNIVERSITY

PREPRINT 2007:42

Using an adaptive FEM to determine the optimal control of a vehicle during a collision avoidance manoeuvre

KARIN KRAFT
STIG LARSSON
MATHIAS LIDBERG

Department of Mathematical Sciences
Division of Mathematics
CHALMERS UNIVERSITY OF TECHNOLOGY
GÖTEBORG UNIVERSITY
Göteborg Sweden 2007

Preprint 2007:42

Using an adaptive FEM to determine the optimal control of a vehicle during a collision avoidance manoeuvre

Karin Kraft, Stig Larsson, Mathias Lidberg

Department of Mathematical Sciences
Division of Mathematics
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden

Göteborg, November 2007

Preprint 2007:42
ISSN 1652-9715

Matematiska vetenskaper
Göteborg 2007

Using an adaptive FEM to determine the optimal control of a vehicle during a collision avoidance manoeuvre

Karin Kraft and Stig Larsson
Mathematical Sciences, Chalmers University of Technology and Göteborg University
karin.kraft@chalmers.se
stig@chalmers.se
Mathias Lidberg
Applied Mechanics, Chalmers University of Technology
mathias.lidberg@chalmers.se

Abstract

The optimal manoeuvering of a vehicle during a collision avoidance manoeuvre is investigated. A simple model where the vehicle is modelled as point mass and the mathematical formulation of the optimal manoeuvre are presented. The resulting two-point boundary problem is solved by an adaptive finite element method and the theory behind this method is described.

Keywords: Vehicle dynamics, collision avoidance manoeuvre, optimal control, boundary value problem, adaptive finite element method.

1 Introduction

Historically, active safety systems for vehicles are designed to ensure that the driver can steer and brake the vehicle. Automatic controls are being incorporated in conventional safety systems such as ESC with the ability to minimise driver errors. It is important to evaluate how such systems perform in various situations. For this purpose the American National Highway Traffic Safety Administration has proposed a test called "sine with dwell" to evaluate the performance of a car during a collision avoidance manoeuvre. In such a manoeuvre the driver of a vehicle tries to avoid an object that suddenly appears in front of the vehicle [1].

This article is a theoretical investigation of how to combine braking and steering to perform a collision avoidance manoeuvre in an optimal way. The optimisation function has two goals. The primary objective is to achieve a vehicle trajectory distance to avoid collision. If the primary objective cannot be met, then a secondary objective is to minimise the final velocity of the unavoidable collision. This is an optimal control problem, since we want to find controls and states which minimise a quantity subject to constraints consisting of a dynamical system.

Methods for solving optimal control problem can be classified as either a *direct* or an *indirect* approach [4]. The direct approach approximates the dynamical system and then looks for a solution, such that the objective function is minimised. The indirect approach determines the necessary conditions for optimality, and then seeks their solution. Taking the indirect approach means that we have to derive the adjoint equations and optimality conditions explicitly. However, we use this approach because the indirect approach in combination with the finite element method gives us the possibility to control the error in the numerical solution of the optimality conditions over the entire interval. We believe that this is important for an efficient solver. Our first attempts in this direction are described in the present work. The most common numerical methods for solving optimal control problems based on either a direct or an indirect approach are multiple shooting or collocation methods [4]. However, in this work we use an adaptive finite element method similar to the one in [8] to solve the necessary optimality conditions that arise in an indirect approach. In the presented adaptive finite element method we derive an a posteriori error estimate which is used as a basis for error control and adaptive mesh refinement. Since we estimate the error over the entire time interval we can use the computational power where it is best needed. This gives us the ability to choose the level of modelling for the FEM solver and we also believe that we will be able to solve optimal control problems for more advanced vehicle models.

2 The collision avoidance manoeuvre

A traffic situation that presents a safety risk is defined for the investigation. A vehicle is driven on a plane homogeneous surface. There is an obstacle in front of the vehicle. How shall the driver manoeuver the vehicle in the best way in order to avoid collision and, if that is not possible, minimise the collision severity? In Figure 1 we show a picture of the steering in a collision avoidance manoeuvre. The driver performs the avoidance manoeuvre by braking and steering simultaneously. The manoeuver starts at time $t = 0$, at a distance a from the obstacle and with velocity U_0. After the manoeuvre the car hits or passes the obstacle at time T, with velocity U_T and at distance b from the original track.

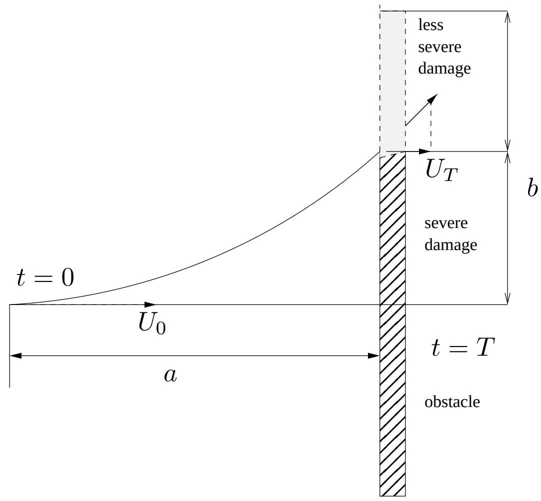

Figure 1: The collision avoidance manoeuvre

We know that the higher speed the vehicle has at the time of collision, the more severe the accident. Therefore we want to determine the best braking and steering strategy to avoid collision or minimise the speed perpendicular to the object at impact. This optimisation problem can be formulated as follows: given the manoeuvre distances a and b determine the braking and steering strategy that minimises the final velocity component U_T.

3 Point mass vehicle dynamics model

The driver controls the braking and steering but it is the friction forces acting on the car tyres that makes the vehicle move in a certain direction. For our purposes, the dynamics of the vehicle due to these forces can be modelled as a point mass [10]. We introduce the X-axis as the direction of the original track and the Y-axis as the axis perpendicular to the X-axis. The equations of planar motion for the vehicle then become

$$\begin{aligned} \ddot{X} &= -\mu g \cos(\beta)\,, \\ \ddot{Y} &= \mu g \sin(\beta)\,, \end{aligned} \tag{1}$$

where β is the angle between the X-axis and the sum of the forces between the tyres and the road, μ is the friction coefficient and g is the gravitational acceleration.

4 Optimal control theory for the collision avoidance manoeuvre

4.1 State-space formulation

To derive the necessary conditions of optimality, the final speed optimal control problem is formulated in state space by transforming differential equations (1) to first order differential equations. The equations of planar motion for the vehicle then become

$$\dot{z} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{U} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} U \\ V \\ -\mu g \cos(\beta) \\ \mu g \sin(\beta) \end{bmatrix}, \tag{2}$$

where U and V are the velocities in the X and Y directions, respectively.

We want to minimise the speed at the time of the accident in order to reduce the damage. Therefore we formulate an optimal control problem: Find the state $z(t) \in \mathbb{R}^n$ and control $\beta(t) \in \mathbb{R}^m$ which fulfill the following minimisation problem

$$\begin{aligned} \min \quad & \mathcal{J}(z,\beta) = c^{\mathrm{T}} z\,(T) \\ \text{s.t.} \quad & \dot{z}(t) = f(z,\beta), \\ & J_0 z(0) = z_0, \quad J_T z(T) = z_T. \end{aligned} \tag{3}$$

Here J_0 and J_T are diagonal matrices with zeroes or ones on the diagonals and f is given by the right hand side of (1) and $c^{\mathrm{T}} = (0, 0, 1, 0)$.

Since this problem has a free terminal time we transform the time interval $t \in [0, T]$ into a normalised time interval $\tau \in [0, 1]$ by introducing the new independent variable

$$\tau = \frac{t}{T}, \tag{4}$$

rewrite the equations in (3) for the new variable τ and add the trivial equation $\dot{T} = 0$. This results in a problem of the form (3) but with a fixed time interval.

4.2 Necessary conditions of optimality

Introducing the Hamiltonian,

$$H = \lambda^{\mathrm{T}} f(z,\beta),$$

and then applying variational calculus [6] to (3) leads to the following necessary conditions of optimality.

The optimal solution $(z^*(t), \lambda^*(t), \beta^*(t))$ fulfills the optimality conditions

$$\dot{z} = \frac{\partial H}{\partial \lambda} = f(z), \tag{5}$$

$$\dot{\lambda} = -\frac{\partial H}{\partial z} = -\left(\frac{\partial f}{\partial z}\right)^{\mathrm{T}} \lambda, \tag{6}$$

$$0 = \frac{\partial H}{\partial \beta} = \left(\frac{\partial f}{\partial \beta}\right)^{\mathrm{T}} \lambda, \tag{7}$$

the boundary conditions

$$J_0 z(0) = z_0, \quad J_T z(T) = z_T, \tag{8}$$

and the transversality conditions

$$(J - J_0)\lambda(0) = \lambda_0, \quad (J - J_T)\lambda(T) = \lambda_T, \tag{9}$$

where λ_0 and λ_T are obtained from $\mathcal{J}$. We note here that $x_0 \in R(J_0)$ and $x_T \in R(J_T)$ which means that the components of the adjoint variable λ that have boundary values are the ones complementary to the components of x that have boundary values. To simplify the problem we assume that the optimality condition (7) can be solved explicitly for β^*.

4.3 Reformulating the boundary value problem into standard form

General purpose software for treating boundary value problems for ordinary differential equations usually requires the problem to be reformulated into standard form [3]. We make this conversion by joining the states z and the costates λ into a new variable $x \in \mathbb{R}^d$ for $d = 2n$, and then redefining f by merging the right hand sides of (5) and (6). The resulting system is a two point boundary value problem with fixed time interval and separated linear boundary conditions,

$$\begin{aligned} &\dot{x} = f(x), \\ &I_0 x(0) = x_0, \quad I_T x(1) = x_T, \end{aligned} \tag{10}$$

where $\dot{x}$ denotes the derivative of x with respect to the new independent variable τ.

5 An adaptive finite element method

5.1 Weak formulation

In this section we derive an adaptive finite element method. It consists of the discretisation of the problem with definitions of the right function spaces and an a posteriori error estimate. We start with the so called weak formulation. To obtain the weak formulation we multiply (10) by a test function $v \in V = C^1([0,T])$, integrate over the interval $[0,T]$ and the weak formulation of the problem is: Seek $x \in V$ such that

$$\begin{aligned} &I_0 x(0) = x_0, \quad I_T x(T) = x_T, \\ &F(x, v) = \int_0^T (\dot{x} - f(x), v)\, dt = 0, \quad \forall v \in V, \end{aligned} \tag{11}$$

where $(\cdot,\cdot)$ is the Cartesian scalar product in $\mathbb{R}^d$.

5.2 Discretisation of the problem

The problem in (11) is an infinite dimensional problem which we discretise as follows to get a finite problem. We discretise the time axis and introduce the trial and test spaces as follows.

- Mesh: $0 = t_0 < t_1 < t_2 < \ldots < t_N = T$, $h_n = t_n - t_{n-1}$ and $I_n = (t_{n-1}, t_n)$.
- Trial space: $W_h = \mathbb{R}^d \times \{w : w|_{I_n} \in P^0(I_n)\} \times \mathbb{R}^d$, discontinuous piecewise constant functions.
- Test space: $V_h = \{v : v|_{I_n} \in P^1(I_n) \cap C^0([0,T])\}$, continuous piecewise linear functions.

The notation $P^k(I_n)$ refers to the $\mathbb{R}^d$-valued polynomials of degree k on the interval I_n. We also introduce the left and right limits $w_n^{\pm} = \lim_{t \to t_n^{\pm}} w(t)$, and jumps $[w]_n = w_n^+ - w_n^-$. The two factors $\mathbb{R}^d$ in W_h contain the boundary values w_0^- and w_N^+. Now our finite element problem can be stated: Find a function $X \in W_h$ which fulfills

$$\begin{aligned} &I_0 X_0^- = x_0, \quad I_T X_N^+ = x_T, \\ &F(X, v) = \sum_{n=1}^{N} \int_{I_n} (\dot{X} - f(X), v)\, dt \\ &\qquad + \sum_{n=0}^{N} ([X]_n, v_n) = 0, \quad \forall v \in V_h. \end{aligned} \tag{12}$$

Here the definition of the form F from (11) has been extended to include the contributions from the jump terms which appear since we use discontinuous trial functions. Since the trial space consists of piecewise constant functions, $\dot{X} = 0$. Hence, (12) results in a system of $(N+2)d$ equations that have to be solved, more precisely, d boundary conditions and $(N+1)d$ equations. With boundary conditions at both ends, the equations are coupled and thus we cannot use time stepping and therefore the equations in the system have to be solved simultaneously.

5.3 An a posteriori error estimate

An adaptive finite element method gives us the possibility to control the error in the numerical solution.

In order to derive an a posteriori error estimate we introduce ϕ as the solution to the adjoint problem to (10) with data functional G. We want to construct an equation for the error, $e = X - x$ where $e \in W = \mathbb{R}^d \times \{w|_{I_n} : w \in C^1(I_n)\} \times \mathbb{R}^d$, the difference between the real and the computed solution. The details of the a posteriori error estimate are given below.

5.3.1 Proof of the error estimate

We subtract (11) from (12),

$$\begin{aligned} F(X,v) &- \underbrace{F(x,v)}_{=0,\forall v\in V} \\ &= \sum_{n=1}^{N} \int_{I_n} (\dot{X} - \dot{x} - (f(X) - f(x)), v)\, dt \\ &\quad + \sum_{n=0}^{N} ([X-x]_n, v_n). \end{aligned} \tag{13}$$

Since f is nonlinear we linearise $f(X)-f(x)$ by rewriting it as follows

$$\begin{aligned} f(X) &- f(x) \\ &= \int_0^1 \frac{d}{d\theta} f(\theta(X(t) - x(t)) + x(t))\, d\theta \\ &= \underbrace{\int_0^1 Df(\theta(X(t) - x(t)) + x(t))\, d\theta}_{=A(t)} (X(t) - x(t)). \end{aligned}$$

Inserting this in (13) we get

$$\begin{aligned} F(X,v) & \\ &= \sum_{n=1}^{N} \int_{I_n} (\dot{X} - \dot{x} - (f(X) - f(x)), v)\, dt \\ &\quad + \sum_{n=0}^{N} ([X-x]_n, v_n) \\ &= \sum_{n=1}^{N} \int_{I_n} (\dot{e} - A(t)e, v)\, dt \\ &\quad + \sum_{n=0}^{N} ([e]_n, v_n), \quad \forall v \in V. \end{aligned} \tag{14}$$

Since (14) is linear in both e and v we introduce a bilinear form to simplify the notation. The bilinear form B is defined as

$$\begin{aligned} B(w,v) &= \sum_{n=1}^{N} \int_{I_n} (\dot{w} - A(t)w, v)\, dt + \sum_{n=0}^{N} ([w]_n, v_n) \\ &+ (I_0 w_0^-, v_0) - (I_T w_N^+, v_N), \quad w \in W, v \in V, \end{aligned} \tag{15}$$

Now we can write the equation for the error (14) with the bilinear form as

$$\begin{aligned} & e \in W \\ & B(e,v) = F(X,v), \quad \forall v \in V. \end{aligned} \tag{16}$$

Partial integration of (15) gives us the backward form of the bilinear form

$$\begin{aligned} B(w,v) &= \sum_{n=1}^{N} \int_{I_n} (\dot{w} - A(t)w, v)\, dt \\ &\quad + \sum_{n=0}^{N} ([w]_n, v_n) \\ &\quad + (I_0 w_0^-, v_0) - (I_T w_N^+, v_N) \\ &= \sum_{n=1}^{N} \int_{I_n} (w, -\dot{v} - A(t)^{\mathrm{T}} v)\, dt \\ &\quad - (w_0^-, (I - I_0)v_0) + (w_N^+, (I - I_T)v_N), \\ &\quad w \in W, v \in V. \end{aligned} \tag{17}$$

This suggests the dual problem with arbitrary data functional G

$$\begin{aligned} & \phi \in V \\ & B(w,\phi) = G(w), \quad \forall w \in W. \end{aligned} \tag{18}$$

We put $v = \phi$ in (16) and $w = e$ in (18) to obtain

$$G(e) = B(e,\phi) = F(X,\phi), \tag{19}$$

that is

$$\begin{aligned} G(e) &= B(e,\phi) \\ &= F(X,\phi) = \sum_{n=1}^{N} \int_{I_n} (\dot{X} - f(X), \phi)\, dt \\ &\quad + \sum_{n=0}^{N} ([X]_n, \phi_n). \end{aligned} \tag{20}$$

Subtracting a Lagrange node interpolant $\tilde{\phi} \in V_h$ from ϕ in the right hand side of (20) using (12) gives us

$$G(e) = \sum_{n=1}^{N} \int_{I_n} (\dot{X} - f(X), \phi - \tilde{\phi})\, dt + \sum_{n=0}^{N} ([X]_n, \phi_n - \tilde{\phi}_n).$$

Hence,

$$\begin{aligned}
|G(e)| &\le \Big| \sum_{n=1}^{N} \int_{I_n} (\dot{X} - f(X), \phi - \tilde{\phi})\, dt \Big| \\
&\quad + \Big| \sum_{n=0}^{N} ([X]_n, \phi_n - \tilde{\phi}_n) \Big| \\
&\le \underbrace{\sum_{n=1}^{N} \int_{I_n} \|\dot{X} - f(X)\| \|\phi - \tilde{\phi}\|\, dt}_{I} \\
&\quad + \underbrace{\sum_{n=0}^{N} \| [X]_n \| \|\phi_n - \tilde{\phi}_n\|}_{II}.
\end{aligned} \tag{21}$$

Now we have the basis for an error estimate, but we want the method to be symmetric, meaning that we want each interior node to contribute to the error estimate on both sides of the node. To do this we rewrite the last term II in (21) as follows

$$\begin{aligned}
\sum_{n=0}^{N} \| [X]_n \| \|\phi_n - \tilde{\phi}_n\| &= \| [X]_0 \| \|\phi_0 - \tilde{\phi}_0\| \\
&\quad + \frac{h_1}{h_1 + h_2} \| [X]_1 \| \|\phi_1 - \tilde{\phi}_1\| \\
&\quad + \sum_{n=2}^{N-1} \Big(\frac{h_n}{h_n + h_{n+1}} \| [X]_n \| \|\phi_n - \tilde{\phi}_n\| \\
&\qquad + \frac{h_n}{h_n + h_{n-1}} \| [X]_{n-1} \| \|\phi_{n-1} - \tilde{\phi}_{n-1}\| \Big) \\
&\quad + \frac{h_N}{h_{N-1} + h_N} \| [X]_{N-1} \| \|\phi_{N-1} - \tilde{\phi}_{N-1}\| \\
&\quad + \| [X]_N \| \|\phi_N - \tilde{\phi}_N\|.
\end{aligned} \tag{22}$$

At this stage we introduce the notation $\|v\|_{I_n} = \max_{I_n} \|v\|$ for the maximum norm of a function on an interval. Now we note that $\phi - \tilde{\phi} \in V$ is continuous and the following estimates $\|\phi_n - \tilde{\phi}_n\| \le \|\phi - \tilde{\phi}\|_{I_n}$ and $\|\phi_n - \tilde{\phi}_n\| \le \|\phi - \tilde{\phi}\|_{I_{n+1}}$ hold. Using this we can estimate the last term in (22) by

$$\begin{aligned}
&\sum_{n=0}^{N} \| [X]_n \| \|\phi_n - \tilde{\phi}_n\| \\
&\le \Big(\| [X]_0 \| + \frac{h_1}{h_1 + h_2} \| [X]_1 \| \Big) \|\phi - \tilde{\phi}\|_{I_1} \\
&\quad + \sum_{n=2}^{N-1} \Big(\frac{h_n}{h_n + h_{n+1}} \| [X]_n \| \\
&\qquad + \frac{h_n}{h_n + h_{n-1}} \| [X]_{n-1} \| \Big) \|\phi - \tilde{\phi}\|_{I_n} \\
&\quad + \Big(\frac{h_N}{h_{N-1} + h_N} \| [X]_{N-1} \| + \| [X]_N \| \Big) \|\phi - \tilde{\phi}\|_{I_N}.
\end{aligned} \tag{23}$$

The term I in (21) (where $\dot{X} = 0$) can be estimated as follows

$$\begin{aligned}
&\sum_{n=1}^{N} \int_{I_n} \|\dot{X} - f(X)\| \|\phi - \tilde{\phi}\|\, dt \\
&\quad \le \sum_{n=1}^{N} h_n \|\dot{X} - f(X_n^-)\|_{I_n} \|\phi - \tilde{\phi}\|_{I_n}.
\end{aligned} \tag{24}$$

Collecting the estimates (24) and (23) we now have

$$\begin{aligned}
|G(e)| &\le \sum_{n=1}^{N} h_n \|\dot{X} - f(X)\|_{I_n} \|\phi - \tilde{\phi}\|_{I_n} \\
&\quad + \Big(\| [X]_0 \| + \frac{h_1}{h_1 + h_2} \| [X]_1 \| \Big) \|\phi - \tilde{\phi}\|_{I_1} \\
&\quad + \sum_{n=2}^{N-1} \Big(\frac{h_n}{h_n + h_{n+1}} \| [X]_n \| \\
&\qquad + \frac{h_n}{h_n + h_{n-1}} \| [X]_{n-1} \| \Big) \|\phi - \tilde{\phi}\|_{I_n} \\
&\quad + \Big(\frac{h_N}{h_{N-1} + h_N} \| [X]_{N-1} \| \\
&\qquad + \| [X]_N \| \Big) \|\phi - \tilde{\phi}\|_{I_N}.
\end{aligned}$$

According to [5] we have the following error bound for the interpolant, $\tilde{\phi}$,

$$\left\|\phi - \tilde{\phi}\right\|_{I_n} \le C h_n \int_{I_n} \left\|\ddot{\phi}\right\| dt.$$

With

$$\begin{aligned}
\mathcal{R}_1 &= h_1 \|\dot{X} - f(X)\|_{I_1} \\
&\quad + h_1 \| [X]_{n-1} \| + \frac{h_1}{h_1 + h_2} \| [X]_1 \|, \\
\mathcal{R}_n &= h_n \|\dot{X} - f(X)\|_{I_n} \\
&\quad + \frac{h_n}{h_n + h_{n+1}} \| [X]_n \| \\
&\quad + \frac{h_n}{h_n + h_{n-1}} \| [X]_{n-1} \|, \quad n = 2, \dots, N-1, \\
\mathcal{R}_N &= h_N \|\dot{X} - f(X)\|_{I_N} + \\
&\quad \frac{h_N}{h_{N-1} + h_N} \| [X]_{N-1} \| + \| [X]_n \|,
\end{aligned}$$

and

$$\mathcal{I}_n = C h_n \int_{I_n} \left\|\ddot{\phi}\right\| dt,$$

we can write the error estimate as

$$|G(e)| \le \sum_{n=1}^{N} \mathcal{R}_n \mathcal{I}_n, \tag{25}$$

where $e = X - x$ is the error and $\mathcal{R}_n$ is essentially the residual, $\dot{X} - f(X)$, expressing how well the differential equation is satisfied by the numerical solution. The weights $\mathcal{I}_n$ depend on the solution to the adjoint problem, ϕ, and express the sensitivity of the error quantity $G(e)$ to the local residuals. The functional G is chosen to be the quantity in which we want to measure the error, for example, $G(e) = \frac{e}{\|e\|}$. The error resulting from approximate nonlinear equation solver is small compared to the error resulting from the discretisation and is therefore neglected in this estimate. Checking which intervals give large contributions to the error estimate (25), we can refine the intervals where the contributions are large and vice versa. Using (25) we obtain an adaptive procedure where we refine those intervals that give large contributions to the estimate and vice versa, see below.

5.4 Implementation

The finite element discretisation of (10) results in the system (12) to be solved. Since we have boundary conditions at both ends it is a coupled system of equations that we have to solve simultaneously. The system is also nonlinear and the nonlinearity is handled using a damped Newton method [7] to extend the convergence region. The initial guess is decided by a homotopy process [3]. Once a solution is calculated the error estimate above is computed. Then we apply the criterion

$$\sum_{n=1} \mathcal{R}_n \mathcal{I}_n \leq \delta,$$

where δ is a given tolerance. If the error is too large compared to the tolerance an iteration is made over the intervals and the mesh is refined where the error is large. New nodes are inserted according to the *principle of equidistribution*, that is, we want to insert nodes such that the contribution to the error is the same from each time interval. A new solution is calculated on the refined mesh and so on until the solution has reached the desired accuracy. In theory the mesh can also be coarsened but we have not implemented this.

The error estimate is dependent of the solution to the dual problem. We need to approximate the unknown data $G(e) = \frac{e}{\|e\|}$ to solve the dual problem numerically. We do this by a Richardson extrapolation using twice the number of nodes. Then the dual problem is solved with the finite element method.

The solver is a prototype solver and it has been implemented in Matlab. More information regarding the implementation can be found in [2].

6 Results

The indirect approach to our optimal control problem results in a boundary value problem. This makes it possible to compare our new approach to the boundary value solver `bvp4c` in Matlab [11]. In Table 1 we can see the results from various choices of the manoeuvre distances a and b. We have used the same initial velocity $u_0 = 90$ km/h and constant friction μ for all cases. We can see that the FEM code is almost always about three times slower than `bvp4c` but it always uses fewer nodes. There is also a remarkable case where `bvp4c` solves the problem in about 30 seconds and with 3415 nodes compared to 1.3 seconds and 21 nodes for the finite element solver. The problem becomes difficult to solve but our FEM solver performs well, maybe due to the adaptivity. There is also one problem that the finite element solver can solve but `bvp4c` cannot.

In some cases where `bvp4c` finds a solution the FEM solver seems to compute the wrong one, maybe by missing a singularity. On the other side of the singularity it continues on another solution. Some results about existence and uniqueness of solutions to boundary value problems can be found in [7] and [9]. This aspect of our solver is something that we have to investigate further.

Figure 2 and 3 show some results from the case with initial velocity $u_0 = 90$ km/h (25 m/s) and the manoeuvre distances $a = 50$ m and $b = 8$ m. We see in the figures that the solutions from the FEM solver and `bvp4c` coincide. The final velocity is 53.32 km/h from `bvp4c` and 53.37 km/h from the FEM solver.

a [m]	b [m]	FEM CPU [s]	FEM nodes	`bvp4c` CPU [s]	`bvp4c` nodes
40	6	2.54	15	0.27	25
50	9	1.59	90	0.36	41
50	8	0.61	10	0.24	28
50	5	0.52	10	0.27	37
50	3	1.45	10	-	-
60	8	1.45	21	18.39	1287
60	6	1.46	10	0.63	81
60	5	3.45	10	3.06	286

Table 1: Performance of the FEM solver and `bvp4c` measured in CPU time and number of nodes for different combinations of manoeuvre distances.

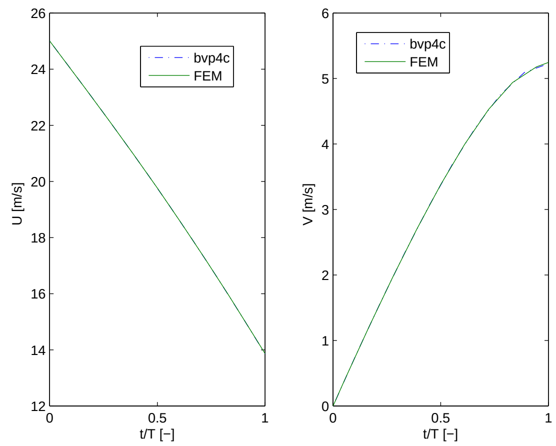

Figure 2: The optimal velocities in the X and Y-directions for the manoeuvre distances $a = 50$ m and $b = 8$ m.

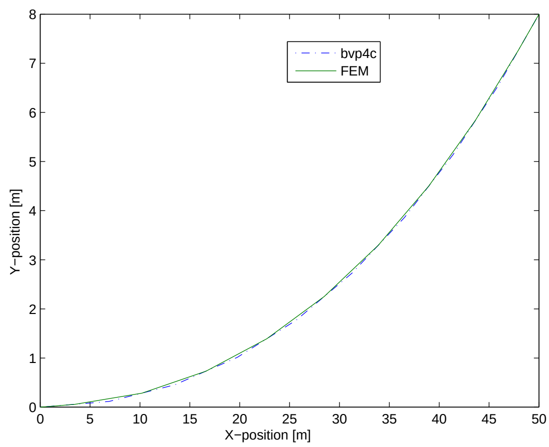

Figure 3: The position of the vehicle when it is manoeuvered in the optimal way for the manoeuvre distances $a = 50$ m and $b = 8$ m.

7 Conclusion

In this article we have presented an adaptive finite element method for solving optimal control in vehicle dynamics. Modelling the vehicle as a point mass, we obtain a system of ordinary differential equations which is solved, together with the constraints imposed by the manoeuvre, using the adaptive finite element method. With this approach, we can control the error and concentrate our resources to the most sensitive parts of the computations.

We have compared the finite element method to the Matlab solver `bvp4c` and found that there are at least some cases where our solver outperforms `bvp4c`. It is noteworthy that in all studied cases, our method uses fewer nodes to find the same solution. At the moment the finite element solver is slower than `bvp4c`, but up to this point no extra effort has been put in optimising the code. Thus, it is expected that the computation time can be reduced by a more efficient implementation. Further, these comparisions are very preliminary, since the accuracies of both methods depend on error tolerances that are not directly comparable. We are not sure that the settings are equal. Still, since the quality of the solutions have been similar throughout our computations, we feel confident that our comparision is reasonable.

We have also noted that our solver is sensitive to the initial guess. If we give the solver a poor initial guess for some component, it may fail to solve the problem or show a dramatical increase in computation time. To make the solver more useful we have to make it more robust to poor initial guesses.

The model used in this article may look simple. However, when it comes to evaluation of combined steering and braking versus braking and then steering, the behaviour of this model gives insights into the behaviour of the more realistic vehicle models and manoeuvres we will consider. In our future work we also intend to compare the performance of our indirect approach to the direct approach.

References

[1] National Highway Traffic Safety Administration, *Laboratory Test Instruction for FMVSS 126 Stability Control Systems*, `http://www.nhtsa.dot.gov`.

[2] C. Andersson, *Solving optimal control problems using FEM*, Master's thesis, Chalmers University of Technology, 2007.

[3] U. M. Ascher, R. M. M. Mattheij, and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, first ed., Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1988.

[4] J. T. Betts, *Survey of numerical methods for trajectory optimization*, AIAA J. Guidance Control Dynam. **21** (1998), 193–207.

[5] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, second ed., Springer-Verlag, New York, 2002.

[6] A. E. Bryson, Jr and Y. Ho, *Applied Optimal Control*, revised printing ed., Hemisphere Publishing Corporation, Washington, D.C, 1975.

[7] P. Deuflhard and F. Bornemann, *Scientific Computing with Ordinary Differential Equations*, Springer-Verlag New York Inc., New York, 2002.

[8] D. J. Estep, D. H. Hodges, and M. Warner, *Computational error estimation and adaptive error control for a finite element solution of launch vehicle trajectory problems*, SIAM J. Sci. Comput. **21** (1999), 1609–1631.

[9] H. B. Keller, *Numerical Methods for Two-Point Boundary-Value Problems*, Dover Publications, Inc., New York, 1992.

[10] B. Schmidtbauer, *Sväng och bromsa samtidigt (in Swedish)*, Teknisk Tidskrift (1971).

[11] L. F. Shampine, J. Kierzenka, and M. W. Reichelt, *Solving Boundary Value Problems for Ordinary Differential Equations in Matlab using* `bvp4c`, Tech. report, MathWorks, 2000, `http://www.mathworks.com`.