Keywords: vehicle dynamics, state and parameter estimation, extended Kalman filter, genetic algorithm

Abstract

This paper describes the development of a hybrid approach to estimate the states and parameters of a vehicle. The parameter estimator is based on a genetic algorithm in conjunction with a bank of extended Kalman filters, which are simultaneously utilised to estimate the states of the system.

This paper shows results of the approach when applied to a four-wheel-vehicle model and the advantages and limitations are discussed.

1 Introduction

1.1 Background

The last two decades has seen a significant increase in the use of electronic control systems to enhance the safety of automotive vehicles. Such systems include antilock braking system and electronic stability programs, as described by Bauer [1]. Common to these systems is a requirement for accurate knowledge of the vehicle states, e.g. slip rate of the wheels or body slip angle and yaw rate of the vehicle.

Traditionally, component suppliers have produced systems/subsystems having their own built-in state estimators and these systems operate as independent modules within the overall system. Additionally, as the introduction of new subsystems modules increases, there is increased concern over the interoperability of these modules and a subsequent need to reconsider the structure/architecture has been recognised.

In this work, an approach to develop a new total vehicle model-based state estimator is described. This state/parameter estimator is to be placed hierarchically above the vehicle safety control systems so that this single estimator supplies all electronic control systems simultaneously, as shown in Figure 1.

1.2 Vehicle model

The implemented model is a four-wheel-vehicle model with four degrees of freedom: the longitudinal direction $x$, the lateral direction $y$, the yaw $\psi$ around the vertical axis $z$ and the roll $\phi$ around the longitudinal axis $x$. Table 1 lists the main vehicle states that are to be regarded.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_x$</td>
<td>longitudinal velocity</td>
<td>[m/s]</td>
</tr>
<tr>
<td>$v_y$</td>
<td>lateral velocity</td>
<td>[m/s]</td>
</tr>
<tr>
<td>$a_x$</td>
<td>longitudinal acceleration</td>
<td>[m/s$^2$]</td>
</tr>
<tr>
<td>$a_y$</td>
<td>lateral acceleration</td>
<td>[m/s$^2$]</td>
</tr>
<tr>
<td>$\Gamma$</td>
<td>torque around $z$-axis</td>
<td>[Nm]</td>
</tr>
<tr>
<td>$\delta$</td>
<td>steer angle</td>
<td>[rad]</td>
</tr>
<tr>
<td>$\beta$</td>
<td>body side slip angle</td>
<td>[rad]</td>
</tr>
<tr>
<td>$\alpha_{ij}$</td>
<td>wheel slip angles</td>
<td>[rad]</td>
</tr>
<tr>
<td>$\dot{\psi}$</td>
<td>yaw rate</td>
<td>[rad/s]</td>
</tr>
<tr>
<td>$\phi$</td>
<td>body roll angle</td>
<td>[rad]</td>
</tr>
<tr>
<td>$s_{ij}$</td>
<td>slip rate of each wheel</td>
<td>[1]</td>
</tr>
<tr>
<td>$u_{ij}$</td>
<td>forward wheel speed</td>
<td>[m/s]</td>
</tr>
<tr>
<td>$\omega_{ij}$</td>
<td>wheel rotational velocity</td>
<td>[rad/s]</td>
</tr>
<tr>
<td>$F_{x_{ij}}$</td>
<td>longitudinal forces on each wheel</td>
<td>[N]</td>
</tr>
<tr>
<td>$F_{y_{ij}}$</td>
<td>lateral forces on each wheel</td>
<td>[N]</td>
</tr>
<tr>
<td>$F_{z_{ij}}$</td>
<td>vertical forces on each wheel</td>
<td>[N]</td>
</tr>
</tbody>
</table>

Figure 1: Vehicle control systems

Table 1: Main vehicle states
Furthermore, there are some parameters, which influence the behaviour of the vehicle. The most important are listed in Table 2.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$</td>
<td>mass of vehicle</td>
<td>[kg]</td>
</tr>
<tr>
<td>$m_s$</td>
<td>sprung mass</td>
<td>[kg]</td>
</tr>
<tr>
<td>$I_z$</td>
<td>moment of inertia around vert. axis</td>
<td>[kgm$^2$]</td>
</tr>
<tr>
<td>$t_f$, $t_r$</td>
<td>front and rear track width</td>
<td>[m]</td>
</tr>
<tr>
<td>$b$, $c$</td>
<td>distances between centre of gravity and front and rear axles</td>
<td>[m]</td>
</tr>
<tr>
<td>$h$</td>
<td>height of centre of gravity</td>
<td>[m]</td>
</tr>
<tr>
<td>$h_s$</td>
<td>sprung mass’s height of centre of gravity</td>
<td>[m]</td>
</tr>
<tr>
<td>$\ell$</td>
<td>length of wheel base, where $\ell = b + c$</td>
<td>[m]</td>
</tr>
<tr>
<td>$r$</td>
<td>tyre radius</td>
<td>[m]</td>
</tr>
<tr>
<td>$g$</td>
<td>gravitational acceleration</td>
<td>[m/s$^2$]</td>
</tr>
<tr>
<td>$\beta_\phi$</td>
<td>roll damping constant</td>
<td>[Nms]</td>
</tr>
<tr>
<td>$\kappa_\phi$</td>
<td>roll stiffness constant</td>
<td>[Nm]</td>
</tr>
</tbody>
</table>

Table 2: Main vehicle parameters

The vehicle model used here is based on the formulation of differential equations for the calculation of the vehicle states. E.g. Wong [12] and Will and Zak [11] give differential equations for the calculation of acceleration, yaw and roll:

\[
\begin{align*}
\dot{v}_y &= \frac{1}{m} \left( F_{y,fl} \cos \delta + F_{y,fr} \cos \delta + F_{y,fl} \sin \delta + F_{y,fr} \sin \delta + F_{y,rl} + F_{y,rr} \right) - v_x \dot{\psi} \\
\dot{\psi} &= \frac{\Gamma}{J_z} \\
\dot{\phi} &= \frac{1}{J_x} \left( -m_s h_s a_y + \phi(m_s h_s - \kappa_\phi) - \dot{\phi} \phi_0 \right) \\
\end{align*}
\]

where

\[
\begin{align*}
\Gamma &= \frac{t_f}{2} F_{z,fl} - \frac{t_r}{2} F_{z,fr} + \frac{t_r}{2} F_{z,rl} - \frac{t_r}{2} F_{z,rr} + b F_{y,fl} + b F_{y,fr} - c F_{y,rl} - c F_{y,rr} + M_{z,fl} + M_{z,fr} + M_{z,rl} + M_{z,rr} \\
ax &= \dot{v}_x - v_y \dot{\psi} \\
ay &= \dot{v}_y + v_x \dot{\psi} \\
\end{align*}
\]

Using the knowledge of these states, further states, such as the slip angles of each of the tyres and the vehicle body, can be calculated:

\[
\begin{align*}
\alpha_{f,l/r} &= \delta - \arctan \left( \frac{v_y + b \dot{\psi}}{v_x \pm \frac{1}{2} t_f \dot{\psi}} \right) \\
\alpha_{r,l/r} &= \arctan \left( \frac{-v_y + c \dot{\psi}}{v_x \pm \frac{1}{2} t_r \dot{\psi}} \right) \\
\beta &= \arctan \left( \frac{v_y}{v_x} \right) \\
\end{align*}
\]

Furthermore, it is to be noted that the lateral acceleration at the centre of gravity is not measured; rather it is the lateral acceleration acting on the accelerometer, which is measured. A simplified relationship for this is given by Chee [2]:

\[
a_y,\text{sensor} = (a_y + \dot{\psi} x_a) \cos \phi + g \sin \phi
\]

where $x_a$ is the longitudinal distance between the accelerometer and the centre of gravity.

### 1.3 Tyre model

The tyre-road interface is a major feature, which has to be regarded. Since, ignoring aerodynamic effects, the vehicle motion results from forces generated in the four, ‘palm-sized’ contact patches between tyre and road, the tyre model is of high importance in the calculation of vehicle dynamics.

There are different approaches to calculate the forces acting on the tyres. The work described in this paper makes use of the so-called ‘TMeasy’ tyre model, developed by Hirschberg et al. [4].

Tyre models vary in terms of their complexities, i.e. numbers of tyre model specific coefficients. Such coefficients are usually obtained via experimental trials using test rigs. The important factors of tyre slip and vertical forces are also required to be known. These particular states may be obtained, see Milliken and Milliken [6], as follows:

\[
\begin{align*}
F_{z,fl/r} &= \frac{1}{2} \left( m g \pm m \frac{a_y h}{t_f} c \right) \ell - m a_z \ell \\
F_{z,rl/r} &= \frac{1}{2} \left( m g \pm m \frac{a_y h}{t_r} b \right) + m a_x \ell \\
\end{align*}
\]

### 2 Extended Kalman filter

The Kalman filter algorithm was first developed by R. E. Kalman [5] in 1960 for estimating the non-measurable states of linear systems. The Kalman filter algorithm was later extended for estimating the states of non-linear systems. In this extension the state equations are linearised at each working point.

The Kalman filter operates by successive prediction followed by correction, as is illustrated in Figure 2:

![Figure 2: Scheme of Kalman filter](image-url)
The main steps of the extended Kalman filter (EKF) can then be formulated as e.g. described by Haykin [3]. It assumes a formulation of the non-linear system with the input states \( u \), the output states \( y \) and the non-measurable, internal states \( x \). In the following formulation, \( w \) and \( v \) are the process and output noise vectors, respectively.

\[
\begin{align*}
\mathbf{x}(t+1) &= f(\mathbf{x}(t), u(t), w(t)) \\
\mathbf{y}(t) &= h(\mathbf{x}(t), v(t))
\end{align*}
\] (14)

Using this non-linear system, the procedure of the Kalman filter can be described by five main steps:

1. Prediction of state estimate
   \[
   \mathbf{x}^\ast(t) = f(\hat{\mathbf{x}}(t-1), u(t))
   \] (16)

2. Prediction of error covariance
   \[
   \mathbf{P}^\ast(t) = J(t)\mathbf{P}(t-1)J^T(t) + \mathbf{Q}
   \] (17)

3. Kalman gain matrix
   \[
   \mathbf{K}(t) = \mathbf{P}^\ast(t)H^T[R + PH\mathbf{P}^\ast(t)H^T]^{-1}
   \] (18)

4. Correction of state estimate
   \[
   \hat{\mathbf{x}}(t) = \mathbf{x}^\ast(t) + \mathbf{K}(t)[\mathbf{y}(t) - H\hat{\mathbf{x}}^\ast(t)]
   \] (19)

5. Correction of error covariance
   \[
   \mathbf{P}(t) = [I - \mathbf{K}(t)H]\mathbf{P}^\ast(t)
   \] (20)

Here \( \mathbf{Q} \) and \( \mathbf{R} \) are the user-specified process noise and output noise covariance matrices, while \( J \) and \( H \) are the Jacobian matrices of the systems functions:

\[
\mathbf{J} = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_m}
\end{bmatrix}
\] (21)

\[
\mathbf{H} = \begin{bmatrix}
0 & 0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & \cdots & 1
\end{bmatrix}
\] (22)

The input vector \( u \) and the output vector \( y \) consist of the only measurable vehicle states:

\[
\mathbf{u} = \begin{bmatrix}
\delta \\
v_{cog}
\end{bmatrix}
\] (23)

\[
\mathbf{y} = \begin{bmatrix}
\dot{v} \\
a_y
\end{bmatrix}
\] (24)

The state vector comprises the internal states, required by the dynamic safety control systems.

\[
\mathbf{x} = [v_x \ v_y \ \dot{v} \ a_y \ \Gamma \ \beta \ a_x \ \cdots \\
a_{ij} \ F_{z,ij} \ s_{ij} \ \phi \ \dot{\phi} \ a_y, sensor]^T
\] (25)

Based on the estimation of wheel slip angles, slip rates and vertical forces, the TMeasy tyre model is then utilized to calculate the longitudinal and lateral tyre forces.

\section{Genetic Algorithms}

\subsection{General}

Essentially a genetic algorithm is a guided random search technique, which follows a scheme of random selection, evaluation and evolution, and a good summary may be found in Reeves [7]. In this work a genetic algorithm is applied to the process of parameter estimation.

Instead of evaluating a single parameter set, a whole group, known as a ‘population’, of parameter sets is evaluated. A binary string, the so-called ‘chromosome’, represents each parameter set.

The following scheme shows an example for a chromosome, which comprises genes for the parameters mass \( m \), moment of inertia \( J_z \) and position of centre of gravity \( b \).

\[
0110010100111010100101011001101101
\]

Each parameter set is assigned with a ‘fitness’, which describes the ‘quality’ of the parameter set.

Once all parameter sets are evaluated, the process of evolution or ‘breeding’ takes places, i.e. a new population of chromosomes is generated. There are numerous approaches, however typically two chromosomes are chosen by a weighted random and using crossover technique, a new chromosome is created.

As the following scheme demonstrates, a new chromosome is produced by combining a part of each existing chromosome’s string.

\[
\begin{array}{c}
\text{Current chromosome 1} \\
110010100111010101011001101
\end{array}
\]

\[
\begin{array}{c}
\text{Crossover point} \\
\times
\end{array}
\]

\[
\begin{array}{c}
\text{Current chromosome 2} \\
00101011000
\end{array}
\]

\[
\begin{array}{c}
\text{Next chromosome} \\
11001101100
\end{array}
\]

The mutation operation, which is effectively realised by random mutation of one or more binary bits in one or more of the chromosomes in a given population, attempts to prevent the algorithm from becoming trapped in local optimums.

The procedure of a standard genetic algorithm can be summarised as follows:

1. Initialise a first population
2. Evaluate fitnesses of all chromosomes
3. Generate new population using selection of fittest
4. Apply mutation to new population
5. If convergence criteria is not fulfilled, continue in step 2
3.2 Combination with extended Kalman filter

There have been already approaches to combine a genetic algorithm with other estimators, e.g. Warwick and Kang [10] used a genetic algorithm in combination with a recursive least squares algorithm.

The hybrid approach combining genetic algorithms with the extended Kalman filter is implemented by executing the Kalman filter operations several times in parallel at each time step.

The parameters to be estimated by the genetic algorithms are the mass \(m\), moment of inertia \(J\), and the position of the centre of gravity \(b\).

For each parameter set the actual prediction error \(\Delta y\) from equation 19 is calculated:

\[
\Delta y = [\dot{y}(t) - H\dot{x}^- (t)]
\]

The chromosome with the best genes has a mean prediction error. Thus as basis for fitness \(F_i\) of the \(i\)th chromosome the difference between its prediction error \(\Delta y_i\) and the mean prediction error \(\overline{\Delta y}\) is selected:

\[
F_i = \frac{1}{(\Delta y_i - \overline{\Delta y})}
\]

Depending on the fitness values a new generation of parameter sets is then generated for use at the next time step.

Since all evaluated chromosomes differ slightly, it is difficult to choose a best one. Therefore the ranges, in which the genes can vary, are discretized into several subdivisions or bins. Each time the fitness of a chromosome is evaluated, then for each parameter the corresponding bin in the histograms, see Figure 5, is incremented. Following such a method, the ‘best’ estimates of the model parameters are obtained.

3.3 Advantages and limitations

The advantage of using genetic algorithms is that a large range of parameters can easily be covered, without an exact a-priori knowledge of the actual parameters.

The limitation is the high computational power, which is needed for the parallel execution of several Kalman filters. This is not a problem, while doing the calculation offline and no real-time processing is required. But for an intended online estimation, the calculation has to be done in less than real-time, thus requiring several parallel processors, which would increase the cost of implementation.

4 Results

The state and parameter estimator was implemented using the industry standard MATLAB [9] environment. For validation, data of a rear wheel drive executive saloon car was available.

Data was on hand from both simulation by other software packages as well as actual measured data from a test drive.

For the examples shown below, data was used, which was generated by the simulation software-package ve-DYNA [8]. The advantage of using simulated data rather than actual measured data is the availability of non-measurable states as reference information.

The performed driving manoeuvre was double cornering on a racetrack with rather large lateral acceleration (\(4\text{m/s}^2\)). Figure 3 shows the steer angle and vehicle velocity inputs, as well as the reference outputs yaw rate and lateral acceleration.

For the genetic algorithms a chromosome length of 31 bits and a probability of 99.9% that at each position not only the same alleles, i.e. bits are chosen, is selected. This results in a population size of 20. For each gene a range of \(\pm 20\%\) of the actually known value was chosen and split up into 50 subdivisions, by this limiting the exactness of the final estimations to less than 1%.

During testing of the estimator it can be noticed, that the quality of the parameter estimation depends not only on the input data, but also on the user-specified settings such as population size and definition of the fitness function.

Figure 4 shows the parameter estimation versus time. It is noted that the final parameter estimates differ less than 3% from the actual values.

The histogram information in Figure 5 shows the occurrence of accumulated parameter bin values, here at time step \(t=12\text{s}\).
It is interesting to note (Figure 4) that after approximately 3 seconds (or 6000 chromosomes at a sample time of 10ms) the model parameter values have converged. As such it is clear that the model parameters need not be updated as frequently as the states.

Figure 6 shows the estimates for the most important vehicle states. It should be noted that no reference signals were available for the vertical tyre forces and the tyre slip rates.

Figure 6 shows the estimates for the most important vehicle states. It should be noted that no reference signals were available for the vertical tyre forces and the tyre slip rates.

5 Conclusion

This paper has demonstrated a hybrid approach for the realisation of a model-based vehicle state and parameter estimator, using combined genetic algorithm and extended Kalman filter. The system states are estimated at each time step whilst a population of ‘candidate’ model parameter values are implemented / evaluated within a genetic algorithm routine. Results to date are encouraging, indicating that system states may be updated at each time step whereas model parameter values can be less frequently updated as new histogram information becomes available. This is considered to be particularly beneficial given that model parameters vary much slower than the system states. Further work is currently ongoing to enhance the stability and efficiency of this approach.

References