Kungliga tekniska Högskolan

Eco Cars

State of the art

Adam Lang, Andreas Fröderberg, Sanel Ferhatovic,
Alexander Ramm, Emil Hjelm, Richard Odell

May 23, 2016

Contents

Abstract

This paper presents the literature study of predictive off-line drive optimization for control of a hybrid vehicle with ethanol/gasoline and electric propulsion. The purpose of the optimization is to reduce fuel consumption for the vehicle on a predetermined track with known topology and distance. An heuristic optimized driving algorithm using dynamic programming can be found by simulating the system backwards in time. Positioning can be solved with a particle filter where the positioning data is filtered and compared to multiple sensor input. Dynamic programming is supported by a wide range of literature and is suitable for this type of optimization problem. The simplicity of a particle filtering algorithm makes it a good candidate for a first implementation with sensor fusion. A more advanced method is to fuse a Kalman filter with particle filter in order to increase accuracy.

1 Introduction

This report aims to cover the necessary state of the art research for the KTH Eco Cars project. In the project, a parallel hybrid electric vehicle (PHEV) with an ethanol combustion engine is designed and built. The vehicle will compete in the international competition Shell Eco Marathon. The objective is to consume as little energy as possible on a known track layout.

There is an already existing car, ELBA, that presents a platform to build upon and improve. The platform is developed using model based development (MBD) in Mathworks Simulink, allowing for modularity. It has the physical system architecture in place as well as a basic control system.

The goal of the project is to implement an optimal drive cycle algorithm that intelligently chooses drive mode to be as fuel efficient as is possible. This entails considering inputs and states of the physical plant to decide the torque demand on each of the systems motors.

2 Scope

The topology of the trajectory is known beforehand which is why the state of the art research will focus on off-line optimization algorithms where the optimal solution is calculated beforehand using Dynamic programming (DP). The focus on offline solutions and DP is because the algorithm used in the system is already set by an external part.

To be able to use the optimization during the race, the vehicle needs to be able to accurately estimate its position. Therefore the state of the art will concentrate on optimization algorithms used in HEVs and filtering position sensor inputs.

3 Drive mode optimization

There is currently a sizable body of recent research on the subject of drive mode control and optimization. In essence, the optimization problem that needs to be solved regards using the least amount of a finite resource under certain physical constraints and boundary conditions. For posterity, the mathematical description of this problem is given as

$$\begin{aligned} &\min_{u(t)} J\left(u(t)\right) \\ &\text{subject to} \\ \dot{x}(t) &= F(x(t), u(t), t) \\ x(0) &= x_0 \\ c(x(t), u(t), t) &\leq 0 \\ b(x(t_0), t_0, x(t_f), t_f) &= 0 \end{aligned} \tag{1}$$

where c is the path constraint and b marks the boundary conditions [11] [10]. $J(u(t))$ is the cost function, given as

$$J\left(u(t)\right) = G(x(t_f)) + \int_0^{t_f} H(x(t), u(t), t)\mathrm{dt} \tag{2}$$

Compared to a pure ICE powered drive train, a hybrid electric vehicle (HEV) has the ability to reuse kinetic energy when braking by regenerating the energy to an energy storage [6]. A hybrid system gives more possibilities to optimize but optimization algorithms can suffer from the curse of dimensionality [6] [10] i.e. more degrees of freedom (possible states) cause the computation time to increase exponentially. Therefore efforts have been made to simplify the problem and thereby reduce computation time, using for example heuristics [13] or by quasi steady-state models [10].

Solving the Eco Driving Optimal Control Problem (ED-OCP) of consuming the least amount of fuel during driving can be divided into two contexts [13]. If the environment of the trip is not known beforehand, a dynamic control problem arises that needs to be solved *online* during the trip. On the other hand, the path parameters and position dependent constraints can be known beforehand, leading to a problem that can be solved *offline*. Solving the problem online carries the additional constraint that the calculations have to be quick enough to be done in real time. Having a prior knowledge of the track, the OCP in ELBA during Shell Eco Marathon is suitable for offline optimization. The rest of the state of the art will therefore focus upon this.

Dynamic Programming (DP) is a method for solving the type of problems where a maximum yield or minimum spent resource needs to be found [1]. It can be applied to a wide range of mathematical problems, however it was initially meant to be used on problems that arise in situations where a sequence of decisions are made to reach a desired result [1]. These decisions are not independent, meaning that decisions taken early on greatly affects the outcome later in the process. This class of problems are called *multi-stage decision problems* [15]. Bellman introduced a concept called the *Principle of Optimality*, PO, which states that

> An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision [15] [9].

This means that at every step in the solution process, the following decisions must themselves be an optimal policy [15], i.e. that the solution at each step only depends on the residual amount [9]. This means that the algorithm exploits the problem to first find local optimum and then find the global optimal control policy [16]. In [9] the algorithm based on PO is summarized by three properties, namely that the state of the system is determined by a small set of parameters, that the effect of a decision is to transform the set into a similar set and finally that the past history of the system is not of importance when making a decision. It is the last feature that is unique to the PO. In his paper, Simpson makes further distinctions on the features of the multi-stage decision process problems that affect the DP solution. The first is whether the problem regards a deterministic or stochastic system. In deterministic systems, the outcome of a decision is known beforehand whereas in a stochastic system, there are multiple possible outputs with different probabilities associated with them. Secondly, an important aspect of the system is if there is a finite or infinite number of decisions at each stage. By extension, this property says if the problem is discrete or continuous at each stage. Finally, one may consider the amount of steps that are required in the solution. These can be finite or infinite. In some cases, the number of steps can be known beforehand but in some cases, they are given by the solution. As stated by Simpson, DP can be used to solve all these types of problems, with slight variations of the algorithm. It also works for problems where time is not explicitly given [15].

Dynamic programming, though versatile, is not without weaknesses. Like many other optimization algorithms, DP suffers from the curse of dimensionality [9] [15] [11] [6]. To circumvent this, [13] use situations where only one of the states of time and position affect constraints and state perturbations. In these situations, a state reduction can be implemented by using the transformation

$$\frac{d}{dt} = v\frac{d}{ds}. \tag{3}$$

These scenarios might arise in for example highway och urban driving without traffic lights where constraints, such as top speed, and perturbations, such as distance-depending slopes, depend only explicitly on position [13].

A more straightforward state reduction is proposed in [10]. When optimizing the fuel consumption for Formula One cars, the racing line is supplied to the algorithm rather than calculated which yields reduced computation time. This, of course, is only applicable in racing conditions since there is little need to optimize the exact path when driving on public roads.

Two novel approaches with many similarities are given in [13] and [6]. Both focus on Eco Driving optimization in HEVs and exploit the properties of the ED-OCP to decouple the solutions into levels [13] or layers in a hierarchy [6]. This is possible by using the fact that different parts of the optimization operate at different time scales which allows for varying degrees of abstraction in the vehicle model[6]. Both approaches suggest that the speed reference be in the top level of the optimization [13] [6]. In [6] it is proposed that the top layer be dedicated to the predictive optimization of energy buffers. This means that the top layer controller determines a speed profile and plans the use of the energy in the battery. This information is then fed into the middle control layer which controls the scheduling of gear and powertrain. Since the energy state is abstracted away at this stage, a simplified model of the drivetrain is used in the optimization to find the correct gear and the mode of the motor (on/off/neutral). The lowers layer of the optimization is the real-time decisions. Here, an acceleration request is sent to the system, along with the reference speed. The torque split is either determined by the maximum allowed torque of the electric motor (EM), of the ICE or by finding it using an optimization algorithm [6]. When combined, this method provides a sub-optimal solution to the problem. However, results of simulations show that the use of predicting the optimal drivemode based on the topography of the track yields up to 5% fuel saving potential. In [7], the concept is further developed with predictions of how surrounding traffic will affect the requested torque, i.e. how much acceleration/retardation is needed to keep a safe distance to the surrounding vehicles on the road. The optimal control algorithm prioritizes the use of the motor and driveline retarder for retardation as to not use the mechanical brakes which would cause losses [7]. The report shows that surrounding traffic can effectively be incorporated into the predictive controller.

A general control algorithm is presented in [11]. It can be used to solve deterministic optimality problems in discrete time. The continuous-time model used by DP must therefore be discretized in time to be able to solve this type of problems. The optimal control problem is described in Equation (1) and (2). Since the problem solved is deterministic, it is inherently non-causal since it expects the future disturbances to be known, which they are generally not. Nonetheless, the solution gives valuable information [11]. The optimal solution is found by simulating the system backwards in time, which is possible thanks to the Principle of Optimality.

4 Position estimation

To provide the optimization algorithm with relevant data, the vehicle must estimate its position in real time with suitable accuracy. A multitude of sensors can be used to determine ones position. Some of these are GPS [19] [8], Inertial Measurement Units [19] [17], Glonass [8], integrated wheel speed (dead reckoning) [3] and digital compass [19]. Multiple sensor inputs can be combined to increase the accuracy of the measurements. This technique is called sensor fusion [19] [8]. For fusing sensor inputs, two algorithms are prominent in the literature, particle filters and Kalman filters. These can be combined to form hybrid solutions. All three approaches are described below.

4.1 Particle filter

A particle filter (PF) is a type of filter that uses a Monte Carlo Simulation to estimate a chosen state [3]. Monte Carlo Simulation is a collective term for methods that uses randomness to estimate a mathematical relation. In the case of positioning the algorithm randomly distributes a number of particles (possible locations), on a known map. When the car senses its environment it compares the particles to the sensor data and gives each particle a weight, a measurement of how closely the particle resembles the sensor data. Then a new particle distribution is generated, with higher particle distributions close to the locations where the largest weights where. After this multiple particle clusters are possible, i.e. the algorithm can't be certain where the car is on the map. The estimated speed of the car is fed into the filter and all particles are updated according to a normal distribution around the speed estimate, due to non exact speed measurements. These steps are done every iteration and the probability density function of the cars location (where the car is in space) converges towards the actual position of the car.

General algorithm as described by [3]:

- First:
  - *Generate* N samples (particles) $x_0^i \sim p_{x0}, i = 1, ..., N$ randomly across all possible positions.
  - *Evaluate* the particles weights depending on their closeness to sensor values.
- Then repeat:
  - *Resample* N new particles distributed according to previous weights.
  - *Move* each individual particle according to a normal distribution of the measured speed and time since last measurement and update its weight according to the probability of the speed used to estimate its position.
  - *Update* weights using new measurements from sensors according to $w_t^i = w_{t-1}^i p_{et}(y_t - h(x_t^i))$

The main computational steps are the time update

$$x^i := Ax^i + B_u u + B_f f^i$$

and the measurement update

$$w^i := w^i p_e(y - h(x^i))$$

The complexity of a PF for large n_x (denoting the dimensions of the state vector) is $\mathcal{O}(Nn_x^2)$ [3]

Performance of PFs used for estimation and vehicles position depends on many parameters and techniques. With pure dead-reckoning (only integrating position change in the wheels over time) , Gustafsson [3] achieves an error that reaches ca 100 meters after 160 seconds on a kilometer scale map. Error is increasing linearly with time. But in addition with a GPS the error is never larger than 10 meters. Any combination of sensors can be used, depending on what information is available as a map or similar [2]. The fusion of the sensor comes in when weighting the particles before resampling. Further supporting positive results, in [12] it is found that sensor fusion with IMU and an Odometer provide accurate positioning when fused with a particle filter.

4.2 Kalman filter

A Kalman filter (KF) is used for taking advantage of linear and Gaussian structure within a system [3]. It is one of the most commonly used filtering techniques to integrate an inertial measurement unit (IMU) together with a position sensor [17]. Since position estimation using only IMUs or dead-reckoning is subject to bias, one can use one or more sensors fuse the sensor outputs and with the help of KF eliminate the accumulated error that would have occurred using only IMUs or dead-reckoning [8].

The complexity of a KF is $\mathcal{O}(2n_x^3)$ [3], where n_x is the dimension of the state vector.

The KF algorithm works in two steps. The first step is the prediction step where the KF produces estimates of the current state variables, together with their uncertainties. When the next measurement is made, these estimates are updated using a weighted average, where more weight is being given to estimates with higher certainty. [17] presents the algorithm steps as the following:

Prediction:

Predicted state:

$$\tilde{x}_k = \Phi_k \cdot \hat{x}_{k-1} + \Gamma_k \cdot u_{k-1}$$

Prediction covariance

$$\tilde{P}_k = \Phi_k \cdot \hat{P}_{k-1} \cdot \Phi_k^T + Q_{k-1}$$

Update:

Kalman gain:

$$K_k = \tilde{P}_k \cdot H_k^T \cdot [H_k \cdot \tilde{P}_k \cdot H_k^T + R_k]^{-1}$$

Estimated covariance:

$$\hat{P}_k = [I - K_k \cdot H_k] \cdot \tilde{P}_k$$

Estimated state:

$$\hat{x}_k = \hat{x}_k + K_k \cdot (z_k - H_k \cdot \tilde{x}_k).$$

Here k represents the iteration number, x_k is the state, u_{k-1} is the deterministic input, ϕ_k is the system transition matrix from time t_{k-1} to t_k, Γ_k is the input matrix, H_k is the measurement matrix, Q_k is the covariance of the process noise and R_k is the covariance of the measurement noise.

In the literature, there are several examples of Kalman filters used in sensor fusion techniques. In [18], it is shown that the use of sensor fusion with KF increase the accuracy in applications where one sensor is subject to drift and one sensor has absolute errors. Similarly it is concluded in [14] that sensor fusion based on a Kalman filter that measurements from several sensors of similar type can get boosts in accuracy.

4.3 Hybrid techniques

KF is often used in combination with a PF to reduce the computational complexity [17]. When a vast amount of particles are needed for the PF to perform well, one way of dealing with the computational power needed would be to estimate as much as possible analytically using KF [4]. Then the remaining part of the reduced dimension can be estimated using PF. The combined filter would make the problem tractable by reducing the amount of particles significantly [4][17].

GPS can pinpoint the absolute longitude and latitude coordinates of a position on earth. However there are some limitations due to slow updates, signal interference and limited accuracy. In many cases, for example tunnels, the GPS radio waves cannot reach the GPS unit and the position of the unit cannot be identified [5]. To overcome these problems, reference [8] proposes a fuzzy-logic-tuned KF sensor fusion technique. The method appropriately combines measurements from a vehicle differential GPS witch a yaw-rate gyro and a speed sensor with respect to the reliability of of each sensor information. The KF equation builds on a dynamical relationship among the sensors and the fuzzy logic tunes the parameters of the KF algorithm. By tuning the parameters of the KF the error caused by consistent and sensor inaccuracies can be minimized [8]. In [5], several approaches to mitigating inaccuracies or losses in GPS data are given. These include using dead reckoning, increasing the accuracy by using differential GPS, also seen in [8]. By the results in the report, one can conclude that GPS is of high importance when determining position but that the reliance of accurate GPS data can be reduced by using multi-sensor fusion.

5 Conclusion

There are several ways to solve the OCP in a hybrid driveline. As has been found in this state of the art study, most of these suffer from exponential growth in computation time when the amount of states increase. To mitigate this, different methods such as using heuristics, state reduction and decoupling of the problem into layers. When using hierarchical solutions, one can lever the time frame properties of the HEV optimization problem and use different plant models for the different levels. This reduces the computation time while producing a good (though sub-optimal) estimation of the optimal solution. There are other methods of solving the optimality problem, amongst else Pontryagins minimum prinicple and analytical solutions. These methods have only been mentioned in passing or not at all. The reason for this is that the choice of optimization algorithm is not a choice of the authors of this report, but is given by an external source. Viewing the literature, it is clear that the use of dynamic programming is widespread and accepted as a good method of computation for HEVs. This is supported by the fact that the reports using the hierarchical solution structure are very recent and on the cutting edge of HEV optimal control.

Based on the resolution of the optimization the position estimation method choice can be made. For position estimation a PF is a feasible solution. With the sensors and data we have available, enough precision is possible. This precision depends on how many particles are used which in turn is dependent on processing power available. PF is a simple algorithm and would be fairly easy to implement and is a tempting choice. Tests have to be run to see if the processing power in the ECUs is sufficient.

Sensor fusion based on Kalman filters is well proven in the literature. There are several variations of the algorithm, one being fuzzy logic KF. The method is proven to be useful for very precise positioning and could be implemented on ELBA with one major difference. The researchers in [8] are forced to use differential GPS due to the fact that when their research was conducted, the GPS system was subjected to an intentional frequency noise (SA) introduced by the satellites for security reasons. The ELBA car could use the absolute position from the GPS and the measured distance from the encoder. Using Fuzzy Logic and KF the accuracy of the GPS and the encoder can be weighted with respect to the accuracy of them at any given point of time. The final absolute position after the filtering process should be sufficiently accurate.

Moreover, it is shown in the literature that hybrid techniques can be used where a Kalman filter is used to reduce the dimensionality of the Particle filter algorithm, yielding shorter computation times. This method could allow for larger amounts of particles, increasing the accuracy of the measurements while utilizing ECU processing capacity as much as possible.

References

[1] Richard Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 1957.

[2] Fredrik Gustafsson. Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronics Systems magazine*, 2010.

[3] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and Per-Johan Nordlund. Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*, 2002.

[4] Fredrik Gustafsson and Per-Johan Nordlund. Secuential monte carlo filtering techniques applied to integrated navigation system. *Procedings of the American Control Conference*, 2001.

[5] Chengwei Huang, Yong Liu, Jia Yunyi, and Heping Chen. Position estimation for an unmanned ground car (ucg) by multi-sensor fusion under random loss of gps signals. *IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, 2015.

[6] Johannesson, Murgovski, Jonasson, Hellgren, and Egardt. Predictive energy management of hybrid long-haul trucks. *IFAC, the International Federation of Automatic Control*, 2015.

[7] Lars Johannesson, Magnus Nilsson, and Nikolce Murgovski. Look-ahead vehicle energy management with traffic predictions. *IFAC, the International Federation of Automatic Control*, 2015.

[8] Kobayashi K., Cheok K., Watanabe K., and Munekata F. Accurate differential global positioning system via fuzzy logic kalman sensor fusion technique. *IEEE Transactions on Industrial Electronics*, 1998.

[9] E.S Lee. History and development of dynamic programming. *Control Systems Magazine*, 1984.

[10] Rao Limebeer. Faster, higher, and greener. *IEEE Control Systems Magazine*, 2015.

[11] L. Guzzela O. Sundstrm. A generic dynamic programming matlab function. *IEEE Multi-conference on Systems and Control*, 2009.

[12] Berk Pelenk and Tankut Acarman. Object detection and tracking using sensor fusion and particle filter. *IEEE International Conference on Imaging Systems and Techniques*, 2013.

[13] Ojeda Scarrietta, De Nunzio. Optimal ecodriving control. *IEEE Control Systems Magazine*, 2015.

[14] Rahul K Sharma, Daniel Honc, and Frantisek Dusek. Sensor fusion for prediction of orientation and position from obstacle using multiple ir sensors an approach based on kalman filter. *International Conference on Applied Electronics*, 2014.

[15] M.G Simpson. An introduction to dynamic programming. *Journal of the Royal Statistical Society*, 1960.

[16] Koos van Berkel, Bram de Jager, Theo Hofman, and Maarten Steinbuch. Implementation of dynamic programming for optimal control problems with continuous states. *IEEE Transactions on Control Systems Technology*, 2014.

[17] Seong-hoon Peter Won, Wael William Melek, and Farid Golnaraghi. A kalman/particle filter-based positioning and orientation estimation method using a position sensor/inertial measurement unit hybrid system. *IEEE Transaction on Industrial Electronics*, 2010.

[18] Sang Won Yoon, Seong-Bae Park, and Jong Shik Kim. Kalman filter sensor fusion for mecanum wheeled automated guided vehicle localization. *Journal of Sensors*, 2015.

[19] Pifu Zhang, Jason Gu, Evangelos Milios, and Peter Huynh. Navigation with imu/gps/digital compass with unscented kalman filter. *IEEE Conference on Mechatronics and Automation*, 2005.

Status Report

The car that will be used in the race and development project, ELBA, was supplied by last years team. The system had the basic architecture in place, i.e. a hybrid driveline with a BLDC, a DC motor, an ICE and a novel clutch solution. Development had been conducted by the previous years team, with a focus on a general, modifiable control system. The car was in a working state, and was test driven with the old driveline early on in the project. There were known weaknesses in the car that needed to be solved in order to be able to complete the race. Therefore, the work conducted on the car has had two focuses. The first has been to lay the groundwork for this years project, to implement a control algorithm that optimizes the fuel efficiency of the system. It was decided to have a basic control algorithm implemented by the time of the race.

The second focus has been on fixing the physical limitations of the system to be able to complete the race. It is crucial to the project to be able to gather test data at the race to have data to evaluate during the continued development of the system during the fall semester.

To have a reference frame of how the project is progressing, the initial time plan is given below.

- **May 11:** System test with new ICE mounted with a control system in place, based upon last years control system.
- **May 27:** Full system test with system that will be used at the race (simple version of optimized control system).
- **June 15:** Full system test with small modifications made since last test.

The course goals mainly concern the optimization of the control system. By extension, this means that the state of the physical system can not be left out of the status report since a working car is needed to gather relevant data and test the control algorithm. Therefore, the progress on both parts is reported.

Optimization algorithm

Currently, the Simulink model can be used to partially compile code to the ECUs of the car, however there are still some sections of the code compilation that have not been made to work. Amongst these are the logging system, the CAN communication and some aspects of the front ECU. Being able to compile ECU code was a part of the May 11 goal and we are therefore behind two weeks on this goal. Effort has been put into understanding how to use and modify the control system but the combination of the use of advanced Simulink features and lacking documentation in the handover has caused this to take more time than expected. Effort is continuously put into solving this since it is vital to the success of the project. Progress is made every day and the group has a positive view on the possibilities of solving the problems.

It was planned to have the race ready system done by May 27. The state of the art has been very helpful in determining which methods to use to gather the position data needed for the optimization algorithm. The things mentioned in the state of the art might not fully be in place for the race so a compromise will most likely be made where the aim is to have some optimization done by the exams. We also aim to evaluate how good our position estimate is through logging data during the runs at the track. The deadline is likely to be missed here also, with reasons related to the setbacks early in the project. A lot of progress has been made in the last few weeks with only minor hurdles to overcome in the Simulink system before having a working system. The group recognizes that the implementation of an optimal control until the race may prove difficult which is why different compromises are discussed to make the deadline of the race. Since the optimal control algorithm is based on layers, one layer at a time can be implemented. This would mean that the optimal speed reference would first be implemented using last years drive mode control algorithm. When this is successfully implemented, the second part of the optimization can be implemented, i.e. the torque split and drive mode controller.

The car

According to the time plan, it should have been possible to drive the car now. Even though this is not the case, a lot of progress has been made. The ICE is mounted in the car, the old clutch is mounted and a lot of the improved software is implemented. However there are some improvements to be made, the ICE needs to run on injection to be efficient, that means that the carburetor needs to be exchanged and a new control system needs to be made. This work has come a long way, but there is still some work to be done. The new and improved clutch also needs to be fitted and implemented. The parts have been manufactured and are waiting to be implemented after testing. When it comes to the ICE and clutch is the Mechatronics team not responsible other than the control systems, but these parts indirectly affect the course because of the requirement for a physical system. It is crucial to test the full system with the ICE since car must be running at a certain speed and drag the engine to start it.

The overall weakness of the car have been it's lack of robustness, large amounts of time have been spent just to make the whole system more reliable. Since there are parts of the old system still in place it is hard to exclude errors relating to this which makes the troubleshooting much more difficult.