Recherche Opérationnelle
Sud Ouest (ROSO)

École Polytechnique
Fédérale de Lausanne

Energy Management Optimization for a Solar Vehicle

Master's Thesis

February 24, 2006

Alexander Scheidegger

Professor
Prof. Thomas M. Liebling

Assistants in Charge
Emma Frejinger
Michael Themans
Gauthier Stauffer

The content of this report is confidential until October 2007.

Contents

Chapter 1

Introduction

1.1 The Research Project and its Context

This master thesis contributes to a student project at EPFL that aims at participating in the competition 'World Solar Challenge' 2007, a race for vehicles powered by solar energy. The goal of this race is to drive across the Australian continent from north to south as fast as possible. This competition has been held biannually since 1987. Competitors from universities and companies from all over the world participate. The EPFL team is developing an innovative solar car in order to participate.

The participating vehicles are powered exclusively by electricity. The energy used has to be converted from solar radiation by photovoltaic cells during the race. The vehicles can use batteries for energy storage. Computers external to the vehicle can be used to optimize their use during the race. The goal of the present work is to develop and test methods for this purpose. This work is done for a vehicle with given parameters. This is, we suppose that its construction can not be influenced.

More concretely, the characteristics of the race relevant to the present optimization problem are the following:

- A specific road from Darwin to Adelaide of 3000 km is given. The location of hills, the state of the road and direction of the road at each location are initially known.
- All vehicles will start on October 21, 2007 at 8 am in Darwin. The vehicles can be driven from 8 am until 5 pm each day. In the morning before 8, as well as in the evening after 5, the collected solar energy can be stored in the batteries. The car that arrives first in Adelaide wins the race.
- At the start, the batteries are full. The rules allow batteries that can store maximally 5 kWh of nominal energy. All teams use batteries that can store exactly 5 kWh.
- The race rules force some additional stops at specific locations. These are known in advance.
- The highway is open to regular traffic, so legal circulation rules have to be respected as well as limitations caused by other vehicles. Most road tracks are without speed limitations. The model considered in this work takes only into account speed limitations.

The goal is to allocate the electric energy collected from solar radiation optimally to the drive system or to the batteries on one hand side, respectively to allocate the energy stored in the battery optimally to the drive system over the time of the race. In order to optimize this allocation at one moment, we need to plan the energy management for the whole race.

To do so, the problem is formulated as follows:

- We use a model linking energy input, energy consumption and dynamics of the car. Estimations of the parameters for this model are assumed to be available. They are assumed to be constant during the whole race. A more sophisticated model could take into account the dynamics of these parameters as stochastic process during the race.
- Accurate energy management planning needs an estimation of some unforeseeable environment variables. In fact, the future intensity of solar radiation influences the energy input into the system. Future wind speed and wind direction influence the resistive forces acting on the vehicle. A stochastic model will be used to predict their evolution. This model uses information from climate data time series and weather forecasts. Taking into account other unforeseeable external variables such as the amount of traffic on the road is beyond the scope of this work.
- The system is controlled by a one dimensional control variable, which is the driving power. In an approximation, the vehicle's speed can be considered as control of the system.
- The objective of optimization is to choose a control that minimizes the time-to-go until the finish. For a stochastic problem formulation, it is to minimize the expected value of the time-to-go.

1.2 Methodology

1.2.1 Restriction of the Work

The first step is to introduce a formulation for the optimization problem, provided that the uncertain quantities for the future are known. Once this complete problem formulation is available, the problem is simplified in different manners. This simple formulation is extended to a stochastic formulation at the subsequent step. For some relevant simplified problems, optimization methods are developed, and some of them are implemented and tested.

Meyer (2005) studies first approaches to the numerical resolution of the optimization problem. The problem is splitted up into two partial problems that are initially considered as independent.

- A *long term energy management problem.* It consists in allocating the available energy over the road tracks.
- A *short term energy management problem.* This is to optimize control over the next road track, using the energy that is allocated to it at the previous step.

Here, the same basic structure is adapted.

The main challenge in long term optimization is to face uncertainty. A useful method to solve this problem should provide a solution with the following properties.

- If the conditions are suddenly different than expected, one should immediately know which quantity of energy should be allocated to the current road track, given the new circumstances.
- If the weather prediction changes, the solution should contain enough information such that the optimal control can easily be updated on-line.

- This requires that strategies should be planned as well for states that have not been foreseen in an initial planning.

The issue of this work is to develop, implement and test such methods. The second important challenge concerning long term optimization is to develop on-line policy update methods. This is outlined, but only a trivial solution is treated here.

The short term optimization problem is not yet considered in this work. In fact, we think that for the subsequent hour, relatively exact information about the weather will be available during the race, and the optimization problem. We hope that this causes the problem to be simple compared to the long term optimization problem.

The two optimization problems are not independent. The long term energy management should use short term behavior values under the hypothesis that for any period, short term control is optimized. Unforeseen optimization results for short term behavior should allow to adapt quickly the resources allocated to the concerned time period.

1.2.2 Optimization Methods

Meyer (2005) proposes to use Dynamic Programming (DP) to solve this problem. In fact, the broad methodology of DP and approximate DP, as exposed in Bertsekas (1995a), Bertsekas (1995b) and Bertsekas & Tsitsiklis (1996), seems to offer many alternatives that are applicable to this dynamic optimal control problem. Therefore we restrict the optimization methods used in this work to DP related methods.

1.2.3 Contributions

All work has been been done on a model that is simplified, in order to avoid unnecessary complication for the method development. At the same time, we think that the model is sophisticated enough, such that the main difficulties of the problem occur. The principle of the approaches found should be extendable to a more complicated formulation. The following results that contribute to the solution of the long term optimization problem have been found.

- We develop two methods that determine off-line the optimal discharge profile, provided that the future values of the uncertain quantities are deterministically known. They have been implemented, tested and compared.
- We give a stochastic model of weather forecast information based on climate data analysis. This model is especially useful for testing optimization methods.
- An off-line stochastic DP algorithm with respect to that model has been developed. However, this algorithm is likely not accessible for computation. Two approximative methods have thus been developed and implemented, and their performance is compared.

1.3 Outline

In the second chapter, a formulation of the deterministic optimization problem is presented in detail. Model simplifications for working out efficient optimization methods are proposed.

In the third chapter, we study DP methods for solving the deterministic long term energy management problem.

In Chapter 4, we introduce a stochastic weather forecast information model. Two approximative off-line stochastic DP methods are discussed, tested and compared.

This is followed by a conclusion in Chapter 5, where the present work is set into the context of the whole research project.

Chapter 2

Deterministic Problem Formulation and Analysis

Throughout this chapter, all quantities relevant to the optimization problem are supposed to be known in advance with certainty. This is, we consider a deterministic approximation of the problem.

In Section 2.1, a formulation of the deterministic optimization problem is presented. It includes the relation between the energy flows and dynamics of the vehicle, a model of the system's control and of the objective of optimization.

Section 2.2 is a review of Pudney (2000). This is a theoretical approach to this problem, where properties of the optimal control function are analyzed.

In Section 2.3, we decompose the optimization problem into a short term optimization problem and a long term optimization problem. Afterwards, we propose appropriate model simplifications that facilitate the development of optimization methods.

2.1 Formulation of the Optimal Control Problem

The model does not describe in detail the processes in the vehicle electronics and in the motor. It is designed to describe the relation between *energy flow* into the system, energy flow from and to the batteries, and the *dynamics* of the vehicle. Here, dynamics expresses only the one dimensional motion of the vehicle from the start towards the finish of the race.

- p_{sol} is the energy flow from the environment into the system. The value of this flow expresses the quantity of solar energy transformed into electric energy. This is the energy to be allocated to the *drive system* and to the *battery*. p_{sol} depends on the environment variables.

- I is the energy flow from or to the battery. It is positive if energy flows to the battery, and negative if energy flows from the battery. This energy flow is the unique control of the system.

- p_{in} is the energy flow to the drive system. Given the environment variables and the vehicle parameters, p_{in} determines the dynamics of the system.

This system is represented in Figure 2.1. In the following, the model and the corresponding optimization problem are described in detail.

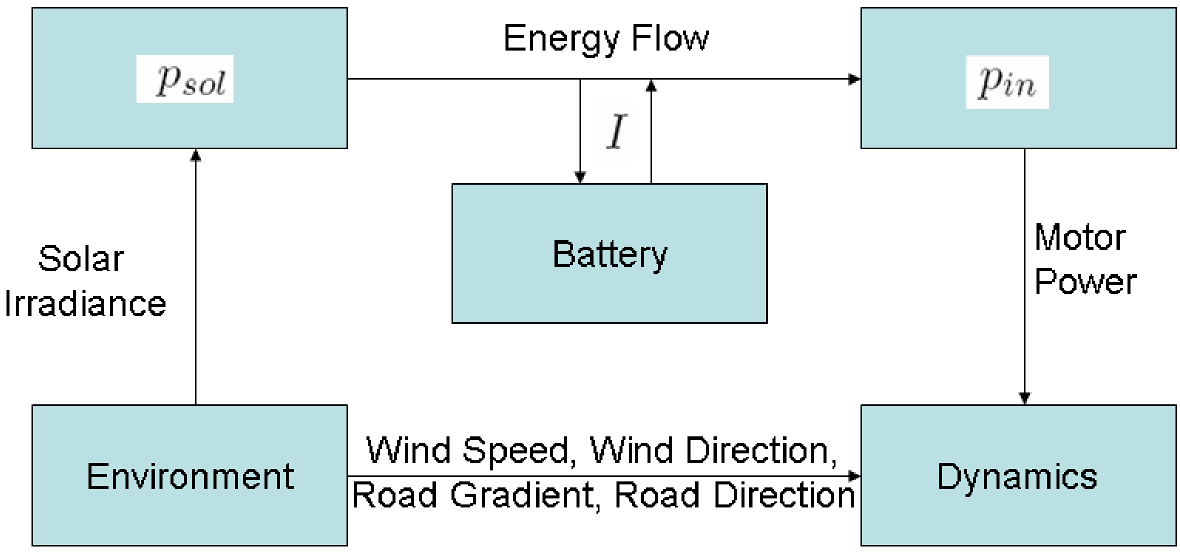


Figure 2.1: This scheme summarizes the energy flows of the present model, as well as its relation to the vehicles dynamics.

The State of the System

The independent variable of the system is the time t. The state of the system is described by three state variables, s_t, v_t and q_t.

- s_t The distance of the vehicle from the start of the race at time t. We refer to this quantity as *location* of the vehicle.
- v_t The *speed* of the vehicle at time t.
- q_t The nominal energy stored in the batteries at time t. The symbol q is chosen, as the common notation for charge. We will refer to it as *charge of the battery* or simply as *charge*.

For simplicity of the notation, we drop the index t sometimes, and the state of the system is denoted by

$$x = (s, v, q) = x_t = (s_t, v_t, q_t). \tag{2.1}$$

Energy Flow into the System

There are several alternatives in modeling the power of solar irradiance incident on a panel. They are not discussed here in detail. For more information, we refer to Duffie & Beckman (1980), Pudney (2000) and Meyer (2005). The present model is taken from Meyer (2005). The panel is modeled as a horizontal surface of 1.8 m × 5 m. The energy flow into the system, p_{sol}, decomposes in two parts. The first one, I_{horiz} is the solar power available under perfect atmospheric conditions.

It is called *clear-sky intensity* and is approximated by

$$I_{\text{horiz}} = \frac{3}{8} W_{\max} \left(3\sin(\alpha_{\text{sol}}) - \frac{1}{4}\sin(2\alpha_{\text{sol}}) + \frac{1}{3}\sin(3\alpha_{\text{sol}}) \right), \tag{2.2}$$

where α_{sol} is the angle of incidence of the beam irradiance, a function of time and location, and $W_{\max}$ is a constant. To obtain p_{sol}, I_{horiz} is multiplied by some loss factors that are explained in Table 2.1.

The energy flow into the system is then given by

$$p_{sol} = \eta_{\text{MPPT}} \eta_{\text{panel}} A_{\text{panel}} B I_{\text{horiz}}. \tag{2.3}$$

B	The brightness factor B is the ratio between the irradiate power per horizontal surface and the clear-sky intensity. This quantity is determined by the brightness of the atmosphere, that diminishes especially if the sky is cloudy. It varies in function of time and of location. In this chapter, this function is treated as if it would be known in advance. An accurate model must treat it as a random function. This is discussed in Chapter 4.
A_{panel}	The surface of the vehicles solar panel.
η_{solar}	The energy efficiency of the solar cells. This can be modeled as a function of temperature, of the irradiate spectrum and of the solar incidence angle. In this model, it is approximated by a constant factor η_{solar}.
η_{MPPT}	The Maximum Power Point Tracker links the solar panel to the electric chain of the vehicle. Its energy efficiency is approximated by a constant factor.

Table 2.1: Factors for the energy input model.

The Model of the Drive System

This part of the model links the energy flow to the drive system, p_{in}, to the driving force at the wheels F. All losses in the vehicle electronics and the motor are summarized in this dependence. The most simple model uses a constant efficiency factor η, that is

$$F = \frac{p_{\text{out}}}{v} = \frac{\eta p_{\text{in}}}{v}, \tag{2.4}$$

where p_{out} is the output power of the drive system (Meyer 2005).

This model is used here, in order to simplify the development of optimization methods. However, a more accurate model would use an empiric loss function

$$p_{\text{in}} - p_{\text{out}} = L(v, p_{\text{out}}). \tag{2.5}$$

The Battery Model

The battery model links the energy flow I into the battery to the difference $p_{\text{sol}} - p_{\text{in}}$. The simple model (Pudney 2000) that we use here approximates I by two constants, the battery charge efficiency η_{charge}, and the battery discharge efficiency $\eta_{\text{discharge}}$. Explicitly, this is

$$I = \begin{cases} (p_{\text{sol}} - p_{\text{in}})\eta_{\text{charge}} & \text{if } p_{\text{sol}} - p_{\text{in}} > 0; \\ \frac{(p_{\text{sol}} - p_{\text{in}})}{\eta_{\text{discharge}}} & \text{otherwise}. \end{cases} \tag{2.6}$$

Models where I depends in addition on the battery charge q are more accurate (Pudney 2000). This dependence is neglected here. The reason is again that we wish to have a simple model for developing optimization methods.

Dynamics

Resistive forces due to friction with the road and to aerodynamic forces are modeled by

$$R(s, v, w_x) = mgc_{rr1}\cos(\theta(s)) + Nc_{rr2}v + \frac{1}{2}\rho S_x C_x (v - w_x)^2, \tag{2.7}$$

where s and v are the state variables location and speed. w_x is the wind speed component in the direction of motion of the vehicle. This is, similar to the brightness factor, a quantity that is treated as uncertain in the model from Chapter 4. Here, we treat it as if it would know its evolution with respect to time and location with certainty. It depends on the wind speed w and the wind direction w_α, that are both functions of t and s, and as well on the driving direction α_{road} that is a function of s.

The parameters/constants of the model are

m	The mass of the vehicle.
g	Gravity acceleration, $g \approx 9.81\frac{\text{m}}{\text{s}^2}$
$\theta(s)$	The gradient of the road at location s. $\cos(\theta(s))$ is approximated by 1.
c_{rr1}/c_{rr2}	Coefficients of rolling resistance.
N	Number of wheels of the vehicle.
ρ	Density of the air. $\rho \approx 1.22\frac{\text{kg}}{\text{m}^3}$.
S_x	Frontal surface of the vehicle.
C_x	Aerodynamics coefficient.

We suppose that these parameters are known and constant during the whole race. This is an approximation that can be revisited. For an accurate energy management planning, the (stochastic) degradation of their values during the race might have to be considered.

Gradient

The gradient force acting on the car is

$$G(s) = -mg\sin(\theta(s)). \tag{2.8}$$

The State Equations

Using this model, the evolution of the state of the system follows the law:

$$\frac{ds}{dt} = v \tag{2.9}$$

$$\frac{dv}{dt} = a = \frac{1}{m_R}(F - R + G) \tag{2.10}$$

$$\frac{dq}{dt} = I, \tag{2.11}$$

where m_R is the effective mass of the vehicle, compensating forces due to its rotational dynamics. It is the sum of the vehicle's mass m and a relatively small constant term.

Control

The system has a one dimensional control variable. It is equivalent whether I, p_{in}, p_{out} or a is controlled. In fact, using equations 2.2, 2.3, 2.4, 2.7, 2.8 and 2.10, an implicit relation between each of those two quantities is established. From an intuitive understanding of the system, it is clear that each one of these mutual relations must be strictly increasing, thus bijective. The existence of the inverse function explains that it is mathematically equivalent to choose any of those controls.

Boundary Conditions

The race starts at $t_i = 0$ with full batteries. It it finishes at location $s_f = 3011$ km. The boundary conditions are thus

$$s_i = 0; \quad v_i = 0; \quad q_i = q_{\text{full}}. \tag{2.12}$$

$$s_T = s_f. \tag{2.13}$$

Here, T is the time when the race finishes, and $q_{\text{full}} = 5$ kWh $= 18$ MJ.

Constraints

The constraints on the state variables to be respected are

$$0 \leq v \leq V_{\max}(s); \tag{2.14}$$

$$0 \leq q \leq q_{\text{full}}. \tag{2.15}$$

The upper speed constraint $V_{\max}(s)$ describes speed limitations forced by the rules of the race or by the shape/state of the road. At locations where the speed is not limited, it is set to the maximal speed the vehicle can reach.

The constraints on the control are

$$B_{\text{charge}} \leq p_{\text{in}} - p_{\text{out}} \leq B_{\text{discharge}} \tag{2.16}$$

$$p_{\text{in}} \leq P_{\max}, \tag{2.17}$$

the constraints on the maximum power flow from/ to the battery and the maximum motor power constraint.

Objective Function

The objective of optimization in this model is to minimize T, the time when the race finishes. In the stochastic approach we follow in Chapter 4, we consider the objective to minimize the expected value of T.

With respect to the competitive character of the race, an other objective function could be formulated: Namely, to maximize the probability to win or to arrive faster than a certain competitor at finish line. In this case, the problem would fit into the framework of stochastic games (Bertsekas & Tsitsiklis 1996). This is beyond the scope of this work. In fact, our approach is to consider the simplest relevant formulation of this optimization problem.

2.2 Literature Review: Analysis of the Optimal Control Function

This section is a review of the findings presented in Pudney (2000). He studies properties of the optimal control for the deterministic problem formulated above, by adding different hypotheses. This section is essential to get a deeper intuition about the problem.

Optimality of a Constant Speed Strategy

First, we consider the restrictive hypotheses

1. solar irradiance does not depend on location;
2. driving force does not depend on location;
3. perfect drive system;
4. perfect battery; and
5. infinitely big battery.

Under these hypotheses, it is shown that the optimal control of the problem contains only three types of sequences: Maximal power driving, constant speed and maximal regenerative breaking.

Inefficient Battery

We relax the perfect battery hypothesis, and battery charging is supposed to be constantly efficient, at efficiency $\eta_{\text{charge}} < 1$. All other above hypothesis are maintained. An optimal strategy contains two constant driving speeds, a low discharging speed if the instantaneous irradiance is 'small' and a high charging speed if it is 'big'. During transition between the speeds or if irradiance is between 'big' and 'small', the car is driven exactly with solar power.

By introducing hills into this model, the above strategy principles do not change, provided that the maximum power of the motor is big enough to maintain the charging speed at the steepest point of the road. This result might be surprising, but it is due to the perfect drive system hypothesis.

A more general battery model than the one described before allows the energy efficiency of the battery to vary with I, the energy flow from/to the battery. If we allow this variation under the above hypothesis, the speed that maximizes the distance traveled during a clear day on a level road is an increasing continuous function of the apparent solar power. For values of the Aurora solar car (2001), it increases from 80 to 90 $\frac{\text{km}}{\text{h}}$, when solar power increases from 0 to 2000 W (Pudney 2000).

Inefficient Drive System

The next hypotheses we consider are

1. Solar irradiance does not depend on location;
2. Driving force does not depend on location;

3. Perfectly efficient battery.

The principle of maintaining a constant speed does not change. The driving modes of maximal power and maximum regenerative braking are adapted under the realistic hypotheses that power losses increase with driving power. If we consider hills, the principle of constant speed changes. The optimal speed policy is to accelerate before a hill from this constant speed. The driving speed variation and the driving power variation can be kept small that way.

Spatially Varying Irradiance

The brightness factor can vary considerably spatially, if parts of the sky are covered by clouds. With a perfectly efficient battery, an optimal strategy accelerates under clouds and slows down under clear sky. With drive system parameters of the Aurora car and perfectly efficient battery, under noon solar conditions, 10 s can be gained on 50 km, where the second half of the track is under clouds. Under realistic battery assumptions, this is opposed to the fact presented above, that speed should increase with solar power. If we accelerate under clouds, the energy flow from the battery $-I$ increases, and with it the energy losses.

This effect is not only to consider at the scale of some kilometers, but also at large scale. In fact, it has often been told that the big difference between the winning car and the second car of the 2005 edition was (partially) due to the 'better weather' they had at the end of the race. This refers to an effect of spatially varying irradiance. In fact, given a time, the weather was better on the last kilometers of the race than at other locations, and other teams should have accelerated before. To take into account such effects, the spatial variation of irradiance must be modeled.

Strategic Principles

The following strategy principles result from the above analysis:

1. Constant speed: Select an average speed that can be maintained during the race;

2. Battery pampering: Increase the speed (slightly) with solar power;

3. Hill riding: Accelerate before hills;

4. Sun chasing: Accelerate under clouds (or accelerate, break to avoid future clouds).

Pudney (2000) remarks that the importance of varying the speed not too much during the race, and to select this speed such that the batteries are empty at the end of the race, is primordial with respect to the other principles.

2.3 Appropriate Model Simplifications

In the following section, we point out appropriate simplifications of the model. In the first subsection, we present the option to eliminate the speed from the state space. This means that the dimension of the state space is reduced, but some accuracy of the model is lost.

In Subsection 2.3.2, we divide the problem into two subproblems, a short term optimization problems and a long term optimization problem. We motivate our decision to only study here the long term problem.

In order to study the basic algorithmic aspects of the problem on a simple model in this first approach, we specify some further work hypothesis in the Subsections 2.3.3 and 2.3.4.

2.3.1 Neglecting Acceleration

Assume that an optimal strategy contains a priori 'few' speed variation. The problem could thus be approximated by neglecting the 'energy cost' caused by acceleration of the vehicle, and take into account the energy needed to maintain a given speed during a time interval. As described by Meyer (2005), this hypothesis allows to reduce the dimension of the state space as follows.

Setting acceleration to 0, the explicit relation for p_{out} respectively p_{in} as function of v,

$$p_{\text{in}} = \frac{v}{\eta}\{mgc_{rr1} + Nc_{rr2}v + \frac{1}{2}\rho c_x A_x (v - w_x)^2 + mg\sin(\theta(x))\} \tag{2.18}$$

can be derived from the relations 2.10, 2.4, 2.8 and 2.7.

Such an explicit relation exists even if the drive system losses (equation 2.5) are expressed as a polynomial in v and $\frac{p_{\text{out}}}{v}$, as suggested by Pudney (2000). This gives the following simplified optimization problem:

- The state of the system is $x = (s, q)$.
- The control of the system is v.
- The state evolution is given by $\frac{ds}{dt} = v$, and $\frac{dq}{dt}$ is given by Equation 2.6. $p_{\text{in}}(v)$ is given by the above relation 2.18.
- The objective function is unchanged with respect to the formulation of Section 2.1.

Applying this simplification influences some imprecision into our model. However, this imprecision can be reduced by dividing the optimization problem into two subproblems, a short term optimization problem and a long term energy management problem, as we describe it below, if we apply the present simplification only to the long term energy management problem.

2.3.2 Long Term and Short Term Energy Management

Meyer (2005) mentions the idea to divide the optimization problem into subproblems that are distinguished by their time horizon. However, he gives no concrete formulation to this approach. In turn, this idea plays a key role in our approach. Let us motivate and describe this approach informally.

Consider the present problem, but not with a continuous time, as it is presented above, but considering only a discrete set of time stages $\{0, \dots N\}$. Let the time interval between two neighbor time stages be relatively long, say 30 minutes. Suppose that we are in a race situation at a certain time t_0 and at a fixed state. For the subsequent time interval, we have a set of possible controls. Using a given control, we can compute at what state we arrive after 30 minutes. Suppose now that given this state after 30 minutes, we know how much time we need to complete the race. If we compute this 'time-to-go' for all possible controls for the next 30 minutes, the control that arrives at the state that minimizes this time-to-go is optimal.

This is the way we divide our problem. Let us define a key notion for dividing the problem. Consider a fixed time stage k, a state (s_k, q_k) and some control v_t between the time stages k and $k+1$. To this control, we can associate the *energy gain per time stage* $\Delta q = q_{k+1} - q_k$.

Let us consider the two subproblems:

- Let us consider a fixed charge gain per stage at stage 0, Δq_0. The *short term energy management problem* is to compute the control that maximizes the distance Δs traveled during the 30 minutes time period, among all controls that have an associated charge gain per stage Δq_0.

- Once the short term energy management problem is solved for all feasible Δq, we have some optimal relation $\Delta s_0(\Delta q)$ for the next 30 minutes time period. The *long term energy management problem* is to choose, for the next 30 minutes, a charge gain Δq that optimizes the time-to-go.

Compared to the long term energy management problem, the short term problem is easy to solve. Solving the long term energy management problem includes to solve the short term energy management problem many times. In fact, an accurate energy management planning for the whole race must suppose that the future short term control is optimized in any situation the system meets.

However, in our initial situation when we ignore which control is optimal, we do not know which situations will be met, and thus for which situation the short term problem must be solved. If no approximation is made, the short term problem must be solved for very many situations. Precisely, its solution can depend on time, location, battery charge, brightness factor, wind speed and wind angle.

The question is how much resources we will be able to invest in solving the short term problem. Thus, we will study the long term problem first, using a very rough approximation for the relation $\Delta s_k(\Delta q)$, the optimal solution of the short term problem.

Once an accurate algorithmic method for the long term problem is available, we will be able to estimate how accurately the short term problem can be solved.

Solving the long term energy management problem only contains the following approximations:

- We discretise time, into larger intervals that is suitable for a final solution of the problem.
- We suppose that the relation $\Delta s_k(\Delta q)$, is known for any stage k, and for any state (s_k, q_k).
- For the tests for long term energy management methods, we need thus a relation that replaces $\Delta s_k(\Delta q)$, as the latter is unknown at this stage of work. For one time interval, we allow only controls with a constant speed. By the above relation, the charge consumption that corresponds to such a control can be computed. This gives a relation $\Delta q(\Delta s)$ resp. $\Delta q(\Delta s)$ that is used as if it would be the optimal solution for the short term energy management problem.
- The same that we constrain the control not to vary during an interval, we force the environmental conditions to be constant. This is, the brightness, the wind speed and the wind angle do not vary during a discretization interval.

For the rest of this work, we only treat the long term energy management problem. The energy gain per stage, Δq, is considered as the control of the system. In the two subsections below, we set some additional stronger work hypotheses that are used in the rest of this analysis of the long term energy management problem.

2.3.3 Elimination of Spatial Dependency of the Environment Variables

In this section, we introduce a work hypothesis. The goal is to eliminate the spatial dependency of the relation $\Delta s(\Delta q)$. This relation depends on s through the relevant environment variables B and (w, α_w), which are functions of t and s. They have a certain 'continuity' with respect to t and s.

s_t and t are closely related, in the sense that by heuristically estimating an average speed of the vehicle, we can predict an approximate location $\hat{s_t}$ in function of t. These considerations lead to the idea to neglect the spatial variation of the weather variables B, w and α_w, the solar beam incidence angle α_{sol} and the drive direction α_{road} in a first approximation, in order to simplify computations by optimization methods. Instead of a correct forecast, we will at time t a forecast at an estimated location $(t, \hat{s_t})$ for all locations. This simplification does not come without risks, and we can argue for and against it:

- We have analyzed that spatial variation of solar irradiance can change considerably the optimal solution of the deterministic problem. This effect can not be captured if the present simplification is applied.
- However, the approximated resolution of the deterministic problem we present is designed to be used as auxiliary function to approximate the solution of the stochastic problem. For this solution, the effect of spatial variation diminishes. In fact, the precision of long term weather forecast is quite low. Already with a horizon of one day, the weather forecast for two locations in the Australian desert that are close enough to each other can hardly be distinguished.
- On the other hand, the previous argument does not hold fully for the relative wind speed w_x, because the driving direction α_{road} is deterministically known in function of s. Given a forecast for the average wind direction for a region, we can precisely locate where w_x changes.

The second argument supports the present simplification as a possible option. However, if we use it, we must be conscious of the precision we loose.

To make this simplification more powerful, we can be tempted to use a model where gradient forces are also modeled as function of time. That is, the road inclines at $\hat{s}_t$ instead of s. This leads to a model where neither dynamics nor input power depends on the location.

Using this simplification, hills will be modeled at the wrong location. Still, for testing methods, this hypothesis simplifies implementation and validation, and the character of the problem might not change much. However, it would be imprudent to apply this simplification when results to be used in practice are computed. We thus insist to reintegrate it later into the model.

We apply the above simplification to approximate $\Delta s_k(\Delta q)$ for the tests presented in Chapter 3 and 4. Let us precise how they are applied:

- The solar incidence angle α_{sol} at stage k is computed as a function of $(k, \hat{s_k})$.
- We give directly the wind speed component $w_x(k)$ in the direction of the vehicle's motion, without considering the drive direction and the wind direction.
- We consider no hills: The road gradient $\theta = 0$.
- The brightness factor B is given as a function of k.

This is valid throughout the Chapters 3 and 4.

2.3.4 Elimination of Constraints from the Problem

The rules of the race preview some constraints in the race. Stops are forced stops at several locations during the race. Moreover, traffic lights and speed limitations have to be respected. These constraints are removed for method development. They must be reintegrated later to obtain solutions that are feasible in practice.

Chapter 3

The Deterministic Problem: Long Term Energy Management

In this chapter, the goal is to point out dynamic programming methods to solve the simplified long term energy management problem formulated in Chapter 2. Let us recall the simplifications discussed in Chapter 2 are applied:

- The weather is known in advance as deterministic function of t and s.
- We do not consider short term optimization of the system. During one time discretization interval, we suppose that the relation $\Delta s(\Delta q)$ is given. For tests, we estimate it as described in Chapter 2.
- For long term optimization, it is natural to eliminate the speed from the state space. The control of the system is Δq, or equivalently Δs. They are linked as described in Subsection 2.3.1.
- All environment variables are treated as a function of time only.

These restrictions are too strong that the algorithms developed here can be applied directly for solving the original problem. The present algorithms are motivated to be used for auxiliary purposes. Firstly, we want to start by applying a dynamic programming method to a simple example, to get some experience in dealing with the method. Secondly, the stochastic problem is to be approximated deterministically in Chapter 4. These approximations require to solve the deterministic problem repeatedly. This indicates that efficient methods for this purpose should be developed.

Meyer (2005) has solved the deterministic problem using weaker simplifications than ours. His methods can be applied for solving our problem. We present methods adapted to the simpler problem formulated here, with increased efficiency and precision, to be able to study the stochastic problem efficiently in the following.

In the deterministic model, the control can be optimized at the beginning of the race and does not have to be updated. The deterministic problem has thus to be solved only once, using as initial state the state at the beginning of the race, and as initial time the time at the beginning of the race. In this chapter, we present the algorithms for solving the deterministic problem that use always this initial state. However, this is just a convention. In Chapter 4, we will use the same algorithms up from other initial states.

In Section 3.1, we give the necessary theoretical introduction to the deterministic shortest path problem and its resolution by dynamic programming. In Section 3.2, the present problem is abstractly formulated in that framework. Two ways to concretize that are given in Section 3.3, and the corresponding results are discussed in Section 3.4.

3.1 Theoretical Background for the Deterministic Shortest Path Problem

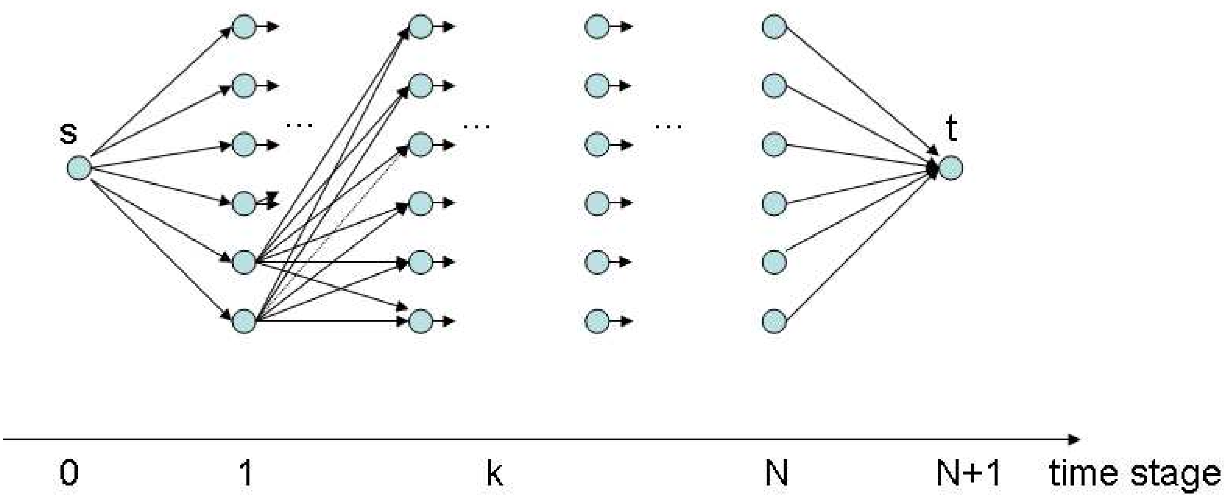


Figure 3.1: The deterministic shortest path problem can be represented by a graph. At each stage k, the state space is represented by a set of vertices. The possible transitions are represented by an arc. Each arc has an associated cost. The objective is to find the shortest path from s to t in terms of cost.

In this section, we will explain the notion of *deterministic shortest path problem*. This includes to introduce some notions from deterministic Dynamic Programming (DP) and to point out DP algorithms to find the optimal solution of the problem.

The introduction is held very briefly here. If you are new in DP, you might want to consult Bertsekas (1995a), a very readable introduction. The theory that follows in this section is a review of that book.

First we treat the finite time horizon deterministic shortest path problem. Consider a finite state space X that evolves over the discrete time stages $0, \dots N$. The transition to move from a state $x_k \in X$ at stage k to a state at stage $k+1$ depends on a control $u_k \in U_k(x_k)$. The state $x_{k+1} \in X$ where we arrive at time $k+1$ is described by a transition function $f_k(x_k, u_k)$. At each stage k, an additional cost $g_k(x_k, u_k)$ incurs. At the terminal stage N, a cost $g_N(x_N)$ may incur.

Let us now add a terminal state t to X. After the stage N, the system transits to t from any state, with a transition cost $g_N(x_N)$. Considering an initial state x_0 at stage 0, the system can be represented by the graph in Figure 3.1. The *finite time deterministic shortest path problem* is to move from state s at stage 0 to state t, with a minimal accumulated cost

$$\sum_{k=0}^{N-1} g_k(x_k, u_k) + g_N(x_N).$$

This problem can be solved by a *dynamic programming algorithm.* It proceeds by a recursion backward in time. In fact, let

$$J_N(x_N) = g_N(x_N), \tag{3.1}$$

$$J_k(x_k) = \min_{u_k \in U_k(x_k)} g_k(x_k, u_k) + J_{k+1}(f_k(x_k, u_k)) \text{ and} \tag{3.2}$$

$$u_k^* = \arg \min_{u_k \in U_k(x_k)} g_k(x_k, u_k) + J_{k+1}(f_k(x_k, u_k)). \tag{3.3}$$

$J_k(x_k)$ is called the *optimal cost-to-go* at the state x_k. The optimal solution of the finite time deterministic shortest path problem is given by $J_0(s)$. The optimal control is given by u_k^*, for all $k \in \{1, \dots N\}$.

The equation 3.2 is called *Bellman-Equation.* It is valid because all paths that are contained in a shortest path are shortest paths between their respective extremities. At each stage k, we must solve it for each state x_k, as far as a path from x_k to t exists.

A particularity which is of principal interest to us is that the finite time deterministic shortest path problem can be equivalently solved by a forward recursion. In fact, consider the system where each arc of the graph represented in Figure 3.1 is inverted. The cost of an arc is unchanged. Consider the finite time deterministic shortest path problem to find the shortest path from t to s. The same recursion as before solves this problem optimally. For the initial problem, the equations to be solved become

$$J_0(x_0) = 0, \tag{3.4}$$

$$J_{k+1}(x_{k+1}) = \min_{u_k \in \tilde{U}_k(x_{k+1})} g_k(\tilde{f}_k(x_{k+1}, u_k), u_k) + J_k(\tilde{f}_k(x_{k+1}, u_k)) \text{ and} \tag{3.5}$$

$$u_k^* = \arg \min_{u_k \in \tilde{U}_k(x_{k+1})} g_k(\tilde{f}_k(x_{k+1}, u_k), u_k) + J_k(\tilde{f}_k(x_{k+1}, u_k)), \tag{3.6}$$

where $\tilde{f}_k$ is the 'inverse' transition function that associates to x_{k+1} its unique predecessor by u_k, x_k, such that $f_k(x_k, u_k) = x_{k+1}$, and $\tilde{U}_k(x_{k+1})$ is the corresponding set of feasible controls.

We find the optimal solution $J_{N+1}(t)$ that is equivalently an optimal solution for the initial problem. The same is valid for the optimal control.

3.2 Problem Formulation as Deterministic Shortest Path Problem

In this section, we formulate the long term energy management problem in as a deterministic shortest path problem. In the following subsection, we consider a formulation that follows immediately from the problem model we consider. A more sophisticated formulation that allows to reduce the dimension of the state space without loosing accuracy is presented in Subsection 3.2.2.

3.2.1 Basic Model

Consider now the time stages $\{0, \dots N\}$ for the long term problem. Let the stage 0 correspond to the beginning of the race. Let the time interval between k and $k+1$ be the length of one interval of our time discretization. N is chosen large enough that we can be sure to arrive at the finish of the race before N. A possibility to choose N is the time we need to complete the race without using the battery.

Consider the state space consisting in some discretization points (s, q). The values of s discretize the interval $[0, s_f]$ from the beginning to the end of the race. The values of q discretize the interval $[0, q_{\text{full}}]$ from an empty to a full battery.

Consider the relation $\Delta s_k(\Delta q)$. In a general deterministic problem formulation, this relation depends additionally on s, but this dependence is excluded here by the hypothesis that the environmental variables can be described as function of time only.

This relation describes the transition function. In fact, let us for first neglect the difficulties that arise with the disretization of the state space. Let us give a state (s, q) such that we will not arrive at the final location s_f at the next transition. Then, the transition function is given by

$$f_k((s,q), \Delta q) = (s + \Delta s_k(\Delta q), q + \Delta q). \tag{3.7}$$

The set of feasible controls $U_k((s, q))$ is restricted by the battery constraints and by the speed constraint given by the maximal speed of the vehicle. The other constraints are neglected under the present hypotheses. The cost per transition, $g_k((s, q), \Delta q)$, is uniformly given by the time between two stages.

Let us now add one terminal state T to the state space. Consider a state-control pair $((s, q), \Delta q)$ where $s + \Delta s_k(\Delta q) \geq s_f$. In this case, we define

$$f_k((s,q), \Delta q) = T. \tag{3.8}$$

The cost g_k that corresponds to this transition is the proportion of the time interval that is need to arrive at s_f, using the control Δq.

Once the system is in the state T, it stays there until time N, and any further transition is cost free.

This determines a deterministic shortest path problem, which can be solved either by a forward or by a backward recursion. However, we will use this recursion. In the next subsection, we point out an equivalent, but simpler formulation.

3.2.2 Reduction of the State Space Dimension

The present hypotheses enable us to change formulation of the optimization problem, such that we have a one dimensional state space. In this subsection, we describe how this can be done.

Let us consider a fixed time stage k, and the set of controls U_{q_k} of the form $(\Delta q_0, \ldots, \Delta q_{k-1})$, such that the vehicle arrives with a charge q_k at stage k. To each $\mu \in U_{q_k}$, there is a corresponding location s_k that we reach applying μ. Meyer (2005) states that if μ is initial segment of an optimal control for the whole race, it must maximize the distance among all controls of U_{q_k}. Inversely, all controls in U_{q_k} that maximize s_k are interchangeable, because they arrive at the same state (s_k, q_k), at the same time.

We can thus, at any stage k, for any charge q_k, maximize the distance traveled so far and choose one control that maximizes the associated s_k. Considering only one such control for all q_k, we can be sure to consider an initial segment of an optimal control.

We reformulate now our deterministic shortest path problem such that we benefit from the above property:

- Consider the same time stages $\{0, \ldots N\}$ as before.

- Let us contract the above state space. q_0 of the new state space $\tilde{X}$ represents the class of all states (s, q_0) from the above state space.
- The transition from (s, q) to (s', q') in the above state space is replaced by a transition from q to q'.
- The cost that incurs at a transition is $-\Delta s_k(\Delta q)$. Instead of using a negative cost, we can also change the problem from a minimization to a maximization problem.

3.2.3 The Corresponding Forward Algorithm

Let us formulate the DP-forward algorithm that solves the above deterministic shortest path problem. At a stage k, such an algorithm associates an optimal s_k to each q_k. That way, we can iterate on k until the terminal state T is reached. At this stage, the value of k gives the minimal time needed for the whole race, the optimal solution of the problem. We are thus not constrained to fix in advance the time stage N. The recursion is iterated until T is reached.

We store, at each stage and for each q_k, the optimal Δq for the previous step. A backward recursion on these stored values allows to reconstruct the optimal control that leads to the optimal solution.

In this formulation, we have a one dimensional state space that is formed by values of the charge of the battery only. Abstractly, we have formulated an algorithmic method that gives the optimal solution. To get from here to a concrete algorithm, we have the following points at are open to be discussed:

- For each stage k and each state q_{k+1}, we have to solve an optimization problem. It is given by

  $$\max_{\Delta q}\{\Delta s_k(\Delta q) + s_k^*(q_{k+1} - \Delta q)\}, \tag{3.9}$$

  where $s_k^*(q)$ denotes the maximal location we can reach at stage k with charge q.

  The naive way to solve this problem is to check the value for each Δq of a discrete set that replaces the continuous set of all feasible Δq, and to select the maximizer. However, we can use the properties of the transition function to increase the performance of the algorithm.
- We have to choose a way to discretize the state space. According to this discretization, we have to choose a way to discretize the control space.

In the rest of this Chapter, we will discuss two methods of particularizing these points, and the results that we obtain when we test them.

3.3 Two Forward DP Algorithms

We discuss two variants of the above algorithm.

3.3.1 Solving the Discrete-Space Problem

In this Subsection, we particularize the DP forward method in Algorithm 1. Basically, we discretize the state space and the control space, and we solve the discrete problem, without using any particular properties of J_k, or of $\Delta s_k(\Delta q)$. This is opposed to the approach in Subsection 3.3.2.

Algorithm 1 Naive DP forward method

Input: An instance of the long term energy management problem
Output: s_k^* and μ_k^*, for all k.
$q_{min} = q_f$
$s_0^*(q_f) = 0$
for $k \in \{1, \dots, N\}$ **do**
  $min = \arg\min_i \{q_i \geq q_{min} + \min \tilde{U_k}\}$
  for $j \in \{min, \dots, f\}$ **do**
    $s_{k+1}^*(q_j) = \max_{\Delta q \in \tilde{U_k}} \{s_k^*(q_j - \Delta q) + \Delta s_k(\Delta q)\}$
    $\mu_k^*(q_j) = \arg\max_{\Delta q \in \tilde{U_k}} \{s_k^*(q_j - \Delta q) + \Delta s_k(\Delta q)\}$
  end for
  if $s_k^*(0) > s_f$ **then**
    BREAK
  end if
end for

Let the state space be a discrete set of values for the charge of the battery, $\{0 = q_0, \dots q_f = q_{\text{full}}\}$. Notice that opposed to the above notation, the index of q does not refer to a time stage here.

The set $\tilde{U_k}(q_i)$ is the set of all feasible controls to arrive at state q_i after transition k. This set has to satisfy the following constraints:

- Δq must be such that $q_i - \Delta q$ equals to an element of the discretized state space. This restriction includes already the full battery constraint and the empty battery constraint.
- In Section 2.3, we give the explicit relation of the energy flow from/to the battery in function of the speed. Here, we need the inverse relation to get our approximation of $\Delta s(\Delta q)$. Let us suppose that the vehicle drives with a minimal speed of 50 km/h, and a maximal speed of 130 km/h. These are reasonable assumptions that are confirmed not to be binding in the following. Let us choose a discretization of this speed interval. Now, for each time stage k, we can compte the value Δq for driving a constant speed v during the interval from k to $k+1$. The inverse relation gives the desired $\Delta s_k(\Delta q)$. In Figure 3.3, we have fitted a polynomial of degree 4 with the MATLAB function 'polyfit' to some data from this inverse relation. The residuals are small, thus we use this approximation for the relation $\Delta s_k(\Delta q)$. By the bound on the speed interval, we define the relation $\Delta s_k(\Delta q)$ on a limited interval. The value of feasible controls $U_k(q_k)$ are thus limited additionally to be in the range where $\Delta s_k(\Delta q)$ is defined.

Now, all elements are defined to establish the forward DP recursion solving the deterministic shortest path problem, given by Algorithm 1. Its output μ_k^* is the optimal Δq we associate to a state q at time stage $k+1$. A backward recursion, starting at the terminal state T, at the stage it is first reached, allows to establish the optimal control Δq for ny stage.

3.3.2 Using the Slope of the Cost-to-go

An obvious default of the above algorithm is that good properties of the cost-to-go function as well as the continuity of the state space are not used. As stated at the beginning of this Chapter, we want to solve the deterministic problem efficiently to be able to solve this problem many times repeatedly.

Below, we develop an necessary optimality condition for the maximization problem that has to be solved for all states, at any stage of the forward recursion in the algorithm. This leads to a DP forward algorithm that is potentially more efficient and more precise than Algorithm 1.

Bertsekas & Tsitsiklis (1996) give the analog of the Bellman equation for continuous state spaces with differentiable cost-to-go functions, called *Hamilton-Jacobi equation.* Using the notations from Section 3.1, it is given by

$$0 = \min_{u_k}(g_k(x_k, u_k) + \nabla_{x_k} J^*_{k+1}(x_k)^T f_k(x_k, u_k)). \tag{3.10}$$

Optimal control at stage k for is approximated by

$$u^*_k(x_k) = \arg\min_{u_k}(g_k(x_k, u_k) + \nabla_{x_k} J^*_{k+1}(x_k)^T f_k(x_k, u_k)). \tag{3.11}$$

Notice first that these equations are given for the case of a backward recursion in a minimization problem, and we would like to apply them to a forward recursion in a maximization problem. We have to proceed as follows to apply the equation:

- We are treating a maximization problem. In our problem formulation, the optimal cost-to-go function $J^*_{k+1}(x_k)$ is thus given by $-s^*_k(q_{k+1})$.
- We want to solve the equation for a forward recursion. We have seen in Section 3.1 how to transform the backward recursion for a deterministic shortest path problem into a forward recursion, and vice-versa. The directions of the arcs in the graph from Figure 3.1 have to be inverted. This is, the transition function $f_k(x_k, u_k)$ from Equation 3.11 must not be replaced by the transition from x_k to x_{k+1} in our case, but by the inverse transition $\tilde{f}_k$ from x_{k+1} to x_k. To adapt the Hamilton-Jacobi equation correctly to the case of a forward recursion, we must thus use $q_k = \tilde{f}_k(q_{k+1}, \Delta q) = q_{k+1} - \Delta q$ to express the transition at stage k.
- $g_k(x_k, u_k)$ is the cost that incurs at the transition, thus the negative distance traveled, $-\Delta s_k(\Delta q)$.

We have thus

$$\begin{aligned} u^*_k(q_{k+1}) &= \arg\min_{\Delta q}\{-\Delta s_k(\Delta q) + \nabla_{q_{k+1}}(-s^*_k(q_{k+1}))(q_{k+1} - \Delta q)\} && (3.12) \\ &= -\Delta s_k(\Delta q) - s^{*'}_k(q_{k+1})(q_{k+1} - \Delta q), && (3.13) \end{aligned}$$

where $s^{*'}_k$ denotes the derivative of s^*_k with respect to q.

Let us fix the value of q, as we do in the forward recursion algorithm, and discuss the Δq that minimizes the expression. We suppose that $\Delta s_k(\Delta q)$ is continuously differentiable. The first order optimality condition is obtained by setting the derivative of the expression with respect to Δq to 0. Denoting by $\Delta s'_k$ the derivative of Δs_k with respect to Δq, this condition is

$$\Delta s'_k(\Delta q) - s^{*'}_k(q) = 0. \tag{3.14}$$

3.3.3 An Algorithm using Hamilton-Jacobi Equation

In this subsection, we use the above property to deduct Algorithm 2 that works efficiently if we assume that both s^*_k and Δs_k are concavely decreasing functions, and that a feasible solution for Equation 3.14 exists for all q where s^*_k is defined.

Intuitively, the assumptions about concavity and decrement should not be far from being true:

- The hypothesis that Δs_k is concavely decreasing expresses that if we decrease the energy flow to the battery, the distance traveled during a time interval increases. The more we decrease the energy flow, the slower the distance increases.

  This is, if we consume more energy, the vehicle goes faster, but the more we increase consumption, the less speed increases.

- The assumption that s_k^* is concavely decreasing states the same for the distance accumulated under optimal control, given a fixed t: First, the more the battery is empty at a given time, the further the vehicle can be. Second, the more it is empty, the less the optimal solution increases.

We remark that by concavity of Δs_k, we have additionally that the solution of Equation 3.14 is unique. Furthermore, by concavity as well, the first order optimality condition is also sufficient, and that for each fixed q_k, the solution of Equation 3.14 maximizes the Hamilton-Jacobi equation of the problem.

Below, we give the missing ingredients to have the DP-forward Algorithm 2 that uses the Hamilton-Jacobi equation:

- An algorithm that solves Equation 3.14 efficiently for all q : To solve the equation once, we can use Newton's method, as Δs_k is concave. By the given hypotheses, the solution has a certain continuity with respect to q. If we solve the equation step by step for a decreasing q, and we use the former optimum as initial value for the next iteration, Newton's method should converge using very few iterations.

- A method to estimate $s_k^{*'}$: We use the scaled difference between the function values at two neighbor discretization points as estimation.

- We have assumed that a feasible solution of Equation 3.14 exists. This is not the case if the battery constraints are met. If the lower battery constraint is met, that is, if the Equation 3.14 has a solution such that $q_{k+1} + \Delta q < 0$, we let $\Delta q = -q_{k+1}$ instead. In the case where the upper battery constraint is met, let $\Delta q = q_{\text{full}} - q_{k+1}$. Other cases where no feasible solution exists are discussed in Section 3.4.

3.4 Test Results

3.4.1 Summary of the Results

We have tested both above methods. The naive Algorithm 1 gives valid results for all instances it has been tested. In the present form, it has the default that only a discrete set of charge consumptions is feasible, and that the optimal solution is thus not found precisely. Moreover, it is slower than Algorithm 2.

In turn, Algorithm 2 does not always give valid results. For some instances, it yields no valid solution if applied as described above. We propose a slight modification within this Section. The modified version is of better robustness, it performs faster than Algorithm 1 and slightly better solutions are found.

We keep thus back the following from the present Section for the whole optimization problem:

- If we need to solve the present problem very rapidly and precisely, we will invest in improving a method based on Algorithm 2, such that it performs robustly.

Algorithm 2 DP-Forward Algorithm using Hamilton-Jacobi

Input: An instance of the long term energy management problem
Output: s_k^* and μ_k^*, for all k.
Use algorithm 1 to initialize, and for the transition at k=1
for $k \in \{2, \dots, N\}$ **do**
  $min = \arg\min_i\{q_i \geq q_{min} + \min \tilde{U_k}\}$
  $\Delta q_{init} = 0$
  for $j \in \{f, \dots, min\}$ **do**
    Determine $\mu_k^*(q_j)$ by solving $s_k^{*'}(q_j) = \Delta s_k'(\Delta q.)$ Use Newton's method with initial value Δq_{init}.
    Force $\mu_k^*(q_j)$ to satisfy the battery constraints.
    $\Delta q_{init} = \mu_k^*(q_j)$.
  end for
  $s_k^*(q) = s_{k-1}^*(q - \mu_k^*(q)) + \Delta s(\mu_k(q))$. Interpolate s_{k-1}^* linearly for evaluation.
  if $s_k^*(q_0) > s_f$ **then**
    BREAK
  end if
end for

- If we only need robustness, but the computing time and precision is not important, we will use Algorithm 1.
- If need robustness and precision, but computing time is not important, we can either refine the discretization of Algorithm 1 or develop a method to interpolate results.

In the following, we describe in detail how these results are found.

3.4.2 Parameter Specifications for Our Tests

We have implemented both algorithms in MATLAB. We have set the parameters of the implementation to the following values:

- The time interval between two discretization points is chosen to be 100 s. This is very short for testing a long term optimization method. This causes that we have to optimize over many stages, which makes that the disadvantages of the methods appear more clearly.
- The discretization interval between two points for the charge of the battery is chosen to be 20000 J. This is also small. The batteries having a capacity of 5 kWh, this step corresponds to a grid of 901 points.
- In these first test, brightness and wind are varied discontinuously, and only at the end of time intervals.
- For testing the methods, we do not necessarily have to optimize over the whole distance of the race. We have chosen to maximize the distance that can be traveled during 6 hours.

3.4.3 Adaption Needed for Algorithm 2

Firstly, let us analyze Figure 3.3. The plots represent the graph of the set of functions s_k^*, one plot computed by Algorithm 1 and the other by 2, for a particular instance of the problem. The goal is to see that Algorithm 2 does not give a valid solution for this instance.

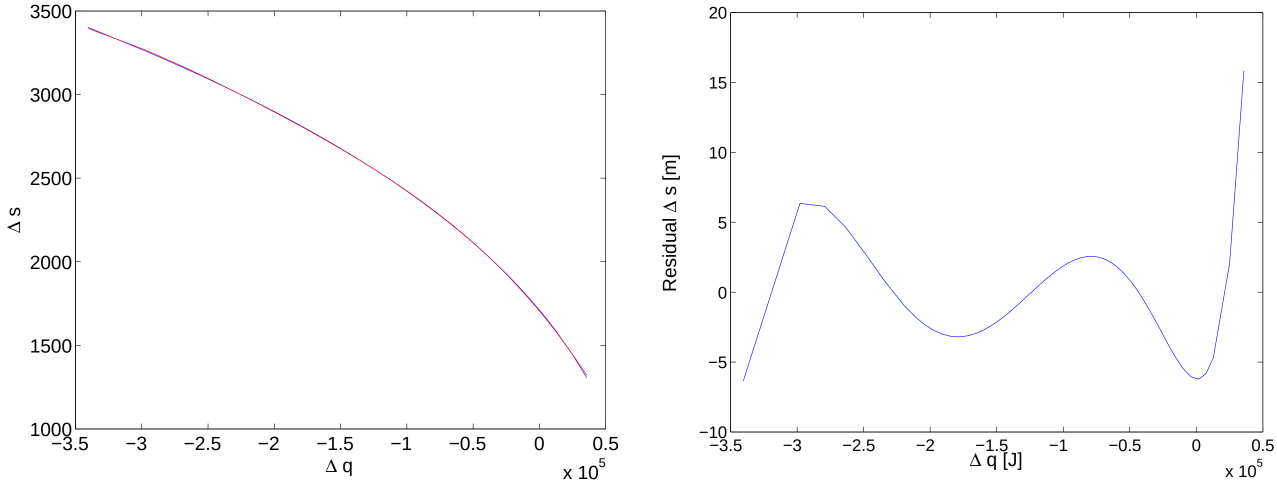


Figure 3.2: The plot on the left shows a polynomial least square fit to values computed for the function $\Delta s_t(\Delta q)$, with parameters $t = 3600$, $s = 0$, $B = 0.8$ and $w_x = v + 10$. The blue graph maps data points, the red one the corresponding values of the polynomial of degree 4. This seems to approximate data nicely in the investigated cases, as in the one mapped here. On the right, the corresponding residuals are plotted. For the most relevant values of Δq close to the center of the intervals, they are close to 0. In fact, they are in the scale of 0.1 % of the distance traveled during the time interval.

From the visual impression, the function s_k^*, describing the maximal location reached at stage k in function of the battery charge, is effectively decreasing and concave if it is computed by Algorithm 1. However, if we compute the same by Algorithm 2, for values of k larger than some value, s_k^* oscillates. The more k increases the more these oscillations are important. Notice as well that once for a certain k, in a certain region of q an oscillation is visible, it is reproduced in the graph of s_{k+1}^*.

Explication of the Lack of Robustness

Let us justify that it is inherent to the present method that once there is a perturbation it is reproduced. Consider that we have a small error in the value of $s_k^*(q)$. In this situation, we estimate the derivative $s_k^{*'}(q')$ for some neighbor q' of q by a difference between $s_k^*(q)$ and some other neighbor $s_k^*(q'')$. However, $s_k^*(q'')$ and $s_k^*(q)$ are close to each other by continuity of s_k^* and the difference is small, and divided by the small $|q'' - q|$. The error on the estimate of $s_k^{*'}(q)$ is much more important than the one on $s_k^*(q)$ that caused it. However, we determine the optimal control using the estimate of $s_k^{*'}(q)$ in order to solve Equation 3.14. A perturbed $s_k^{*'}(q)$ causes a perturbed control, which causes that we loose control over the error on s_{k+1}^*.

Moreover, initial small perturbation can be influenced by various factors. Namely, we cite that Newton's method is approximative, and that the derivative $s_k^{*'}(q)$ is estimated from a discrete set of points.

These arguments underline that the present phenomenon is not surprising.

Improvements

We consider only a quick and simple improvement of the method. The reason for not working out such improvements in detail is that we want to focus our ulterior work on more general issues of

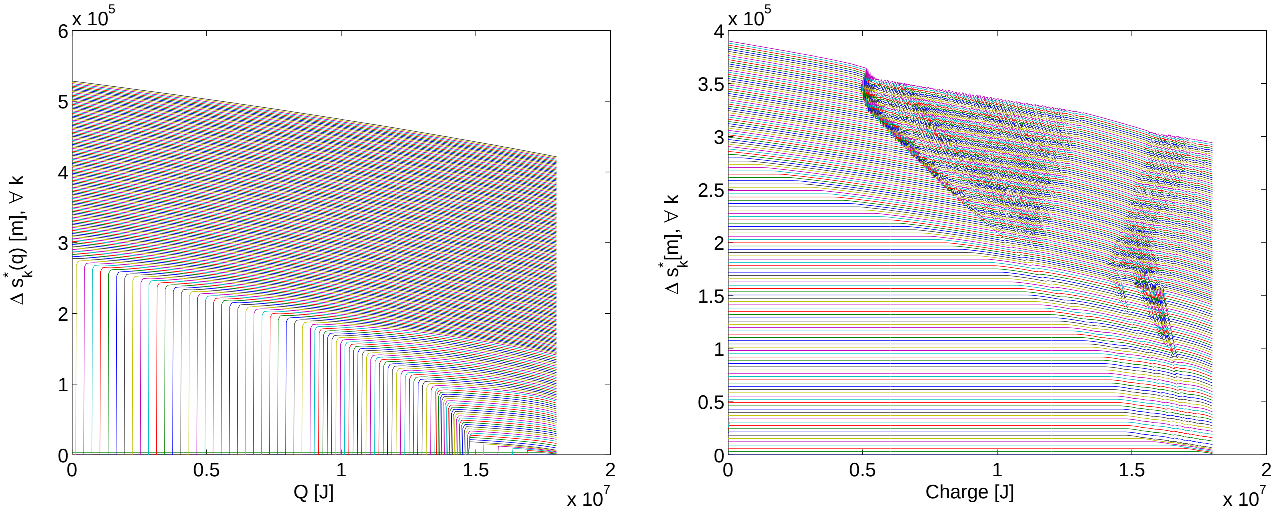


Figure 3.3: This plot on the left shows the values of $s_k^*(q)$ computed by Algorithm 1 for each k, for an instance of the problem. The data points for one fixed time point k are connected by a line. For q and k where the function does not exist, the values are set to 0. The fact that $s_k^*(q) < s_{k+1}(q)$ yields that the plotted lines are ordered 'bottom up' with increasing k. Notice that the function s_k^* seems to be effectively concavely decreasing, as we suppose it for Algorithm 2. On the right, $s_k^*(q)$ is computed by Algorithm 2, for the same instance. Once the solution has small perturbations, it begins to oscillate. This algorithm needs to be changed such that its solution is valid more likely.

the optimization problem, and improve this method only in the case it is needed. However, the present improvement of Algorithm 2 shows that the robustness of this method can be improved already with small adaption, and that the approach can be powerful.

Instead of using at each stage Algorithm 2 to find the optimal control, we use Algorithm 1 at some stages. For the variant where Algorithm 1 is used at the initial stages 1,2,3,4,5, and at the stages 10, 20, 30, ..., and Algorithm 2 at all the other stages, we have found no instance of the problem where the above phenomenon has become visible. However, this has not been systematically tested.

The s_k^* computed by the new method, for the same instance where we have the above oscillations with the above method, are plotted in Figure 3.4.

3.4.4 Analysis of the Computed Solution

In this subsection, we compare the controls of the system that we obtain using both Algorithm 1 and the improved version of Algorithm 2 on a particular instance of the problem. In Figure 3.5, the computed charge gain per stage and the distance traveled per stage are plotted for each one of the algorithms. The details of this instance are summarized in Table 3.1.

Let us analyze the result in two stages. First, we consider the particularities that distinguish the outcome of the two methods. Second, we consider the nature of the solution itself that has been computed.

Differences of the Solutions

The principal difference between the two computed solutions lies in the fact that Algorithm 1 allows only a discrete set of controls. In fact, the control space is discretized. Two grid points

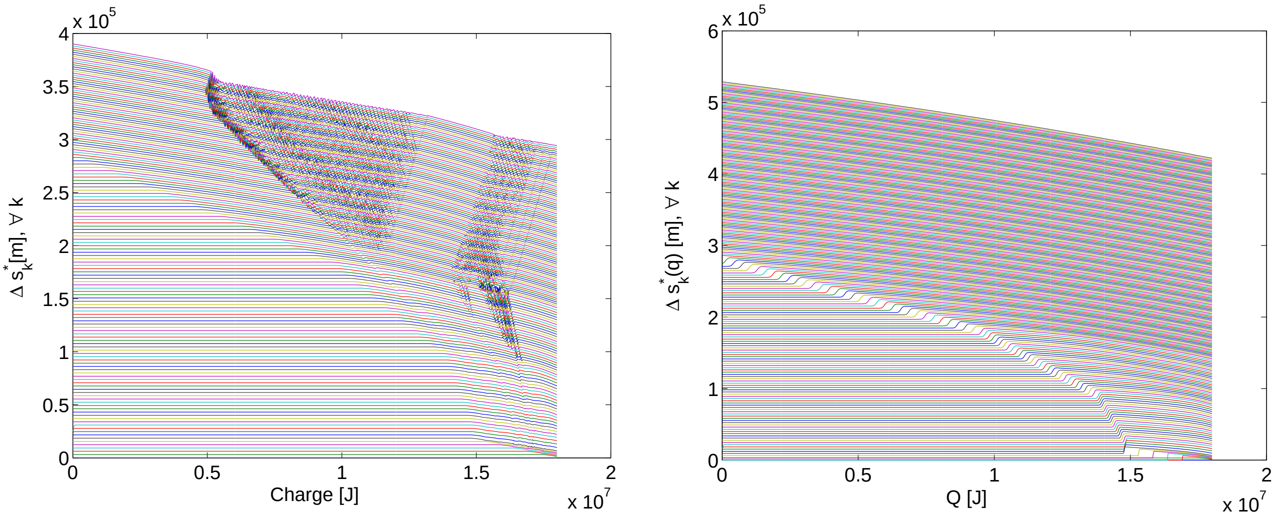


Figure 3.4: These plots show the values of $s_k^*(q)$, for a given instance of the problem, for each k. The solution on the left is computed by Algorithm 2, the solution on the right by its modified version. The solution of the improved version is visibly less perturbed than the solution of Algorithm 2. As the solution computed with Algorithm 1, $s_k^*(q)$ appears to be decreasing and concave, for each k.

Stages	Front Wind Speed	Brightness Factor
0-49	+5	0.8
49-96	+10	0.8
97-148	-10	0.8
149-178	+10	0.8
178-216	0	0.8

Table 3.1: Details of the problem instance thats results are plotted in Figure 3.5.

have a distance of 20000 J. At each stage k, the solution for the charge gain per stage Δq can only take values such that $q_k + \Delta q_k$ is a grid point again. Algorithm 2 has the same restriction, but only each tenth stage, when Algorithm 1 is used to compute the optimal control. We see in both plots that the solution of Algorithm 1 approximates the solution of Algorithm 2. The fact that the independent methods to compute the optimal solution have trajectories that follow each other closely is an indication that the computed solution is close to the optimal one.

Analysis of the Computed Solutions

The computed solutions plotted in Figure 3.5 are as expected. The computed control trajectories have the following properties:

- At all stages where the front wind increases discontinuously, the computed speed diminishes discontinuously.
- During the day, the apparent solar irradiance increases until noon, then diminishes again. If other environmental factors are constant and a constant speed is maintained, the charge gain increases until noon, then decreases. We now that under these circumstances, the optimal

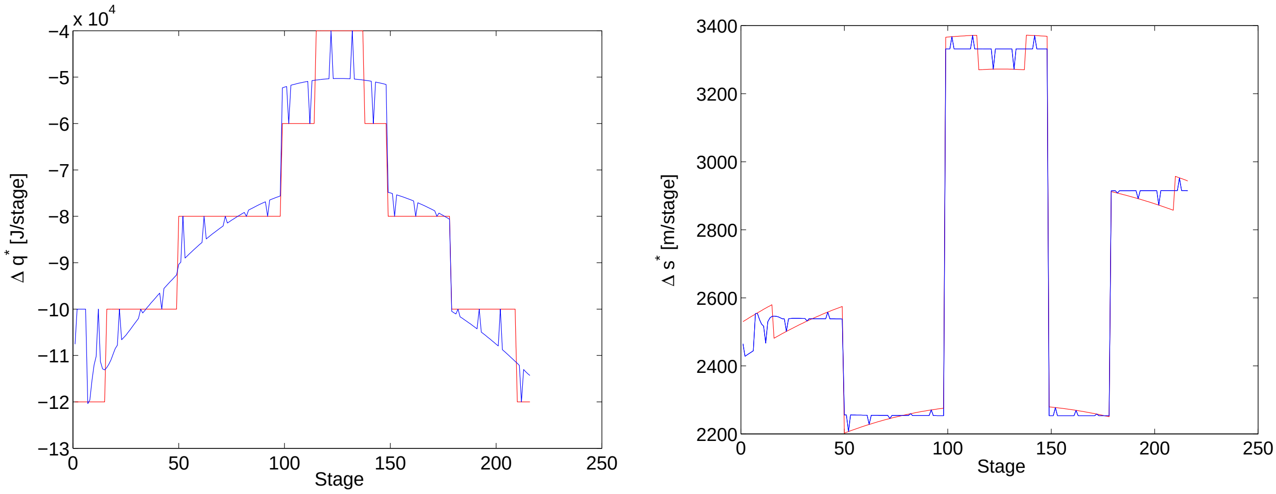


Figure 3.5: On the left, the charge gain trajectories computed by the Algorithms 1 and 2 for some instance of the problem are given in function of the stage. Approximately, both algorithms give the same trajectory. However the control space for Algorithm 1 is discrete, in the present version. This gives the graph mapped in red. For algorithm 2, the control space is continuous. At any stage when the Algorithm 1 is used, the algorithm optimizes only among a discrete set of controls. At these stages, the computed control is the same as for Algorithm 2. The right figure shows the computed speed trajectories that correspond to the left charge trajectory. Again, the red trajectory is the result Algorithm 1. Even if a constant speed should be maintained, the algorithm can not adapt charge consumption continuously.

strategy is close to maintaining a constant speed, and rather the charge gain than the speed is adapted. This can be observed in intervals where the apparent front wind speed is unchanged.

- With respect to the above point, we can observe that our model simplifications lead to the situation that during these intervals, it is optimal to maintain a constant speed. We conclude this from the fact that the charge gain values computed by Algorithm 2 are adapted at each stage, such that a constant speed is maintained. The fact that the speed values play no direct role in the computations by Algorithm 2, and that the trajectory of charge gain is adapted such that the speed values stay exactly constant, is an indication that Algorithm 2 performs very precisely, once is robust.

- The two approaches give average speeds that are very close one to each other. The solution of Algorithm 1 has an average speed of 26.68 $\frac{\mathrm{m}}{\mathrm{s}}$, the one of Algorithm 2 is at 26.70 $\frac{\mathrm{m}}{\mathrm{s}}$, for this instance. The fact that these independent computations with two different methods lead to results that are so close is an indication that this solution is close to the optimum.

These considerations lead to the conclusion that Algorithm 2 gives very precise results for instances where it performs robustly. In turn, Algorithm 1 is robust and gives correct results, but the precision is limited by the discretization of the state space.

3.4.5 Computing Time

We have evaluated the computing time consumed by each one of the methods for some instance of the problem. The MATLAB program has been run on a machine with a Pentium 4 2.2 GHz processor, and with 512 MB RAM. The results are graphically represented in Figure 3.6.

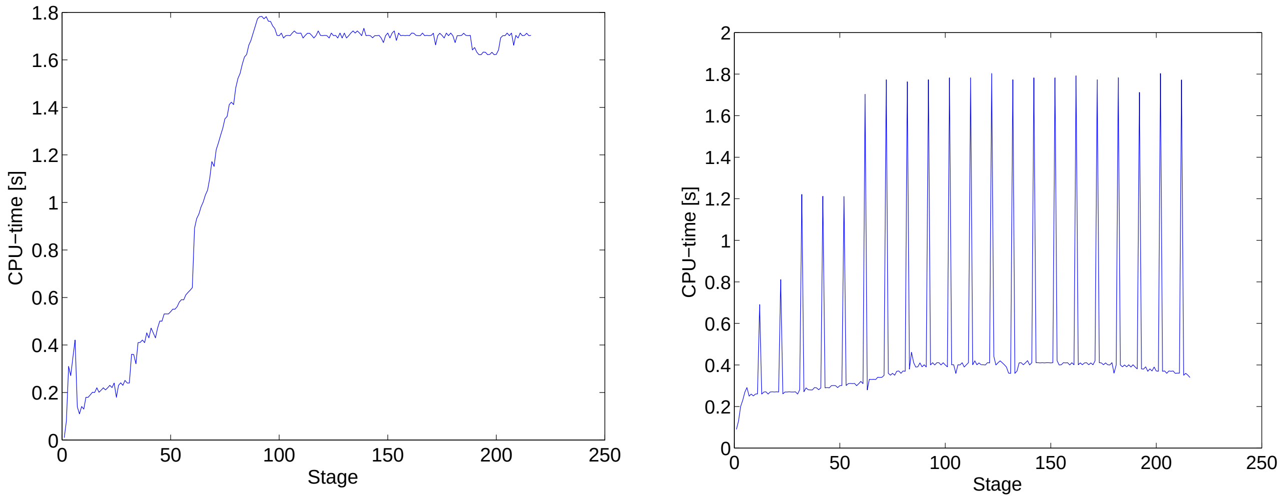


Figure 3.6: In these figure, we compare the computing time per stage for both Algorithm 1 and 2. The figure on the left maps the values for Algorithm 1, the the figure on the right is the analogue for Algorithm 2.

Analyzing the graphs leads to the following results:

- Computing time per stage for Algorithm 1 increases until a certain stage is reached. Then it is approximately constant. This is because for the first stages, s_k^* is not defined for all q. In fact, it is defined for states inside the interval $[q_{\text{min}}, q_{\text{full}}]$, and $q_{\text{min}} > 0$. Therefore, the maximization of s_k^* at these stages does not have to be done for all states. Once q_{min} reaches zero, the number of stages where s_k^* has to be computed does not change anymore.

- The peaks on the plot for Algorithm 2 are at the stages where Algorithm 1 is used. At these stages, it reaches approximately the same values as Algorithm 1.

- Let us neglect these values for this analysis. The increment of the computing time in function of k is also visible here in the same range of values, but less drastically than for Algorithm 1. This indicates that the number of states where the solution has to be computed does less affect its performance than the one of Algorithm 1. In fact, once an optimal Δq for some states has been computed, the algorithm can determine the solution of neighbor states very quickly.

- For stages where both algorithms reach a stable computing per stage, Algorithm 2 is approximately 6 times faster than Algorithm 1, if we use the methods with the current parameters. This difference could increase if the resolution of the state space is increased. In fact, computing time for the Algorithm 2 depends less on the number of states where the optimal Δq has to be found, for the reason discussed in the point above.

Validation of the Solution

In this last point, we give another indication that the present results are valid solutions with respect to the current model, and that they are close to optimality.

First, we have simulated the 6 hours journey by controlling the vehicle according to the optimal speed at each stage, as generated by both Algorithm 1 and Algorithm 2. The optimal charge profile that corresponds to this speed profile in the approximation used by the Algorithms is such that the

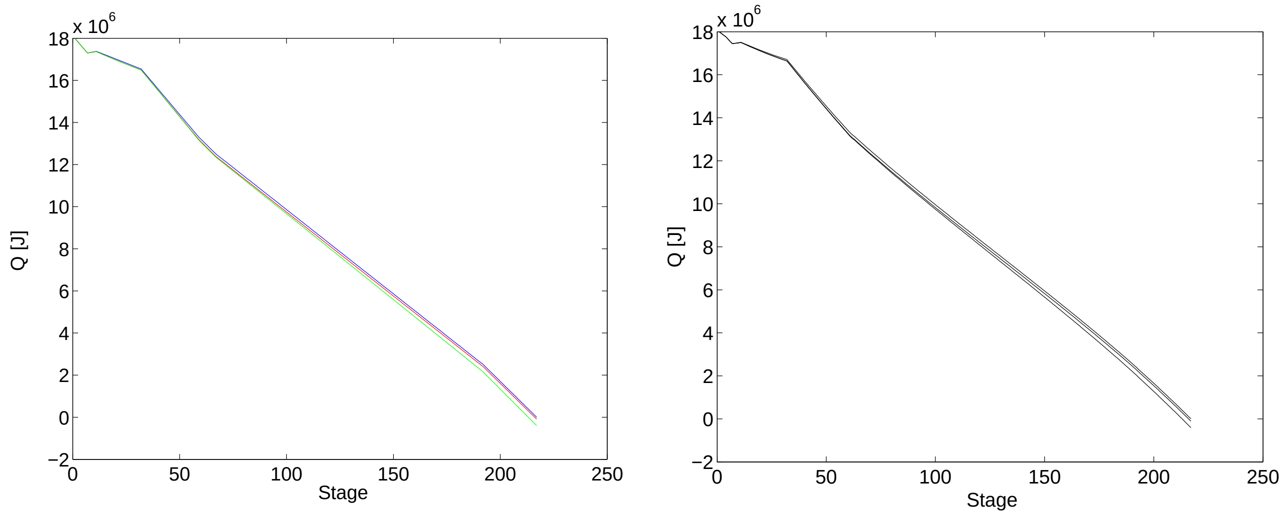


Figure 3.7: This Figure serves to analyze the battery charge values that we obtain by simulating the race by using the optimal speed computed for each stage. The figure on the left is a plot for values from some instances of Algorithm 1, the Figure on the right is the same for Algorithm 2. For both figures, the blue trajectory is the charge trajectory that in the approximations made by the algorithms corresponds to the gives speed trajectories. The red trajectory is the outcome of the simulation by the present model. This model uses an estimated location to determine solar irradiance. The green trajectory is the outcome from the simulation that uses a model where solar irradiance is computed at the true location of the vehicle.

battery is empty after the journey. If the charge profile computed by the algorithm corresponds to the charge profile that is issued by our simulation, this is an indication that the assumptions made by the algorithm are close to the model. This in turn indicates that the control is close to the optimal control among all controls that that are possible solutions an algorithm can furnish. Especially, for Algorithm 1, if all approximations would correspond to the model, the outcome would be exactly the optimal solution of the problem.

Let us look at Figure 3.7. We compare the approximative trajectory that is used by the algorithms, plotted in blue, to the trajectory generated from the simulation on the corresponding model, plotted in blue.

The trajectories correspond approximately. At the end of the race, the approximate difference between the values is 100000 J for Algorithm 2, and 10000 J for Algorithm 1. This is, in the first case, less than 1 % of the whole battery charge, in the latter case less than 0.1 %. This is in the scale of the approximations made by approximating Δs as a polynomial in $Deltaq$. This supports the validity of the results.

Chapter 4

The Stochastic Problem: Long Term Energy Management

In this Chapter, we introduce uncertainty into the model. Our approach is complementary to the one followed by Meyer (2005). He introduces a realistic model for gradient driving forces, the legal speed constraints that are to be respected during the race. However, he neglects the fact that we are in general not able to precisely predict the relevant weather variables for the duration of the race. He shows that under the assumption that a prefect deterministic weather forecast is available, the long term energy management problem he formulates can be solved quite efficiently.

For a first study of the effects caused by uncertainty in our knowledge about future weather, we keep the model of the car, the dynamics and the environment as simple as formulated it in Chapter 3. Our purpose is to study the following questions that arise with respect to the long term energy management problem:

- What should we do if the weather forecast changes?
- How can we know immediately the optimal speed if the weather changes suddenly?
- How to deal with the fact that information about wind predictions and weather predictions might be correlated.
- How to take into account the fact that we know that information gains in reliability when approaching a future time point? More precisely, this question expresses the following: Consider the time points $t_0 < t_1 < t_2$. How to take into account at time t_0 that the decisions at time t_1 will be based on a weather forecast for t_2 that is systematically more precise than the one we have at t_0?
- As last, we have to consider that weather forecast for different time and location points may be interdependent? This is, if we assume at (t_0, s_0) that the weather at (t_1, s_1) is bad, the weather forecast for (t_2, s_2) is different than if we assume that the weather at time (t_1, s_1) is good.

We organize this discussion as follows. In Section 4.1 we explain briefly our notion of weather forecast. However, studying weather forecast in depth is beyond the scope of this work. We describe in Section 4.2 the parametric model of weather information that we assume to be valid for the rest of this Chapter. We hope that by using this model, the properties of the optimization problem are similar to the ones arising when using a more accurate model. In Section 4.3, we formulate the optimization problem with respect to this model. In Section 4.4, we introduce stochastic DP, and in Section 4.5, we give an algorithm that solves the optimization problem for a fixed information at a fixed initial state and time. This algorithm is most likely inaccessible for computation, especially because without further considerations, it has to be executed many times during the race, in order to update the optimal solution. We introduce thus approximation schemes at different levels of accuracy and complexity in Section 4.6. Each of them uses deterministic approximation to solve some subproblem, where the algorithms from Chapter 3 can be used. In Section 4.7, some initial considerations for updating the solution with respect to changing information are done. This leads to Section 4.8, where the outcome of a deterministic approximation is compared to the one of an algorithm where stochastic optimization is done.

4.1 Weather Forecast

In this Section, we explain firstly what we know about weather forecast. This knowledge is based on Linder (Private Communication). However, in this domain, a considerable amount of research is to be done until we will be ready to participate at WSC 2007.

Secondly, we explain how we use in this work the knowledge we currently have in order to test optimization methods.

As far as we know, there are two kind of information available that can be used to predict the weather variables B, w and α_w :

- The first are weather forecasts from *local* or *global atmospheric models.* These are models that take into account measurements of various atmospheric variables at various locations and that allow to compute a forecast of those variables for a period of some days, with decreasing precision with evolving time. To quantify uncertainty, the initial conditions of the models are slightly varied, and the predicted trajectories change.

- The second are results from data analysis from historical data. From the cumulative data set for one factor such as B for example, a cumulative distribution for such a factor can be derived. The characteristic frequencies of the process can as well be described by time series analysis.

Pudney (2000) proposes to use this information in two phases. For the long term energy management problem, he suggests to use the model forecast data for the first one or two days, as if it would represent a deterministic knowledge of the weather. For the resting days, he suggests to use the data predicted by a stochastic process estimated by time series analysis of climate variables.

Within the present Chapter, we will only use a stochastic model to predict the brightness factor B. Wind speed and wind direction are set constantly to 0, for simplicity. We assume that this introduces uncertainty in an appropriate way into the model, in order to test optimization methods.

Our model for the brightness factor is only based on time series analysis, for the following reasons:

- We have climate data analysis results from Australia from Boland (1995), but we have not studied models for the accuracy of an Australian weather forecast so far. Studying those would be beyond the scope of this work. Therefore, we are not able to formulate a stochastic model based on weather forecast information at this stage of work.

- The weather uncertainty is biggest for a long term forecast. Therefore we assume that considering an accurate model for long term forecasts is most important to point out the differences between a stochastic and a deterministic information model. However, for long term forecasts, Pudney (2000) suggests to choose climate data distributions in any case. To choose model forecasts would therefore be hardly much more accurate. We hope thus that, for three and four day forecasts, such a model has a similar 'degree' of uncertainty as we will meet during the race, and that we obtain reasonable results when developing methods with this model.

4.2 A Parametric Brightness Factor Information Model

In this section, we present a stochastic model that describes the information we have about future evolution of the brightness factor B. It is based on climate data analysis from Boland (1995). We use it here for testing optimization methods, assuming that it represents uncertainty of our knowledge about the future in an way that tests for the performance of the methods are accurate.

4.2.1 Prediction of the Daily Average Brightness Factor at a Fixed Location

As a first step, we consider a fixed location s. We take a look at the time series of the daily integrated solar radiation on a horizontal surface, H. This quantity is proportional to the *daily average brightness factor,* a quantity of importance to our model. We denote it by D, throughout this report. In fact, D is defined by Duffie & Beckman (1980) as the ratio between H and the daily integrated clear sky irradiation.

Boland (1995) identifies an annually steady periodic part for H that he subtracts from the time series to analyze the series of residuals. He shows that, once a given non-linear transformation is applied on the series of the residuals, it follows a first order autoregressive process. He derives the following method to generate data that is 'statistically indistinguishable from the original time series.' We will use this method, given right below, for predicting a distribution for future H, respectively for the proportional future D.

- Let

$$a_j \sim \mathcal{N}(0, 1 - \beta^2),$$

$$z_0 = a_0,$$

$$z_j = \beta z_{j-1} + a_j,$$

  where $\mathcal{N}$ stands for the Gaussian probability law. This random process for z_j is the first order autoregressive process described in Boland (1995). β is a location specific coefficient. Boland (1995) gives the method to estimate it.

- For all j, compute

$$F(z_j) = \frac{1}{2}\left(1 \pm \sqrt{1 - e^{\frac{-2z_j^2}{\pi}}}\right). \tag{4.1}$$

  The plus sign is for values $z_j > 0$, and the minus sign is for values $z_j < 0$.

- Consider the cumulative empirical distribution G of the residuals of daily solar radiation, after subtracting the annually steady periodic part. Let x_j be the values that solve $G(x_j) = F(z_j)$.

- Adding the steady periodic part to x_j, we obtain the requested series for H.

In this work, we need to generate such data series that are similar to the real ones we will meet in Australia to test our algorithms. However, not all data we need is yet available to us. Therefore, we have solved these problems provisorily as follows:

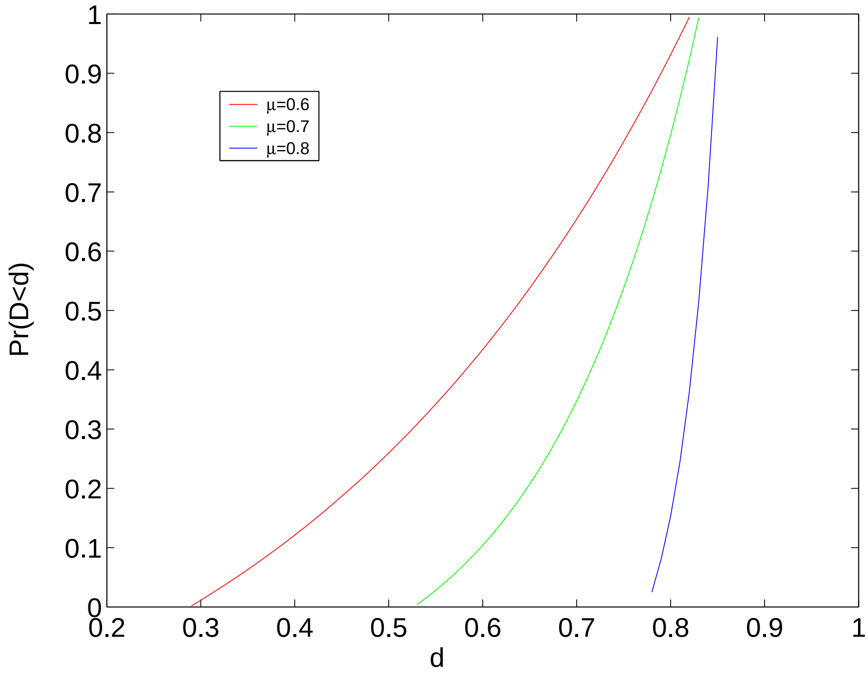


Figure 4.1: These are plots of values from the universal distribution function for D, as given by Duffie & Beckman (1980). The parameter μ is the long term average of D, estimated from of the location- and season-specific data. The values are computed using an approximate explicit formula. We use this distribution function instead of a location specific empirical distribution of D. The same, we use it as error distribution for the half hourly brightness factor, as described in Subsection 4.2.2.

- For the autoregressive coefficient, Boland (1995) gives values for some locations close to the race track. We use them for the needed locations without further considerations about their validity.

- Instead of using the empirical distribution of H, we can use the parametric universal distribution function for D that is given by Duffie & Beckman (1980). Its only parameter μ is the long term average of the season-specific average D. This data is given in *WORLD SOLAR CHALLENGE, AUSTRALIA, SEPTEMBER - OCTOBER 2005, Climate Information Package* (2005) for relevant locations. For some values of this parameter, we have plotted this function in Figure 4.1.

  According to Duffie & Beckman (1980), some authors claim this universal distribution to be valid for all locations on the world, while others question it. To assume that it is approximately valid is sufficient for our purposes.

Consider now the initial value x_0 for H. We get z_0 by applying the inverse of transformation 4.1. We generate z_j starting from z_0 by the autoregressive process, and get x_j from it.

If we generate D with the above method, we can use its distribution conditional to the value of the current day as a stochastic weather forecast. In fact, we can generate an independent sample of D up from the given initial value that is distributed according to that distribution.

This means that we have a first element that we can use to simulate the information situation as it is in the race. At day 1, we have a stochastic weather forecast for a given location, for the subsequent days. We use this forecast to control the vehicle at day 1. Then, we generate the weather at day 2 according to the above method. This can be iterated until the race ends.

4.2.2 Prediction of the Half Hourly Average Brightness Factor at a Fixed Location

A first improvement of the above model concerns its temporal resolution. The distribution of brightness over the day matters for our optimization problem. The best temporal resolution of brightness data measurements that is commonly available are half hourly integrated values of solar radiation. We denote them by I.

Let us define the *half hourly average brightness factor* in analogy to the daily average brightness factor, as the ratio between the half hourly clear sky radiation and the apparent solar radiation I. We denote it by E.

Boland (1995) points out that the conditionally expected value of I, knowing D for the corresponding day, can be found by integrating DI_{horiz} over the concerned half hour. In this terminology, we have that the conditionally expected E, knowing D, is equal to D. To this conditionally expected value, an independent error is added to obtain E. However, Boland (1995) gives no distribution of the error.

In turn, Duffie & Beckman (1980) affirms that the cumulative distribution of E can be assumed to be distributed as well according to the function plotted in Figure 4.1. The parameter μ is now estimated as location- and season-specific average E.

We assume here that we can equally use this function as conditional distribution of E, knowing $D = d$, with $\mu = d$. This assumption is not claimed to be accurate, but again, we hope that it produces data series that are 'similar' to the real ones.

4.2.3 Varying Location

So far, we have considered weather evolution for one given location. However, the vehicle moves in space, an we have to consider effects of varying location. We will use the following simple model:

- To each day i, we associate one location s_i. This location is chosen in a region where we estimate to travel during day i.
- For any time stage on day i, the brightness is predicted and also generated by the transformed autoregressive process for the location s_i.
- The processes for the different locations are independent.

This choice can be justified as follows:

- We have chosen a model simplification where all environmental variables only depend on t, and where the dependence on s is expressed through the dependence of s on t.
- The distance traveled in one day is around 800 km. Weather seems to have a certain continuity not only in time, but as well in space. Without having studied this in detail, we will use one

forecast that gives the average daily brightness factor at all locations visited during a day. For this, the forecast from an 'average' location over all locations visited that day is used. This expresses the continuity in space and time during a day.

- The night is a natural discontinuity of the process. This is used by the model. For each day, we choose thus an average location for which an independent random process as described above takes place, which gives an average brightness factor for the day at a location that varies with the motion of the vehicle.
- The half hourly values are generated as if the location would be constant.
- This independence assumption does not restrict us in considering the prediction at the future location. It is thus a conditional independence assumption.

While the independence assumption for time-location couple met in the evening of day k and in the morning of day $k+1$ limits the validity of the model, this is likely not the case for further distant couples, because the location gap increases rapidly, and also the time gap becomes bigger. For the cumulated average brightness factor that we expect to meet during a day, this independence assumption might be quite likely close to accuracy.

4.2.4 Model for the Growth of Information

An accurate model of weather forecast information is not static. Let us consider the weather forecast that we obtain at time t_0, for a fixed location s. We predict the weather at time t_2 with the above model. This forecast contains random parameters for the integrated daily radiation for the days between t_0 and t_2. The formulation of our optimization problem depends on when we think these parameters become known. If we assume that the daily average radiation D for day i becomes known at the evening of day i, the model is different than if we assume that a sufficiently good prediction for it will be available in the morning of the day i, such that we can suppose it to be known for optimizing control in the morning.

We make the following assumptions about the time when D and E become known:

- D the becomes known before the corresponding race day begins.
- E becomes known before the corresponding half hour begins.

In analogy to the assumption we have made for long term optimization for the deterministic case, we assume that the brightness factor is constant at the value of E during the corresponding 30 minutes.

We consider thus the following brightness information model for a fixed location:

- Consider a system with time stages $k \in \{0, 1 \dots N\}$. The time stage 0 corresponds to the current instant, t_0. The time difference between two neighbor stages is 30 minutes. This system expands over the days $i \in \{0, \dots N_d\}$. The day that contains the time stage k is denoted by i_k. To each day i, there is an associated location s_i.
- We have N_d independent transformed autoregressive processes to consider, the processes that correspond to the locations $s_i, i = 1, \dots N_d$. For each one of these processes, the current day value of D is assumed to be known, and is denoted by D_0^i. The random variable for the daily average brightness factor for the process i, j days after the current day, is denoted by D_j^i.

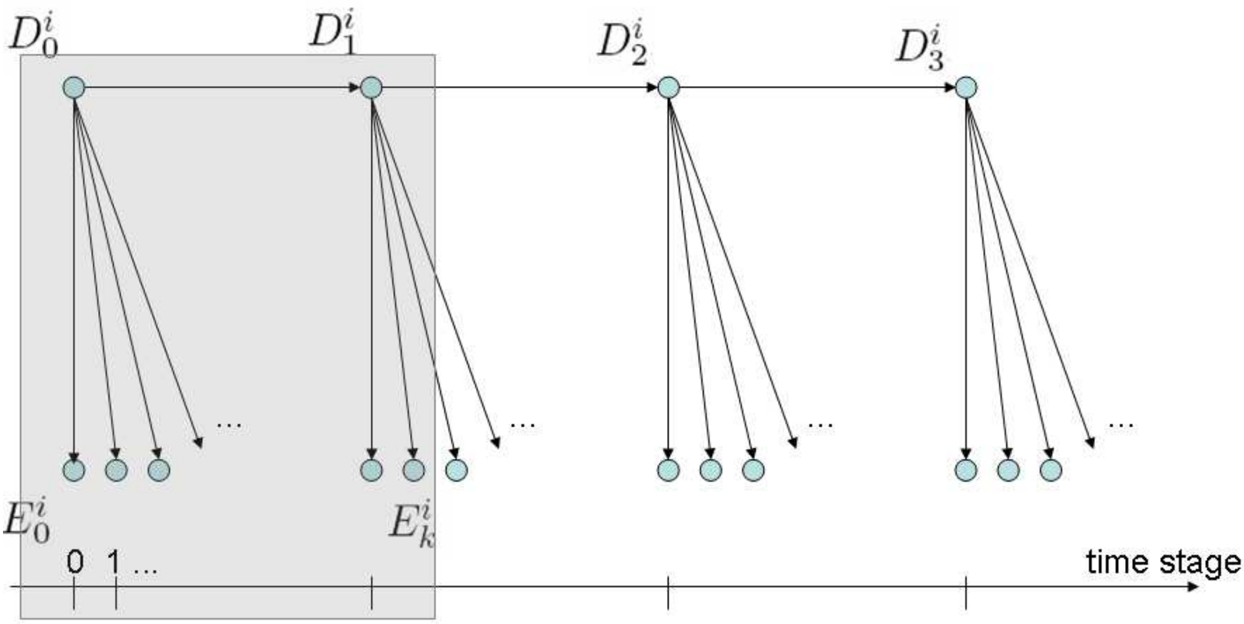


Figure 4.2: Each node of this graph represents a random variable that is relevant to describe the evolution of the brightness factor at a fixed location i. The set of all variables relevant to our problem formulation is a set of N_d copies of this graph. The nodes represented in the above row represent daily average brightness factors. In the row below, half hourly average brightness factors are represented. The position over the time axis where a node is placed gives the time point when the corresponding random variable becomes known. For example, the nodes in the grey font rectangle are the ones that are known at time stage k. If the arcs of the graph are regarded as non oriented edge, the resulting non oriented graph gives the dependencies between two variables. They are dependent if and only if they are connected. Conditionally to knowing some of the variables, the graph of dependence becomes the subgraph generated by suppressing the known variables. The direction of the arc gives the paths that we use to explain the variables.

Its distribution is defined by the initial value D_0^i and the autoregressive process specific to location i. Notice that by our model assumptions, the brightness factor is treated as if the vehicle would stay at location s_i during the whole day i. The prediction for the daily average brightness factor that the vehicle will meet on day i is thus given by the random variable D_i^i.

- The half hourly average brightness factors for the time interval after stage k, at location i is denoted by E_k^i. Its distribution depends on $D_{d_k}^i$, where d_k is the day that contains the stage k. Knowing the value of D_j^i, the variables E_l^i for stages l on day j are conditionally independent. This information situation is visualized in Figure 4.2.
- For the current time stage $k = 0$, the values

$$W_0 = (D_0^1, E_0^1, D_0^2, E_0^2, \dots D_0^{N_d}, E_0^{N_d})$$

are known, for all locations $\{s_0, \dots s_{N_d}\}$.

For each future time stage $k > 0$, we have the random variables

$$W_k = (D_k^1, E_k^1, D_k^2, E_k^2, \dots D_k^{N_d}, E_k^{N_d})$$

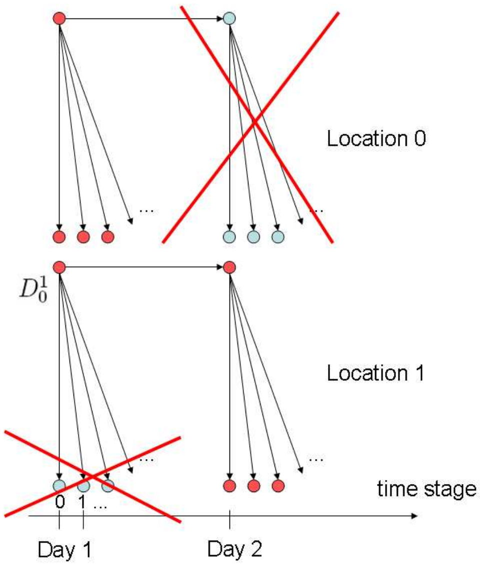


Figure 4.3: The nodes of this graph printed in red represent the variables relevant for a two day journey. At day 0, the brightness factor information for location s_0 is met by the vehicle. At day 1, this is the information for location s_1. The arcs and nodes that are crossed out in red are not relevant for the problem model. Notice especially that on day 0, the only information we can use to predict the weather on day 1 is the daily average brightness factor D_0^1.

for all relevant locations.

- For a time stage k, the function $\Delta s_k(\Delta q)$ depends on k and on $E_k^{d_k}$, the random variable describing the half hourly brightness factor for stage k at the corresponding location. In some cases, we add the index $E_k^{d_k}$ to denote the additional dependence of this relation, and we consider thus the notation $\Delta s_{k,E_k^{d_k}}(\Delta q)$.

  Once $E_k^{d_k}$ is known, $\Delta s_{k,E_k^{d_k}}(\Delta q))$ becomes deterministic for the long term energy management problem.

The goal for the subsequent Sections is to give a stochastic DP formulation for the long term energy management that allows to find the optimal control at each time, taking in account all information that is available. In the next Section, we formulate this optimization problem. In Section 4.4, we introduce stochastic DP, and in Section 4.5 we give a DP algorithm that solves the problem.

4.3 Problem Formulation

In this section, let us consider one fixed situation during the race. We will suppose that we are at a given time t_0. At this time, the vehicle is at the location s_0, and the batteries are charged at q_0. Moreover, we have a given information W_0, the weather at the current time stage.

For the long term energy management problem, time is discretized. The length of one time interval is 30 minutes, and we consider the time stages $\{0, \dots N\}$. The stage 0 corresponds to the time t_0. N is chosen 'large enough' that we arrive certainly at the finish before N. We denote by N_d the day at time stage N.

We formulate the long term stochastic optimization problem that uses the weather information model from Section 4.2. For that purpose, we will use the state space as we had it for the deterministic problem. However, we have to add a weather state variable. At stage k, this variable should contain all information we have about the future weather at this time. This is, all information that we have about the weather at stage k, and all information available at time k that contributes predicting the ulterior weather.

To do so would mean, for the most general model without any conditional independence assumptions, to add $W_0, \dots W_k$ to the state space, following the description from Bertsekas (1995a) for modeling the state space when forecasts are available. Thanks to our independence assumptions, we can break the complexity of the state space:

Breaking the Complexity of the State Space

W_k has been defined as a 8 dimensional random variable before. In this paragraph, we justify that it is sufficient to add a state variable the value $W_k' = (D_{d_k}^{d_k}, E_k^{d_k})$ to the state space at stage k. Besides the justification in words that is given below, the Figures 4.2 and 4.3 are helpful to understand that all relevant information about the weather that the vehicle will meet in the future with respect to a stage k is contained in W_k.

- Let us fix a day d such that the stages k and l belong to day d. Given a D_d^i, E_k^i and E_l^i are conditionally independent from all other information. For explication, have a look at Figure 4.2.
- Suppose now that we know $E_k^{d_k}$. Then $\Delta s_k(\Delta q)$ is a deterministic function, given by the model from Chapter 2.
- At stage k, also D_k^i for $i \neq d_k$ becomes known. This knowledge adds some information to the forecast of D_i^i, with respect to the knowledge of D_0^i we already have. However, $\Delta s_k(\Delta q)$ depends only on $E_k^{d_k}$, which does not depend on D_k^i. The information about D_k^i can thus be marginalized out without that $\Delta s_k(\Delta q)$ would be affected. Mathematical rigor would demand this justification to be formalized. However, for pragmatical reasons, we leave it like this. If the fact that eliminating D_k^i from the state space does not lead to information losses is doubted, we declare it as an additional assumption.

For the above reasons, let us redefine the state variable W_k as $W_k = (D_{d_k}^{d_k}, E_k^{d_k})$.

4.3.1 Formulation of the Stochastic Long Term Energy Management Problem

Now, we can formulate the stochastic long term energy management problem that we intend to solve.

The State Space

The state space consists in triplets (s, q, W). At stage k, W describes the predicted average brightness factor $D_{d_k}^{d_k}$ for the current day, and the predicted half hourly average brightness factor for the current stage. It evolves over a discrete set of time stages $\{0, 1, \dots N\}$.

The Transition

The transition of (s, q) is described by the deterministic $\Delta s_{k,E_k^{d_k}}(\Delta q)$. W decomposes into two state variables D and E.

For the transition of D, we distinguish two cases:

- If $k+1$ and k are on the same day, the corresponding daily average brightness factor D does not change at transition between the stages k and $k+1$.
- If $k+1$ is on day $d+1$, while k is on day d, W_{k+1} is conditionally independent to w_k. It is given by D_{k+1}^{k+1}, knowing W_0.

For the transition of E, we distinguish the same two cases.

- If $k+1$ and k are on the same day, E is a random variable that depends on the known $D_{d_k}^{d_k}$, which is a state variable at stage k.
- If $k+1$ is on day $d+1$, while k is on day d, $E = E_{k+1}^{d+1}$ depends on D_{d+1}^{d+1}, which is unknown at stage k, but becomes known at the transition to stage $k+1$. In this case, we can describe the transition of E conditionally to knowing the value of D at stage $k+1$.

Control

The control of the system is Δq.

Objective Function

The objective of optimization is to minimize the expected time.

As next, we introduce Stochastic Dynamic Programming, in order to give the algorithm solving the problem.

4.4 Stochastic Dynamic Programming

Complementary to the introduction to deterministic DP in Chapter, we introduce here to *Stochastic Dynamic Programming*, according to the theory from (Bertsekas & Tsitsiklis 1996). It is not presented in its whole generality.

General DP formulations of discrete-time, finite horizon dynamic optimization problems have a state $x \in X$ that evolves over the discrete time stages $0, \ldots N$, according to given transition probabilities. The transition probability to move from a state i at stage k to a state j at stage $k+1$ depend on a control $u \in U_k(i)$, and are denoted by $p_{ij}^k(u)$. The transition from state i under control u at the time stage k can be expressed equivalently by a transition function $f_k(i, u, w)$, where w is a random variable. The transition from i to j under control u entrains a cost $g_k(i, u, j)$. The cost is accumulated additively over N stages. At the stage N, a terminal cost $g_N(x)$ incurs.

The rule by which a control at time k, at state i is selected is denoted by $\mu_k(i)$. A policy is a set π of such control rules for all stages, in the finite case, $\pi = \{\mu_0, \ldots \mu_N\}$. The expected cost-to-go of the state i, under the policy π, at stage k is given by the recursion

$$J_k^\pi(i) = \mathrm{E}_w[g_k(i, \mu_k(i), f_k(i, \mu_k(i), w)) + J_{k+1}^\pi(f_k(i, \mu_k(i), w))]. \tag{4.2}$$

In the notation using transition probabilities, this is equivalent to

$$J_k^\pi(i) = \sum_{j\in X} p_{ij}(\mu_k(i))g_k(i,\mu_k(i),j) + J_{k+1}^\pi(j). \tag{4.3}$$

The objective of Dynamic Programming is to find a policy π^* that minimizes the expected cost-to-go. This optima satisfy some form of Bellman's equation, for the cost-to-go function

$$J_k^*(i) = \min_{u\in U_k(i)} \mathrm{E}_w[g_k(i,u,f_k(i,u,w)) + J_{k+1}^*(f_k(i,u,w))], \text{ and for the control} \tag{4.4}$$

$$\mu_k^*(i) = \arg\min_{u\in U_k(i)} \mathrm{E}_w[g_k(i,u,f_k(i,u,w)) + J_{k+1}^*(f_k(i,u,w))]. \tag{4.5}$$

These notations can be simplified by dropping some indices if there is no ambiguity.

Finite horizon DP problems can be algorithmically solved by using the above backward recursion on the cost-to-go.

A lot of methods for approximate DP are available, if exact resolution is not possible in practice. They will be introduced step by step, when they are applied to a problem.

4.4.1 Solving the Long Term Problem

Below, we use the stochastic DP method to solve the long term energy management problem. This is, we suppose that for the initial time stage, an optimal solution of the short term energy management problem is available, in the form of a relation $\Delta s_0(\Delta q)$. The purpose of long term energy management is only to decide which value of Δq is long-term optimal to be chosen at this first step, and not to plan an optimal control for the rest of the race. This decision can be made once the optimal expected time-to-go at time stage 1 is known for all states of the form $(s_0 + \Delta s(\Delta q), q_0 + \Delta q, W_1)$. It is the Δq that minimizes the optimal expected time-to-go at stage 1,

$$\mathrm{E}_{W_1}[J_1^*(s_0 + \Delta s(\Delta q), q_0 + \Delta q, W_1)].$$

Until the end of this Chapter, we will thus discuss DP methods that estimate this time-to-go for (relevant) states at time stage 1, but that do not concern in specifying the optimal control for ulterior stages.

4.5 A DP Algorithm for Fixed Information, Initial State and Time

In this Section, we give a stochastic DP algorithm that solves the problem formulated in Section 4.3.

The State Space

To the states of the form (s, q, W), an artificial terminal state T is added.

The Transition

Consider first the transition function f_k of the system.

First, in the case where $s + \Delta s < s_f$, where s_f is the finish location. In this case, the transition function is

$$f_k((s,q,W_k),\Delta q) = (s + \Delta s_{k,W_k}(\Delta q), q + \Delta q, W_{k+1} \mid W_k), \tag{4.6}$$

where $W_{k+1} \mid W_k$ is the random variable that gives the weather state variable at time stage $k+1$, conditional to the knowledge that it's value at stage k is W_k.

In the case where $s + \Delta s \geq s_f$, the system transits to the terminal state T that is added to the system analogously to the DP-formulation in Chapter 3. Once the state T is reached, it does not change anymore until time N.

The Incurring Cost

Now, let us define the cost function g_k. The approach is slightly different than in the deterministic problem in Chapter 3, but the outcome is the same.

Recall that $g_k(i, u, j) = 0$ is defined as the cost of a transition from state i to state j under the control u. We distinguish 3 cases:

- If $i = T$, then $g_k(i, u, j) = 0$.
- If $j \neq T$, we let $g_k(i, u, j) = 0$, too.
- The only case where a cost incurs is if $i \neq T$ and $j = T$. This is the transition when the system enters from a state (s, q, W) into the terminal state. At that transition, the cost corresponds to the total time spent for the race. If the transition is at stage k, this cost is thus $k\Delta t$, where Δt is the length of a time interval, plus the fraction of the last time interval needed until arrival at s_f.

4.5.1 Backward Recursion for a Fixed Day

Let us first remark that the time horizon of this stochastic optimization problem is not a priori specified. However, we are sure to finish the race within finite time, even in the worst weather scenario. Let thus N be the time stage that corresponds to the time when we finish the race in case of the worst weather scenario, by using with some arbitrary control.

We can now compute at each stage the expected cost-to-go for any state by a backward recursion. For first, let us fix some day i, and some corresponding daily average brightness factor $D = d$. Let us study the backward recursion formula at stages $k+1$ and k that are both on the day i.

Under the assumption of a fixed brightness factor, E_k^i are independent random variables for all k, and we do not need to consider the state variable D. Let us thus denote E_k^i simply by w_k for this discussion.

Let us denote by N_i the time stage at the end of day i, and by 0_i the one at the beginning. Assume that the optimal expected cost-to-go function $J_{N_i}^*$ at the end of day i is known.

We have

$$J_k^*(s, q, w_k) = \min_{\Delta q} \mathrm{E}_{\,w_{k+1}}[g_k + J_{k+1}^*(s + \Delta s, q + \Delta q, w_{k+1})]. \tag{4.7}$$

In the case where $s + \Delta s \geq s_f$, we enter into state T and the cost-to-go is clearly 0. In that case, the cost function g_k has a nonzero value, as described above. In this case $J_k^* = g_k$.

Consider now the case where $s + \Delta s < s_f$. Then, $g_k = 0$, thus we have

$$J_k^*(s, q, w_k) = \min_{\Delta q} \mathrm{E}_{\,w_{k+1}}[J_{k+1}^*(s + \Delta s, q + \Delta q, w_{k+1})].$$

Now, let us set

$$\overline{J}_k^*(s, q) = \mathrm{E}_{\,w}[J_k^*(s, q, w)]. \tag{4.8}$$

We can express the backward recursion 4.7 equivalently in terms of $\overline{J}_k^*$, instead of expressing it in terms of J_k^*.

In fact,

$$\begin{aligned}
\overline{J}_k^*(s,q) &= \mathrm{E}_{w_k}[J_k^*(s,q,w_k)] && (4.9)\\
&= \mathrm{E}_{w_k}[\min_{\Delta q}\{\mathrm{E}_{w_{k+1}}[J_{k+1}^*(s+\Delta s, q+\Delta q, w_{k+1})]\}] && (4.10)\\
&= \mathrm{E}_{w_k}\{\min_{\Delta q}\{\overline{J}_{k+1}^*(s+\Delta s, q+\Delta q)\}. && (4.11)
\end{aligned}$$

Notice that given a fixed $D = d$, this allows to reduce the dimension of the state space to 2 again. However, the optimal control at stage k depends on w_k, and we can not express the optimal control in function of the state (s,q). Fortunately, this is no restriction of the performance of our approach. According to the discussion in Subsection 4.4.1, the interest of the long term energy management problem is not to compute an optimal control for each stage k, but only to determine of the optimal expected cost-to-go at stage 1.

Given the cost-to-go at the end of the day i, $\overline{J}_{N_i}^*$, the cost-to-go $\overline{J}_{0_i}^*$ at the beginning of the day i can be computed, under the assumption that $D = d$ is fixed, once we can evaluate the expression for the backward recursion (4.11).

Evaluation of the Optimal Expected Cost-to-go

Our next concern is thus to evaluate the expression 4.11. We use Monte Carlo integration. Let $y_1, \dots y_p$ be an independent sample of w_k. This can be generated by the universal distribution function from Section 4.2, knowing the parameter $\mu = d$.

Then we approximate

$$\overline{J}_k^*(s,q) \approx \frac{1}{p}\left[\sum_{i=1}^{p}\left(\min_{\Delta q}\{\overline{J}_{k+1}^*(s+\Delta s_{k,y_i}(\Delta q), q+\Delta q)\}\right)\right]. \tag{4.12}$$

In the case where for some pairs y_i and Δq, the finish can be reached within the time interval between k and $k+1$, the cost-to-go $\overline{J}_{k+1}^*$ is simply replaced by the incurring cost g_k in the right member of equation 4.12.

4.5.2 Complete Backward Recursion

Now, we establish the backward recursion over all the stages $0, \dots N$. The above daily recursion allows to compute the optimal expected cost-to-go $\overline{J}_{0_i}^d$, conditional to $D = d$, for any state (s,q). This is the cost-to-go at stage 0_i, conditionally to the state of the variable d at stage 0_i.

Given the density $f_D(d)$, the expected cost-to-go at stage 0_i is thus

$$\overline{J}_{0_i}^* = \int f_D(d)\overline{J}_{0_i}^d(s,q)db, \tag{4.13}$$

for any fixed state (s,q).

Let us consider an independent sample $y'_1, \dots y'_{p'}$, distributed according to f_D. Using Monte-Carlo Integration once again, 4.13 can be approximated by

$$\tilde{J}_{0_i}^*(s,q) = \frac{1}{p}\sum_{i=1}^{p'}\tilde{J}_{0_i}^{y'_i}(s,q). \tag{4.14}$$

Algorithm 3 Stochastic DP-Algorithm

Input: Initial brightness values W_0, for the locations that correspond to each day.
Output: The approximated cost-to-go at stage 0, $\tilde{J}_0^*(s,q)$, for all states (s,q).
for $i \in \{N_d, \dots, 0\}$ **do**
  Generate a sample $y_1, \dots y_p$ of daily integrated radiation D_i^i, for day i and the corresponding location s_i.
  for $j = \{1, \dots p\}$ **do**
    for $k \in \{N_i, \dots, 0_i\}$ **do**
      Generate a sample $y'_1, \dots y'_{p'}$ of half hourly integrated E_k^i, conditional to the known daily integrated radiation y_i.
      for all states (s,q) such that there is a Δq where $\overline{J}_{k+1}^{y_i}(s + \Delta s_{y'_{j'}}, q + \Delta q)$ is defined for all j' **do**
        Compute $\overline{J}_k^{y_i}(s,q)$, using Monte Carlo Integration as in relation 4.12.
      end for
    end for
  end for
  for all states (s,q) where $\overline{J}_{0_i}^{y_j}(s,q)$ is defined for all j **do**
    Use Monte Carlo Integration as in relation 4.14 to find $\tilde{J}_i^*(s,q) = J_{0_i}^*(s,q)$.
  end for
end for

In this notation, $\tilde{J}_{0_i}^*$ is the cost-to-go approximation at the begin of day i we use for our algorithm. The cost-to-go function at the end of day $i-1$, $J_{N_{i-1}}^*$, is the same as $J_{0_i}^*$. We can thus pose

$$\tilde{J}_{N_{i-1}}^*(s,q) = \tilde{J}_{0_i}^*(s,q).$$

This establishes the backward recursion over i, for $\tilde{J}_{N_i}^*$. Alternatively, we can denote $\tilde{J}_{N_i}^*$ as $\tilde{J}_i^*$, the cost-to-go at the end of day i.

Algorithm 3 is based on this backward recursion. It serves to determine the cost-to-go at stage 1, which in turn can be used to determine the optimal control at stage 0. Notice that to simplify the notations, Algorithm 3 is formulated to give the cost-to-go at stage 0 instead of stage 1.

4.6 Certainty Equivalent Approximation

Algorithm 3 becomes expensive in computational time once we consider random samples that are sufficiently big and discretization grids for (s,q) that are sufficiently fine. However, it gives an illustration of the complexity of optimizing the system using stochastic DP, with respect to an accurate information model. This is the starting point for developing convenient approximation methods. In this Section, we introduce two of them.

4.6.1 Certainty Equivalent Control

Certainty Equivalent Control (CEC) is a control heuristic for stochastic DP-problems that is described in Bertsekas (1995a). It is to transform the stochastic problem into a deterministic one by identifying each random variable w of the problem by its expected value $\overline{w}$. This is, we replace the transition function $f_k(x,u,w)$ by $f_k(x,u,\overline{w})$, for each state-control pair (x,u).

No performance guarantee for CEC is theoretically given, but Bertsekas (1995a) points out that CEC performs nice in many problems. For our case, comparing the performance of CEC to the one of the stochastic DP method that is given right below, is the subject of tests in Section 4.8.

Using CEC, the problem to be solved becomes a deterministic DP problem as discussed in Chapter 3. Up from a given time t_0, a given location s_0, a charge q_0 and an initial weather for all locations W_0, E at time stage k is predicted at the expected value of the distribution given by the present model, knowing W_0.

4.6.2 Solving a Medium-Term Problem by CEC

This above approximation neglects all aspects of the discussion in this Chapter. We oppose it thus to a sort of partial CEC where some aspects of the stochastic problem are kept.

Let us consider the optimization problem for one future day i, under the assumption that the state variable $D_i^i = d$ is known for that day. In Algorithm 3, this assumption is made many times repeatedly. For some fixed d, the cost-to-go at the beginning of day i, $J_{0_i}^d$, has to be computed up from the given cost-to-go $J_{N_i}^*$.

We proceed analogously as we have divided the short term optimization problem and the long-term optimization problem. To get started, let us approximate the one day horizon optimization problem by Certainty Equivalent Control.

- The brightness factor is fixed for the whole day at d. Given initial charge and time, the maximal traveled distance during day i, in function of the final charge, is computed using a forward algorithm of Chapter 3. This is possible because the energy consumption does not depend on the location s by hypothesis. We obtain a relation $\Delta s_{i,q,d}(\Delta q)$. We define that computing such a relation is called solving the *medium term optimization problem.*

- Once this relation is given, the initial cost-to-go can be approximated by

$$\tilde{J}_{0_i}^d(s, q) = \tilde{J}_{N_i}^d(s + \Delta s_{i,q,d}(\Delta q), q + \Delta q). \tag{4.15}$$

- Integrating this with respect to d as in relation 4.14 allows to establish a backward recursion over the days $\{1, \dots N\}$.

The corresponding algorithm is Algorithm 4. It has the same form as Algorithm 3, but the daily backward recursion for all elements of the sample generated for D is replaced by the CEC solution of the medium term optimization problem. We have implemented this algorithm. The results are given in Section 4.8.

4.6.3 Remark Concerning the Energy Management from Nuon Solar Team

Meyer (2005) states that the Nuon Solar Team, the Dutch WSC Champion from 2001, 2003 and 2005, uses a three level model involving short-term, medium-term and long-term optimization to plan energy management. Unfortunately, neither a motivation nor a discussion of this model seems to be publicly available.

We fall back to this approach at this stage of work. The approach follows naturally from our assumptions about weather information availability. Instead of an independent approach, as we have conceived it, the above discussion can thus be viewed as an interpretation and a formulation of this approach in terms of stochastic dynamic programming.

Algorithm 4 Approximate Stochastic DP-Algorithm

Input: Initial brightness values W_0, for the locations that correspond to each day.
Output: The approximated cost-to-go at stage 0, $\tilde{J}_0^*(s,q)$, for all states (s,q).
for $i \in \{N,\ldots,0\}$ **do**
  Generate a sample $y_1,\ldots y_p$ of daily integrated radiation D_i^i, for day i and the corresponding location s_i.
  for $j = \{1,\ldots p\}$ **do**
    for all pairs j,q **do**
      Solve the medium-term optimization problem to get $\Delta s_{i,q,y_j}(\Delta q)$.
      for all s such that $\tilde{J}_{i+1}(s+\Delta s, q+\Delta q)$ is defined **do**
        $\overline{J}_{0_i}^{y_j}(s,q) = \tilde{J}_{i+1}(s+\Delta s, q+\Delta q)$.
      end for
    end for
    for all states (s,q) where $\overline{J}_{0_i}^{y_j}(s,q)$ is defined for all j **do**
      Use Monte Carlo Integration as in relation 4.14 to find $\tilde{J}_i(s,q)$.
    end for
  end for
end for

4.7 Regular Policy Update

So far, we have always considered the optimization problem to decide for one given initial race situation which speed is optimal. However, during the race, this problem is changing continuously. One has to decide how often the solution has to be recomputed. To enable a frequent update of the optimal control, the algorithmic challenge is to recycle as much as possible of the computations used for solving a problem to solve ulterior ones. This question is important, but has not been treated in detail so far.

However, we would like to compare the above optimization methods over the whole race. To compare results that make sense, some policy update methods must be used.

To make this more accessible for computations using the present methods, we simplify our model again, with the following additional hypothesis. Instead of considering a model where the brightness changes half hourly, we force the brightness factor to stay constant during one whole day. For this model, Algorithm 4 becomes an exact stochastic method.

At the beginning of each day i, the weather for this day is deterministically known. One iteration of this policy update algorithm is thus as follows:

- Start at a state at the beginning of day i.
- Apply either the CEC or the stochastic method to determine the optimal Δq for day i.
- Generate the state at the beginning of day $i+1$, using the the optimal control. This is, the transition of s and q are deterministic, and the weather W_i is randomly generated.

This is iterated until the finish location s_f is reached.

4.8 Method Comparison: Exact Method versus CEC Approximation

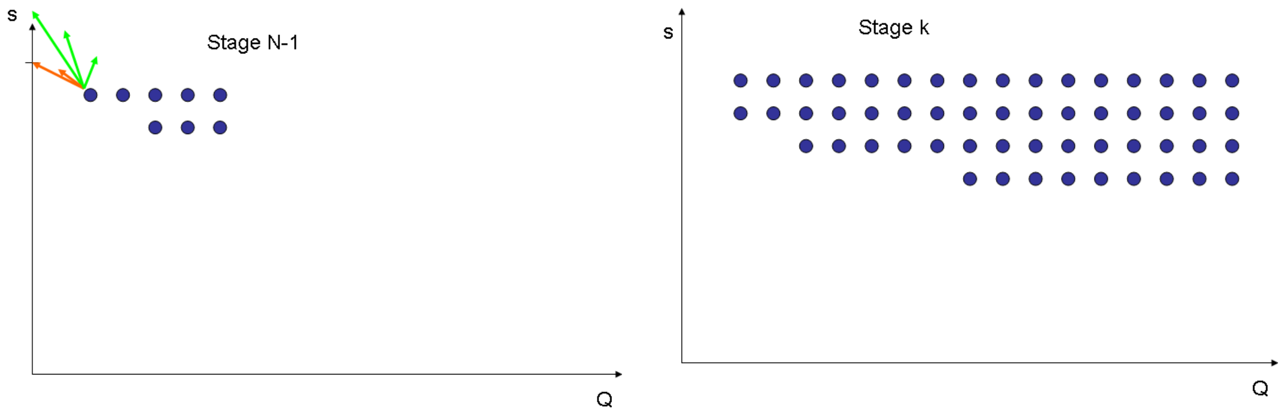
Using MATLAB, we have implemented two methods for controlling the stochastic system. As it is described above, the optimal control is updated at each stage. The control is optimized at each stage using both methods, the CEC approximation from Section 4.6, and Algorithm 4. In this Section, we give an analysis of the results that are available at this stage of work.

In Subsection 4.8.1, we precise the open parameters for applying Algorithm 4. In Subsection 4.8.2, we do the same for the CEC reference method. This is followed by the discussion of the results in Subsection 4.9.

4.8.1 Precision of the Usage of Algorithm 4

Let us first specify the open parameters for Algorithm 4.

Limiting the number of days



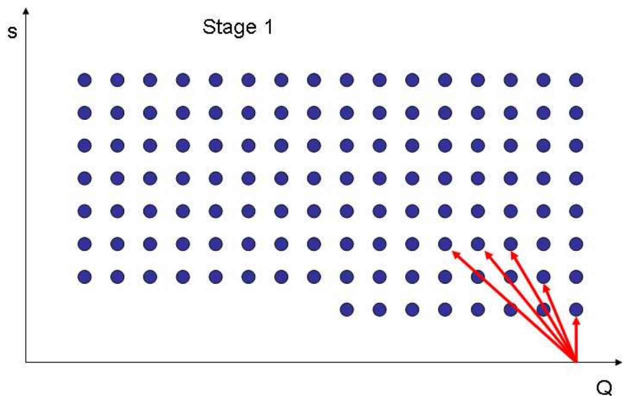


Figure 4.4: On this figure, the blue vertices represent states (s, q) where the cost-to-go $\tilde{J}$ is typically defined. The Figure in the first row on the left shows a typical situation at stage $N-1$. $\tilde{J}$ is only defined at states where there is a Δq such that the finish is reached for all y_i. The green arcs represent some feasible transitions for an elevated y_i, the red ones do the same for a lower y_i. In the first row on the right, the situation is represented for an intermediate stage. $\tilde{J}$ is defined for all states where there is a Δq such that either a state in the 'domain' where $\tilde{J}_{k+1}$ is defined, or the location s_f can be reached. At stage 1, $\tilde{J}_1$ should be defined for each state that can be reached with a feasible control (red arcs), such that the controls can be compared.

For stochastic DP, we are constrained to fix in advance a maximal time stage N_d where we start the backward recursion on the cost-to-go. In Algorithm 4, stage 0 corresponds to the morning of the first race day, and stage k corresponds to the evening of the kth race day. In order to estimate the expected cost-to-go correctly, N_d must be big enough that for all possible weather scenarios, the optimal control leads to a solution where the finish is reached before the end of day N_d. We have not studied it, but an N_d that satisfies this condition is likely to be 5 or more. However, we would like to limit this number to 4 for the present tests, to save computation time. $N_d = 4$ is chosen because in almost all situations, the race can be completed within four days. To enable this choice, Algorithm 4 must be adapted.

In the formulation of Algorithm 4, we a the requirement for the expected time-to-go $\tilde{J}_k(s,q)$ to be defined, which is that $\tilde{J}_k^{y_i}(s,q)$ is defined for the whole sample $(y_i)_{i=1,\ldots p}$ of randomly generated B^d. This requirement expresses the above statement that the cost-to-go can only be estimated correctly if we arrive at the finish before the end of day N_d independently of the weather. If we maintain this requirement when setting $N_d = 4$, we risk that the cost-to-go at stage 1 will not be defined for states that are relevant to our optimization problem. See Figure 4.4 as an illustration of this situation. Let us thus adapt the estimation method for the time-to-go, in order to have a less strong requirement.

Consider $y_1 \leq y_2 \ldots \leq y_p$ to be the ordered sample of D_k^k. There is an index l such that $\tilde{J}_k^{y_i}(s,q)$ is defined if and only if $i \leq l$. This can be justified recursively, using the fact that for a fixed k and Δq, $\Delta s_{k,y_i}(\Delta q) \geq \Delta s_{k,y_j}(\Delta q)$, for $y_i \geq y_j$. Without justifying it formally, we refer to Figure 4.4 as a graphical explication. We define $\tilde{J}_k(s,q)$ if and only if $l \geq \frac{p}{2}$, as

$$\tilde{J}_k(s,q) = \frac{1}{2l-p} \sum_{i=p-l}^{l} \tilde{J}_k^{y_i}(s,q). \tag{4.16}$$

This is, $\tilde{J}_k(s,q)$ is estimated by a truncated mean. This estimator can biased in general for states where $l < p$, namely if the distribution of $\tilde{J}_k^{y_i}(s,q)$ is not symmetric. However, we hope that the states where $l < p$, and especially where it is close to $\frac{p}{2}$, are not often relevant to our optimization problem. This is the case if the weather is good enough that the finish is reached at least some hours before the end of day 4.

We fix the value of the sample size $p = 300$ each time we compute $\tilde{J}$ in Algorithm 4. In the following, we will precise how to compute the involving $\overline{J}_k^{B^d}$. As well, we will precise how the state space is discretized, which means to precise the states where we compute $\tilde{J}$. Furthermore, where ever we need to evaluate $\tilde{J}$ between the grid points, we give an interpolation method. This will establish completely the backward recursion.

Look-Up Table for $\Delta s(\Delta q)$

In a naive implementation, one would, for each instance of the problem, for each time the policy updated, for each state (s,q) at stage k, and for each generated brightness sample y_i, optimize the relation $\Delta s_{k,y_i}(\Delta q)$ by CEC in order to solve the medium-term optimization problem, using an algorithm from Chapter 3. This involves very much computation. We propose an alternative solution. We solve the medium-term optimization problem before we start executing Algorithm 4, in order to represent the maximal distance Δs in a look-up table, in function of some values of the four factors influencing it. These values are chosen as follows

- Δq: The range of feasible values of Δq depends on the initial state q. It is given by the interval $[-q, q_{\text{full}} - q]$. The maximal feasible Δq, over all initial q is thus q_{full}, which is reached if the batteries are initially empty. The minimal feasible Δq is $-q_{\text{full}}$, reached if the batteries are initially full. We will discretize the interval $[-q_{\text{full}}, q_{\text{full}}]$ into 37 grid points. This is, the difference between two neighbor grid points is 1 MJ.

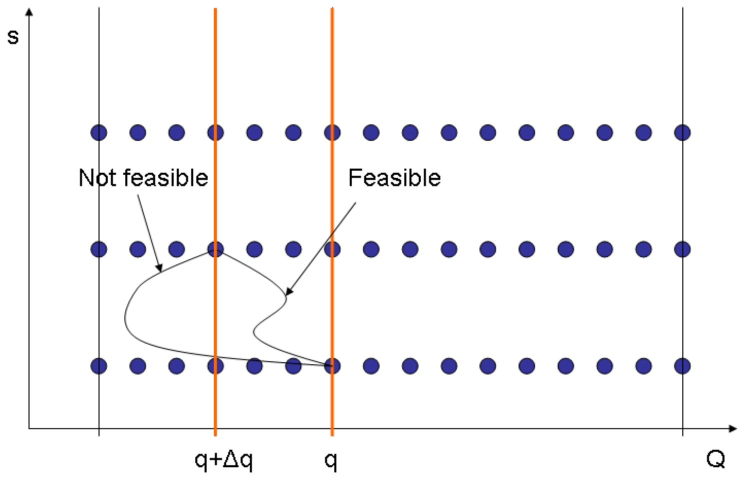


Figure 4.5: On this Figure, the blue vertices represent states (s, q) again. We consider a situation with initial charge q. The black lines show the true battery constraints of the problem. The red lines show the supplementary constraints that are introduced for a medium-term control Δq. The control labeled with 'Not feasible' would be feasible in an accurate formulation, in the present one, it is excluded. This is to reduce the dimensionality of the grid where Δs has to be computed. We think that in most of the instances, the optimal control is not excluded.

- D : We discretize the interval $[0.1, 0.9]$, where more than 99 % of the generated values for D are in, into 81 grid points $\{0.1, 0.11, \ldots 0.9\}$. Each generated sample point y_i where we evaluate Δs is truncated to be one of the grid points.
- $k \in \{0, \ldots N_d\}$: The time stages.
- Δt : On one hand side, the medium term optimization problem is solved to give the maximal Δs that can be traveled during the 9 hours of a day. However, in the case we have $s + \Delta s \geq s_f$, we need to evaluate the fraction of the day we need to arrive to the finish location s_f. Therefore, we need to store the intermediate solution of the forward recursion to solve the medium-term optimization problem as well. Let t_{k-1} be the time at the beginning of the race in the morning of day k. We store the maximal Δs that can be traveled during the intervals $[t_k, t_k + \Delta t]$, for $\Delta t \in \{30 \text{ min}, 60 \text{ min}, \ldots 9 \text{ h}\}$.

To define the optimal solution of the medium term optimization problem in function of Δq only, and not additionally in function of q, contains one further simplification that we have not mentioned so far. Let us give the initial charge q and the consumption Δq. When computing the optimal solution $\Delta s(\Delta q)$, we need to specify the upper and the lower battery constraint. We can assume that q and $q + \Delta q$ satisfy those constraints. However, we do not know if the constraints are far from $q + \Delta q$ or not. See Figure 4.5 as an illustration. However, most of the good solutions of the medium-term problem vary in between the values q and $q + \Delta q$. We exclude thus solutions that limit this bounds, to ensure that we consider only feasible controls, risking that for some instances, the optimal solution could be excluded.

Once the medium-term optimal values for this grid are specified, no medium-term optimization has to be done anymore. Instead, the optimal solution for the medium-term problem for grid points can be read from the table. For points outside the grid where we need to evaluate Δs, we will specify appropriate interpolation methods in the following.

Discretizing the State Space and Computing $\tilde{J}_k^d$ (Case $s + \Delta s < s_f$)

The state space of the problem is discretized as follows:

- The state variable q is constrained to be in the interval $\{0, 1\text{ MJ}, 2\text{ MJ}, \ldots q_{\text{full}}\}$. This is a grid of 19 values. Values of Δs for all Δq that we need for the transition between these grid points are directly available in the above look-up table.
- The time-to-go is computed at locations s on a grid of 32 points, $\{s_0, 100$ km, 200 km, ... 3000 km,$s_f\}$. This is a coarse resolution. However, at each step of the backward recursion evaluation the function $\tilde{J}_{k+1}(s + \Delta s, q + \Delta q)$ is quite precise. In fact, the optimal speed does not vary much during the race, thus time-to-go is close to linear with respect to s, for a fixed q and a fixed time stage. We can therefore use linear interpolation to evaluate $\tilde{J}_{k+1}(s + \Delta s, q + \Delta q)$ quite precisely.

Computing $\tilde{J}_k^d$ (Case $s + \Delta s \geq s_f$)

Let us consider the case where $s + \Delta s_{k,y_i}(\Delta q) \geq s_f$. In this case, the corresponding cost-to-go is the time we have needed to complete the race. We have thus to evaluate the time we need for $\Delta s = s_f - s$. Δq, B^d and k are fixed values of the grid. As we have stated above, we have an approximately linear relation between Δs and Δt. Now, we use the fact that its inverse is linear, too. Consider the smallest Δt of the grid such that $s + \Delta s \geq s_f$, and the biggest one such that

$$s + \Delta s < s_f.$$

In between the two values of Δt, we can interpolate linearly to get approximately the time-to-go.

Optimal Control

With these specifications, we have all elements to compute the backward regression on the time-to-go at all states on the grid where it is defined. Suppose that we are in a race situation at stage k, at state (s, q). We can compute the time-to-go at stage $k + 1$, then evaluate the estimated time-to-go function $\tilde{J}_{k+1}$ for each feasible control Δq at stage k. We choose the control that minimizes this time.

Updating the Control

This algorithmic procedure is repeated at the beginning of each day k, and we obtain an optimal control Δq. Applying this control, we arrive either to the state $(s + \Delta s, q + \Delta q)$, the initial state for the optimization at stage $k + 1$, or to the finish.

4.8.2 Precision of the Usage of the CEC Reference Method

As a second method, and as a reference for the above method, we implement the CEC approximation described in Section 4.6. To ensure the solutions of the two methods to be comparable, the discretization of the state and the control space, as well as the discretization for D are chosen equally in both methods. The values for the transition function are computed up from the same look-up table we use for Algorithm 4. This entrains that the set of feasible solutions for the medium term optimization problem as it is described above for Algorithm 4

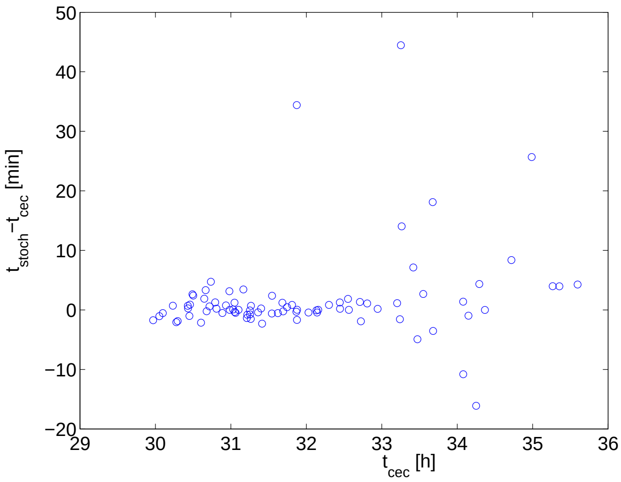
In the CEC approach, the brightness factor is fixed at its expected value. The expected value of the brightness factor is computed by taking the mean of the same sample y that we use for the Monte Carlo integration over $\tilde{J}_k^d$ in Algorithm 4.

For the resolution, we use a forward DP method from Chapter 3. Given the discrete domain of the charge variable, we are constrained to use Algorithm 1, as Algorithm 2 needs a continuous control space to be a useful method.

The optimal control is updated by repeated execution of the method at the beginning of each day, as it is done for Algorithm 4.

4.9 Result Comparison

We generate a sample of 100 weather situations and optimize the control with both methods as described above. We compare the time needed to complete the race with each one of the two approaches. The resulting time differences are plotted in Figure 4.6.



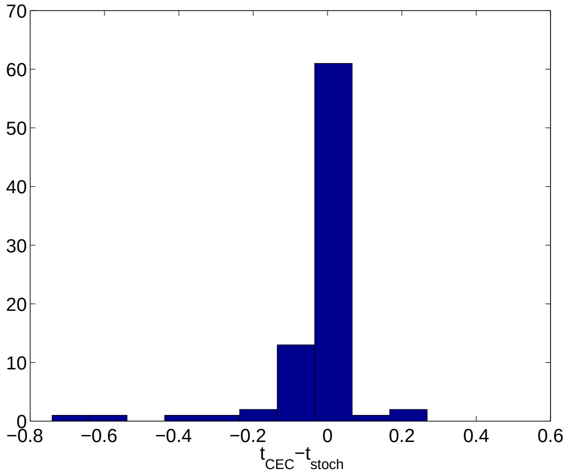


Figure 4.6: In the plot on the right hand side, $t_{\text{CEC}} - t_{\text{stoch}}$ is plotted against the differences of the total time needed to complete the race, for each sample. The tendency to positive values stands for a better performance of the CEC algorithm. However, the difference is small. Notice that especially for samples that have an optimal total race time close to 36 hours (the race time on 4 days), the CEC method performs better. On the left hand side, the histogram for the same time difference is given.

Let us define t_{stoch} as the time needed to complete the race when using stochastic method for some instance, and t_{CEC} the analogue for the CEC method. Let us suppose that the time difference between the two approaches are normally distributed, with unknown variance. We are surprised to see that the CEC method, that is designed as an approximative, suboptimal solution, performs significantly better than Algorithm 4. With confidence 95%, the expected value of $t_{\text{stoch}} - t_{\text{CEC}}$ for the whole race lies within the interval $[0.2\text{min}, 3.5\text{min}]$.

Notice that this time is practically nothing, with respect to the race duration of more than 30 hours, and notice as well that the confidence interval is very short. These facts are important to be analyzed later on, but first, the surprising outcome needs conceptual explication.

Possible Explications of the Suboptimal Performance of Algorithm 4

Without having made the simplifications in the present section, Algorithm 4 would be designed such that it minimizes the expected value of the time to complete the race. The fact that the expected time of its solution is significantly estimated to be superior than the one we obtain with a suboptimal approximation must thus be explained by a simplification that we have made.

It can not be justified in an approximation of the model we have made to enable the look-up table representation for Δs_k. In fact, both optimization methods we test are with respect to this simplified model. This is, the stochastic DP algorithm furnishes solution that minimizes the expected time with respect to this approximative model.

Therefore, there are only two possible explications left to justify the surprising outcome.

- The procedure to limit $N_d = 4$. For some samples where the race time is close to 36 hours, states where the estimation of the time-to-go is biased might be relevant for determining the optimal control. This biased estimation can lead systematically to a control that is suboptimal.
- From Bertsekas & Tsitsiklis (1996), we know that the DP algorithm using a time-to-go function estimated by Monte-Carlo integration converges when the sample size converges towards infinity. However, this does not automatically justify a good performance for our limited sample size. Actually, we think that there is no reason that this could cause systematically suboptimal controls, but we have no proof.

Discussion of the Concentration of the Difference around 0

Despite the fact that Algorithm 4 has, under the present assumptions, a suboptimal performance, the important conclusions can be made from the present results:

- For most of the instances, the time difference between the two methods is concentrated very close to 0. The fact that the approximations made in each one of the algorithms are of 'different nature' entrains that we would be surprised to have such a concentration far from the optimal solution. Moreover, the fact that this concentration is principally encountered for instances that have a total race time fairly smaller than 36 hours, in which case Algorithm 4 has a 'bigger chance' to perform well. Let us thus assume that for most instances where the time difference is close to 0, both methods give solutions that are close to be optimal.
- The fact that the latter argument holds for most of the instances supports that the average value of a CEC-optimal solution is close to the average value of an optimal solution in a stochastic sense.

Stochastic DP envisages exact optimization with respect to the expected value of the solution. However, on the present problem, the possible average gain with respect to CEC appears to be very small.

Further analysis of stochastic DP on the present model could be done. However, we have few hope for findings that improve the performance of the system that way. Instead, we will discuss in the following chapter, which conclusions for the continuation of the present work are reasonably made.

Chapter 5

Conclusion

This master's thesis sets the basis to develop energy management optimization methods for a solar vehicle during the World Solar Challenge. The novel approach is that uncertainty about the future weather information is modeled explicitly. To conclude this report, we will set our particular findings in the context of the general problem that has been introduced in Chapter 1. This is the subject of Section 5.1. In Section 5.2 we point out some research that we have initiated, but that is not ready to be presented in this report. In the final Section 5.3, we discuss what research could be done in future.

5.1 Discussion of the Results with Respect to a General Problem Formulation

The three principal contributions of this work are:

- Our test results support the thesis that the optimal control computed by CEC-approximation furnishes a solution for the long term optimization problem that is in average very close to the solution obtained when applying a control minimizing the expected time-to-go.
- We have developed a model to express weather information stochastically. Moreover, if one still wants to apply stochastic DP, we have developed the stochastic DP Algorithm 3 that can be applied to optimize with respect to this present model, or can be adapted to a similar parametric model.
- We have developed Algorithm 2, an algorithm that is based on the approximative resolution of the Hamilton-Jacobi equation. This enables to optimize the simplified, deterministic long term energy management problem without discretizing the control space, and with a diminished computing time.

In the following, we discuss the role these results play in the more general context we have introduced in Chapter 1, and the consequences they have on the further research.

CEC-approximation versus stochastic DP

Let us justify with respect to the research project as a whole why we opt for focussing future research on the CEC approach. The experimental results presented at the end of Chapter 4 support the quality of optimization results obtained through a CEC approximation. However, the following hypotheses have been made to obtain this result:

- The CEC algorithm has been compared to the stochastic DP Algorithm 4 that only takes into account the stochastic nature daily average weather variation. This is a coarse temporal resolution. We think that this approach would be sufficient to point out an important difference between CEC and stochastic DP, if one would exist. However, we can not proof this.

- The stochastic model we have used is not a model that contains all weather information we have. The important information from atmospheric models is neglected.

- We use an environment model that is only time dependent. Especially, road gradients are neglected.

- We neglect the discontinuities of the problem, as speed limitations and stops during the race that are forced by the race rules.

It is difficult to estimate if this result is still valid with respect to the general long term optimization problem. However, we have no indication that developing a stochastic DP method for the general problem would bring a large benefit.

The quality of the solution is one criteria for the performance of a long term optimization method. In Chapter 1, we have requested some supplementary properties for long term optimization methods, namely

- it should be possible to update the solution rapidly, if long term forecasts change, and

- the solution should enable flexible reaction on unforeseen changes of short term conditions.

With respect to these requirements, we can argue as follows that the CEC-approach is favorable:

- After the presentation of the methods in Chapter 4, it needs no long justification to understand that complexity of the long term energy management problem is drastically reduced using the CEC simplification. Each time we wish to update our solution, this optimization method has to be executed. We conclude that using the CEC approach, the solution of the long term energy optimization can be updated more frequently.

- In the form the two methods are presented in Chapter 4, the ability to react on short term condition changes represents a plus for the stochastic DP-problem. Effectively, it determines a time-to-go at any state (s, q) at stage 1. This corresponds to a complete resolution of the long term energy management problem. For any short term optimal relation $\Delta s(\Delta q)$ that we can obtain for the time interval between the stages 0 and 1, we can immediately determine the Δq that minimizes the time-to-go. The present CEC forward algorithm determines the time-to-go only at one single state at stage 1. However, we can apply the CEC algorithm repeatedly, in order to determine the time-to-go at selected relevant states. In Section 5.2, we point out a path to get started with this. An alternative would be to use a backward CEC recursion on the (s, q) state space, as we use for stochastic DP. Such an algorithm would be similar to the stochastic one, except that we would not have to integrate the time-to-go solution at each stage over various weather scenarios. This indicates that even if the identified disadvantage of the CEC method is completely removed, CEC still performs faster.

- Moreover, from a pragmatical point of view, developing a stochastic DP method ready to apply would consume a lot of research resources. Firstly, it needs, more critically than a CEC approach, an appropriate stochastic weather forecast model. Secondly, given its superior complexity with respect to a CEC approach, we would be forced to invest more research in efficient solution update methods.

From this global point of view, it is well justified to concentrate the next research steps on the development of CEC methods. However, we should keep in mind that this approach has no performance guarantee with respect to minimizing the expected time-to-go.

The Benefit from the Present Stochastic Weather Model

The stochastic weather information model we have developed here is full of assumptions that are not validated at the moment. However, it gives an example of how to consider the structure of dependence in a weather forecast.

This is less relevant to our optimization problem after having decided that we will use a CEC approximation. However, it is still a practical tool to test any long term optimization method for the present problem with respect to a stochastic information model.

Deterministic Forward Algorithm Solving Hamilton-Jacobi

As we will develop a CEC approximation, methods to solve the deterministic forward problem become interesting. Here, we contribute a DP-forward algorithm using the resolution of the Hamilton-Jacobi equation. The algorithm we have developed is not robust in the present form. However, we have shown that a small adaptation increases its robustness considerably. We think that this method has the potential that the present problem can be solved quickly and precision. Unfortunately, the present method is developed for a problem formulation under restrictive hypotheses, and we ignore what difficulties incur if we apply it to more general situations.

5.2 Initiated Research

We have initiated some further research that is not ready to be presented in this report. However, we point out the issues in the follwoing:

- As we have decided to choose a CEC approximation to solve the long term energy management problem, we should be able to relax some hypotheses on the forward methods, such that we are able to take into account at least spatial dependency. Therefore, a model of road gradients in function of the location is needed. We have GPS measurements of altitude at 100000 points of the race track. We have done some first steps in filtering this data. However, the result is not yet accurate. It should be fitted to map data.

- As we have remarked above, the present deterministic forward algorithms that are used to solve the CEC problem compute the time-to-go at stage 1 for only one single state. However, in order to preview unexpected short term condition changes, we must compute the time-to-go for the various states that are reached at stage 1 with the different weather situations that can incur between the stages 0 and 1. Therefore, we should determine relevant states where the time-to-go is computed. Based on Bertsekas & Tsitsiklis (1996), we are designing a system that iterates control optimization and forward simulation, a so called actor-critic system, in order to identify relevant states where the cost-to-go has to be determined.

- So far, we have applied the CEC simplification directly by identifying the weather with its expected value. However, this simplification could also been applied at a higher level. In fact, we can optimize the short-term and the medium-term problem for the various possible weather situations w, to get $\Delta s_w(\Delta q)$. Then, we can integrate this with respect to the density of w, to get the CEC approximated $\Delta s(\Delta q)$. This would request to repeatedly solve the short-term respectively the medium-term optimization problem. This is a possible precaution such that CEC performs nicely with respect to a weather model with a better temporal resolution.

5.3 Organization of Future Research

The further challenge for long term energy management is:

- For accelerating the above repeated resolution of the deterministic problem at relevant states, an algorithm that allows efficient computation of the CEC time-to-go approximation for multiple states that are close to each others could be developed.
- If we see that computing time is critical, we should consider a method where the computations made at a stage can be recycled to update the solution for the next stage. To do so, we can use the property that the optimal control has a certain continuity with respect to the initial state.

Besides, we should concern about short term optimization. A dynamic programming algorithm to solve the short term energy management problem should be developed, possibly based on the resolution of the corresponding Hamilton-Jacobi equation. The model for short term evolution of the system can be discussed. Especially, we could consider a model that takes into account energy losses due to characteristic fluctuations of the apparent wind speed around a short term average value (Daidié 2005).

This serves firstly to optimize the short term control at stage 0, once the long term problem is solved. Secondly, we should establish an empirical relation $\Delta s(\Delta q)$, in order to consider the optimal solution of the short term optimization problem to define the transition function within the long term energy management problem. This can be done using a neural network, as Bertsekas & Tsitsiklis (1996) suggest for similar problems, or by other non parametric statistical methods.

The last important step we identify to be done nextly is to specify more concretely our information situation and our information needs.

- Principally, this concerns the long term weather forecast. Using CEC, a detailed stochastic information model is not needed. However, it is important to analyze witch kind of information from atmospheric models is available, how it can be got and how it can be used.
- We should develop methods such that we will be able to estimate the divers vehicle losses empirically, once the vehicle is built.
- In collaboration with our engineers, we specify which measurements can be taken during the race. We should discuss how they can be evaluated to influence our problem model.

The computing time and the time for method development we save by using CEC instead of stochastic methods might be used to take into account the stochastic nature of the problem in a more implicit way. For example, for cases we have different controls that lead to a CEC solution that is close to optimality, we could systematize evaluation of chances and risks of the different arising long term energy management strategies. This analysis could be done by taking a set of weather scenarios as possible forecasts, and analyzing the robustness of a control with respect to these scenarios. Moreover, robustness could be understood with respect to events that are not preview, but that quite often occur, as for example a flat tire or problems in the vehicle electronics.

A lot of research is left until October 2007. However, during this work, the direction to take has become concrete.

Acknowledgements

During this master's thesis I have not only learnt to control a solar vehicle. I have made an important step in learning how to control the process from the vague idea towards concrete methods that are useful for an engineering project. With his big experience and his excellent pedagogical feeling, Prof. Th. Liebling indicated me the landmarks in this process.

It was a pleasure to work in the ROSO chair. My assistants Emma Frejinger and Michael Themans gave me important advices and were ready to help me at any time. Gauthier Stauffer has contributed to this process with some important ideas, too. The interaction with Niklaus Eggenberg, the master student that shared the office with me, has been very useful for many details.

A very big thank goes to the chief and initiator of our solar race team, Michael Friedrich, and its other active members. With their energy, they make this race project possible, which allows myself to make some first experiences in cooperating in the management of a complex engineering projet.

Moreover, thanks to the people that know how to see the future weather, and who have helped me out with their knowledge. These are especially Wolfgang Linder and John Boland.

Bibliography

Bertsekas, D. P. (1995a). *Dynamic Programming and Optimal Control*, Vol. 1, Athena Scientific.

Bertsekas, D. P. (1995b). *Dynamic Programming and Optimal Control*, Vol. 2, Athena Scientific.

Bertsekas, D. P. & Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*, Optimization and Neural Computation Series, Athena Scientific.

Boland, J. (1995). Time-series analysis of climatic variables, *Solar Energy* **55**(5): 377–388.

Daidié, D. (2005). Modélisation et simulation d'une voiture solaire du point de vue énérgetique, Internal Report.

Duffie, J. A. & Beckman, W. A. (1980). *Solar Engeneering of Thermal Processes*, John Wiley and Sons.

Linder, W. (Private Communication). wlinder@meteomedia.ch.

Meyer, F. (2005). *Speed adaptive system: Côntroleur de vistesse pour une voiture solaire*, Master's thesis, École Polytechnique Fédérale de Lausanne.

Pudney, P. (2000). *Optimal energy management for solar-powered cars*, PhD thesis, University of South Australia.

WORLD SOLAR CHALLENGE, AUSTRALIA, SEPTEMBER - OCTOBER 2005, Climate Information Package (2005). Unpublished, Consultative Services, Australian Bureau of Meteorology, N.T. Regional Office, P.O. Box 40050 (3rd Floor, Met Building, 13 Scaturchio St.) Casuarina, NT 0811, Australia, climate.nt@bom.gov.au.