EVS30 Symposium
Stuttgart, Germany, October 9 - 11, 2017

An intelligent energy management system for an electric bicycle

Stefan van Sterkenburg[1], Bram Veenhuizen[2]

[1]Stefan.vansterkenburg@han.nl

HAN University of Applied Sciences, Ruitenberglaan 29, 6826CC Arnhem, the Netherlands

[2]Bram.Veenhuizen@han.nl

Summary

This paper presents an energy management system (ems) for an electric bicycle that uses route information such as distance of the ride, slope and wind in order to obtain the most efficient energy distribution. The biggest part of the ems runs on a smart phone platform. Only a Bluetooth adapter is needed as extra hardware on the electronics of the bicycle to implement it. We present here the first test results, which show that the ems works satisfying.

Keywords: bicycle, power management, optimization

---

1 Introduction

An electric bicycle is a very popular means of transportation for relatively short ranges. They have many of the advantages of a regular bicycle. Compared to a car, an electric bicycle suffers less from traffic congestion, has less problems with parking, cause no emissions and is healthier. Because of the electric support, it overcomes the inconvenient feeling many cyclists experience on a regular bicycle when cycling uphill or with a headwind. Because of these advantages, the market share of electric bicycles increases each year and reached 28% in 2015 in the Netherlands [1]. Electric bicycles can be considered as the most widely applied hybrid light weight vehicle.

Most commercial electric bicycles use relatively simple power assist algorithms, such as the constant assist power method [2], the proportional power method [3] or the proportional torque method [4]. Also more complex assist strategies are proposed. For instance, [5] relates the assist level to the cyclist’s fatigue in order to reduce the travel time. However, none of the methods mentioned above take into account all relevant route characteristics, such as length of route, slope or wind speed during the route and the available energy from the battery in order to determine the optimal support. If the support level is too high at the start of the route, the cyclist might face an empty battery before he arrives his destination. In this paper we present an EMS that determines the support on the base the route characteristics and available energy.

2 Problem definition and system description

The goal of the ems is defined as follows: “Assume an electric bicycle that drives a given route and that has a limited amount of energy to assist the cyclist during the route. Driving conditions, like wind speed, slope and power input from cyclist are known within a certain level of certainty. Then, the goal of the ems is to control the electrical support to a level that gives the smallest drive time". One specific requirement is that it may not require significant extra hardware or costs. By preference, it must be feasible to implement the ems on already available hardware platforms.

Fig. 1 shows the block scheme of the proposed ems. It exists of two parts: an offline ems and online ems.

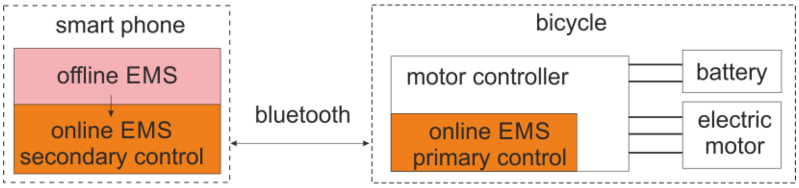

Figure1: Block scheme of the ems

The offline ems and part of the online ems run on a smart phone. At the start of the ride, the user inputs his destination, the amount of energy he wants to use from the battery and the average power he intends to deliver himself. A navigation app determines the route characteristics such as length and slope; a weather app determines the wind speed along the route. Based on this input, the offline ems calculates the power support during the route that leads to the shortest drive time. The calculated support and velocity setpoints are then passed to the online ems. Part of the online ems (the secondary control) runs also on the smart phone, the other part runs on a microcontroller unit of the bicycle. This is explained later in this paper. The online ems is active during the ride and has two main functions: it controls the support level to the setpoint calculated by the offline ems and it adjusts the offline support setpoint if unforeseen events or circumstances lead to a different power consumption than assumed in the offline calculations. The latter function aims to make the energy consumption at the end of the route equal to what has been inputted.

3 Offline EMS

The goal of the offline ems is to determine the optimal support during the route under fully deterministic circumstances. Our design approach is as follows: We defined the bicycle and cyclist model and the system constraints. Next, we tested three optimization algorithms and selected the most suited one.

3.1 Vehicle model

We used a standard vehicle model to describe the bicycle and the road load [6]. The vehicle model consists of the following subsystems (see Fig. 2):

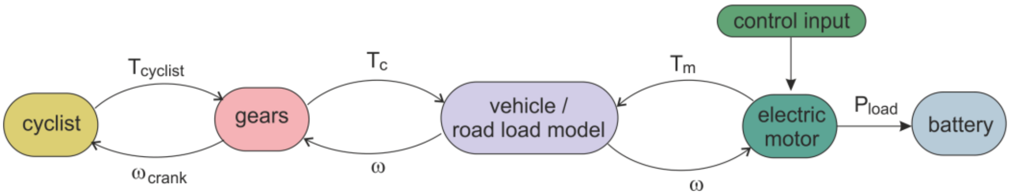

Figure2: Block scheme of the ems

- A battery pack: the battery pack is modeled as a voltage source in series with an internal resistance. The following equations apply:

$$I = \frac{U_{ocv}(soc) - \sqrt{U_{ocv}^2(soc) - 4 \cdot R_i \cdot P_{load}}}{2 \cdot R_i} \quad where: \quad soc = soc_0 - \frac{1}{Q_{bat}} \int_0^t I \cdot dt \qquad (1)$$

where: U_{ocv} = open circuit voltage in [V]
R_i = internal resistance of the battery in [Ω]
P_{load} = power delivered by battery in [W]
I = battery current in [A]
soc = state of charge [-]
soc_0 = state of charge at start of drive [-]
Q_{bat} = battery capacity in [As]

- An electric motor model: The motor is modelled as a power load. Its power consumption is determined from the rotational power delivered by the motor and power losses. It applies:

$$P_{load} = T_m \cdot \omega \ + \Delta P_m(T_m, \omega) \qquad (2)$$

where: T_m = torque delivered by the electric motor in [Nm]
ω = radial speed of the motor and wheel in [rad/s]
$\Delta P_m (T_m,\omega)$ = power losses of motor in [W]

- The road load model. The road load consists of rolling resistance, drag resistance and gravity force. For simplicity reasons we did not take into account the 'crosswind to headwind drag force ratio' as proposed by [7]. The following relations apply:

$$F_d = \ m \cdot g \cdot \left(r_w \cdot cos(\alpha) + sin(\alpha)\right) + C_d \cdot (v - w \cdot \cos(\gamma)) \ \cdot \sqrt{(v - w \cdot \cos(\gamma))^2 + (w \cdot \sin(\gamma))^2} \quad (3)$$

$$T_m + \ GR \cdot T_c \cdot \eta_{tr} = (F_d + m_x \cdot \dot{v}) \cdot R_w \qquad (4)$$

where: F_d = driving resistance in [N]
m_x = total mass of bicycle and cyclist plus reduced inertia of wheels and motor in [kg]
v = velocity of bicycle in [m/s]
R_w = radius of the wheel in [m]
T_c = torque supplied by cyclist in [Nm]
T_m = torque supplied by the electric motor in [Nm]
GR = gear ratio of transmission between crankshaft and wheel [-]
η_{tr} = efficiency of transmission (from crankshaft to wheel)
m = total mass of bicycle and cyclist in [kg]
g = gravity of earth (g=9.81 [m/s^2])
r_w = rolling resistance coefficient [-]
α = slope [rad]
C_d = drag resistance coefficient ($C_d = 0.5 \cdot C_w \cdot A \cdot \rho$) in [Ns/m]
w = wind speed in [m/s]
γ = angle between velocity of cyclist and wind speed in [rad]

- Gears system. In real world, the range of the gear ratio (GR) is discrete. However, in our model we assume that there are enough gears available to allow the approximation of a continuous range. We also assume that the cyclist selects the gear ratio that gives a crank shaft speed as close as possible to the optimum speed where he feels most comfortable.

- The cyclist is modelled as a power source of which the supplied power depends on the crankshaft rotational speed. [8] defines the relationship with an optimal crankshaft speed where the maximum average power output of the cyclist occurs. This maximum average power occurs at 120 [rpm] crankshaft speed and at 30% of the maximal muscle shortening speed. However, this maximum average power is not the typical value for the average cyclist for a comfortable ride. According to [8], [9] and [10] the typical pedaling rate is around 90 [rpm]. Therefore, we assume that the cyclist will always adjust the gear ratio towards a pedaling rate of 90 [rpm]. The maximum cyclist power depends on the physical condition of the cyclist and the effort he is willing to put into the system. In our offline ems model, we have set the maximum power point on 60 [W] (see Fig. 3).

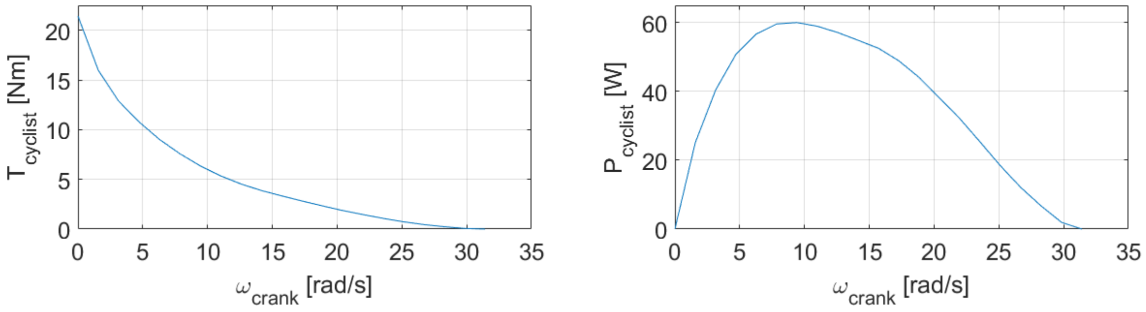

Figure3: The curve Torque-speed and power-speed curve that are used in our cyclist-model [8].

3.2 Constraints

The enumeration below defines the constraints we consider in our offline EMS calculations.

$$E_0 \geq \int_0^{t_{end}} U_{ocv} \cdot I \cdot dt \qquad (5a)$$

$$T_{motor\_min}(\omega) \leq T_m \leq T_{motor\_max}(\omega) \qquad (5b)$$

$$v_{min} \leq v \leq v_{max} \qquad (5c)$$

$$soc_{min} \leq soc \leq soc_{max} \qquad (5d)$$

$$I_{min} \leq I \leq I_{max} \qquad (5e)$$

$$GR_{min} \leq GR \leq GR_{gmax} \qquad (5f)$$

where: E_0 = maximum amount of energy that is available for the ride
T_{motor_min} = minimum torque of electromotor in [Nm]
T_{motor_max} = maximum torque of electromotor in [Nm]
v_{min} = minimum speed of bicycle in [m/s].
v_{max} = maximum speed of bicycle in [m/s].
soc_{min} = minimum value of state of charge
soc_{max} = maximum value of state of charge
I_{min} = minimum value of battery current in [A]
I_{max} = maximum value of battery current in [A]
GR_{min} = minimum value of gear ratio.
GR_{max} = maximum value of gear ratio

3.3 Vehicle model

In order to find the smallest drive time for a predefined route and energy, we first tried two standard optimization algorithm's: the Bellman's dynamic programming algorithm (DP) and nonlinear programming (NLP). DP [11,12] is a typical brute force solving method that discretizes all states, controls and the integration domain within their boundaries for a given resolution. Each possible combination with respect to the system's dynamics is calculated and finally the solution which results in the smallest value of the cost function is chosen.

The base of the NLP algorithm is the standard Lagrangian optimization method where the Karush-Kuhn-Tucker (KKT) method is used to add inequality constraints [14]. The Newton approximation method is used to approach the KKT conditions. The Euler method is used to discretize the derivative of the velocity in equation (2e). A major difference between the DP and NLP is that DP guarantees a global minimum of the cost function whereas NLP might get stuck in a local minimum.

To have a quick assessment about the applicability of both algorithms to solve our problem, we used Matlab 2015b that runs on a regular PC (i5-6600K CPU@3.5GHz,16GB memory). For this feasibility test, we defined a 12 [km] long route with a constant wind speed of 2 [m/s] and a road profile shown in Fig. 4. The available energy for the ride is set on 286.8 [kJ] which corresponds to a 30% discharge of the battery that is used in our model. The motor torque is used as control variable. A calculation step of 10 [m] is chosen. The drive time is defined as the cost function.

For the DP algorithm, we used the Matlab DPM function, described in [13]. The state resolutions are 0.1 [m/s] for the velocity, $200 \cdot 10^{-6}$[-] for the state of charge and 0.5 [Nm] for the control variable. For the NLP algorithm, we used the fmincon function from the Matlab Optimization toolbox [14]. We set the maximum number of function iterations to 300. We found this value to be a good balance between accuracy and calculation time.

Fig. 4. shows the speed and optimum support for this route as calculated by both algorithms. We conclude that the results of DP and NLP are almost the same. We explain the difference in the speed in the range 9 [km] < x < 10 [km] from the relatively large state resolution in the DP algorithm

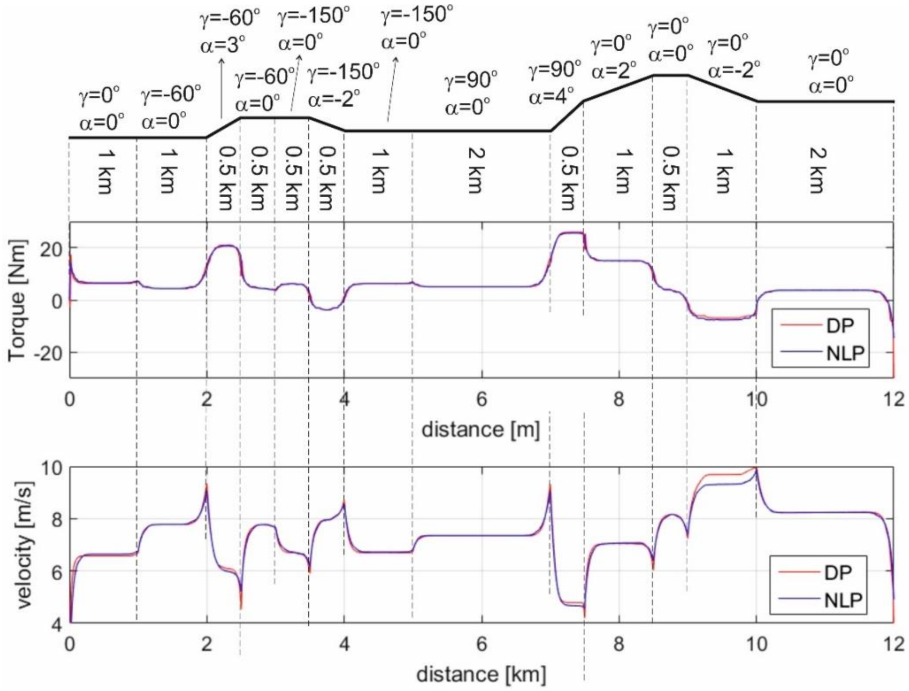

Figure4: DP and NLP test route results. The upper graph shows the height profile of the route. The lower graphs show the velocity and motor torque.

3.4 Static model optimization algorithms.

We also tested two optimization algorithms on a simplified static vehicle model. The static model doesn't take acceleration forces into account and the dependency of the open circuit voltage of the battery on the state of charged is not considered. Instead, the static model uses the average value of the open circuit voltage in the defined state of charge range.

One test is done with the NLP algorithm, discussed in the previous section. We removed the Euler approximation of the acceleration and defined a variable calculation step size that equals the length of intervals of the route with constant driving circumstances. The route of Fig. 4 is divided into 13 intervals. This is a significant reduction of calculation steps and leads to a corresponding reduction of calculation time compared to the dynamic algorithms where we had 1200 calculation steps of 10 [m] length.

The second optimization algorithm we tested is developed by ourselves. We refer to this algorithm as the dt/dE algorithm. The algorithm is named after the assumption on which it is based that states that the derivative dt_i/dE_i is the same for all intervals in the route where the inequality constraints are non-binding. In this, t_i is the drive time to drive interval i at an energy consumption of E_i. The validity of this assumption can be reasoned as follows: Assume we have found the optimum energy distribution for the route and we shift a little energy ΔE from interval i to interval j. Because we are considering a static model, the shift of ΔE only effects the drive time of interval i and j. The change in drive time of interval i and j (Δt_i and Δt_i) and the change of total drive time Δt can then be calculated as follow:

$$\Delta t = \Delta t_i + \Delta t_j \quad with\colon \Delta t_i \approx -\frac{dt_i}{dE_i}\cdot\Delta E \;\; and \;\; \Delta t_j \approx \frac{dt_j}{dE_j}\cdot\Delta E \qquad \rightarrow \qquad \Delta t = \left(\frac{dt_j}{dE_j} - \frac{dt_i}{dE_i}\right)\cdot\Delta E$$

The smallest drive time is obtained if $\Delta t \geq 0$ regardless of the sign of ΔE. This implies that dt_i/dE_i must be the same for all intervals of which the inequality constraints are non-binding.

In order to find the value of dt/dE that leads to the smallest drive time, we first determine for each interval the velocity range that doesn't violate constraints (5b), (5c) and (5e). In this range we define a set of velocities and calculate the drive time t_i and energy E_i for each velocity in that range. Also, we determine the derivative dt_i/dE_i numerically by the central difference quotient method for each velocity point in the range. At this point, we have a set of values of E_i and corresponding dt_i/dE_i values. Next, we use the bisection method to determine the value of dt/dE where the $\Sigma E_i = E_0$ (constraint (5a)). Finally, an iterative loop is implemented in order to meet constraint (5d). The route is split up in into multiple sub routes if the state of charge at the end of an interval doesn't meet constraint (5d). The energy E_x of each sub route is chosen is such a way that constraint (5d) is met and $\Sigma E_x \leq E_0$. Fig. 5 gives an example of this.

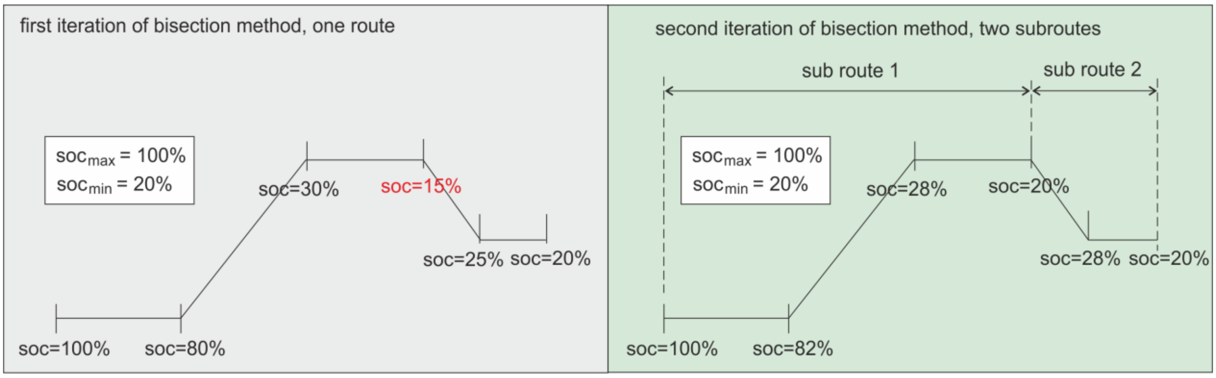

Figure5: Example of a route that is divided into two sub routes to meet constraint (5d). The left figure shows the soc after the first iteration. At the end of interval 3, the soc exceeds the lower soc limit. Therefore, the route is divided into 2 sub routes. The available energy for sub route 1 is E_0 , the available energy for sub route 2 is 0.

Fig. 6 shows the optimized torque obtained from the NLP and dt/dE algorithms applied on the static model for the test route. The NLP optimization on the dynamic model is also shown as a reference. Fig. 6 also shows the simulated velocity that we obtained from a dynamic model (including acceleration forces) that is excited by the optimized torque from the NLP and dt/dE algorithms.

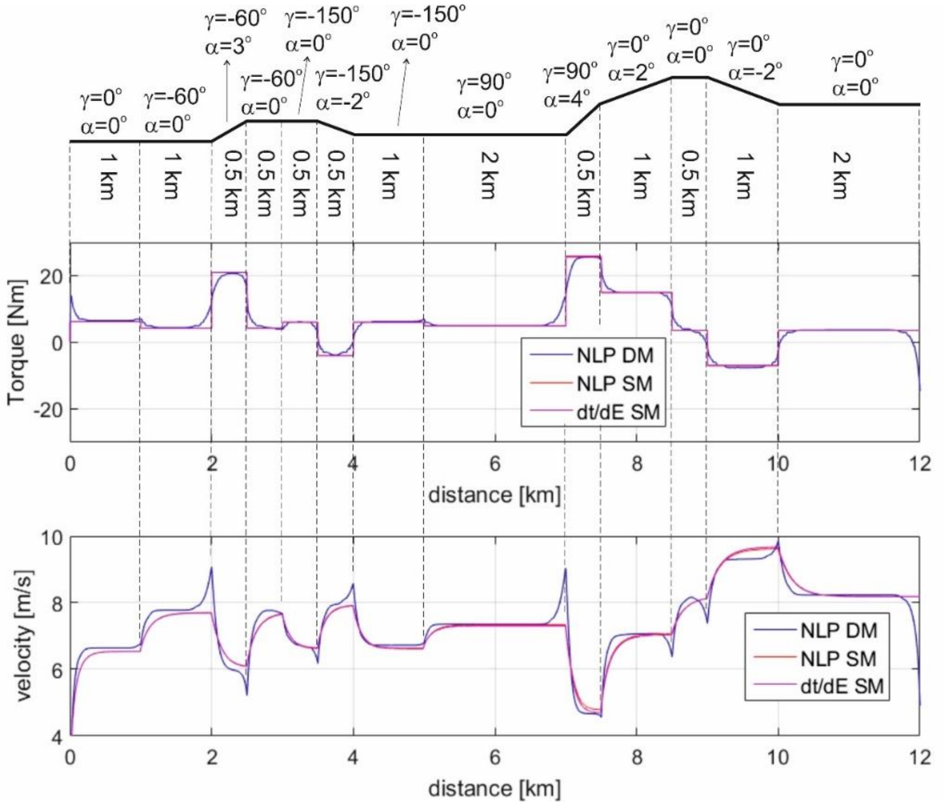

Figure6: Upper graphs: optimized torque of NLP applied on a dynamic model (NLP DM) and static model (NLP SM) and the optimized torque of the dt/dE (dt/dE SM) algorithm. The lower graphs show the simulated velocity based on a dynamic model and excited by the torque of the upper graphs.

With respect to the difference between the optimization on the static and dynamic models, we conclude that the main difference is found at the transition areas of the intervals. The dynamic algorithms make use of kinetic energy buffering in order to optimize the efficiency of the power train. Outside the transition zones, the differences are small.

3.5 Evaluation of offline EMS algorithms

We evaluated the applicability of the algorithms on the following 4 criteria:

1. The quality of the optimization of the drive time. The smaller the drive time, the better the algorithm.
2. The compliance of constraint (5a). The static models don’t include power that is associated with accelerating and therefore this might lead to a violation of constraint (5a). We determined the energy consumption from the battery by the simulations of a dynamic model that is excited by the optimized torque from the NLP and dt/dE algorithms.
3. The CPU calculation time. The offline EMS must run on a smart phone which has in general a much less performance than the PC we used for our tests.

Table 1 shows the results on the assessment criterion on the four tested optimizations.

Table 1. Characteristics of the optimization algorithms based on the test route given in Fig. 5.

Algorithm	Drive time [s]	Trip energy [kJ] (E_0=286.8 [kJ])	CPU calculation time [s]
Dynamic Programming (DP)	1645	286.6 [kJ]	26449
NLP on dynamic model	1651	287.6 [kJ]	1800
NLP on static model	1660	287.0 [kJ]	7.0
dt/dE algorithm	1662	286.1 [kJ]	0.16

The differences of the four optimizations in drive time and energy consumption are small. However, the difference in CPU calculation time is big. The dt/dE algorithm is by far the fastest and therefore we choose this algorithm to be used in our offline ems.

4 Online EMS

The main functions of the online ems are to control the electric assist to the setpoint calculated by the off-line ems and to adjust the assist level setpoint in case of disturbances. We define disturbances here as all events and circumstances that are not taken into account by the offline ems or that differ from the assumptions made by the offline ems. Typical examples of disturbances are:
- traffic situations that force the driver to stop or to drive slower than the offline ems setpoint
- real life model parameters that differ from the values used in the offline ems, for instance power input from cyclist, wind speed, drag and rolling resistance.

The online ems consists of a primary and secondary control system (see Fig. 7).

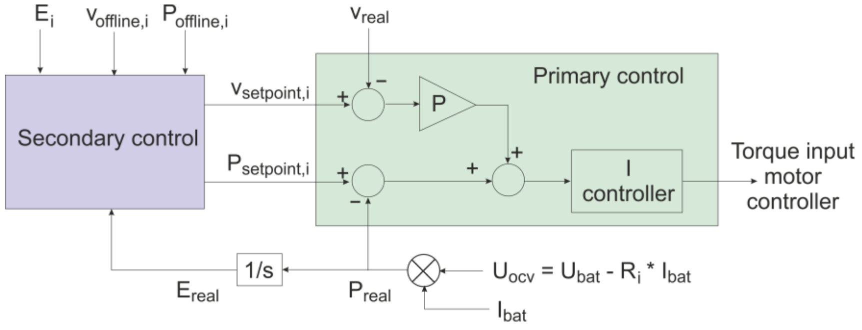

Figure 7: Block scheme of the online ems. v_{real} and I_{bat} are the measured velocity and battery current. U_{ocv} is the open circuit battery voltage and equals the battery voltage minus the voltage drop across the internal resistance. $P_{offline,i}$ and $v_{offline,i}$ are the setpoints of battery power and velocity for interval i, calculated by the secondary control. E_i is the by the offline EMS calculated energy that is available for interval i

The output of the primary control is connected to the torque input of the motor controller of the bicycle. We choose an anti windup integrative controller in the primary control to ensure a continuous output signal which gives a smooth driving experience for the cyclist. The input of the primary I-controller is a combination of the battery power error and the speed error. The P-factor is set to a constant value of P=200 W/(m/s). The following applies:

$$Torque\ input = k_i \int \left(P_{setpoint} - P_{real} + P \cdot (v_{setpoint} - v_{real})\right) \cdot dt \qquad (6)$$

The secondary control is used to ensure that the actual energy consumption at the end of the ride matches E_0, the amount of energy that is reserved for the ride. To explain the secondary control, we first define the following quantities:

- $E_{offline}(x)$ = energy calculated by offline ems at point x in the route; x=0 is starting point of the route.
- $E_{real}(x)$ = real energy consumption at point x in the route
- $\Delta E_x = E_{offline}(x) - E_{real}(x)$
- $\alpha_i = dE_i/dE_0$ (α_i defines the increase of energy available to interval i at an increase of route energy dE_0)
- $\beta_i = dP_{offline,i}/dE_0$ (β_i defines the power setpoint increase of interval i at an increase of route energy dE_0)
- $\gamma_i = dv_{offline,i}/dE_0$ (γ_i defines the speed setpoint increase of interval i at an increase of route energy dE_0)

The factors α_i, β_i and γ_i are determined by executing the offline ems calculations twice: once with the true amount of energy that is available ($E=E_0$) and once with E_0 plus a small difference ($E=E_0+\Delta E_0$). We than calculate α_i , β_i and γ_i as:

$$\alpha_i = \frac{E_i(E_0 + \Delta E_0) - E_i(E_0)}{\Delta E_0},\ \beta_i = \frac{P_{offline,i}(E_0 + \Delta E_0) - P_{offline,i}(E_0)}{\Delta E_0},\ \gamma_i = \frac{v_{offline,i}(E_0 + \Delta E_0) - v_{offline,i}(E_0)}{\Delta E_0}$$

The online secondary online ems aims to control ΔE_x to zero at the end of the route. This is done as follows: Assume that ΔE_x is not zero at point x in interval i of the route. Then we have to increase the energy supply with ΔE_x in the part of the route that is not driven yet. To do so, we determine how much extra route energy ΔE_{0x} in the offline ems gives a difference of ΔE_x from point x to the end of the route. So:

$$\Delta E_x = \left(\left(\frac{x_i - x_{done,i}}{x_i}\right) \cdot \alpha_i + \sum_{j=i+1}^{N-1} \alpha_j\right) \cdot \Delta E_{0x} \quad \rightarrow \quad \Delta E_{0x} = \frac{\Delta E_x}{\left(\frac{x_i - x_{done,i}}{x_i}\right) \cdot \alpha_i + \sum_{j=i+1}^{N-1} \alpha_j} \qquad (6)$$

where: N = number of intervals (first interval is indexed 0)
x_j = length of interval j
$x_{done,i}$ = distance already traveled in interval i $\left(x_{done,i} = x - \sum_{j=0}^{i-1} x_j\right)$

The value of ΔE_{0x} is used to make a correction on the setpoints of power and velocity according to:

$$\Delta P_{sp1,i} = \beta_i \cdot \Delta E_{0x} \quad and \quad \Delta v_{sp1,i} = \gamma_i \cdot \Delta E_{0x} \qquad (7)$$

Apart from this correction, we implemented a second correction in the offline EMS. The purpose of this correction is to adjust possible systematic disturbances. We make an estimation of the structural energy disturbances over the whole route by assuming that the disturbances along the route per meter driving are on average the same. So, if we have driven a distance of x and found a total of disturbances of ΔE_y, then the total disturbance ΔE_{dist} over the whole route is estimated as:

$$\Delta E_{dist} = \frac{x_{ride}}{x} \cdot \Delta E_y \qquad (8)$$

where: x_{ride} = total distance of the ride.

The correction on the power and velocity setpoints is done in the same way as done in equation (7), so:

$$\Delta P_{sp2,i} = \beta_i \cdot \Delta E_{dist} \quad and \quad \Delta v_{sp2,i} = \gamma_i \cdot \Delta E_{dist} \qquad (9)$$

The disturbance ΔE_y in equation (8) is calculated as the sum of ΔE_x plus the amount of energy that has been corrected by the power adjustment terms in equation (7) and (9):

$$\Delta E_y = \Delta E_x + \int \left(\Delta P_{sp1,i} + \Delta P_{sp2,i}\right) \cdot dt \qquad (10)$$

The online power and battery setpoints are calculated as:

$$P_{setpoint,i} = P_{offline,i} + \beta_i \cdot (\Delta E_{0x} + \Delta E_{dist}) \quad and \quad v_{setpoint,i} = v_{offline,i} + \gamma_i \cdot (\Delta E_{0x} + \Delta E_{dist}) \qquad (11)$$

5 Tests

In order to evaluate our ems, we defined two test routes. One test route is a typical recreational route. It has a length of 17.6 [km] and goes for a big part through the Veluwezoom national park in the Netherlands. For Dutch terms, it is relatively hilly route with a total ascending and descending of 227 [m]. Apart from 4 traffic lights and some sharp turns, there are no traffic situations to be expected that force the cyclist to make stops or to drive more slowly than the offline ems prescribes. The second route goes entirely through the city of Arnhem. The route incorporates in total 17 traffic lights. The length of the route is 12.5 [km] and has a total ascending and descending of 94 [m].

In the online EMS, we used the GPS of the smart phone to determine in which interval the cyclist is driving. The elevation along the route has been determined via the google maps elevation API.

Both routes have been tested three times:
- We tested the EMS with the secondary control turned off. We asked the cyclist to drive an average power of 60 [W], which is also inputted in the online EMS. The purpose of these experiments is to get an idea about the difference in energy consumption between the offline calculations and real world, whereby the influence of the cyclist on the total amount of disturbances is tried to keep limited.
- We tested the complete EMS whereby we asked the cyclist to drive an average power of 60 [W], that was also inputted in the online EMS. This experiment is done in order to investigate the working of the secondary online EMS, again with a limited influence of the cyclist on the total amount of disturbances.
- We tested the complete EMS, whereby we asked the cyclist to drive on average a significant higher power than 60 [W] that has been inputted in the online EMS. This experiment is done in order to investigate the working of the secondary online EMS, whereby relative large disturbances are present.

For the tests, we rebuild a regular all terrain bicycle to an electric bicycle (see Fig. 7). We used a 48 [V]/1 [kW] direct drive brushless DC hub motor with a maximum efficiency of 86%. The use of a direct drive motor enables regenerative braking, which is limited in our case by the motor controller to a maximum brake torque of 17 [Nm] from speeds above 3 [m/s]. Further, we used a 20 [Ah]/53 [V] battery with a measured internal resistance of 80 [mΩ]. We determined the relevant road load parameters of the bicycle by means of a deceleration curve measurement.

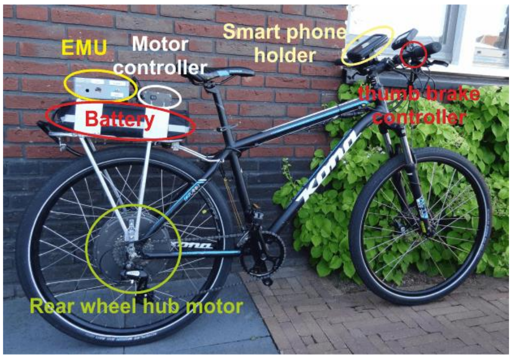

Figure 7. The bicycle that is used for our tests

We mounted a torque and speed sensor in the crankshaft in order to measure the power input of the cyclist. Since the front tooth wheel of the bicycle is mounted directly on one side of the crankshaft, only the torque and power exerted by one leg is being measured. In order to get the cyclist power, we doubled the measured power of the sensor, assuming that both legs contribute equally to the cyclists power. We equipped the bicycle with an electronic measurement unit (EMU) that consists of sensor amplifiers, a microcontroller, an actuator that is connected to the torque input of the motor controller and a Bluetooth serial interface adapter. The EMU basically has three functions:
- it serves as an interface between the smart phone and the sensors
- it implements the primary control of the online EMS
- it overrules the EMS in case of the following situations:

a. The cyclist has stopped pedaling for more than 1 [s]. In this case, the electric support from the motor is controlled to zero [Nm] in order to meet Dutch legislation which allows electric assist only when the cyclist is pedaling.

b. The speed is less than 1.5 [m/s]. In that case, we assume that the cyclist is stopping or has already stopped. Also in this case the torque is controlled to 0 [Nm]

c. The driver is using the thumb brake controller that we mounted on the bicycle. By turning the thumb controller a little, the cyclist can turn off the assist. By turning it further, regenerative braking is activated.

6 Test results

Fig. 8 shows the measured velocity and energy consumption and the by the offline ems calculated velocity and energy consumption of the six test drives. Table 2 show the most interesting characteristics of the tests.

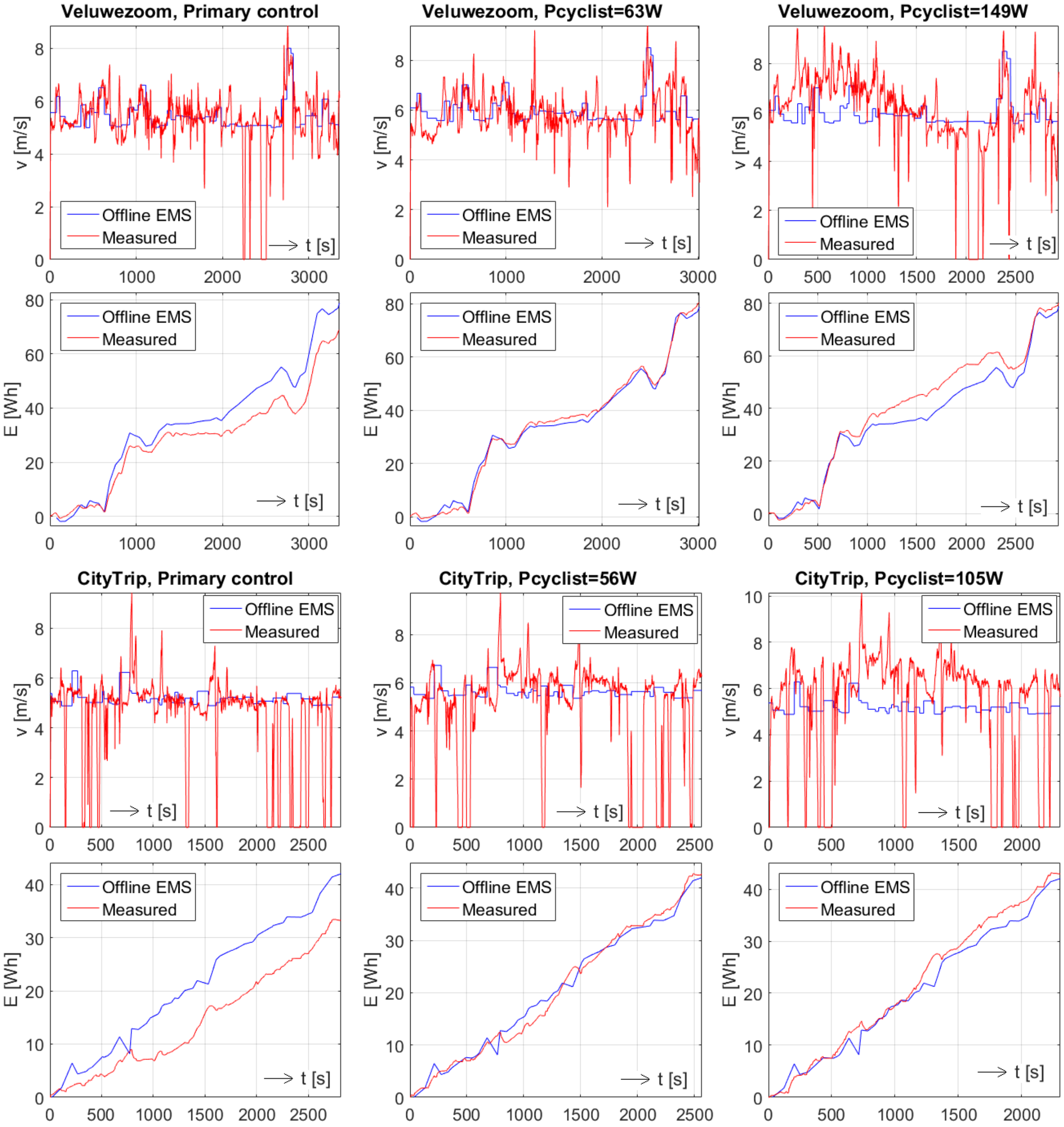

Figure 8: The measured and the by the offline EMS calculated velocity and energy as a function of the time.

Table 2. Characteristics obtained from the tests on the Veluwezoom trip and city trip.

Veluwezoom trip	Primary control	Pcyclist ≈ 60W	Pcyclist >> 60W
Energy from battery [Wh]	68.8	80.3	80.2
Difference with E_0 [%][1]	-14	1.9	1.8
Energy from cyclist [Wh]	57.3	52.7	121.7
Average cyclist power [W]	61.5	62.9	149.2
Percentage of time that ems is turned off [%][2]	4.2	-	7.0
wind speed [m/s] an direction	1.0 / 85°	0.0 / 30°	0.0 / 122°
Regenerative brake energy [Wh]	-24.0	-20.9	-20.7
City trip	Primary control	Pcyclist ≈ 60W	Pcyclist >> 60W
Energy from battery [Wh]	33.3	42.8	42.9
Difference with E_0 [%][3]	-14.4	1.97	2.1
Energy from cyclist [Wh]	49.4	40.0	67.4
Average cyclist power [W]	63.1	56.2	105.3
Percentage of time that ems is turned off [%]	19.2	-	-
wind speed [m/s] an direction	1.2 / 63°	0.9 / 9°	1.2 / 63°
Regenerative brake energy [Wh]	-7.9	-7.9	-7.9

[1] The available energy in the offline calculations for the Veluwezoom trip equals: E_0 = 78.7 [Wh]

[2] The ems is turned off when he cyclist stops pedaling, the speed is less than 1.5[m/s] or if the thumb controller is used.

[3] The available energy in the offline calculations for the city trip equals: E_0 = 42.0 [Wh]

7 Conclusions and further research

We have designed an ems that takes route information into account in order to obtain the most efficient assist during the route. Tests show that the real energy consumption at the end of the trip differs about 2% from the value that has been inputted in the offline ems, even at relative high disturbances. The implementation of the ems requires a relative simple microcontroller unit, a speed sensor, a battery voltage and current sensor and a Bluetooth adapter. Apart from the Bluetooth adapter, most electric bicycles have all the above mentioned hardware. By using microcontrollers with an embedded Bluetooth module, the extra costs to implement this ems are minimal. Further work on the ems will focus on the following points:

1. More tests need to be carried out to investigate the behaviour of the ems on more factors such as wind speed, mass, type of cyclist, etc..

2. At this moment, the offline ems calculations are carried out via a Matlab script on a regular PC. We have to write the ems in an IDE for a smart phone platform. If this is done, we might incorporate a part the offline ems in the secondary online ems in order to improve the accuracy. Also, the offline ems still needs to be coupled to a navigation and weather app.

4. At this moment, the ems works fully autonomous. The cyclist has no control over the amount of electrical assist while driving. Sometimes, this is awkward. For instance, a cyclist needs extra assist when he wants to increase the speed to overtake another road user. The primary control of the online ems, however, reduces the assist at an increase in speed (see eq. (6)). In order to make the ems more user friendly, some control of the cyclist to the electric assist needs to be present, for instance button(s) to increase or decrease the assist temporarily.

5. We have used the Google Maps elevation API to obtain elevation data. We found this service not always accurate enough. At some points in our test routes, uphill and downhill parts of the route started and ended almost 50 [m] later than indicated by the API. Another example where the service fails it at viaducts or bridges. Instead of returning the height of a bridge, the service returns the elevation of the river level. On

the other hand, we also found that the primary control of the online EMS deals well with these inaccuracies. If the EMS encounters a positive slope where a negative one is expected, the loss of speed automatically increases the electric assist as may be expected on a positive slope.

References

[1] https://www.bovag.nl/BovagWebsite/media/BovagMediaFiles/Cijfers/2016/Fietsverkoop-2007-2015.pdf (accessed on 24-06-2017)

[2] E. A. Lomonova, A. J. A.van den Put, J. Rubacek, B. d'Herripon and G. Roovers, "Development of an improved electrically assisted bicycle," in Conf. Rec. IEEE IAS Annu. Meeting. 2002, vol. 1, pp. 384-389.

[3] K. Masatoku, "Regulations on electric bicycles in Japan" in Proc. Jpn. Conf. Safety Popularization Electric Assistance Bicycles, 2001, pp. 1-5.

[4] https://electricbikereview.com/guides/ebike-mid-drive-motor-comparison/ (accessed on 24-06-2017)

[5] S. A. Fayazi, N. Wan, S. Lucich, A. Vahidi, and G. Mocko, "Optimal pacing in a cycling time-trial considering cyclist's fatigue dynamics" in American Control Conference (ACC), 2013. IEEE, 2013, pp. 6442-6447.

[6] Federico Millo, Luciano Rolando and Maurizio Andreata, "Numerical Simulation for Vehicle Powertrain Development", Researchgate chapter september 2011, https://www.researchgate.net/publication/221916059 (accessed on 24-06-2017)

[7] O. Isvan, "Power optimization for the propulsion of lightweight vehicles", Harris Cyclery, Report, 2011.

[8] R. R. Neptune, C. P. McGowan, and J. M. Fiandt, "The influence of muscle physiology and advanced technology on sports performance, "Biomedical Engineering, vol. 11, no. 1, p. 81, 2009.

[9] H. Gonzalez and M. Hull, "Multivariable optimization of cycling biomechanics" Journal of biomechanics, vol. 22, no. 11, pp. 1151-1161, 1989.

[10] S. Cheng and I. Huang, "Real-time dynamic power management of electrically assisted bicycle", in Mechatronics and Automation (ICMA), 2011 International Conference on. IEEE, 2011, pp. 313-318.

[11] R. Bellman, "Dynamic programming princeton university press", Princeton, NJ, 1957.

[12] D.P. Bertsekas, "Dynamic programming and optimal control". Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.

[13] Sundstrom, O. and Guzzella, L., "A Generic Dynamic Programming Matlab Function", In Proceedings of the 18th IEEE International Conference on Control Applications, pages 1625-1630, Saint Petersburg, Russia, 2009

[14] Optimization Toolbox User's Guide, The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098, 3 2016.

Authors

Stefan van Sterkenburg received his master's degree and PhD degree from the Eindhoven University of Technology in 1987 and 1991 respectively. Currently he is researcher in the fields of battery technology, electrical drive trains and energy management systems at the HAN University of Applied Sciences in Arnhem, the Netherlands

Bram Veenhuizen received his masters degree from the Free University in Amsterdam and PhD degree from the University of Amsterdam in 1984 and 1988 respectively. He joined SKF focussing on electromagnetic and X-ray techniques to characterize materials and material fatigue. In 1995 he joined van Doornes Transmissie (now Bosch Transmission Technology), where he was responsible for the realization of some advanced drive train projects. In 2002 he was appointed assistant professor at the Eindhoven University of Technology. Since 2005 he is professor in vehicle mechatronics at the HAN University of Applied Sciences.