A High-Efficiency Control Algorithm for Electric Vehicles

Richard Walker

Abstract

An electric vehicle consisting of a battery, a PWM controller, a motor, a mass and a drag term is simulated using timestep methods. A variable duty cycle PWM controller is assumed. The instantaneous system efficiency is defined as the ratio of power drawn from the battery to the power delivered to the vehicle. The total power required to accelerate the vehicle from zero to 1/2 maximum speed is computed for several different control algorithms. It is shown that substantial power can be saved by dynamically limiting the duty cycle of the controller to force the system to operate at high efficiency.

1 Controller and Motor model

Motor torque as a function of rotor speed (S) and controller duty cycle (D) is calculated as described in [1].

The fixed system parameters (and some typical values) are

paramter	default value	units	description
vb	24.0	volts	battery voltage
rb	1.0	ohms	battery series resistance
s0	500	rpm	unloaded motor speed
rm	1.0	ohms	motor series resistance
ip	12.0	amps	controller current limit
rc	1.0	ohms	controller series resistance
vd	0.4	volts	schottky voltage
pdc	1.0	watts	controller idle power draw

The model computes total system power draw (PS), efficiency (E) and torque (T) as a function of duty cycle (D) and rotor speed (S).

The maximum possible power draw from the battery is

$$pmax = \frac{vb}{rb + rm + rc} \tag{1}$$

The normalized rotor speed (S) is defined as the actual speed divided by s0. This gives an expression for the motor back electromotive force

$$bemf = vb * S \tag{2}$$

---

[1] http://www.ebikes.ca/simulator/

Practical controllers include short circuit protection, so a maximum duty cyle is calculated to stay below the controller peak current limit

$$dmax = \frac{bemf + ip * rm}{vb - rb * ip} \tag{3}$$

the commanded battery current is calculated using the peak limited duty cycle (dl = min(D,dmax))

$$ib = \frac{vb - bemf/dl}{rb + rm/dl} \tag{4}$$

The controller output voltage is then

$$vm = vb * dl - (1 - dl) * vd - im * rc \tag{5}$$

The total power draw from the battery is

$$ps = ib * (vb - ib * rb) \tag{6}$$

The total power transferred to the vehicle momentum is equal to the motor back electromotive voltage multiplied by the motor current. The motor current im=ib/dl, so the power transferred to the vehicle is

$$pm = ib * bemf/dl \tag{7}$$

The instantaneous system efficiency is then

$$E = \frac{pm}{ps + pdc} \tag{8}$$

2 Examination of the Motor and Controller model

Motor performance is plotted vs speed with D as a parameter. Some clear trends are apparent.

At a speed of s0*D, the back EMF is equal to the effective controller output voltage. The graphs show that for a given duty cycle D, the power falls to zero as the speed approaches s0*D.

Power transfer is equal to force times displacement. A static torque transfers no power. The curves for total power transferred go to zero a S=0 and also at S=s0*D. At S=0, no power is transferred because the rotor is stalled (no displacement). At S=s0*D, no power is transferred because the motor back EMF has driven the motor current and torque to zero. The symmetrical nature of these curves gives the rule of thumb that peak power transfer occurs at S = (s0*D)/2.

If we divide power transferred by total system power draw, we can plot efficiency curves with D as a parameter. Notice that it is extremely inefficient to use a high D drive at low vehicle speeds. A naive PID controller will accelerate to a target speed using full drive (D=1). At low speeds the power efficiency at full duty cycle can be less than 1/3 that which could be obtained with a lower duty cycle.

The torque graphs are somewhat surprising. With the simulation default parameters, a low duty cycles produces nearly twice the torque at low speeds than full duty cycle drive. Why is this? Essentially the controller acts as a DC transformer. When the duty cycle is 1, the battery resistance drops voltage in direct proportion to the motor current. The power lost in the battery resistance subtracts from the total torque available. At a lower duty cycle, the ratio between motor resistance and battery resistance is scaled by D. This causes less power to be lost in the battery resistance at the high current levels. The low duty cycle allows the controller to deliver higher currents with less system voltage drop. The higher efficiency translates to a torque boost at low speeds and low D when compared to operation with full duty cycle.

3 Control System for Improved System Efficiency

The above observations suggest a way to increase vehicle operating efficiency. A typical cruise-control loop accepts a target speed and then manipulates D to servo the vehicle speed to match the target. A simple PID controller will typically use full throttle to minimize the speed acquisition time. This causes most of the acceleration transient to operate in the very poor efficiency region where D=1 and S<<1.

If the control system is designed to limit peak D as a function of S, then it is possible to force the speed acquisition transient to spend most of it's time in the high efficiency regions.

To investigate this possibility, a system simulation is performed which combines the motor/controller model with a vehicle mass and drag. Full acceleration ramps from zero to s0/2 are performed using various control algorithms while integrating total power draw. It will be seen that some simple non-linear modifications to PID control can achieve significant power savings with modest reductions in vehicle acceleration time.

4 System Simulation

To evaluate the system efficiency under simulated operating conditions, a timestep simulator was used. The simulator core was written in C and described in [2]. It allows for efficient simulation of potentially non-linear unilateral blocks connected in an arbitary fashion. The system to be simulated is described as a block diagram. Each element of the diagram is converted to a C subroutine that describes future output node values in terms of the current input state variables. A list of all the blocks, with appropriate input and output node assignments is linked to the core simulator algorithm. The simulator calls each of the blocks in turn at each time step calculating the future output nodes for each block. After all nodes have been computed, the simulator copies the future values to the present values, advances the timestep and repeats the process until reaching the simulation stop time. The simulation algorithm is simple (l00 lines of C), and guarantees forward progress with no possible race conditions. Accuracy can be controlled by the step size used.

The block diagram used for simulating the vehicle and traction system is shown below.

The drive block is responsible for setting a target vehicle speed as a function of time. A PID control loop adjusts the motor controller duty cycle D to make the vehicle speed match the target speed. The vehicle speed is computed by taking the motor torque and using it to accelerate the vehicle mass subject to optional friction and air-drag losses. The system power is integrated to provide a running power usage total.

The next figure shows the results of two simulations. The first is shown in green and uses a standard PID control algorithm. The top plot shows full throttle commanded from 5 to 130 seconds. The vehicle accelerates in a straight-line fashion until it reaches the target speed at 130 seconds at which point the drive falls to D=0.5 which is the steady state for the target speed of S=0.5. During most of the acceleration transient, the efficiency is quite low

---

[2] Richard Walker, "Clock and Data Recovery for Serial Data Communications", pp 62-65, BCTM tutorial, September 27, 1988. (http://www.omnisterra.com/pdfs.talks/bctm2.maker.pdf)

due to the high current in both the battery and controller series resistance. Upon reaching the final speed, the first control algorithm has burned 16300 units of energy.

The second simulation is plotted in red and uses a modified PID with a dynamic output limit. The maximum D value is limited to S*1.05+0.05. This forces the system to stay near peak system efficiency during the entire acceleration transient. The second algorithm takes 321 seconds to reach the target and uses 8359 units of energy.

5 Summary

Dynamically limiting the PID speed controller output as a function of speed produces a significant increase in system efficiency when measured by total power required to ramp the vehicle speed from zero to s0/2. Acceleration time lengthens from 130 seconds to 321 seconds and the total energy falls from 16300 to 8359. On flat land with low aerodynamic drag, the total energy budget can be dominated by acceleration efficiency in stop-and-go traffic. The proposed algorithm nearly cuts in half the total power draw for these example parameters.

Vehicles implementing this system could have an efficiency dial on the dashboard allowing the user to select any degree of “sportiness” vs efficiency. For safety reasons, it might be advisable to add a rate-of-change discrimator to the throttle input. The controller can use this to detect panic speed changes to override the efficiency controls and give full acceleration during emergency maneuvers.

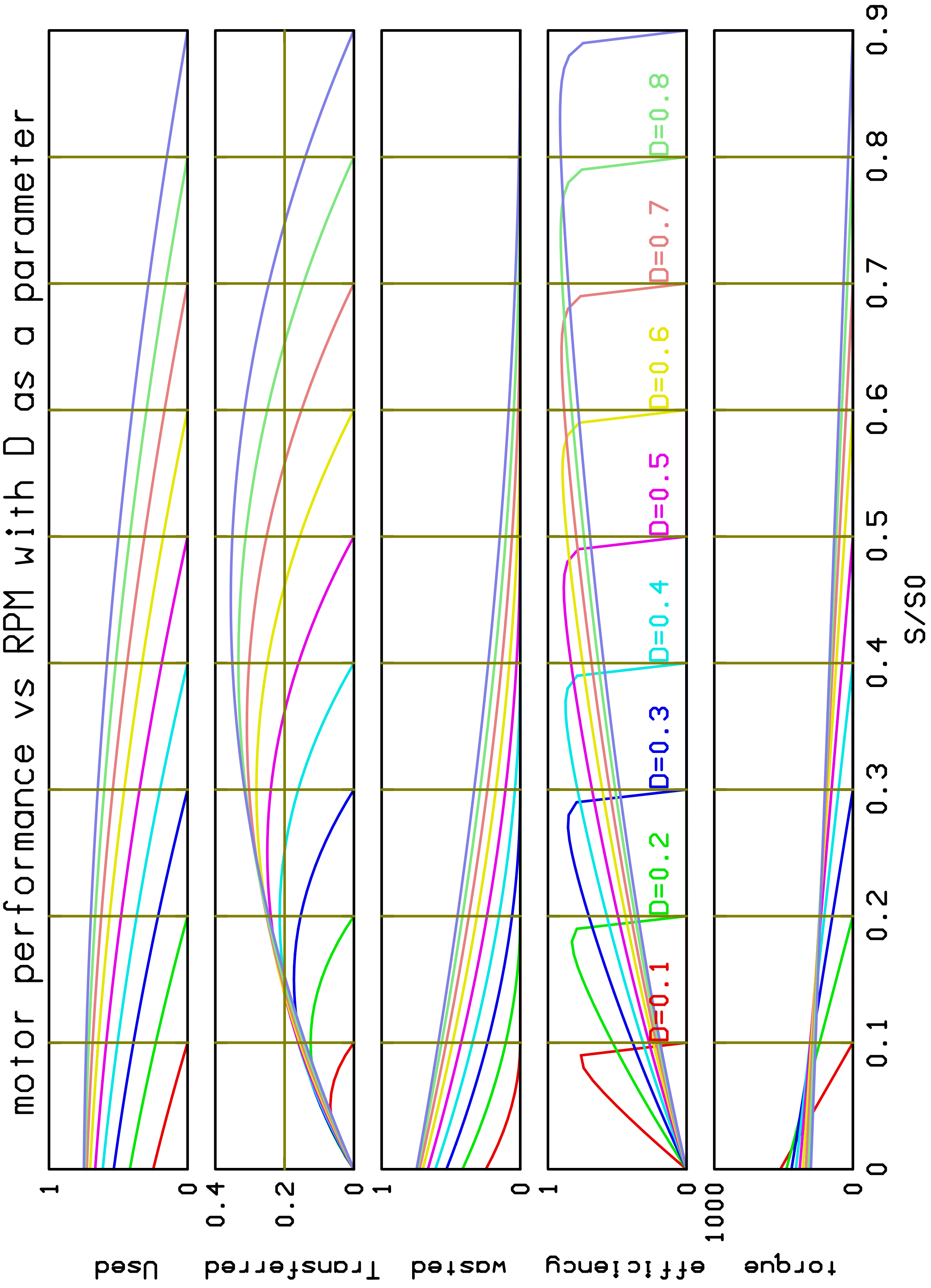

Fig. 1: Calculated performance for default system parameters at various speeds with duty cycle as a parameter

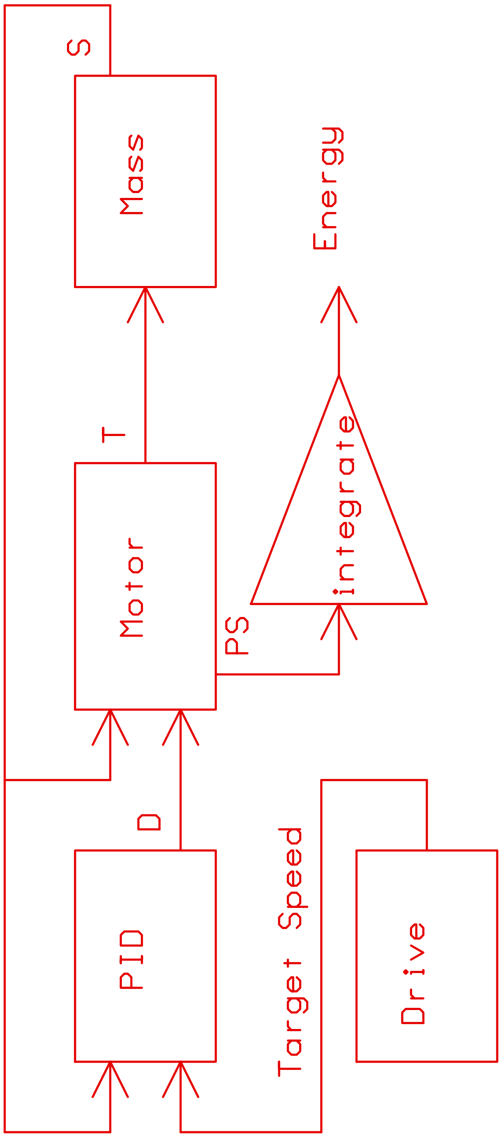

Fig. 2: Simulation Block Diagram

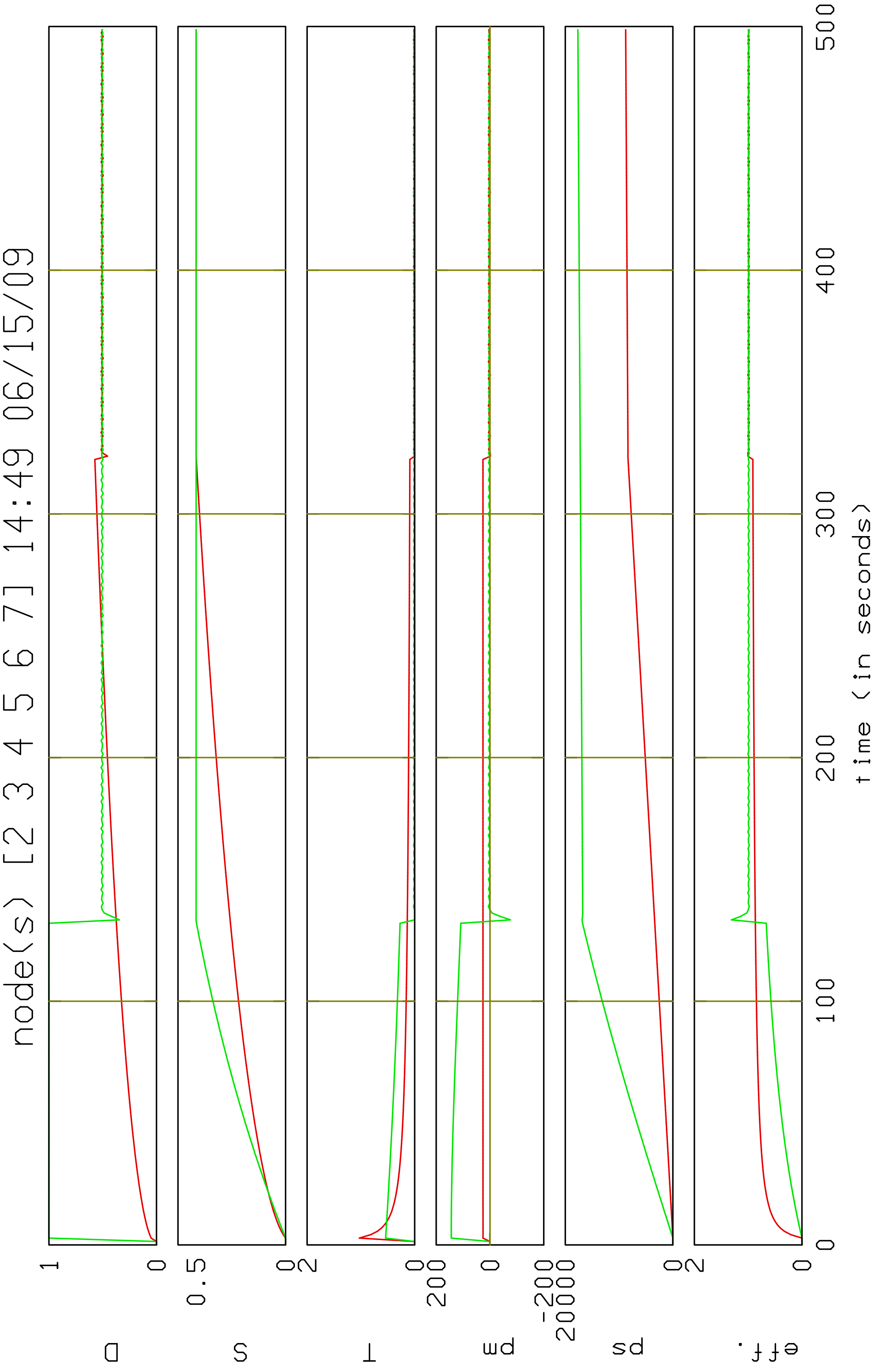

Fig. 3: Power usage for zero-to-half max acceleration test using two different control algorithms