SimBALink

Aaron Bonnell-Kangas
Nathan Lord

December 2014

SimBALink is a motorcycle powertrain and longitudinal vehicle dynamics model. It is written in Simulink, with the goal of being modular and integrating calibration and validation functionality.

This document describes the structure and the validation status of the model as of December, 2014.

Contents

1 Vehicle **3**

2 Chain and gearing **3**
2.1 Inputs and outputs . . . 3
2.2 Background, rationale, modeling strategy . . . 3
2.3 States . . . 4
2.4 Parameters . . . 4
2.5 Functions . . . 4
2.6 Assumptions . . . 4
2.7 Validation . . . 4

3 Brakes **6**
3.1 Inputs and outputs . . . 6
3.2 Background, rationale, modeling strategy . . . 6
3.3 Variables . . . 6
3.4 Assumptions . . . 6

4 Tires **6**
4.1 Inputs and outputs . . . 6
4.2 Background, rationale, modeling strategy . . . 7
4.3 Variables . . . 8
4.4 Parameters . . . 8
4.5 Function . . . 8
4.6 Assumptions . . . 8
4.7 Calibration . . . 9

4.8 Validation 10

5 Wheel forces **11**

6 Chassis **11**
6.1 Chassis 11
6.2 Inputs and outputs 12
6.3 Background, rationale, modeling strategy 12
6.4 States 12
6.5 Variables 12
6.6 Parameters 13
6.7 Assumptions 13
6.8 Calibration 13
6.9 Validation 14
6.10 Validation 16

7 Environment **20**
7.1 Inputs and outputs 20
7.2 Background, rationale, modeling strategy 21
7.3 Parameters 21
7.4 Assumptions 22
7.5 Validation 22

8 Motor controller **24**
8.1 Block diagram 24
8.2 Inputs and outputs 24
8.3 Background, rationale, modeling strategy 24
8.4 Parameters 27
8.5 Assumptions 27

9 Battery pack **28**
9.1 Block diagram 28
9.2 Inputs and outputs 28
9.3 Background, rationale, modeling strategy 28
9.4 Parameters 29
9.5 Assumptions 30
9.6 Battery pack model performance 30

10 Motor **34**
10.1 Block diagram 34
10.2 Inputs and outputs 34
10.3 Background, rationale, modeling strategy 34
10.4 Parameters 36
10.5 Assumptions 36
10.6 Motor model performance 37

1 Vehicle

This system models the forces acting on the vehicle.

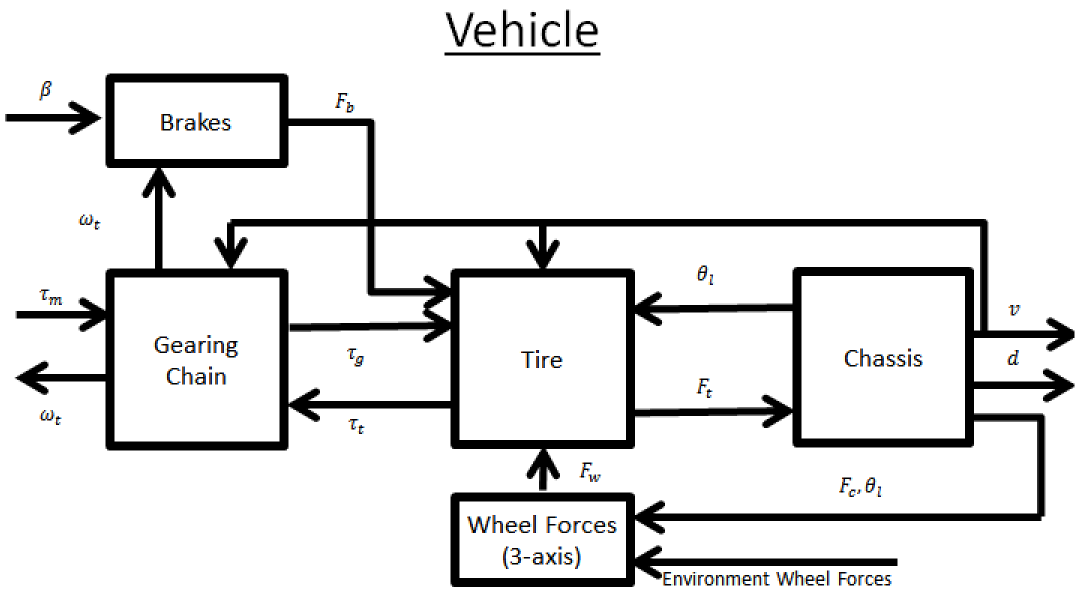

Figure 1: Vehicle Diagram

2 Chain and gearing

This system models the gear and chain in such a way to allow for wheel slip.

2.1 Inputs and outputs

2.1.0.1 Inputs

Input	Symbol	Unit
Motor Torque	τ_m	Nm
Tire Torque	τ_t	Nm

2.1.0.2 Outputs

Output	Symbol	Unit
Tire Velocity	ω_t	rad/s
Gear Torque	τ_g	Nm

2.2 Background, rationale, modeling strategy

The tire, chain, gear, and motor are modeled as a lumped inertia that is accelerated by the motor torque and tire torque(modeled as a load). The chain is modeled lossey through an efficiency map. Gearing is modeled as a ratio that linearly changes motor torque to gear torque. This method of modeling allows for wheel slip down the line.

$$\dot{\omega_t} = \frac{\tau_g - \tau_t}{J_m + J_g + J_t + J_c} \tag{1}$$

$$\tau_g = \frac{\tau_m \eta_c(\omega_t)}{R_g} \tag{2}$$

2.3 States

State	Symbol	Unit
Tire Velocity	ω_t	rad/s

2.4 Parameters

Parameters	Symbol	Unit
Motor inertia	J_m	$kg * m^2$
Gear inertia	J_g	$kg * m^2$
Chain inertia	J_c	$kg * m^2$
Tire inertia	J_t	$kg * m^2$
Gear Ratio	R_g	$\frac{\tau_g}{\tau_m}$

2.5 Functions

$\eta_c(\omega_t)$

Type	Description	Symbol	Unit
Input	Wheel Speed	ω_t	rad/s
Output	Chain Efficiency	n/a	%

The function is modeled as a look up table following the curve below described in the paper "Optimization of Chain Drives in Sports Motorcycles".

2.6 Assumptions

- The chain and gearing is rigid (no chain/gear dynamics)
- Chain efficiency is only a function of wheel speed

2.7 Validation

The model was subjected to a motor torque of 10 and an increasing tire torque. The model works correctly. The wheel increases in speed and the correctly models the losses in the chain.

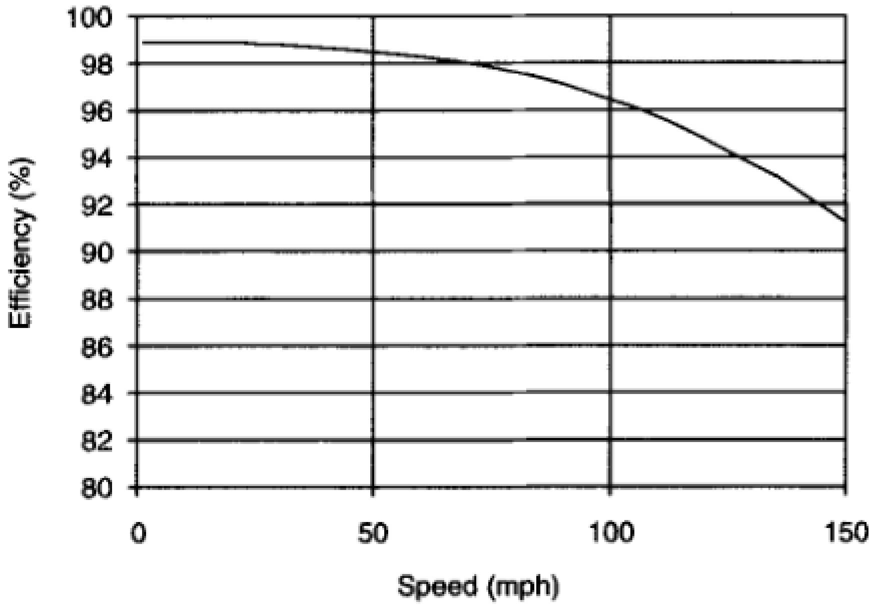

Figure 2: Estimated Chain Efficiency

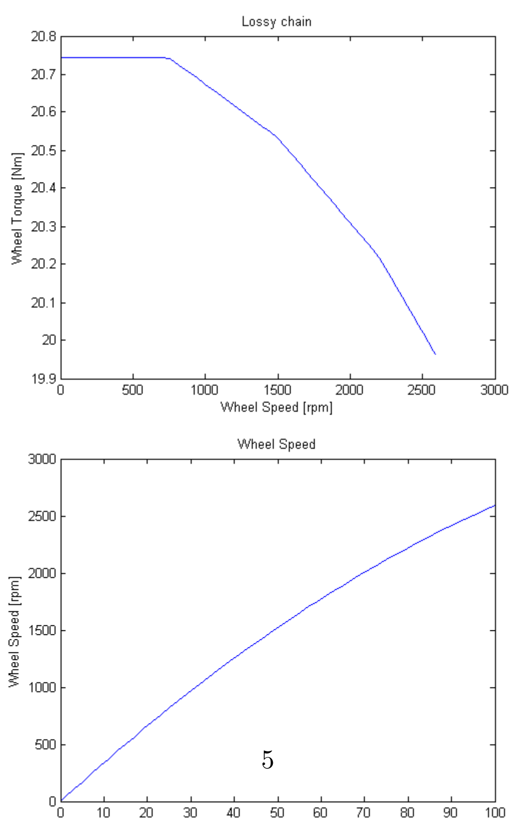

Figure 3: Gear and Chain Validation

3 Brakes

3.1 Inputs and outputs

3.1.1 Inputs

Input	Symbol	Unit
Brake Command	$beta$	%
Wheel Speed	ω_t	rad/s

3.1.2 Outputs

Output	Symbol	Unit
Brake Force on Tire	F_b	N

3.2 Background, rationale, modeling strategy

The brake is modeled as a friction force and a constant that converts β to a force.

$$F_b = \mu_b \omega_t \beta k_b \tag{3}$$

3.3 Variables

Var	Symbol	Unit
Brake Coefficient of Friction	μ_b	$\frac{N}{rad/s}$
Force Constant	k_b	$\frac{N}{\%}$

3.4 Assumptions

- Brake percentage to friction force is linear
- The tire never locks

4 Tires

4.1 Inputs and outputs

4.1.1 Inputs

Input	Symbol	Unit
Brake Force	F_b	N
Gear Torque	τ_g	Nm
Wheel Forces[3]	F_w	N[3]
Vehicle Velocity	v	m/s
Lead Angle	θ_l	rad

4.1.2 Outputs

Output	Symbol	Unit
Tire Torque	τ_t	Nm
Acceleration Force	F_a	N
Acceleration Torque	τ_a	Nm
Tire Road Torque	τ_r	Nm
Wheel Slip	κ	ratio
Max Force	F_{max}	N

4.2 Background, rationale, modeling strategy

The tire is modeled in three parts, rolling resistance, Load and Torque, and Traction Limiting. Force directions are defined as longitudinal(long), lateral(lat), and normal(n). Longitudinal is along the direction of the motorcycle (when moving straight). Lateral is orthogonal to Longitudinal axis. Normal 3-D orthogonal to lateral and longitudinal, in general the axis to the road on no incline.

The tire models slip which in turn is used to calculate the force the tire exerts on to the motorcycle. Wheel slip occurs when because the the tire does not exert a force on to the vehicle until there is some wheel slip, thus causing the tire to spin up causing wheel slip.

Rolling Resistance

$$Frr = \begin{cases} (0.0085 + \frac{0.18}{p_t} + \frac{1.59*10^{-6}}{p_t} v_{kph}^2) F_{w,n} & : v_{kph} \leq 165(km/h) \\ (\frac{0.18}{p_t} + \frac{2.91*10^{-6}}{p_t} v_{kph}^2) F_{w,n} & : v_{kph} > 165(km/h) \end{cases} \tag{4}$$

Wheel Slip

$$\kappa = -\frac{v - \omega_t r_t(\theta_l)}{v} \tag{5}$$

$$\mu_{t,gnd} = D_\kappa \sin(C_\kappa \arctan[B_\kappa \kappa - E_\kappa (B_\kappa \kappa - \arctan B_\kappa \kappa)]) \tag{6}$$

Load and Torque

$$\tau_r = F_{w,long} r_t(\theta_l) + F_b r_b + F_{rr} r_t(\theta_l) \tag{7}$$

Traction Limiting

$$F_a = \mu_{t,gnd} F_{w,n} - F_{w,long} \tag{8}$$

Torque on Chain/Gear

$$\tau_a = F_a r_t(\theta_l) \tag{9}$$

$$\tau_t = \tau_a + \tau_r \tag{10}$$

The tire coefficient ($\mu_{t,gnd}$) is modeled using the "Magic Formula" as shown below. Where D_κ is the maximum tire coefficient of the tire.

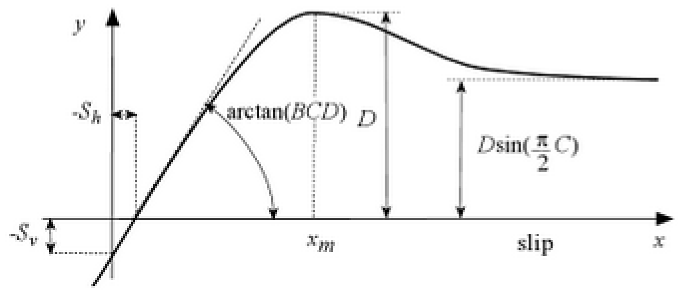

Figure 4: Magic Formula

4.3 Variables

Var	Symbol	Unit
Rolling Constant	K_t	n/a
Tire coefficient	$\mu_{t,gnd}$	n/a
Force	F	N

4.4 Parameters

Param.	Symbol	Unit
Tire Pressure	p_t	*bar*
Brake Caliper Radius	r_b	m
Magic Formula		
	$A_\kappa, B_\kappa, C_\kappa, D_\kappa$	n/a

4.5 Function

$r_t(\theta_l)$

Type	Description	Symbol	Unit
Input	Lean Angle	θ_l	rad
Output	Tire Radius	n/a	m

4.6 Assumptions

- Maximum acceleration force should also depend on lateral forces on the vehicle. However this is not modeled because it requires modeling of high-side and low-side dynamics. The Rider model should control for a safe operating area of the motorcycle to compensate for this assumption.
- No tire deformation
- No tire temperature dynamics
- No change in rolling resistance with lean angle

4.7 Calibration

The magic formula was calibrated using non-linear least squares to data collected from BikeSim. The calibration was fit to data for a normal force of 400 Newtons.

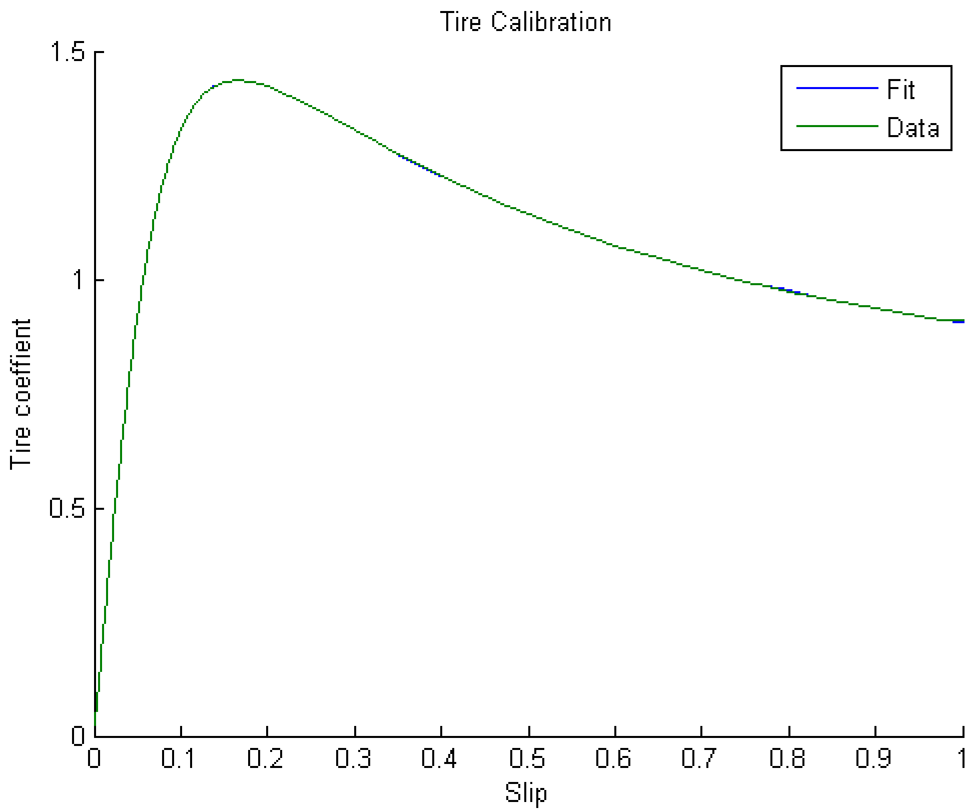

Figure 5: Magic Formula Calibration

It can be seen the fit is very good.
The fit was then compared to other normal forces.

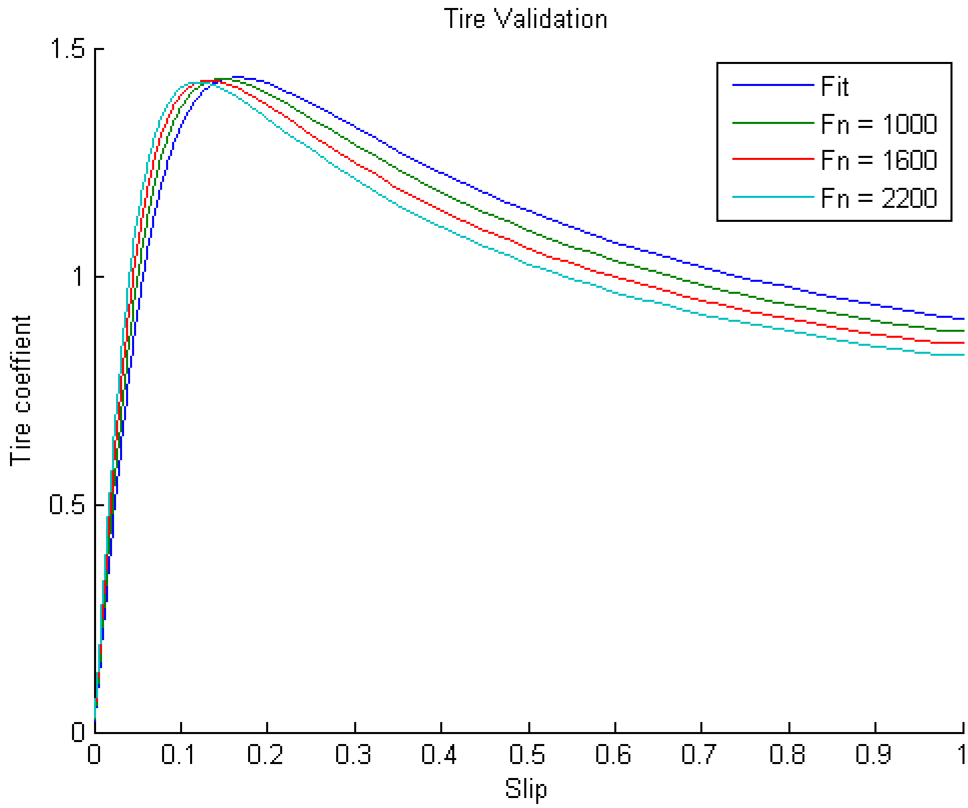

Figure 6: Magic Formula validation

The fit is not longer perfect, it can be seen the tire coefficient changes with normal force but it is not being modeled.

4.8 Validation

The tire model was swept through different slip speeds to validate correct shapes. All validation looks correct but multiple parts need to be connected to check for proper dynamics.

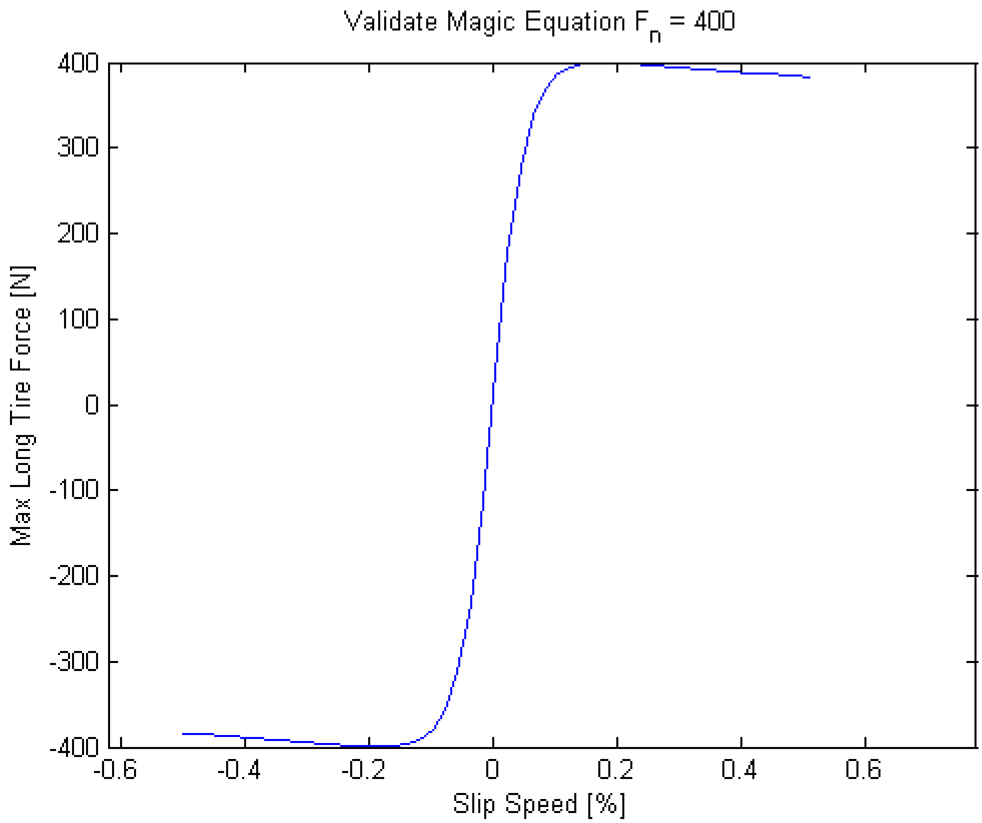

Figure 7: Tire Validation

5 Wheel forces

Wheel Forces does nothing. It would allow for the environment to affect tire forces.

$$F_{\omega,long} = F_{c,long} \tag{11}$$
$$F_{\omega,n} = F_{c,n} \tag{12}$$

6 Chassis

6.1 Chassis

This system models the chassis of the motorcycle including the velocity of the motorcycle and forces on the road.

The forces have a notation of Longitudinal (long), Normal(n) , and Lateral (lat). Longitudinal being the direction the motorcycle is moving. Lateral at a 2D right angle to Longitudinal direction. Normal is orthogonal to others.

6.2 Inputs and outputs

6.2.1 Inputs

Input	Symbol	Unit
Tire Force	F_t	N
Air Density	ρ	kg/m^3
Road Gradient	θ_r	rad
Road Corner Radius	R_c	m

6.2.2 Outputs

Output	Symbol	Unit
Vehicle Velocity	v	m/s
Distance Traveled	d	m
Lean Angle	θ_l	rad
Chassis Forces	F_c	N[3]

6.3 Background, rationale, modeling strategy

The Chassis is modeled point mass with drag.

$$F_a = \frac{1}{2}\rho C_d A v^2 \tag{13}$$

$$F_{c,long} = F_a + gm\sin(\theta_r) \tag{14}$$

$$F_{c,n} = mg\cos(\theta_r) \tag{15}$$

$$\dot{v} = \frac{F_t}{m} \tag{16}$$

$$\dot{d} = v \tag{17}$$

$$O_l = \arctan(\frac{v^2}{gR_c}) \tag{18}$$

6.4 States

State	Symbol	Unit
Distance	d	m
Velocity	v	m/s

6.5 Variables

Output	Symbol	Unit
Drag Force	F_a	N

6.6 Parameters

Param.	Symbol	Unit
Drag Area	C_dA	$\frac{N}{rad/s}$
Gravity	g	m/s^2
Mass of Motorcycle	m	kg

6.7 Assumptions

- The full weight of the motorcycle is always on the correct tire for breaking or acceleration. That is not a bad assumption because maximum braking or acceleration will happen at wheelie or stoppie when there is only one tire on the ground.
- Lean angle does not affect Aero Drag
- No lateral forces
- lean angle is optimal lean angle given corner radius and speed

6.8 Calibration

The model was calibrated against coastdown data using linear least squares. Both C_dA and Rolling resistance values were found using the following equation.

$$Force = 0.5\rho V^2 + mgV\cos(\alpha) + mg\sin(\alpha) \tag{19}$$

$$m = 236.04 + 90.71 \tag{20}$$

$$\rho = 1.187 \tag{21}$$

$$\alpha = -.0157 \tag{22}$$

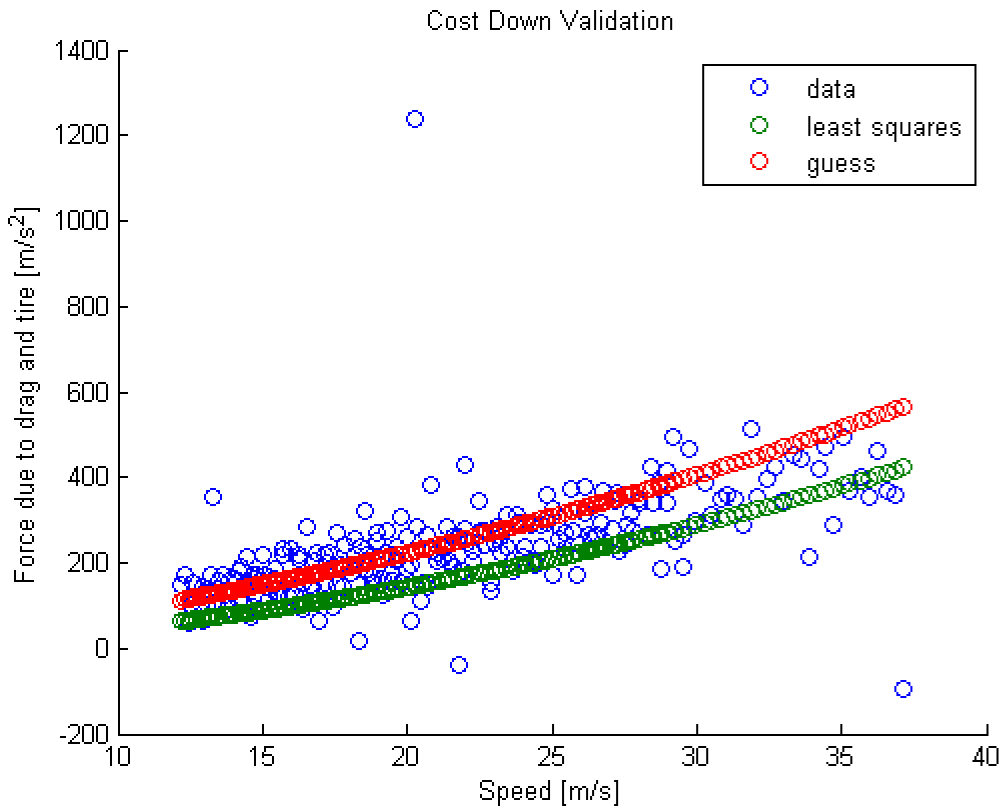

Figure 8: CdA Calibration Validation

The figure above shows the calibrated value and the guess compared to data. The calibrated value is close to the guess and follows the data well.

6.9 Validation

First the Chassis model was validated by checking the lean angle and normal force by changing road gradient and corder radius. Both normal force and lean angle behave correctly

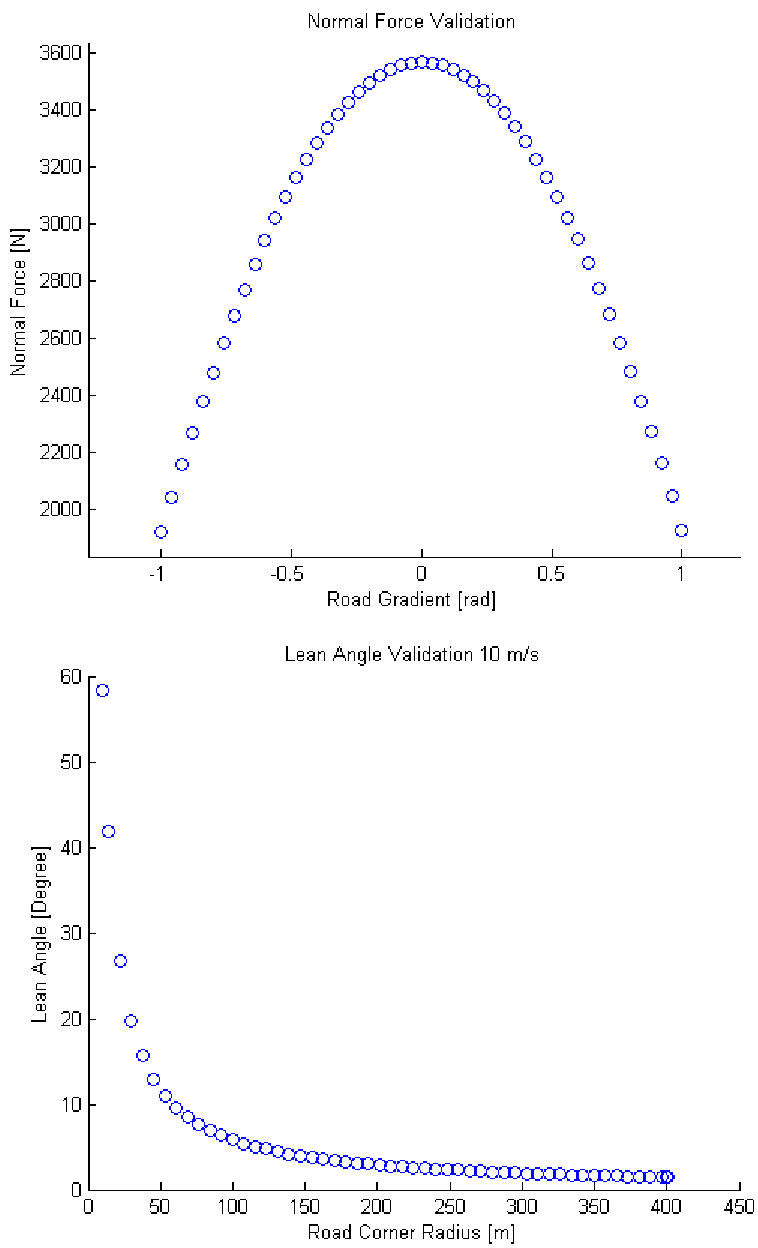

Figure 9: Chassis Validation

Then the Chassis model was validated by simulating a coast down and comparing it against collected data. The data follows the simulation well, but the CdA value needs better calibrated

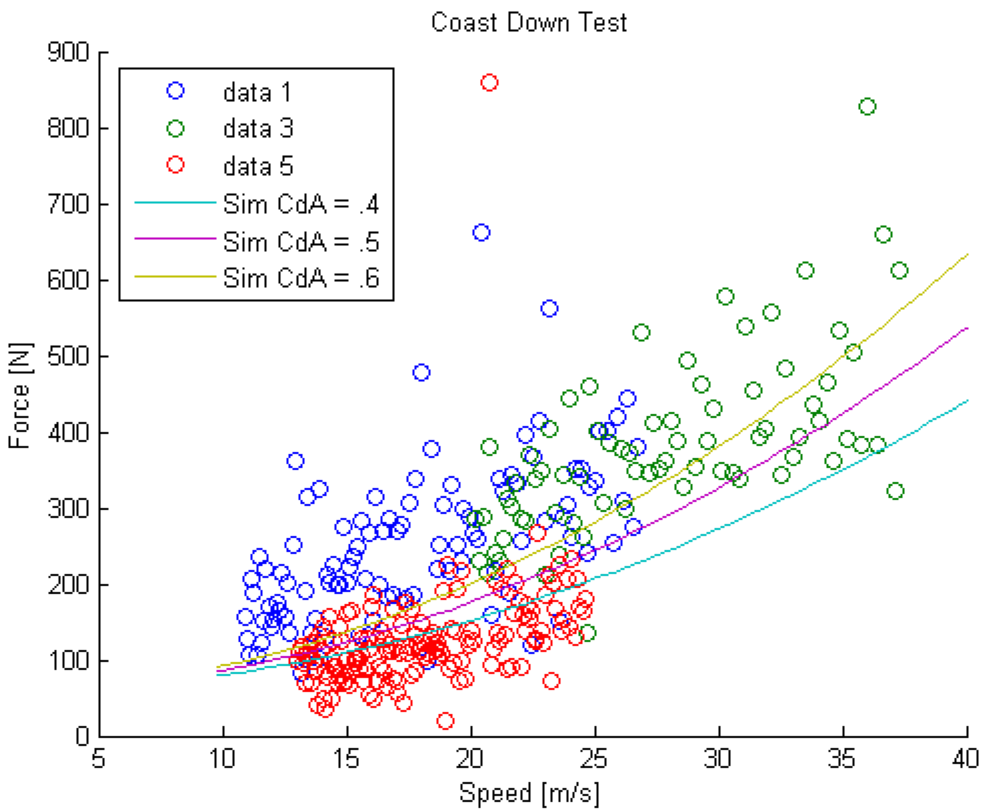

Figure 10: Chassis Validation Coast Down

6.10 Validation

A PI controller was added to the vehicle model as a whole to control for speed. The test shows the vehicle starting at 20 m/s and going to 40 m/s. The model works well.

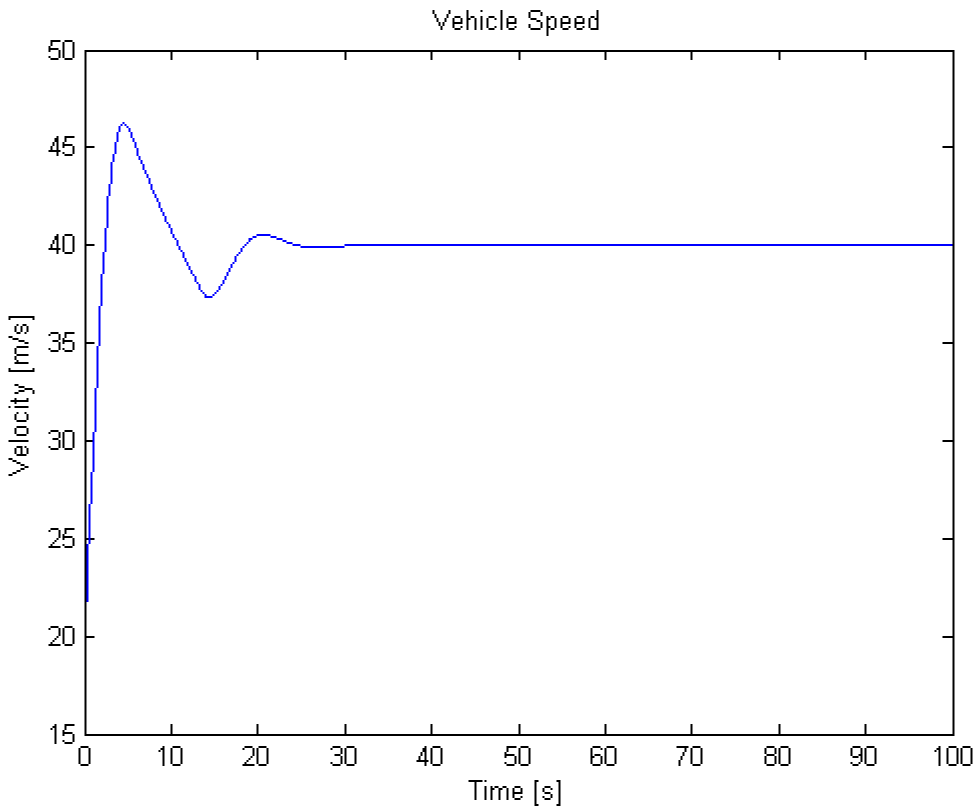

Figure 11: Vehicle Validation Speed

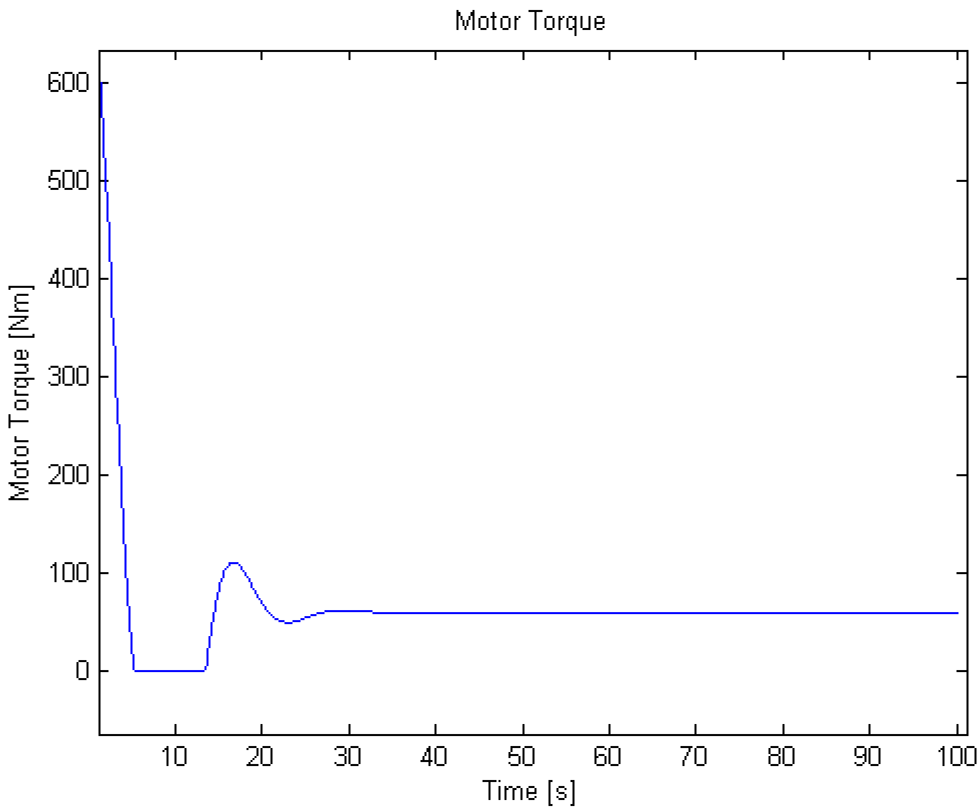

Figure 12: Vehicle Validation Motor Torque

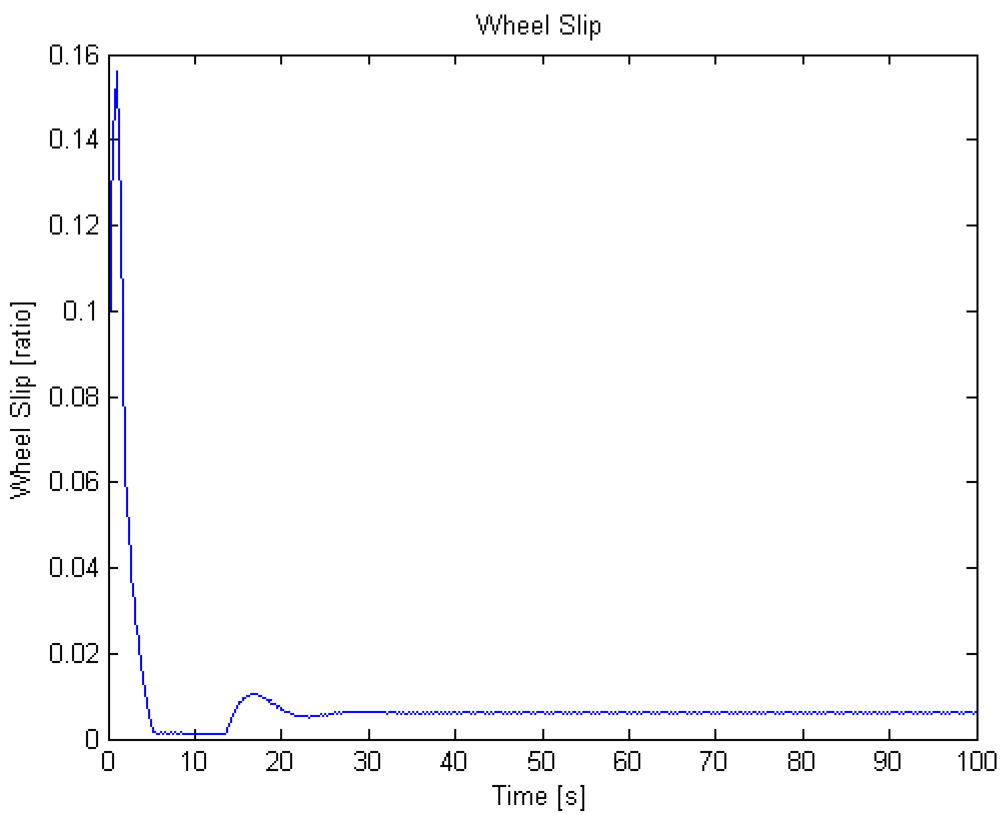

Figure 13: Vehicle Validation Slip

The Model was also validated with coastdown data by making the command velocity 0.

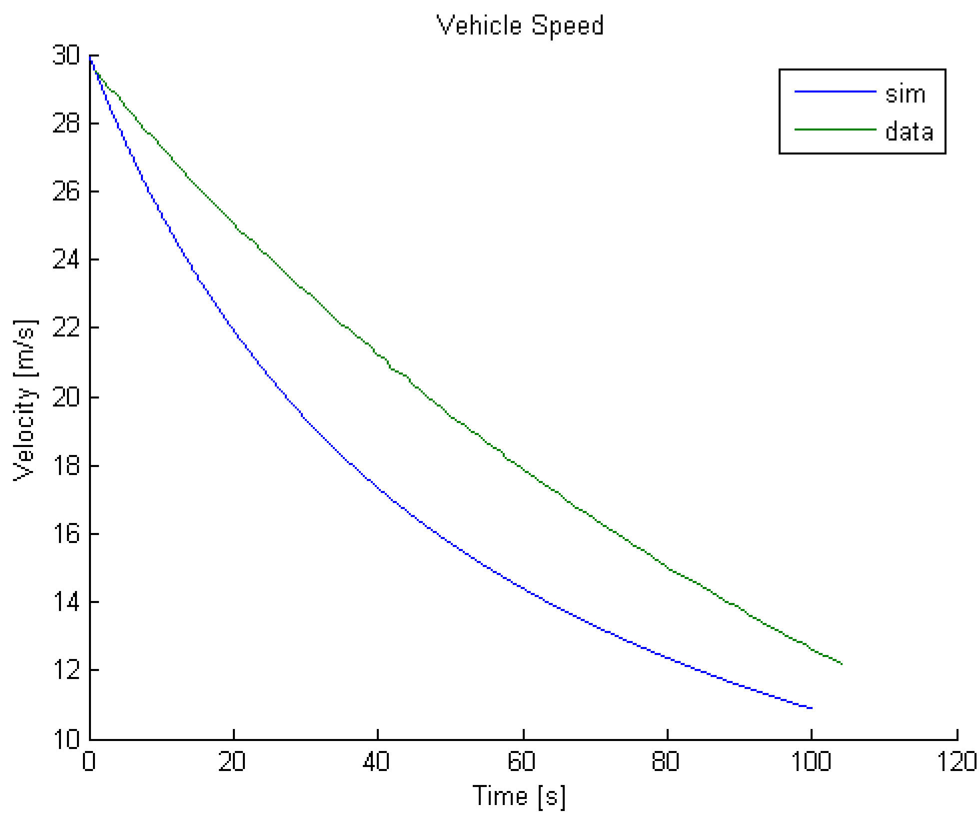

Figure 14: Vehicle Validation Coastdown

7 Environment

This system models the environment of the motorcycle is riding in.

Force directions are defined as longitudinal(long), lateral(lat), and normal(n). Longitudinal is along the direction of the motorcycle (when moving straight). Lateral is orthogonal to Longitudinal axis. Normal 3-D orthogonal to lateral and longitudinal, in general the axis to the road on no incline.

7.1 Inputs and outputs

7.1.1 Inputs

Input	Symbol	Unit
Distance Travel	d	m

7.1.2 Outputs

Output	Symbol	Unit
Environment Forces on Tire[3]	F_t	N[3]
Road Gradient	θ_r	rad
Ambient Temperature	T_{amb}	K
Air Pressure	P	Pa
Air Density	ρ	kg/m^3
Corner Radius	R_c	m

7.2 Background, rationale, modeling strategy

The Environment only models air density, air temperature, and road gradient.

$$\theta_r = \arctan\left(\frac{\frac{d}{dt}h(d)}{\frac{d}{dt}d}\right) \tag{23}$$

$$T_{amb} = T_0 - Lh(d) \tag{24}$$

$$P = P_0\left(1 - \frac{Lh(d)}{T_0}\right)^{\frac{gM}{RL}} \tag{25}$$

$$\rho = \frac{PM}{1000RT} \tag{26}$$

7.3 Parameters

Parameter	Symbol	Unit
Temperature Lapse	L	K/m
Initial Pressure	P_0	Pa
Initial Temperature	T_0	K
Gravity	g	m/s^2
Molar mass of Dry Air	M	kg/mol
Ideal Gas Constant	R	$\frac{J}{mol*K}$

7.3.1 Look up Table

$h(d)$

Type	Description	Symbol	Unit
Input	Distance Travel	d	m
Output	height	n/a	m

$R_c(d)$

Type	Description	Symbol	Unit
Input	Distance Travel	d	m
Output	Corner Radius	n/a	m

7.4 Assumptions

- The air is dry
- Temperature lapse rate right is correct (no inversion)

7.5 Validation

To validate the road gradient the Isle of Man altitude map was supplied to the model and the road gradient was plotted. To validate air density data from Colorado was compared to a simulated data.

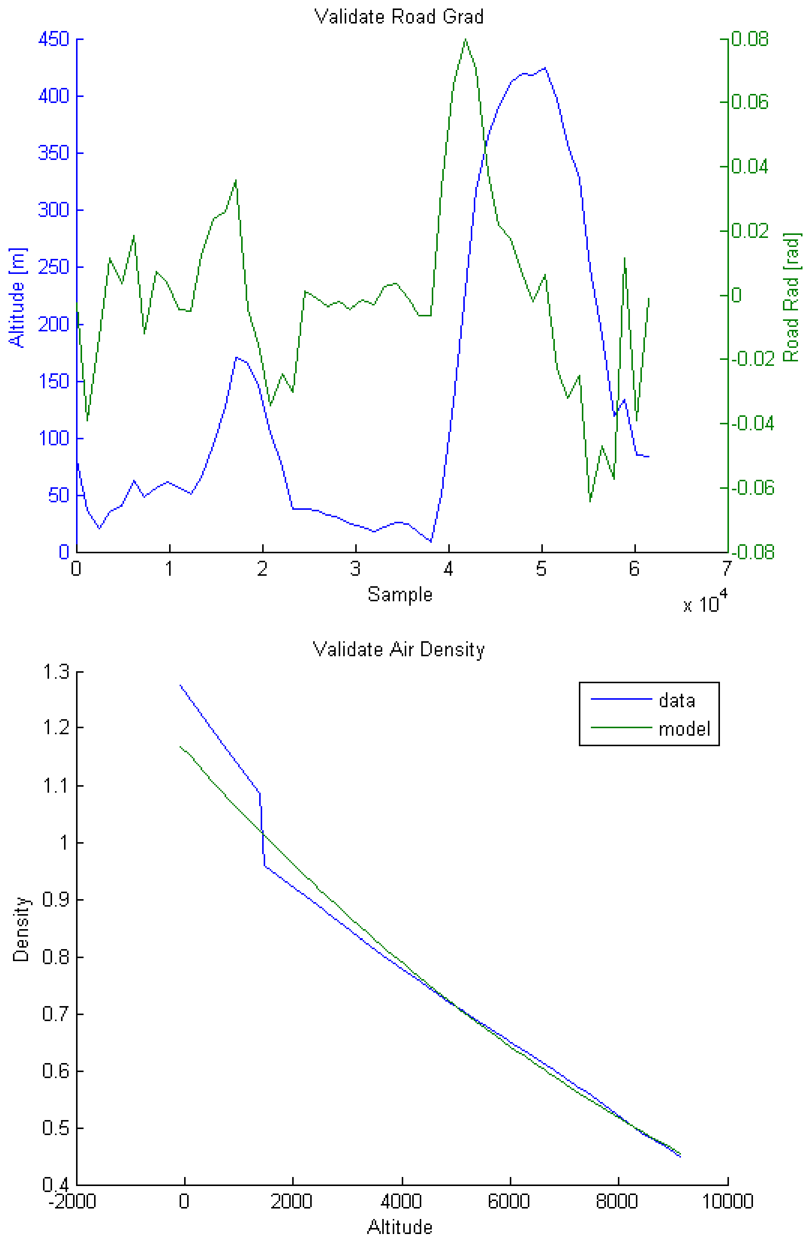

Figure 15: Environment Validation

8 Motor controller

8.1 Block diagram

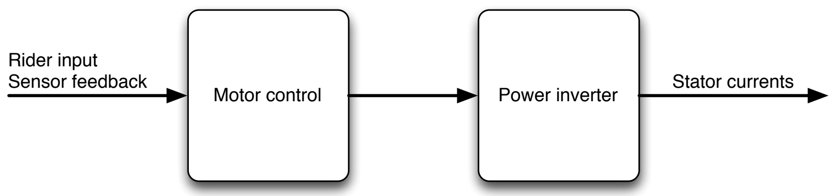

Figure 16: Motor controller - block diagram.

8.2 Inputs and outputs

8.2.1 Inputs

Signal	Symbol	MATLAB variable	Unit
Commanded motor current		`Motor_Current_Command`	%
Commanded motor velocity		`Motor_Velocity_Command`	rpm
Commanded DC bus current		`Bus_Current_Command`	%
Motor speed	ω_m	`omega_m`	rad/sec
Motor temperature	T_m	`T_m`	°C
DC bus voltage	V_{dc}	`Vdc`	V

8.2.2 Outputs

Signal	Symbol	MATLAB variable	Unit
Motor q-axis current	I_q	`Is.Iq`	rms amps
Motor d-axis current	I_d	`Is.Id`	rms amps

8.3 Background, rationale, modeling strategy

The motor controller model is actually two separate models: the motor controller, which makes control decisions based on rider commands and sensor feedback; and the power inverter, which is responsible for power conversion between DC and three-phase AC power.

For this model, a simplified motor controller was developed based on documentation and working knowledge of the Tritium WaveSculptor 200, a commercially available unit used by the team. The power inverter was modeled as a lossless feedthrough pending the availability of efficiency data.

8.3.1 Motor control

A number of strategies for permanent-magnet synchronous motor (PMSM) control are available. Most modern strategies are based on space-vector control using the Park reference frame transformation. At their core, these strategies have the goal of determining the magnitude and phase of the motor current that should be used to achieve a given torque value.

The WaveSculptor 200 uses so-called "$I_d = 0$" control, where the phase angle between the motor's back-emf and the motor phase current is controlled to zero. In the most common configuration, the the rider throttle input is mapped linearly to the motor torque command, which in turn is mapped linearly onto total motor current.

$$I_s = \sqrt{I_q^2 + I_d^2} = I_{s,max} \times \text{(Rider throttle command)} \tag{27}$$

Since $I_d = 0$, the rider throttle input is effectively proportional to I_q. The model assumes that the motor controller has perfect control over stator currents.

In addition to the throttle control loop, the WaveSculptor 200 implements five other control loops that can limit motor current:

- Motor velocity
- DC bus voltage
- DC bus current
- Power inverter temperature
- Motor temperature

In the normal operation of the motor controller, each of these control loops runs simultaneously and generates a stator current command. A supervisory control loop chooses the smallest of all the current commands, and sends it to the inverter.

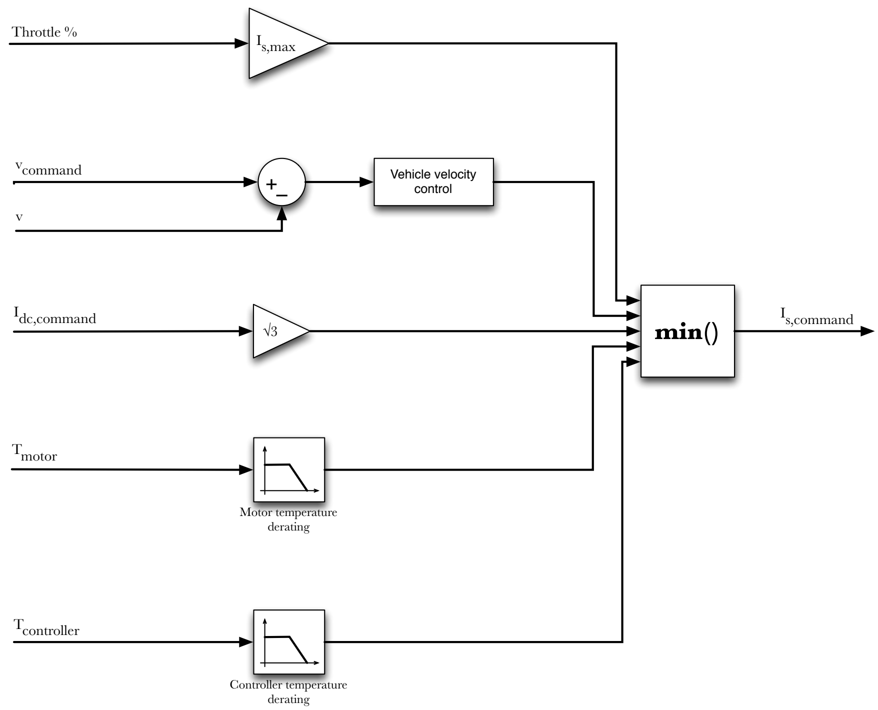

Figure 17: Motor control block diagram.

No calibration or validation data was available for most of the loops, and they do not typically affect motor controller operation. In this model, only the motor temperature and throttle command loops were implemented.

8.3.1.1 Motor temperature derating The motor temperature control loop affects the motor current by linearly derating the current once the motor temperature reaches a programmed value. Figure 18 shows the behavior of the motor temperature control loop.

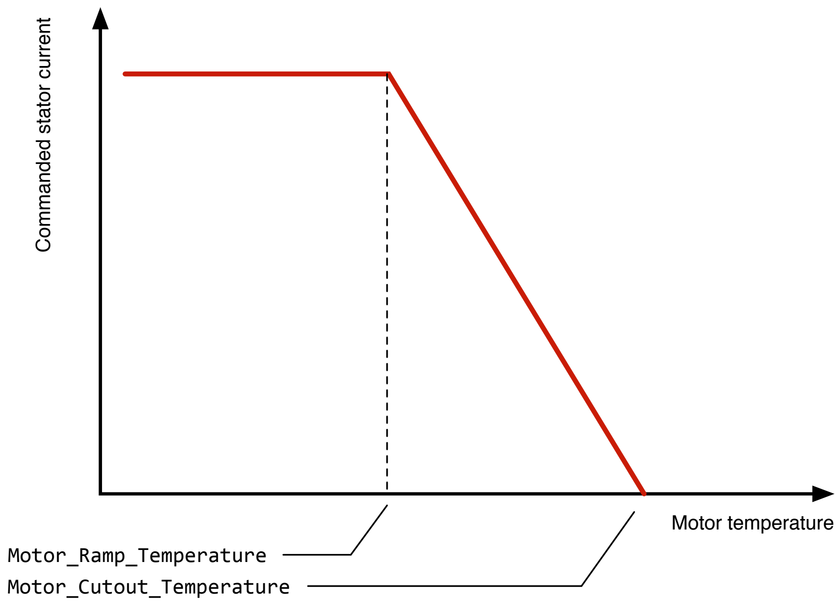

Figure 18: Motor temperature derating behavior.

8.4 Parameters

Parameter	Symbol	MATLAB variable	Unit
Maximum motor current	$I_{s,max}$	`Sine_Current_Limit`	rms amps
Motor temperature ramp point		`Motor_Ramp_Temperature`	°C
Motor temperature cutout point		`Motor_Cutout_Temperature`	°C

8.5 Assumptions

- The power inverter is currently assumed to be lossless (this will be updated after better calibration data is obtained).
- The model assumes that the power inverter has perfect control over the phase currents. This assumption is valid over most of the operating range, but at high motor speeds especially, it may not be true.

  Implementing a more accurate inverter model - specifically, a model that models switching dynamics - would improve the accuracy of the simulation, but this is a complex model operating on a time scale much slower than that of the overall simulation.

- The model currently assumes that the motor controller is cooled well enough that the inverter temperature will never limit operating performance.

9 Battery pack

9.1 Block diagram

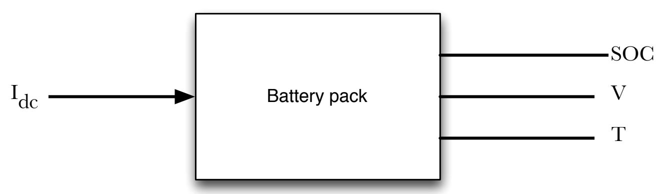

Figure 19: Battery pack block diagram

9.2 Inputs and outputs

9.2.1 Inputs

Signal	Symbol	MATLAB variable	Unit
DC current	I_{dc}	`Idc`	A

9.2.2 Outputs

Signal	Symbol	MATLAB variable	Unit
State of charge	SOC	`SOC`	%
Terminal voltage	V_{dc}	`V`	V

9.3 Background, rationale, modeling strategy

9.3.1 Electrical model

Each battery cell is modeled as an equivalent circuit:

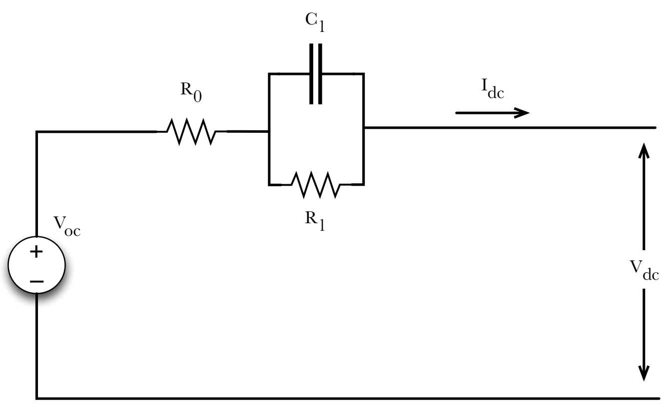

Figure 20: Battery cell equivalent circuit

where:

V_{oc}	is the battery open-circuit voltage in volts
R_0	is the battery zero-order resistance (AC resistance) in ohms
R_1	is the battery first-order resistance in ohms
C_1	is the battery first-order capacitance in farads

The battery open-circuit voltage, V_{oc}, is a function of the remaining cell capacity Q, and is represented by a lookup table.

For a series circuit composed of n identical battery cells, the terminal voltage of the series circuit is $n \times V_{oc}$.

9.3.2 Thermal model

The battery pack thermal model is not implemented. It is assumed that the battery open-circuit voltage has no temperature dependence.

9.4 Parameters

Parameter	Symbol	MATLAB variable	Unit
Initial stored charge	Q_0	`Q_0`	coulomb
Number of series cells	n	`n`	
Zero-order series resistance	R_0	`R0`	ohm
First-order capacitance	C_1	`C1`	farad
First-order resistance	R_1	`R1`	ohm
Open-circuit voltage	V_{oc}(SOC)	`Voc`	volt
Open-circuit voltage lookup: state-of-charge breakpoints		`Voc.SOC`	
Open-circuit voltage lookup: voltage data		`Voc.V`	volt

9.5 Assumptions

This battery pack model assumes that:

- None of the equivalent-circuit parameters are affected by temperature.
- The charging and discharging open-circuit voltage profiles are identical.

9.6 Battery pack model performance

The battery pack model was calibrated against discharge data from three sample cells. The model, with the calibrated parameters, was simulated using a current profile representative of a competition load. The model output was compared to test data.

9.6.1 Calibration

The battery pack model was calibrated using data from two discharge tests.

9.6.1.1 Open-circuit voltage First, a low-rate discharge test was used to approximate the battery open-circuit voltage $V_{\mathrm{oc}}(\mathrm{SOC})$. In this test, the battery was charged to a terminal voltage of 4.2 V (constant-voltage, C/20 cutoff), then discharged at 0.1C until the terminal voltage reached 2.5 V.

Figure 21 shows an example of 0.1C discharge test voltage data. The test was repeated three times on a sample cell at 20°C.

The actual current out of the cell recorded during testing was integrated to calculate the capacity discharged as a function of time. Equation 29 gives the expression of battery state-of-charge, where C is the battery capacity in amp-hours, and $I(T)$ is the measured discharge current from the cell, in amps.

$$C = \int_0^{t_{\mathrm{end}}} I(T)\,\mathrm{d}T \tag{28}$$

$$\mathrm{SOC}(t) = 1 - \frac{1}{C}\int_0^{t} I(T)\,\mathrm{d}T \tag{29}$$

$\mathrm{SOC}(t)$ and $V(t)$, the measured terminal voltage of the cell, were used to create a 1D lookup table of the cell's open-circuit terminal voltage.

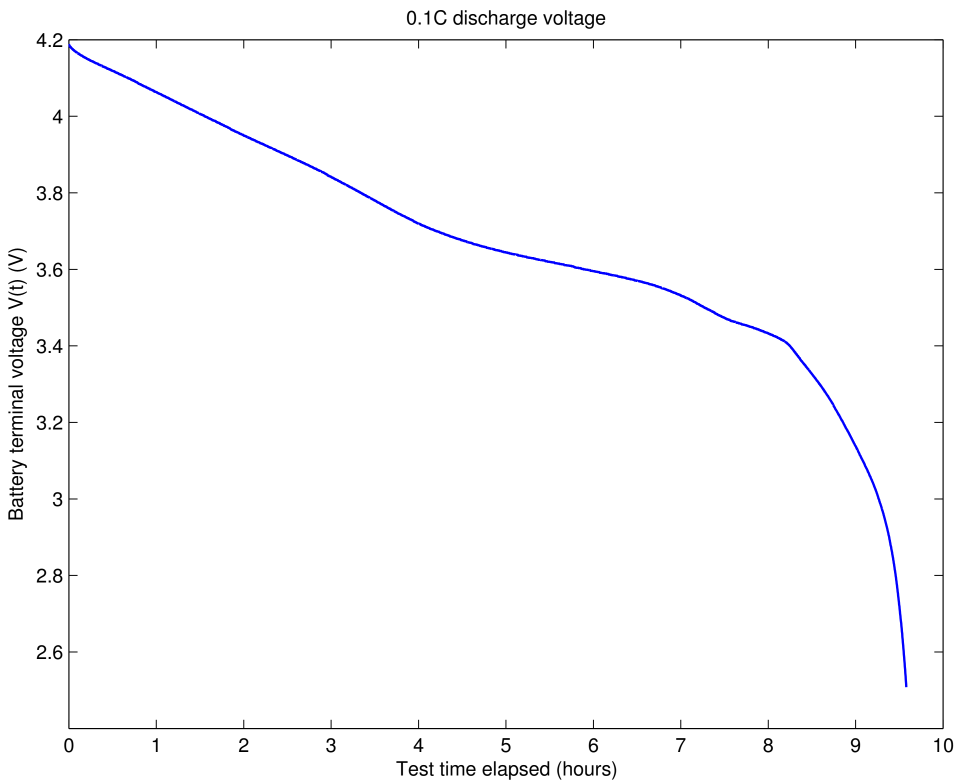

Figure 21: Battery open-circuit voltage discharge test - voltage profile

9.6.1.2 Equivalent-circuit parameters After the open-circuit voltage had been calibrated and loaded into the model, data from a pulse discharge test was used to calibrate the equivalent-circuit parameters R_0, R_1, and C_1. The discharge test current profile, shown in Figure 22, consists of sets of high-magnitude current pulses at different states of charge.

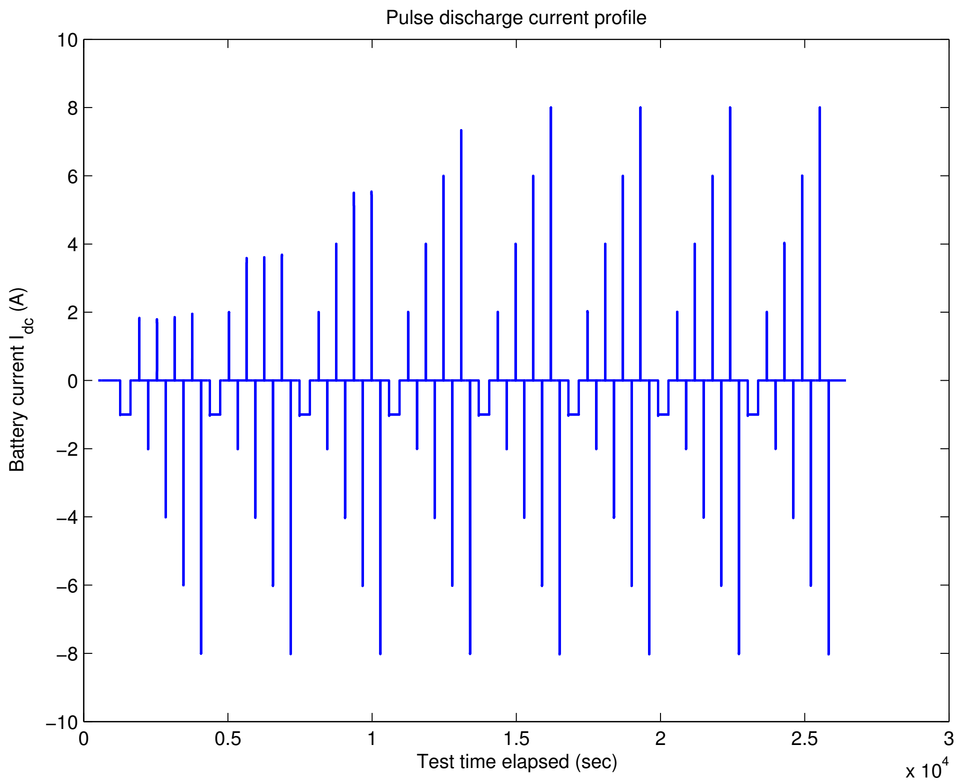

Figure 22: Pulse discharge test - battery current profile

Using the discharge current profile in Figure 22, the model parameters in Table 1 were calibrated using the MATLAB `fminsearch()` function.

Table 1: Calibrated parameters in battery pack model

Symbol	Parameter	Initial guess value	Calibrated value
R_0	Zero-order resistance	25 mΩ	24.5 mΩ
R_1	First-order resistance	25 mΩ	24.1 mΩ
C_1	First-order capacitance	1000 farad	982.9 farad
R_0	Capacity	2.5 Ah	2.35 Ah

The capacity parameter was included as a calibrated parameter because initial tests with the model displayed what appeared to be a scaling error in the model's open-circuit voltage term.

9.6.2 Model validation and quality of fit

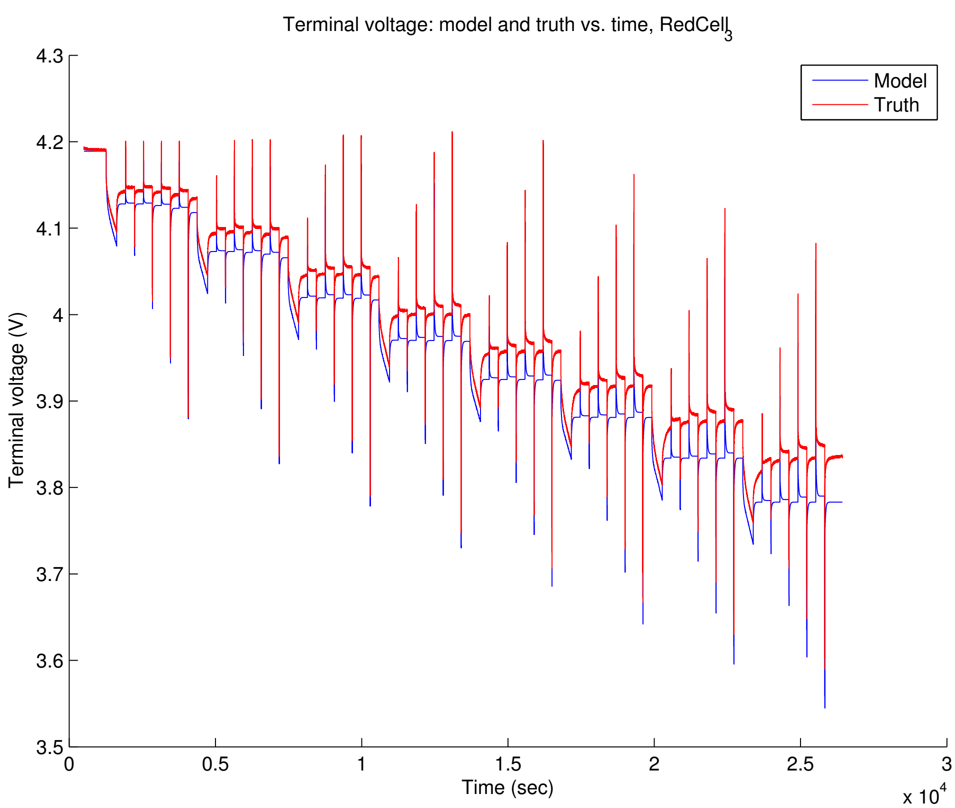

Figure 23: Model agreement - pulse discharge profile

Figure 23 shows the model behavior compared to a pulse discharge test for a different cell. Although the dynamic response of the circuit (RC network) is close to accurate, the open-circuit behavior with SOC displays increasing error as the test continues (and cell state-of-charge decreases).

Measurement error is a likely source of this issue. In Jan. 2014, the calibrations will be repeated with more precise equipment and a larger sample size.

10 Motor

10.1 Block diagram

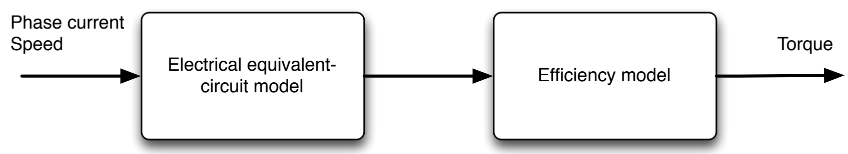

Figure 24: Motor model block diagram.

10.2 Inputs and outputs

10.2.0.1 Inputs -

Signal	Symbol	MATLAB variable	Unit
Stator current	I_s	`Is`	rms amps
Stator current (quadrature axis)	I_q	`Is.Iq`	rms amps
Stator current (direct axis)	I_d	`Is.Id`	rms amps
Motor speed	ω	`omega`	rad/sec

10.2.0.2 Outputs -

Signal	Symbol	MATLAB variable	Unit
Motor torque	τ	`tau`	N m

10.3 Background, rationale, modeling strategy

The motor model has two main components. The torque-generation component is based on an electrical equivalent-circuit model and a 2D efficiency map derived from motor-specific testing. There is also a simple thermal model based on a constant thermal resistance.

10.3.0.3 Torque model The motor's electrical behavior can be represented using two equivalent circuits using the Park transformation [?]. The result of the transformation for a 3-phase permanent-magnet synchronous motor (PMSM) is the dq motor model. Figure 25 shows the constant-parameter dq equivalent circuits.

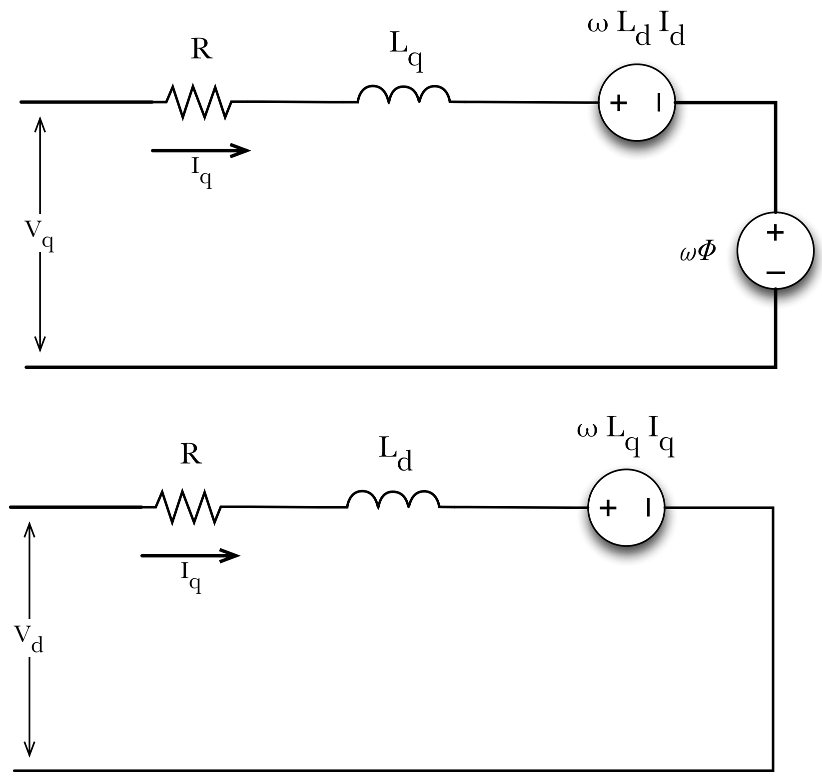

Figure 25: Motor model - dq equivalent circuits.

The motor parameters are assumed to be constant with respect to stator currents and temperatures.

By the equivalent circuits in Figure 25, the stator voltages V_d and V_q can be found:

$$V_d = Ri_d + L_d \frac{di_d}{dt} + \omega L_q i_q \tag{30}$$

$$V_q = Ri_q + L_d \frac{di_q}{dt} + \omega L_d i_d - \omega \phi \tag{31}$$

Calibrating the above model using experimental data could be difficult because of the $\frac{di}{dt}$ terms. After evaluating typical values of $\frac{di_q}{dt}$ recorded in previous testing, the maximum observed value (about -3070 A/sec) was small enough that the inductance term $L\frac{di}{dt}$ could be neglected without introducing significant error. With this modification, 30 and 31 become

$$V_d = Ri_d + \omega L_q i_q \tag{32}$$
$$V_q = Ri_q + \omega L_d i_d - \omega\phi \tag{33}$$

It is assumed that the power inverter has perfect control over the d- and q-axis currents I_d and I_q. With the known stator current $I_s = \sqrt{I_d^2 + I_q^2}$, the input power can be found:

$$P_e = I_s \times V_s \tag{34}$$

Then the motor "electrical torque" can be found:

$$\tau_e = \frac{P_e}{\omega} \tag{35}$$

The motor output torque is the product of τ_e and the motor efficiency:

$$\tau = \eta(\tau_e, \omega) \times \tau_e \tag{36}$$

10.4 Parameters

Parameter	Symbol	MATLAB variable	Unit
Q-axis inductance	L_q	`Lq`	H
D-axis inductance	L_d	`Ld`	H
Equivalent-circuit series resistance	R	`R`	ohm
Permanent-magnet flux linkage	ϕ_m	`phi`	V s rad^{-1}
Motor efficiency	η	`eta`	
Motor efficiency lookup: stator current breakpoints		`eta.Is`	rms amps (I_s)
Motor efficiency lookup: motor speed breakpoints		`eta.omega`	rad/sec
Motor efficiency data		`eta.eta_m`	

10.5 Assumptions

The motor electrical model is significantly simplified.

- The model assumes that the motor magnetization is linear (i.e., the motor never saturates).
- The model assumes that the motor inductances L_d and L_q are constant. In fact, details about behavior of Lq with Id
- The motor efficiency model assumes that motor efficiency η depends only on motor speed and electrical torque.

References

10.6 Motor model performance

The motor model was calibrated against experimental data from vehicle testing in 2014. Although no torque data was available to validate the motor torque model, intermediary signals were measured and validated.

10.6.1 Calibration

10.6.1.1 Efficiency model The motor efficiency model was derived from an efficiency plot provided by the manufacturer.

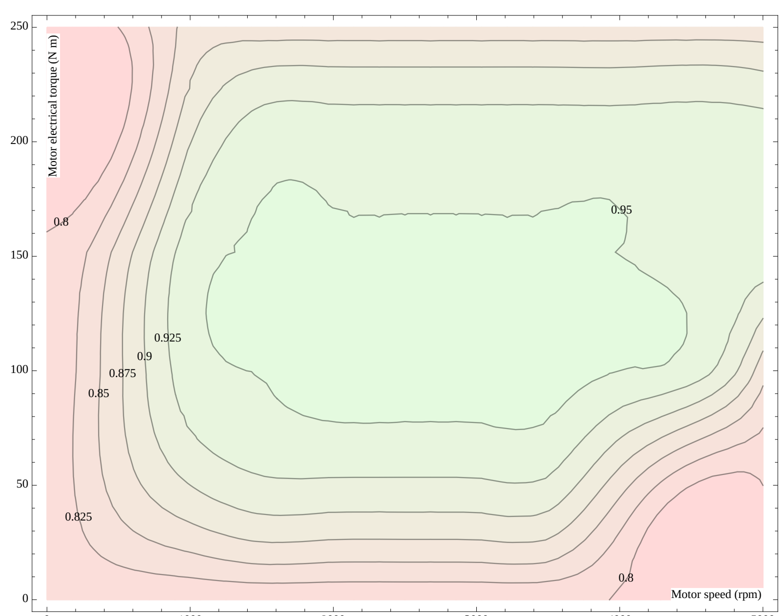

Figure 26: Enstroj EMRAX 228HV - motor efficiency over operating range.

Table 2: Calibrated parameters in motor model

Symbol	Parameter	Initial guess value	Calibrated value
ϕ	Motor flux linkage	1 V rad/sec	-1.051 V rad/sec

The efficiency plot is based on variables which are convenient to measure using an engine dynamometer and a power inverter. In future testing, the efficiency function will be derived from steady-state experimental data.

10.6.1.2 Equivalent-circuit parameters The motor manufacturer specifies values for L_d, L_q, and R. Since the model performance was good without additional calibration of these values, they were used unchanged.

The only equivalent-circuit parameter that was calibrated in the model was ϕ, the permanent-magnet flux linkage.

Since there was no calibration data available that included a motor torque output, the model was calibrated against a subset of the experimental data from RW-2x coastdown testing (the same dataset used for validating the overall vehicle model). The coastdown test data was chosen because it covered a wide range of vehicle speeds and motor currents.

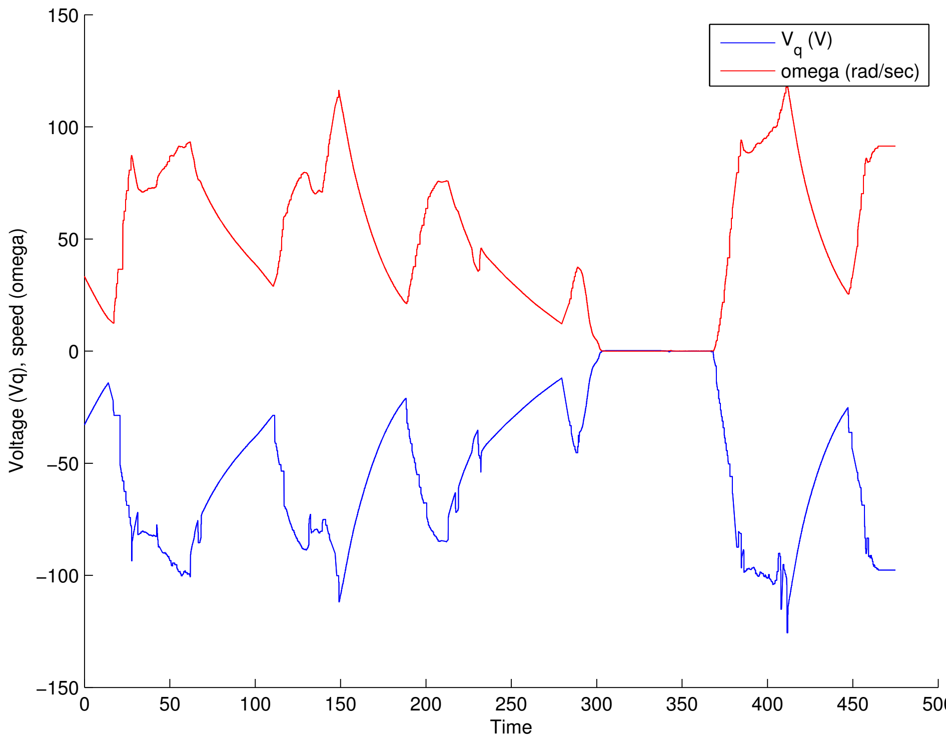

Figure 27: Coastdown dataset used for ϕ calibration.

10.6.2 Model validation and quality of fit

The model was validated using the full coastdown dataset from RW-2x vehicle testing. Motor output torque was not directly measured during this testing, so the validation focused on accurate prediction of the motor equivalent-circuit voltages V_d and V_q.

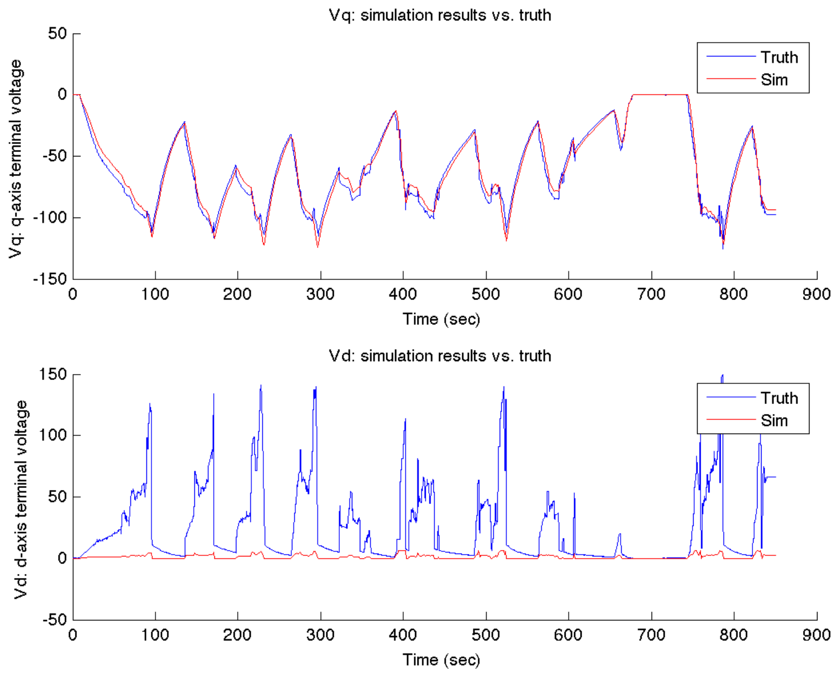

Figure 28: Motor model performance: simulated V_q, V_d vs. measured data.

The model performance for V_q is very good, even during transients.
The model does not predict V_d well. Some possible sources of this error include:

- Large applied voltages by the motor controller in an attempt to control for $I_d = 0$
- Larger than expected coupling between the d- and q-axis circuits
- Larger than expected $\frac{di_d}{dt}$ or L_d