rEnginePowerFactor	A global engine power output scaling.
PEnginePowerOffset	Global offset to engine power output.
MEngineMapData	Either an arbitrarily large set of nEngine, rThrottle, MEngine triplets, defining engine power output over the operating range; or an object containing *nEngineBasis* and *rThrottleBasis* (together defining a grid of points for torque and fuel rate), *MEngineTable* and *dmFuelTable* (each defining their respective values at the points of the grid given by the two basis vectors).
dmFuelMapData	An arbitrarily large set of nEngine, rThrottle, dmFuel triplets, defining engine fuel consumption (in kg/s) over the operating range (overridden by *dmFuelTable* if present)
mFuelMaxPerLap	Fuel mass allowed per lap.
PTurboHarvestMax	Maximum electrical power harvestable from the turbo.
eTurboHarvest	Harvest efficiency of the turbo.
PTurboUnloaded	Electrical power required to keep the turbo spinning while off-throttle.
rApplication	Proportion of torque to be applied at application point.

Electric

This section of the powertrain deals with electrical power deployment and harvest, through an electric motor, and storage in a generic electrical energy store. Any number of Electirc motors can be specified, each given a name.

Motors

In order to accommodate the drivetrain layouts of as many different cars as possible, any number of electric motors can be added to the powertrain, each with its own characteristics and its own torque application position. To allow users to keep track of multiple electric motors, each one is given a name. It is this name which is used when producing the output channels. For example, if you have one motor and name it *FEMotor*, then you will find the channels *PFEMotorDeployment* and *PFEMotorHarvest* in your simulation output (amongst others).

Two types of electric motor parameterisation are available in the model (as distinguished in the drop-down meny next to *Deployment* in the car editor): a simple constant power motor with a fixed maximum power and a constant efficiency, and a variable power motor, with an efficiency map. In the case of the simple motor, the user must simply provide *PMotorMax* and *eMotor*, the maximum motor power, and the motor efficiency respectively. In the case of the more complex model, the user can provide maps for maximum torque at different speeds, and efficiency maps for both deployment and regenerative braking. In both cases (simple and complex) the user can optionally define *PDeploymentRegulatoryLimit* and *PHarvestRegulatoryLimit*, these parameters impose power limits on the motor for deployment and harvest respectively. The point at which the regulatory limit is applied defaults to the output shaft of the motor, but can be specified in the motor's optional parameter *PMotorMeasurementPosition*, which also defines the point at which the powers for the output channels is measured.

Much like the engine maps, the points of the efficiency map do not have to be arranged in a grid, or regularly spaced. The model will fit a smooth surface to interpolate through and between the provided points. Similarly to the engine maps, sharp corners, especially in the motor torque map, should be rounded-off with extra points to achieve a more convex interpolation.

The harvesting of energy through the electric motor is characterised by the maximum harvest power, *PHarvestRegulatoryLimit* and the harvesting efficiency, *eMotorHarvestMapData*. When we control harvesting torque to achieve a target brake balance - the torque demand is governed by the brake pressure and brake balance, and the split between electric motor harvesting torque and mechanical brake torque is given by:

$$PElectricMotorHarvest = min(PHarvestMax, PBrakeDemandR)$$

$$PMechanicalBrake = max(0, PBrakeDemandR - PHarvestMax)$$

In other words, the electric motor gets as much as it can off the rear axle torque, and the brakes make up the rest.

The electric motor can be selected to harvest without targeting a given brake balance, in which case it enables harvesting on the regen paddle (i.e. control of harvest power independent of brake pressure).

A third option "Use harvest map" allows you to specify harvest power as a function of pBrakeF, pBrakeR and/or aSteerAbs.

The remaining parameters govern how much power can be harvested whilst coasting (for example with the Formula E 'regen paddle') and how much charging is done at part throttle (where an ICE can be used to charge the e-store through an electric motor, for example in F1). A full list of electric motor parameters is given in the table below.

Parameter	Definition