1. Introduction

1.1. Background and Motivation

The transportation sector is a major contributor to global fuel consumption and greenhouse gas emissions. Vehicle electrification has emerged as an effective solution to improve energy efficiency and reduce fuel usage. However, fully electric vehicles are constrained by limited driving range and charging infrastructure. Plug-in Hybrid Electric Vehicles (PHEVs) provide a practical compromise by combining electric propulsion with an internal combustion engine for extended range.

In PHEVs, the Energy Management Strategy (EMS) plays a critical role in determining how traction power is split between the electric motor and the engine. This power split directly affects fuel consumption, battery State of Charge (SOC), and overall vehicle fuel economy. Traditional EMS approaches are typically rule-based and rely on fixed heuristics, which do not exploit future driving information or efficiency variations across engine and motor operating points.

This project investigates the use of Unconstrained Linear Time Invariant Model Predictive Control (MPC) with hard coded constraints as an advanced EMS for a P2 parallel PHEV and compares its performance against a baseline 50–50 rule-based controller. The comparison is performed using a backward-looking vehicle model under the Urban Dynamometer Driving Schedule (UDDS).

1.2. Objectives

The objectives of this project are:

- To develop a backward-looking longitudinal vehicle model with realistic nonlinear powertrain components.
- To implement engine and motor models using ADVISOR-based efficiency maps for further improving realism in simulation and results.
- To design and compare two energy management strategies: a 50–50 rule-based controller and a LTI MPC.
- To evaluate controller performance in terms of SOC behavior, fuel consumption, energy losses, and MPG.

2. Vehicle and Powertrain Modeling

2.1. Vehicle Longitudinal Dynamics

A backward-looking longitudinal vehicle model is adopted, in which the required traction power is computed from a prescribed vehicle speed profile. The model determines the power demand necessary to follow the driving cycle, rather than predicting vehicle speed from applied power.

Vehicle Acceleration

Longitudinal acceleration is obtained using a discrete-time approximation:

$$a(k) = \frac{\mathrm{v(k)} - \mathrm{v(k-1)}}{T_s}$$

where $v(k)$is the vehicle speed at time step $k$and $T_s$is the simulation time step.

The resistive forces acting on the vehicle include aerodynamic drag, rolling resistance, and road grade force:

$$F_{drag} = \frac{1}{2} p C_d A_f v^2$$

$$F_{roll} = C_{rr}\ m\ g \cos(\theta)$$

$$F_{grad} = m\ g\ \sin(\theta)$$

where ρ is air density, C_d is the drag coefficient, A_f is the frontal area, C_{rr} is the rolling resistance coefficient, $m$is vehicle mass, $g$is gravitational acceleration, and $\theta$is the road grade angle.

The total torque demand is computed by accounting for inertial and resistive effects, and the required shaft power is obtained through the gearbox and final drive efficiencies.

$$T_{w,req} = mar_w + J_w \dot{\omega}_w + (i_{tot}^2) J_e \dot{\omega}_w + \left(F_{drag} + F_{roll} + F_{grade}\right) r_w$$

where $r_w$is the effective wheel radius, $J_w$and $J_e$are the wheel and engine/motor rotational inertias, respectively, and $i_{\mathrm{tot}} = i_g i_f$is the total gear ratio.

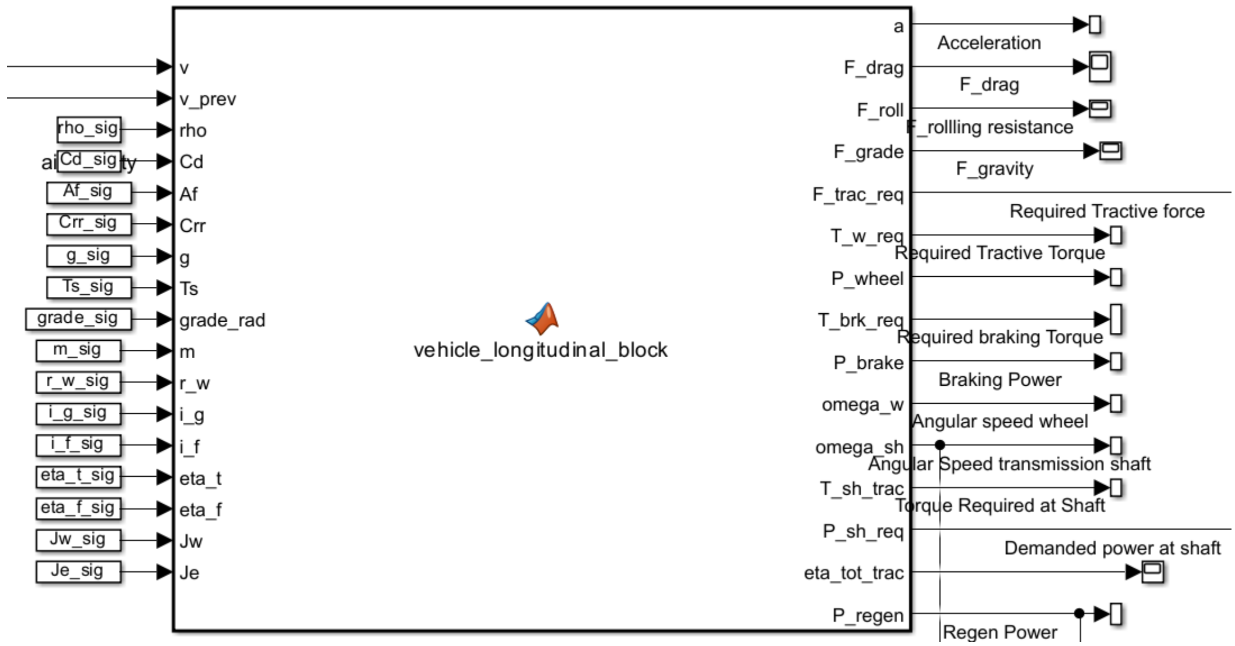
Shaft Power Requirement

The required shaft power is obtained by mapping the wheel power through the transmission and final drive efficiencies:

$$P_{sh,req} = \begin{cases} \dfrac{T_{w,req} \cdot v}{\eta_t\, \eta_f}, & P_{wheel} > 0 \\ 0, & braking \end{cases}$$

$$P_{regen} = \begin{cases} P_{wheel}, & P_{wheel} < 0\ (braking/deceleration)) \\ 0, & only\ traction (no\ regen) \end{cases}$$

where $\eta_t$and $\eta_f$are the gearbox and final drive efficiencies, respectively. During braking events, regenerative braking is now considered after suggestion on the presentation, and hydraulic braking is assumed. Later the increase in SOC is based on the P_{regen} is assumed to be at constant efficiency of 0.65.



2.2. Engine Model

The internal combustion engine is modeled using ADVISOR data from the **FC_SI63_emis** dataset, which represents a 1.9 L naturally aspirated spark-ignition engine suitable for hybrid vehicle operation. This engine is chosen as it can satisfy the power demand even when the battery is exhausted. The model is implemented as a steady-state, lookup-table-based representation (through linear interpolation in 1D and 2D) that captures engine fuel consumption and torque limits as functions of operating speed and load.

The engine model consists of:

- Engine speed breakpoints ranging from **701 to 5500 rpm**.
- Torque breakpoints from **0 to approximately 100 Nm**.
- A two-dimensional fuel consumption map expressed in **grams per second (g/s)**.
- A maximum torque envelope is defined as a function of engine speed.

Given a requested mechanical power $P_{e,\mathrm{req}}$ and shaft angular speed ω, the commanded engine torque is first computed as:

$$T_{raw} = \frac{P_{e,req}}{\omega_{safe}}$$

where ω_{safe} is a minimum enforced operating speed to avoid numerical instability and to ensure idle operation.

The requested torque is then saturated by the maximum available torque:

$$T_e = \min\left(T_{raw}, T_{max}(\omega)\right)$$

The resulting mechanical output power is given by:

$$P_e = T_e \omega_{safe}$$

Fuel mass flow rate is obtained from the two-dimensional fuel consumption map:

$$\dot{m}_f = f_{map}(\omega, T_e)$$

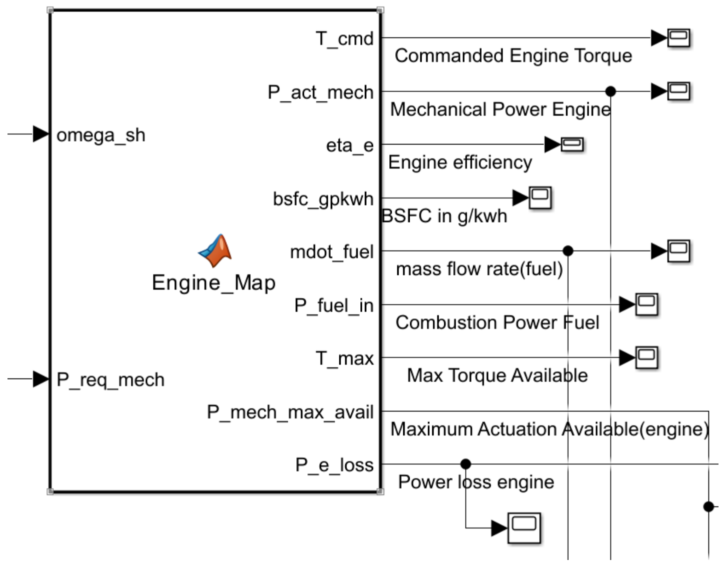
The corresponding fuel power input is computed using the fuel lower heating value (LHV):

$$P_{fuel,in} = \dot{m}_f \cdot LHV$$

Finally, the engine efficiency is defined as the ratio of mechanical output power to fuel input power:

$$\eta_e = \frac{P_e}{P_{fuel,in}}$$

The engine idle power consumption is addressed in this block. At very low load conditions, a fixed idle fuel consumption is enforced to represent engine idling behavior, during which mechanical output power is zero while fuel is consumed.



2.3. Motor Model

The electric traction motor is modeled using ADVISOR **MC_PM49** data, corresponding to a **49 kW permanent magnet motor** commonly used in hybrid electric vehicle applications. The motor is represented using a steady-state, lookup-table-based model in a similar way as engine model that captures efficiency and torque limits as functions of operating speed and load.

The motor model includes:

- A speed operating range from **0 to 8500 rpm**.
- A torque range from approximately **176 Nm to 274 Nm**.
- A two-dimensional efficiency map defined over speed and torque.
- A maximum continuous motoring torque curve as a function of motor speed.

Given a requested mechanical power $P_{m,\mathrm{req}}$ and shaft angular speed ω, the requested motor torque is computed as:

$$T_{\mathrm{raw}} = \frac{P_{m,\mathrm{req}}}{\omega_{safe}}$$

The commanded motor torque is then saturated by the maximum available motoring torque:

$$T_m = \min\left(T_{raw}, T_{max,m}(\omega)\right)$$

The resulting mechanical output power delivered by the motor is:

$$P_m = T_m \omega_{safe}$$

Motor efficiency η_m is obtained from the two-dimensional efficiency lookup table:

$$\eta_m = \eta_{\mathrm{map}}(\omega, T_m)$$

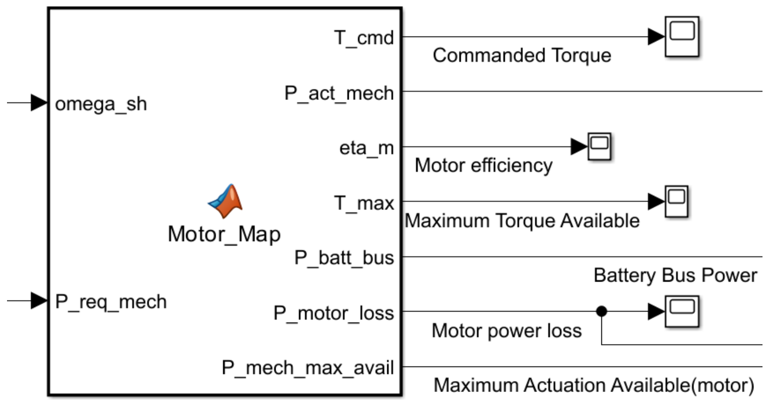
The electrical power drawn from the battery DC bus during traction operation is computed as:

$$P_{\text{batt,bus}} = \frac{P_m}{\eta_m}$$

The motor and inverter losses are then given by:

$$P_{\text{loss,m}} = P_{\text{batt,bus}} - P_m$$

Only the motoring (traction) operation is considered in the motor model, whereas the regenerative operation is handled through available regenerative power signal from vehicle dynamics block to battery SOC model block.



2.4. Battery SOC Model

The battery is modeled using a power-based state-of-charge (SOC) formulation with constant nominal voltage and capacity. The model captures battery energy depletion during traction operation, regenerative energy recovery assuming constant efficiency for regeneration during braking and enforces SOC limits to protect the battery from over-discharge and over-charge.

The SOC dynamics are expressed in discrete time as:

$$SOC_{K+1} = SOC_K - \frac{P_{\text{cell,K}}\, T_s}{E_{\text{nom}}}$$

where T_s is the simulation time step and E_{nom} is the nominal battery energy capacity.

During traction operation, the chemical power drawn from the battery cells is related to the DC bus power by the battery efficiency:

$$P_{\text{cell}} = \frac{P_{\text{batt,bus}}}{\eta_{\text{batt}}}$$

where $P_{\text{batt,bus}}$ is the electrical power supplied to the motor inverter and η_{batt} is the battery discharge efficiency.

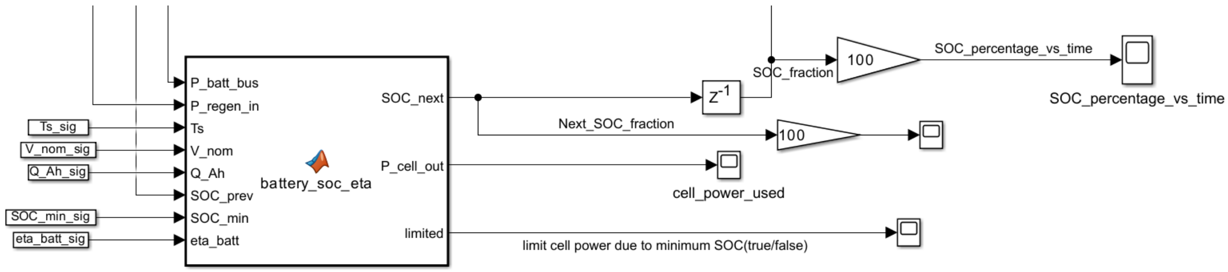
During regenerative operation, negative DC bus power is applied to the battery with a regenerative efficiency factor which leads to increase in SOC:

$$P_{cell} = \eta_{regen} . P_{regen} , \qquad P_{regen} < 0$$

To ensure safe operation, the SOC is constrained to remain above a predefined threshold:

$$SOC_{min} \le SOC_{K+1} \le SOC_{max}$$

If the predicted SOC violates these limits, the battery power is clipped such that the SOC reaches the corresponding bound within the current time step. When discharge power is limited due to low SOC, the energy management controller reallocates the remaining power demand to the engine, if available.



2.5. Fuel Tank Model

Fuel consumption is accumulated using the engine fuel mass flow rate obtained from the engine model. The total fuel mass consumed over time is computed as:

$$m_{f,\text{used}}(t) = \int_0^t \dot{m}_f\,(\tau)\,d\tau$$

The consumed fuel mass is converted to fuel volume using a constant gasoline density and standard unit conversions:

$$m_{\text{kg}} = \frac{m_{\text{g}}}{1000}$$

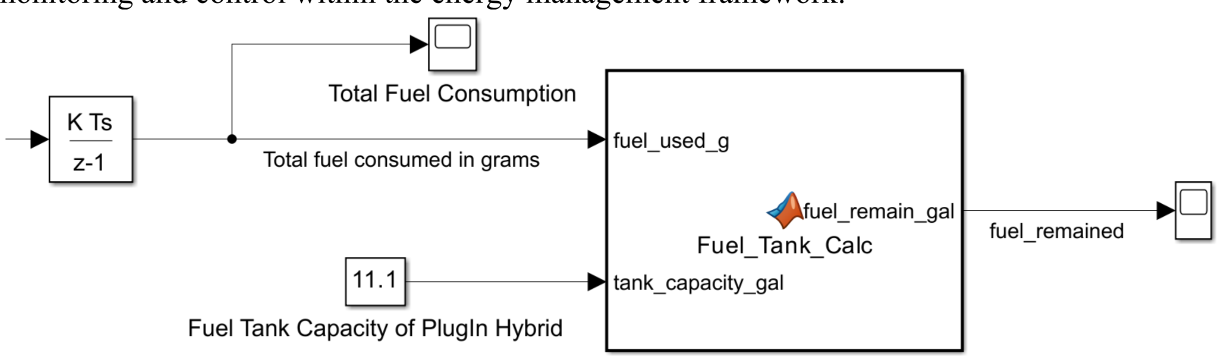
$$V_L = \frac{m_{\text{kg}}}{\rho_{\text{gas}}}$$

$$V_{\text{gal}} = \frac{V_L}{3.78541}$$

The remaining fuel volume in the tank is then given by:

$$\text{Fuel}_{\text{remain,gal}} = \max\left(\text{Tank}_{\text{cap}} - V_{\text{gal}},\, 0\right)$$

This remaining fuel estimate is used to enforce fuel-depletion constraints that is non negative remaining fuel and provides consistent estimation of fuel depletion for use in the system level monitoring and control within the energy management framework.



2.6. MPG Calculation

Fuel economy is evaluated using both instantaneous and average miles-per-gallon (MPG) metrics. Vehicle speed is converted from meters per second to miles per hour:

$$v_{\text{mph}} = 2.236936\, v_{\frac{\text{m}}{\text{s}}}$$

The engine fuel mass flow rate is converted from grams per second to gallons per hour:

$$\dot{m}_{f,g/h} = 3600\,\dot{m}_{f,g/s}$$

$$\dot{V}_{gal/h} = \frac{\dot{m}_{f,g/h}}{2819} = \frac{3600\,\dot{m}_{f,g/s}}{2819}$$

Instantaneous fuel economy is computed as:

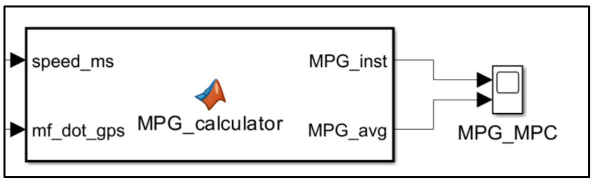
$$\text{MPG}_{\text{inst}} = \frac{v_{\text{mph}}}{\dot{V}_{gal/h}}$$

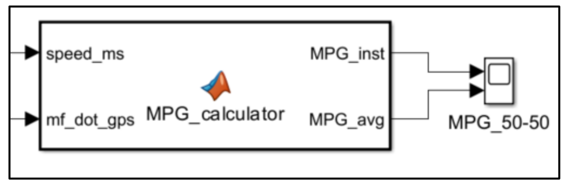
where 2819 g/gal corresponds to the assumed gasoline density.
The instantaneous fuel economy is computed as:

$$\text{MPG}_{\text{inst}} = \begin{cases} \frac{v_{\text{mph}}}{\dot{V}_{gal/h}}, & \dot{V}_{gal/h} > 0 \\ 0, & \dot{V}_{gal/h} = 0 \end{cases}$$

A running average MPG is computed to evaluate overall drive-cycle fuel economy:

$$\text{MPG}_{\text{avg}}(k) = \frac{1}{k}\sum_{i=1}^{k} \text{MPG}_{\text{inst}}\,(i)$$





3. Design of the Energy Management Strategy

3.1. 50–50 Rule-Based Controller

50–50 Rule-Based Controller

This controller is a baseline energy management strategy used for comparison against unconstrained LTI MPC with hard coded constraints. It allocates half of the required traction power between the internal combustion engine and electric motor using fixed heuristics, that is, without predictive optimization or future preview.

The controller operates on required shaft power demanded $P_{\text{sh, req}}$ computed by backward looking vehicle dynamics model. Here, only the traction demand is considered, whereas during braking and idling conditions, both the engine and motor are commanded to zero power as the power generated or more correctly the fuel consumption by the engine to overcome friction and keep the engine running is already inculcated in the engine model itself.

At each time step the traction power required at the shaft along with the current battery SOC and engine and motor power limits is provided as an input to the controller to determine the instantaneous power split between the engine and motor. This power split is 50-50 unless any one power source is maxed out for a specific shaft power demand at the operating shaft speed.

$P_{\text{req}} = \max\left(P_{\text{sh, req}}, 0\right)$.

Braking / idling condition: if $P_{\text{req}} = 0$. No propulsion is required, so:

$$P_e = 0, \quad P_m = 0$$

Low SOC protection (engine-only mode): if $\text{SOC} \le \text{SOC}_{\text{min}}$: When the battery SOC falls below a predefined minimum threshold SOC_{min}, electric propulsion is disabled.

$$P_e = \min\left(P_{\text{req}}, P_{e,max}\right), \quad P_m = 0$$

Power Shortfall: Any remaining unmet demand is recorded as power shortfall.

$$P_{\text{short}} = \max\left(P_{\text{req}} - P_{e,max}, 0\right)$$

Normal Operating Region (50/50 power split): If $\text{SOC} > \text{SOC}_{\text{min}}$, the traction demand is equally divided between the engine and the motor:

$$P_{e,\text{share}} = 0.5\, P_{\text{req}}, \quad P_{m,\text{share}} = 0.5\, P_{\text{req}}$$

Apply powerhouse component's limits to find actual input to the non-linear plant(that is, engine and motor map:

$$P_e = \min\left(P_{e,\text{share}}, P_{e,max}\right)$$
$$P_m = \min\left(P_{m,\text{share}}, P_{m,max}\right)$$

Component shortfalls:

$$P_{\text{short},e} = \max\left(P_{e,\text{share}} - P_{e,\text{max}} \ \ , 0\right)$$
$$P_{\text{short},m} = \max\left(P_{m,\text{share}} - P_{m,\text{max}} \ \ , 0\right)$$

Total shortfall (assuming no redistribution):

$$P_{\text{short}} = P_{\text{short},e} + P_{\text{short},m}$$

3.2. Model Predictive Control Formulation

After suggestion on the presentation. The unconstrained MPC-based EMS uses a linear time-invariant energy model with the state vector where SOC is the only state in the MPC internal model, is defined as:

$$x_k = \ [SOC_k]$$

and the control input vector:

$$u_k = \ \left[P_{e,k}\right]$$

Here, engine mechanical power is the only MPC control variable and the motor power is computed from power balance:

$$P_{m,k} = P_{sh,req,k} - P_{e,k}$$

The required shaft power is treated as a known disturbance, for which a preview vector for required shaft power is constructed offline for future shaft power prediction along the horizon(N=10).

Motor power is computed by power balance (not optimized directly):

$$P_{m,k} = d_k - u_k$$

Discrete-time SOC model:

From the power-based SOC relation, using constant equivalent efficiency $\eta_{\text{eq}} = \eta_{\text{m}}\eta_{\text{batt}}$ and nominal energy $E_{nom} = V_{nom} Q_{Ah} 3600$:

$$x_{k+1} = x_k - \gamma(d_k - u_k) = x_k + \gamma u_k - \gamma d_k$$

$$\gamma = \frac{T_s}{E_{nom}\eta_{eq}}, \quad \eta_{eq} = \eta_m \eta_{bat}$$

LTI state-space form is:

$$x_{k+1} = Ax_k + Bu_k + Ed_k$$

$$A{=}1,\ \ B{=}\ \gamma,\ \ E{=}{-}\gamma$$

The continuous-time dynamics are given by:

$$\dot{\mathrm{SOC}}(t) = -\frac{1}{\eta_m \eta_{batt} E_{\mathrm{nom}}} P_m(t) \dot{E}_{\mathrm{fuel}}(t) = -\frac{1}{\eta_e} P_e(t)$$

These equations are discretized using forward Euler integration. Over a finite prediction horizon, the MPC minimizes a quadratic cost function penalizing SOC deviation from a target value, fuel energy usage, and control effort.

Horizon stacking/prediction matrices:

Define horizon N and stacked vectors:

$$U = [u_k, u_{k+1}, \dots, u_{k+N-1}]^\top \in R^N,$$

$$D = [d_k, d_{k+1}, \dots, d_{k+N-1}]^\top \in R^N,$$

$$X = [x_{k+1}, x_{k+2}, \dots, x_{k+N}]^\top \in R^N$$

Using repeated substitution, the stacked prediction is:

$$X = A_H x_k + B_H U + E_H D$$

A_H:

Since $A = 1 \ \Rightarrow \ A^i = 1$

$$A_H = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{N\times 1}$$

B_H:

$$B_H = \gamma \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix} = \gamma\, \mathrm{tril}(\mathbf{1}_{N\times N})$$

E_H:

Same structure, with $E = -\gamma$:

$$E_H = -\gamma\, \mathrm{tril}(1_{N\times N})$$

These are exactly what your mpc_initialize() builds:

- A_H = ones(N,1)

- B_H = Bd * tril(ones(N,N)), with $B_d = \gamma$
- E_H = Ed * tril(ones(N,N)), with $E_d = -\gamma$

Cost function and QP matrices:

SOC reference along the horizon:

$$X_{ref} = \left[SOC_{ref} \ \vdots \ SOC_{ref}\right]^T$$

Stage cost (your report form):

$$J = \sum_{i=0}^{N-1} \left[q_{soc}\left(x_{k+i+1} - SOC_{ref}\right)^2 + r_e u_{k+i}^2\right]$$

Stacked quadratic form:

$$J = \left(X - X_{ref}\right)^\top Q_H \left(X - X_{ref}\right) + U^\top R_H U$$

with:

$$Q_H = q_{soc} I_N, \quad R_H = r_e I_N$$

Substitute $X = A_H x_k + B_H U + E_H D$. Define the prediction error term:

$$\Delta = A_H x_k + E_H D - X_{ref}$$

Then:

$$J = (B_H U + \Delta)^\top Q_H (B_H U + \Delta) + U^\top R_H U$$

Expand and collect terms in standard QP form:

$$J = \frac{1}{2} U^\top H U + f^\top U + \text{constant}$$

Hessian H:

$$H = 2(B_H^\top Q_H B_H + R_H)$$

(this matches your code: H = 2*(B_H' * Q_H * B_H + R_H))

Gradient vector f:

If you write the QP as $\frac{1}{2}U^{\top}HU + f^{\top}U$, then:

$$f = 2B_H^{\top}Q_H\Delta$$

How you compute the optimal input sequence U^*:

For the unconstrained QP, set gradient to zero:

$$\nabla J = HU + f = 0 \quad \Rightarrow \quad U^* = -H^{-1}f$$

In implementation you solve using a linear system (numerically better than explicit inverse):

$$U^* = -H^{-1}f$$

Then apply receding horizon:

$$u_k^* = U^*(1) \quad \Rightarrow \quad P_e(k) = u_k^*$$

Hard constraints (NOT in the QP; applied after solving):

After obtaining the commanded engine power $P_{e,\text{cmd}}$, power-split feasibility is enforced as follows:

$$P_{m,k} = P_{req,k} - P_{e,k}$$

with the constraints:

$$0 \leq P_{e,\text{k}} \leq P_{e,max,\text{k}}, \quad 0 \leq P_{m,\text{k}} \leq P_{m,max,\text{k}}$$

These constraints imply the equivalent bounds on the engine power command:

$$u_{min,\text{k}} = \max\left(0, P_{req,\text{k}} - P_{m,max,\text{k}}\right)$$

$$u_{max,\text{k}} = \min\left(P_{e,max,\text{k}}, P_{req,\text{k}}\right)$$

The final applied control is:

$$P_{e,\text{k}} = \text{clip}\left(P_{e,cmd}, u_{min,\text{k}}, u_{max,\text{k}}\right)$$

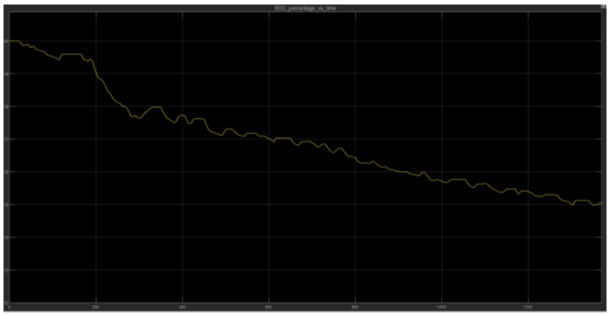
$$P_{m,\text{k}} = \max\left(P_{req,\text{k}} - P_{e,\text{k}}, 0\right)$$

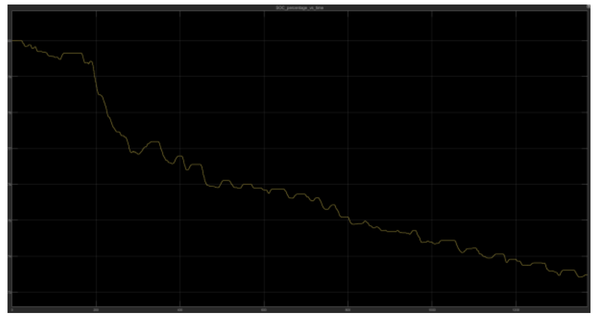
4. Simulation Results and Discussion

4.1. Drive Cycle

All simulations are performed using the UDDS drive cycle with a sampling time of 1 s. For this project, all the simulations were performed with time sample 1s. The cycle represents typical urban driving conditions with frequent acceleration and deceleration events. For this project, the cycle was run only once.

4.2. SOC Behavior



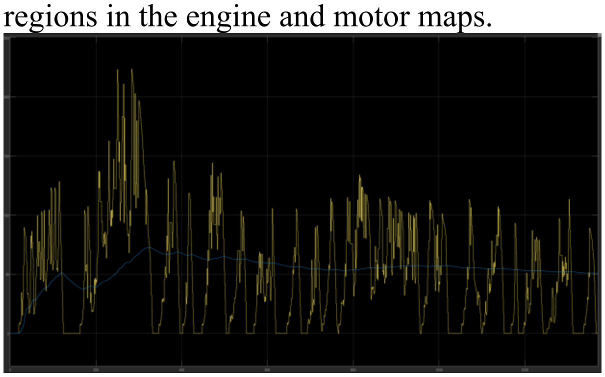


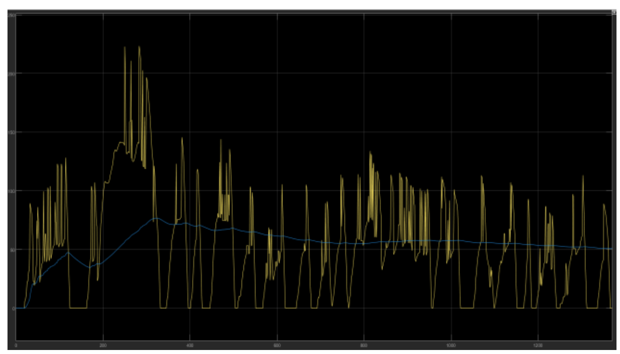
50-50 MPC

Towards the end of driving cycle, the SOC percentage for 50-50 and MPC is approximately 75% and 73.5% which shows that MPC is using motor more for improved power performance.

4.3. Fuel Economy Comparison

The MPC strategy consistently achieves higher instantaneous and average MPG compared to the 50–50 controller. The improvement is attributed to better utilization of high-efficiency operating regions in the engine and motor maps.

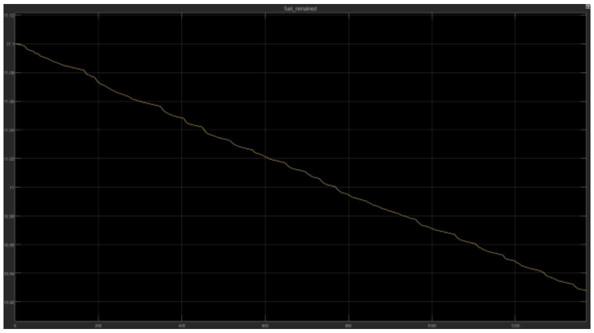


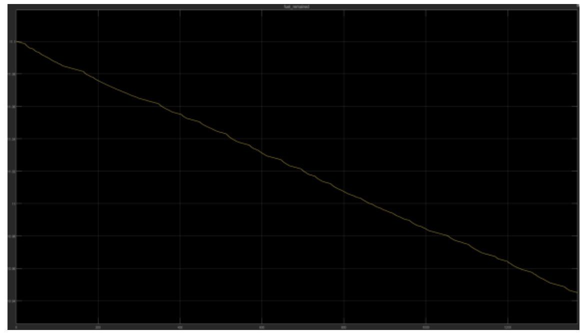


50-50 MPC

After tuning MPC parameters and making so that both engine and motor works, the MPG was similar but the SOC drop shows a significant reduction in energy consumption for the same driving cycle. This MPC is a newer version after the suggestion/review were received based on the presentation.

4.4. Remaining Fuel



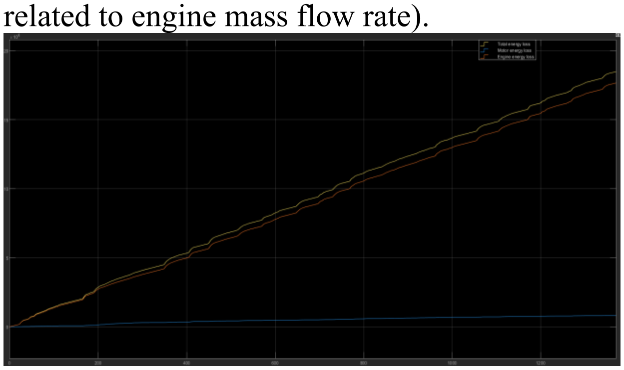


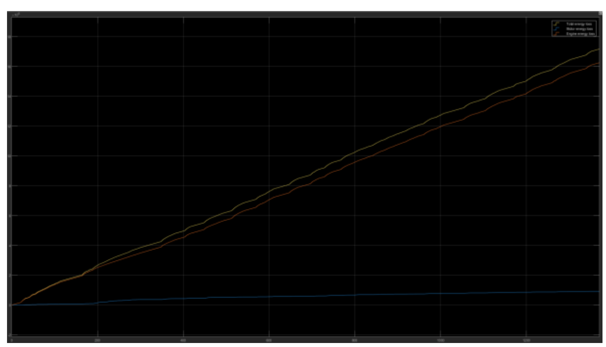
50-50 MPC

Toward the end of the cycle, the fuel tank capacity for 50-50 power split and MPC was approximately 10.927gal and 10.945gal

4.5. Energy Losses

Total energy losses in the powertrain are reduced under MPC operation, when zoomed into the graph for 50-50 power split and MPC, it was found that 50-50 controller caused an energy loss of 1.85MJ and MPC caused the energy loss of 1.715MJ. If the controller is tuned more, the energy loss can be more minimized. Due to the reformulation of the MPC with lesser states and addition of regeneration and the debugging of problems, less time was left for improved tuning of MPC weights especially q_SOC(weight related to battery SOC drift from target) and r_e(weight related to engine mass flow rate).





50-50 MPC

5. Conclusion

This project developed a backward-looking P2 parallel PHEV model with realistic engine, motor, and battery subsystems and compared two energy management strategies. The MPC-based EMS demonstrated superior performance over a simple rule-based controller in terms of SOC regulation, fuel economy, and overall energy efficiency. These results highlight the potential of predictive control methods for improving PHEV energy management.

When the engine cost r_e was set to a negligible value (0.00000001), the controller virtually ignored the fuel penalty and defaulted into a pure Electric Vehicle mode. In this scenario, the engine was only commanded for brief time which was seen as idling, forcing the motor to provide the vast majority of the required dynamic shaft power. This resulted in a controlled, gradual depletion of the battery.

The system exhibited extreme sensitivity to r_e. Increase in engine cost by 100 times (to 0.000001), when paired with a high q_{SOC} (approx. 10^5), caused the strategy to flip to an engine-

dominant or charge-sustaining mode. This high-cost ratio forced the MPC to prioritize using the engine to maintain the battery's state which had overridden the EV-preferred operation.
This result from parameter tuning sensitivity confirms that while the MPC functions correctly as a controller, the effective operating region for a hybrid is narrowly defined and needs extensive tuning.
Future work may includes:

- Improved tuning of parameters. For this now-a-days, neural networks based tuning method are used.
- Using of non-linear MPC formulation.
- Running simulation for multiple drive cycles and running one drive cycle for multiple number of times.

6. Contributions

- Yash Panthri: Controller design(both), controller mathematics, motor and engine selection and formulation of lookup table, Vehicle Dynamics Model formulation with selection of inertias and gear ratios, integration of all matlab functions as simulink block and their interactions, selecting KPI for testing performance and validation, selection of constants and assumption of efficiencies required in MPC by running on engine only mode and EV only mode, report.
- Hesham Aldahbali: Lead presentation, Vehicle dynamics model formulation, initial formulation of project problem, research about types of controllers.
- Tejas Deodhar: Integrated engine and motor models into the overall powertrain framework, led report writing, result interpretation, and presentation preparation, research about types of controllers.

7. Reference:

This report was written without any specific references. The reference papers used for learning more about MPC are in the project folder, but those papers were simplified as much as possible to make it easy to implement for this project. Open AI helped a little.