DELFT UNIVERSITY OF TECHNOLOGY

SC42075 MODELLING OF HYBRID SYSTEMS

Assignment 2021

Vivek Varma (5227828)
Justine Kroese (4547659)

June 15, 2021

Contents

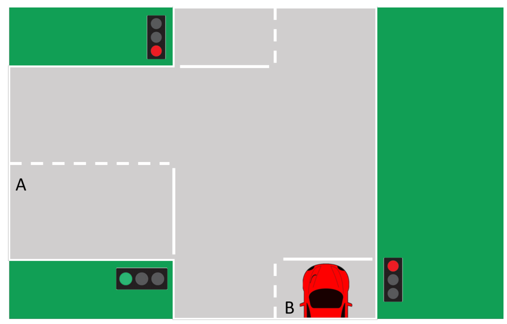
1 Part 1: Hybrid system example

In this chapter an intersection with traffic lights is represented as a hybrid automaton. First, the system itself is described and in after that this description is converted into a hybrid automaton.

1.1 Description of the system

This system that is going to be described is the intersection of the Hugo de Grootstraat and the Krakeelpolderweg in Delft, displayed in figure 1a. This is an intersection with traffic lights, where one light is used for all directions. So if the light is green the cars can choose whether to turn left or right or to drive straight ahead. A more schematic representation of the intersection can be found in figure 1b. The system will only take into account the car movements (pedestrians and cyclists are excluded).

(a) Photograph at intersection



(b) Schematic representation

Figure 1: System of an intersection with traffic lights

1.1.1 Variables and dynamics of the system

To model the intersection, it is split up into two roads. Road A is the road on the left and road B is the road straight ahead. The amount of cars on road A at a certain instance is modelled as y(t), the amount of road B as x(t). To make the system not too complicated, road A and road B will only have two states: closed and open. If the road is open, the light is green or orange, if the road is closed the light is red.

$y(t)$: the amount of cars waiting on road A
$x(t)$: the amount of cars waiting on the road B
$A(t)$: the state of road A $\in$ [Open,Closed]
$B(t)$: the state of road B $\in$ [Open,Closed]
$c(t)$: the time elapsed since both roads are closed

If the road is closed, the line of cars will grow. If the road is open the amount of cars in line will decrease. Only one road can be open at the same time. To switch which road is open, both roads will for safety reasons first be closed for 2 seconds.

This means there will be four states: The first state is where both roads are closed and $y(t) > 5$. The second state is where road A is open, road B is closed and $x \leq 5$. The third state is where both roads are closed and $x > 5$. The last state is where road A is closed, road B is open and $y \leq 5$.

1.2 The system as a hybrid automaton

Using the variables described is subsection 1.1.1, the hybrid automaton in figure 2 is constructed.

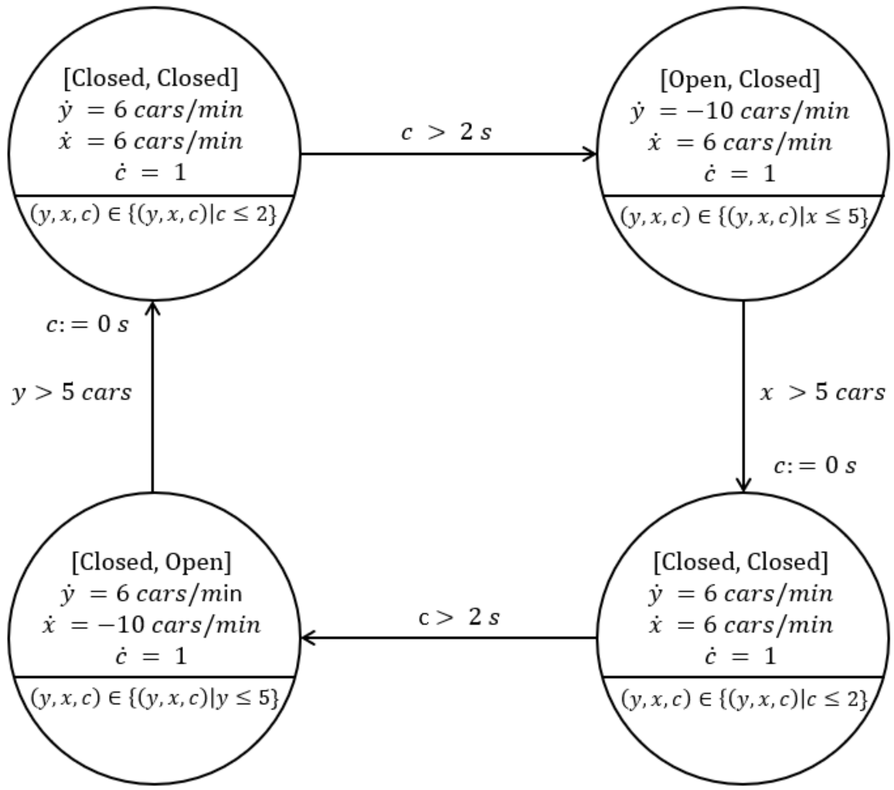


Figure 2: Hybrid automaton of intersection with traffic lights

The amount of cars in line and the amount of cars passing the open road are modelled as a continuous flow. This is of course a simplification of the reality, but it makes it the model easier to comprehend. When there are more than 5 cars waiting on either one of the roads, that road needs to switch to the Open state. Before this can happen, the automaton first has to go through the state where both roads are closed, to make sure cars from the two roads will not hinder each other. After two seconds, the system gets into the next mode where the new road will open.

2 Part 2: Energy management of microgrids

2.1 Discrete-time Piecewise Affine (PWA) Model of a Battery

To determine a piecewise affine (PWA) model of a battery, it is important to first specify the parameters of the battery. These are stated in table 1.

Variables related to a battery	Name	Range	Unit
Stored energy	x_b	$[0,\bar{x_b}]$	kWh
Exchanged power	u_b	$[\underline{u_b},\bar{u_b}]$	kW
Operational mode (charge/discharge)	s_b	$\{0,1\}$	-
Charging efficiency	η_c	CONSTANT	-
Discharging efficiency	η_d	CONSTANT	-

Table 1: Parameters related to a battery

The state of the PWA model is the energy stored in the battery. The input is the exchanged power. The discharging of the battery is defined to have a positive sign and is connected to operational mode 0. Charging has a negative sign and is connected to operational mode 1. In one sampling instant T_s, the exchanged energy is is $T_s \cdot u_b$ kJ. This is than multiplied by the charging or discharging efficiency. This results in the PWA model:

$$x_b(k+1) = \begin{cases} x_b(k) - \eta_d T_s u_b(k) & u_b \geq 0 \quad (s_b(k) = 0) \\ x(b) - \eta_c T_s u_b(k) & u_b \leq 0 \quad (s_b(k) = 1) \end{cases}$$

2.2 Mixed Logical Dynamical (MLD) Model of a Battery

To determine the mixed logical dynamical (MLD) model of the battery, the following constraints are used:

$$0 \leq x_b \leq \bar{x_b}$$

$$\underline{u_b} \leq u_b \leq \bar{u_b}$$

$$if\ s_b(k) = 1\ u_b \leq 0$$

$$if\ s_b(k) = 0\ u_b > 0$$

To obtain the MLD model, we first take a look at the general model:

$$x(k+1) = Ax(k) + B_1 u(k) + B_2 d(k) + B_3 z(k) + B_4$$

$$E_1 x(k) + E_2 u(k) + E_3 \delta(k) + E_4 z(k) \leq g_5$$

Looking at the model from subsection 2.1, it is clear that $x(k) = x_b(k)$ and $u(k) = u_b(k)$. $s_b(k)$ can be interpreted as the binary variable $\delta(k)$.

By using this $\delta(k)$, the PWA model can be rewritten as:

$$x(k+1) = x(k) - \eta_d T_s u(k) + (\eta_d - \eta_c) T_s \delta(k) u(k)$$

s.t.

$$u(k) \leq \bar{u_b}(1 - \delta(k))\ u(k) + \bar{u_b}\delta(k) \leq \bar{u_b}$$

$$u(k) \geq +(\underline{u_b} - \epsilon)\delta(k)\ \ -u(k) + (\underline{u_b} - \epsilon)\delta(k) \leq -\epsilon$$

The constraints can also be written in matrix form:

$$\begin{bmatrix} \bar{u}_b \\ \underline{u}_b - \epsilon \end{bmatrix} \delta(k) + \begin{bmatrix} 1 \\ -1 \end{bmatrix} u(k) \leq \begin{bmatrix} \bar{u}_b \\ -\epsilon \end{bmatrix}$$

In the constraints the ϵ is used to turn the strict inequality $u_b > 0$ into a non-strict equality.

In the equation above there is a product of two variables present: $\delta(k)u(k)$. This can be replaced by the new variable $z(k) = \delta(k)u(k)$. Then the formula for the state evolution is equal to:

$$x(k+1) = x(k) - \eta_d T_s u(k) + (\eta_d - \eta_c) T_s z(k)$$

The constraints are now equal to:

$$z(k) \leq \bar{u}_b \delta(k)$$
$$z(k) \geq \underline{u}_b \delta(k)$$
$$z(k) \leq u(k) - \underline{u}_b (1 - \delta(k))$$
$$z(k) \geq u(k) - \bar{u}_b (1 - \delta(k))$$

or in matrix form:

$$\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} z(k) + \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} u(k) + \begin{bmatrix} -\bar{u}_b \\ \underline{u}_b \\ -\underline{u}_b \\ \bar{u}_b \end{bmatrix} \delta(k) \leq \begin{bmatrix} 0 \\ 0 \\ -\underline{u}_b \\ \bar{u}_b \end{bmatrix}$$

The last constraint that has to written in vector form is $0 \leq x(k) \leq \bar{x}_b$:

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} x(k) \leq \begin{bmatrix} \bar{x}_b \\ 0 \end{bmatrix}$$

By combining all these constraints the form of the general model can be obtained:

$$x(k+1) = \underbrace{1}_{A} \cdot x(k) + \underbrace{(-\eta_d T_s)}_{B_1} u(k) + \underbrace{0}_{B_2} \delta(k) + \underbrace{(\eta_d - \eta_c) T_s}_{B_3} z(k)$$

s.t.

$$\underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}}_{E_1} x(k) + \underbrace{\begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{E_2} u(k) + \underbrace{\begin{bmatrix} \underline{u}_b - \epsilon \\ \bar{u}_b \\ -\bar{u}_b \\ \underline{u}_b \\ -\underline{u}_b \\ \bar{u}_b \\ 0 \\ 0 \end{bmatrix}}_{E_3} \delta(k) + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}}_{E_4} z(k) \leq \underbrace{\begin{bmatrix} -\epsilon \\ \bar{u}_b \\ 0 \\ 0 \\ -\underline{u}_b \\ \bar{u}_b \\ \bar{x}_b \\ 0 \end{bmatrix}}_{g_5}$$

2.3 Modelling a Diesel Generator

The fuel consumption of the diesel generator is given by Equation 1.

$$f\left(u_{\mathrm{d}}(k)\right)=\begin{cases} u_{\mathrm{d}}^2(k)+4 & \text{if } 0 \leq u_{\mathrm{d}}(k)<2 \\ 4u_{\mathrm{d}}(k) & \text{if } 2 \leq u_{\mathrm{d}}(k)<5 \\ -9.44u_{\mathrm{d}}^3(k)+166.06u_{\mathrm{d}}^2(k)-948.22u_{\mathrm{d}}(k)+1790.28 & \text{if } 5 \leq u_{\mathrm{d}}(k)<7 \\ -11.78u_{\mathrm{d}}(k)+132.44 & \text{if } 7 \leq u_{\mathrm{d}}(k)<9 \\ 4.01\left(u_{\mathrm{d}}(k)-10.47\right)^2+17.79 & \text{if } 9 \leq u_{\mathrm{d}}(k) \leq 15, \end{cases} \tag{1}$$

where $f(u_d(k))$ is the consumed fuel of the diesel generator at time step k in [kg/h] and the value of $u_d(k)$ represents the output power of the diesel generator at time step k.
Now the task it to construct a PWA model of the non-linear model in Equation 1. This PWA approximation($\hat{f}\,[0, \bar{u}_d] \rightarrow \mathbb{R}$) consists of 4 regions of approximation of the fuel consumption curve of Equation 1. This is defined as in Equation 2.

$$\hat{f}\left(u_{\mathrm{d}}(k)\right)=\begin{cases} a_1+b_1u_{\mathrm{d}}(k) & \text{if } 0 \leq u_{\mathrm{d}}(k)<u_1 \\ a_2+b_2u_{\mathrm{d}}(k) & \text{if } u_1 \leq u_{\mathrm{d}}(k)<u_2 \\ a_3+b_3u_{\mathrm{d}}(k) & \text{if } u_2 \leq u_{\mathrm{d}}(k)<u_3 \\ a_4+b_4u_{\mathrm{d}}(k) & \text{if } u_3 \leq u_{\mathrm{d}}(k) \leq 15, \end{cases} \tag{2}$$

where $u_1 = 5, u_2 = 6.5, u_3 = 11$. To construct the best possible approximation, we need to ensure that the squared error through the intervals is minimized. Mathematically, this means that the cost function is:

$$c=\min \int_0^{\bar{u}_{\mathrm{d}}}\left(f\left(u_{\mathrm{d}}\right)-\hat{f}\left(u_{\mathrm{d}}\right)\right)^2 du_{\mathrm{d}} \tag{3}$$

We use a Genetic Algorithm(GA) to find the minimum of Equation 3. We define Equation 2 and Equation 3 using the **piecewise**() function in MATLAB. We then use the integration function to complete the cost function definition. The GA parameters defined explicitly are as follows: Population Size-700, Maximum Generations-1600, Elite Count- 25.
The results we obtained(rounded to 4 decimal places) are:

$$\{a_1, b_1\} = \{1.6318, 3.5392\},\ \{a_2, b_2\} = \{-85.6286, 20.8671\}$$

$$\{a_3, b_3\} = \{111.8701, -9.1653\},\ \{a_4, b_4\} = \{-207.2789, 19.6968\}$$

$$c = 128.0206$$

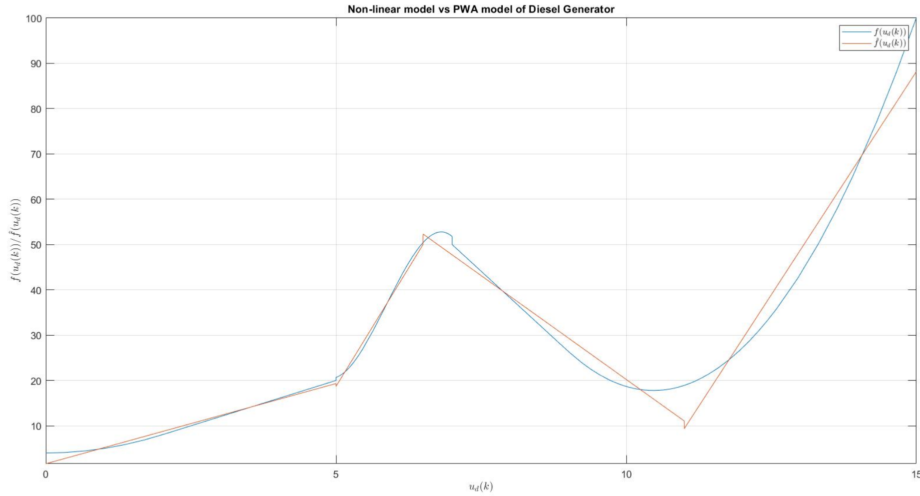


Figure 3: PWA Approximation of the non-linear Diesel Generator

We can visualize our PWA approximation(Figure 3). Considering that we have only 4 PWA models as an approximation, this approximation seems to be a good fit for the non-linear model.

2.4 Variable Limits of PWA Approximation

Now we have the additional freedom that u_1, u_2, u_3 need not be fixed as they were previously. We can now re-model our cost function to have u_1, u_2, u_3 as variables to be optimized in addition to a_i, b_i. This can be done in the piecewise function definition. Now that we have an initial guess(the final result from Question 2.3), we can use another optimization algorithm to find a better cost function value, and thus better values for a_i, b_i, u_j.
The algorithm we use here is Simulated Annealing. The initial guess values were use here are $a_i, b_i \ and \ u_j$ from Question 2.3. Additionally, we define an upper and lower bound, since u_j is bounded. Its lower bound is 0 and upper bound is 15.
Indeed with more freedom of optimization, we obtain better results. The results we obtained(rounded to 4 decimal places) are:

$$\{a_1, b_1\} = \{1.7111, 3.5224\},\ \{a_2, b_2\} = \{-85.6007, 20.7958\}$$

$$\{a_3, b_3\} = \{111.0235, -9.0577\},\ \{a_4, b_4\} = \{-207.4428, 19.705\}$$

$$u_1 = \{5.1091\},\ u_2 = \{6.8142\},\ u_3 = \{11.0738\},\ c = 126.5814$$

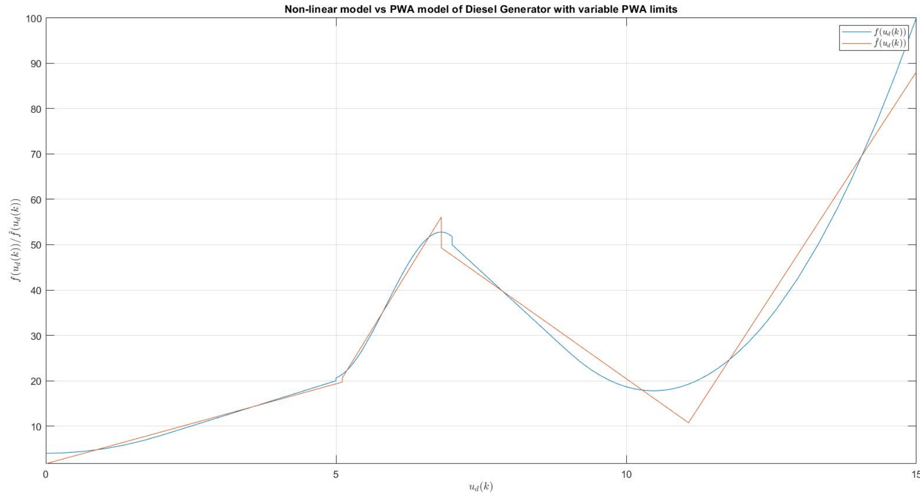


Figure 4: PWA Approximation of the Non-Linear Diesel Generator with Variable Limits

We can visualize our PWA approximation(Figure 4). This does seem to be an even better fit. Minor differences can be noted in the PWA linear intercepts and slopes, but the major difference is the shifting of u_2 from 6.5 to 6.8. Thus with a greater degree of freedom, we have obtained a lower cost function value, thus lesser errors and a better approximation of the non-linear diesel generator fuel consumption model.

2.5 Discrete-Time PWA Model of Diesel Generator

To determine a piecewise affine (PWA) model of the diesel generator, it is important to first specify the parameters of the generator. These are stated in table 2.

Variables related to a diesel generator	Name	Range	Unit
Remaining Fuel	x_d	$[\underline{x}_d, \bar{x}_d]$	kg
Generated power	u_d	$[0,\bar{u}_d]$	kW
Operational mode (on/off)	s_d	$\{0,1\}$	-
Filling rate of fuel tank	R_f	CONSTANT	-

Table 2: Parameters related to a diesel generator

The state of the PWA model is the remaining fuel in the diesel generator. The input is the generated power. The use of the generator is defined consumes fuel and is connected to operational mode 0. Refilling is constantly adding fuel to the system, and the absence of consumed fuel to generate power is connected to operational mode 1. In one step(one hour), the fuel tank is constantly refilled by R_f irrespective of the switch state. This results in the PWA model:

$$x_d(k+1) = \begin{cases} x_d(k) + R_f T_s & \text{if } s_d(k) = 0 \quad (u_d(k) = 0) \\ x_d(k) + R_f T_s - (a_1 + b_1 \left(u_\mathrm{d}(k)\right) T_s & \text{if } s_d(k) = 1 \quad 0 \leq u_\mathrm{d}(k) < u_1 \\ x_d(k) + R_f T_s - (a_2 + b_2 \left(u_\mathrm{d}(k)\right) T_s & \text{if } s_d(k) = 1 \quad u_1 \leq u_\mathrm{d}(k) < u_2 \\ x_d(k) + R_f T_s - (a_3 + b_3 \left(u_\mathrm{d}(k)\right) T_s & \text{if } s_d(k) = 1 \quad u_2 \leq u_\mathrm{d}(k) < u_3 \\ x_d(k) + R_f T_s - (a_4 + b_4 \left(u_\mathrm{d}(k)\right) T_s & \text{if } s_d(k) = 1 \quad u_3 \leq u_\mathrm{d}(k) < 15 \end{cases} \tag{4}$$

2.6 Mixed Logical Dynamical (MLD) Model of a Diesel Generator

To determine the mixed logical dynamical (MLD) model of the battery, the following constraints are used:

$$\underline{x_d} \leq x_d \leq \bar{x_d}$$

$$0 \leq u_b \leq \bar{u_b}$$
$$if\ s_d(k) = 1\ u_d > 0$$
$$if\ s_d(k) = 0\ u_d = 0$$

To obtain the MLD model, we first take a look at the general model:

$$x(k+1) = Ax(k) + B_1u(k) + B_2d(k) + B_3z(k) + B_4$$

$$E_1x(k) + E_2u(k) + E_3\delta(k) + E_4z(k) \leq g_5$$

Looking at the model from subsection 2.5, it is clear that $x(k) = x_d(k)$ and $u(k) = u_d(k)$. $s_d(k)$ can be interpreted as the binary variable $\delta(k)$. To account for the four different operational modes in the PWA model, four δ's are introduced. Only only of these can be nonzero at the same time. If all are zero, the generator is not generating any power.

$$\begin{array}{ll} \delta_1(k) = 1 & 0 \leq u(k) < u_1 \\ \delta_2(k) = 1 & u_1 \leq u(k) < u_2 \\ \delta_3(k) = 1 & u_2 \leq u(k) < u_3 \\ \delta_4(k) = 1 & u_3 \leq u(k) < 15 \end{array} \tag{5}$$

$$\text{s.t. } \delta_1(k) + \delta_2(k) + \delta_3(k) + \delta_4(k) \leq 1$$

By using these $\delta(k)$, the PWA model can be rewritten as:

$$x(k+1) = x(k) + R_fT_s - (a_1 + b_1u(k))T_s\delta_1(k) - (a_2 + b_2u(k))T_s\delta_2(k) - (a_3 + b_3u(k))T_s\delta_3(k) - (a_4 + b_4u(k))T_s\delta_4(k)$$

s.t.

$$0 \leq u(k) \leq \bar{u}_d + \delta_1(k)(u_1 - \bar{u}_d - \epsilon)$$

$$\delta_2(k)u_1 \leq u(k) \leq \bar{u}_d + \delta_2(k)(u_2 - \bar{u}_d - \epsilon)$$

$$\delta_3(k)u_2 \leq u(k) \leq \bar{u}_d + \delta_3(k)(u_3 - \bar{u}_d - \epsilon)$$

$$\delta_4(k)u_3 \leq u(k) \leq 15$$

$$\delta_1(k) + \delta_2(k) + \delta_3(k) + \delta_4(k) \leq 1$$

$$\underline{x_d} \leq x_d \leq \bar{x_d}$$

The equation with its constraints can also be written in matrix-vector form. Then the model is equal to:

$$x(k+1) = x(k) + R_fT_s - T_s\mathbf{affi_d}(k) - T_s\mathbf{bffi_d}(k)u(k)$$

s.t

$$\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\0\\0\\1\\-1\end{bmatrix} x(k) + \begin{bmatrix}-1\\1\\-1\\1\\-1\\1\\-1\\1\\1\\0\\0\end{bmatrix} u(k) + \begin{bmatrix}0 & 0 & 0 & 0\\ -u_1+\bar{u}_d+\epsilon & 0 & 0 & 0\\ 0 & u_1 & 0 & 0\\ 0 & -u_2+\bar{u}_d+\epsilon & 0 & 0\\ 0 & 0 & u_2 & 0\\ 0 & 0 & -u_3+\bar{u}_d+\epsilon & 0\\ 0 & 0 & 0 & u_3\\ 0 & 0 & 0 & 0\\ 1 & 1 & 1 & 1\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix} \delta_d(k) \leq \begin{bmatrix}0\\ \bar{u}_d\\ 0\\ \bar{u}_d\\ 0\\ \bar{u}_d\\ 0\\ 15\\ 1\\ \bar{x}_d\\ -\underline{x}_d\end{bmatrix}$$

where $\mathbf{a} = \begin{bmatrix}a_1 & a_2 & a_3 & a_4\end{bmatrix}$, $\mathbf{b} = \begin{bmatrix}b_1 & b_2 & b_3 & b_4\end{bmatrix}$, $\mathbf{ffi_d} = \begin{bmatrix}\delta_1 & \delta_2 & \delta_3 & \delta_4\end{bmatrix}$

Because the left sides of the constraints and the right side of the last u(k) constraint are already non-strict equalities, there is no ϵ needed there.

In the equation above there is a product of two variables present: $\delta_d(k)u(k)$. This can be replaced by the new variable

$$\mathbf{z_d}(k) = \begin{bmatrix}z_1(k)\\z_2(k)\\z_3(k)\\z_4(k)\end{bmatrix} = \mathbf{ffi_d}(k)u(k) = \begin{bmatrix}\delta_1(k)u(k)\\ \delta_2(k)u(k)\\ \delta_3(k)u(k)\\ \delta_4(k)u(k)\end{bmatrix}$$

.
Then the formula for the state evolution is equal to:

$$x(k+1) = x(k) + R_f T_s - T_s \mathbf{affi_d}(k) - T_s \mathbf{b}\mathbf{z_d}(k)$$

The constraints are now equal to:

$$z_i(k) \leq u_i\delta_i(k)$$
$$z_i(k) \geq u_{i-1}\delta_i(k)$$
$$z_i(k) \leq u(k) - \underline{u}_d(1-\delta_i(k))$$
$$z_i(k) \geq u(k) - \bar{u}_d(1-\delta_i(k))$$

where $i \in [1,4]$, $u_0 = 0$ and $u_4 = 15$.
Here u_i and u_{i-1} are the limits of $u(k)$ in the different operational modes. Because $\underline{u}_d = 0$, the constraints can be simplified to:

$$z_i(k) \leq u_i\delta_i(k)$$
$$z_i(k) \geq u_{i-1}\delta_i(k)$$
$$z_i(k) \leq u(k)$$
$$z_i(k) \geq u(k) - \bar{u}_d(1-\delta_i(k))$$

or in matrix form:

$$\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix} z_i(k) + \begin{bmatrix}0\\0\\-1\\1\end{bmatrix} u(k) + \begin{bmatrix}-u_i\\u_{i-1}\\0\\ \bar{u}_d\end{bmatrix} \delta_i(k) \leq \begin{bmatrix}0\\0\\0\\ \bar{u}_d\end{bmatrix}$$

The last constraint that has to be written in vector form is $\underline{x_d} \leq x(k) \leq \bar{x_d}$:

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} x(k) \leq \begin{bmatrix} \bar{x_d} \\ \underline{x_d} \end{bmatrix}$$

By combining all these constraints the form of the general model can be obtained:

$$x(k+1) = \underbrace{1}_{A} \cdot x(k) + \underbrace{0}_{B_1} u(k) + \underbrace{(-T_s \mathbf{a})}_{B_2} \delta_d(k) + \underbrace{(-T_s \mathbf{b})}_{B_3} z_d(k) + \underbrace{R_f T_s}_{B_4}$$

s.t.

$$\underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}}_{E_1} x(k) + \underbrace{\begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \\ -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{E_2} u(k) + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ -u_1 + \bar{u}_d + \epsilon & 0 & 0 & 0 \\ 0 & u_1 & 0 & 0 \\ 0 & -u_2 + \bar{u}_d + \epsilon & 0 & 0 \\ 0 & 0 & u_2 & 0 \\ 0 & 0 & -u_3 + \bar{u}_d + \epsilon & 0 \\ 0 & 0 & 0 & u_3 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ -u_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \bar{u_d} & 0 & 0 & 0 \\ 0 & -u_2 & 0 & 0 \\ 0 & u_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \bar{u_d} & 0 & 0 \\ 0 & 0 & -u_3 & 0 \\ 0 & 0 & u_2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \bar{u_d} & 0 \\ 0 & 0 & 0 & -15 \\ 0 & 0 & 0 & u_3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \bar{u_d} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{E_3} \delta(k)$$

$$
+\underbrace{\begin{bmatrix}
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0\\
1 & 0 & 0 & 0\\
-1 & 0 & 0 & 0\\
1 & 0 & 0 & 0\\
-1 & 0 & 0 & 0\\
0 & 1 & 0 & 0\\
0 & -1 & 0 & 0\\
0 & 1 & 0 & 0\\
0 & -1 & 0 & 0\\
0 & 0 & 1 & 0\\
0 & 0 & -1 & 0\\
0 & 0 & 1 & 0\\
0 & 0 & -1 & 0\\
0 & 0 & 0 & 1\\
0 & 0 & 0 & -1\\
0 & 0 & 0 & 1\\
0 & 0 & 0 & -1\\
0 & 0 & 0 & 0\\
0 & 0 & 0 & 0
\end{bmatrix}}_{E_4} z(k) \leq \underbrace{\begin{bmatrix}
0\\
\bar{u}_d\\
0\\
\bar{u}_d\\
0\\
\bar{u}_d\\
0\\
15\\
1\\
0\\
0\\
0\\
\bar{u_d}\\
0\\
0\\
0\\
\bar{u_d}\\
0\\
0\\
0\\
\bar{u_d}\\
0\\
0\\
0\\
\bar{u_d}\\
\bar{x_d}\\
\underline{x_d}
\end{bmatrix}}_{g_5}
$$

2.7 System model

For the system as a whole, two batteries and one diesel generator are considered. Their MLD models are repeated below (including corresponding subscripts).

$$
\begin{aligned}
x_{b_1}(k+1) &= A_{b_1} x_{b_1}(k) + B_{b_1,1} u_{b_1}(k) + B_{b_1,2} \delta_{b_1}(k) + B_{b_1,3} z_{b_1}(k) + B_{b_1,4}\\
x_{b_2}(k+1) &= A_{b_2} x_{b_2}(k) + B_{b_2,1} u_{b_2}(k) + B_{b_2,2} \delta_{b_2}(k) + B_{b_2,3} z_{b_2}(k) + B_{b_2,4}\\
x_d(k+1) &= A_d x_d(k) + B_{d,1} u_d(k) + B_{d,2} \delta_d(k) + B_{d,3} z_d(k) + B_{d,4}
\end{aligned}
$$

where $B_{b_1,4}$ and $B_{b_2,4}$ are zero subject to the constraints:

$$
\begin{aligned}
E_{b_1,1} x_{b_1}(k) + E_{b_1,2} u_{b_1}(k) + E_{b_1,3} \delta_{b_1}(k) + E_{b_1,4} z_{b_1}(k) &\leq g_{b_1,5}\\
E_{b_2,1} x_{b_2}(k) + E_{b_2,2} u_{b_2}(k) + E_{b_2,3} \delta_{b_2}(k) + E_{b_2,4} z_{b_2}(k) &\leq g_{b_2,5}\\
E_{d,1} x_d(k) + E_{d,2} u_d(k) + E_{d,3} \delta_d(k) + E_{d,4} z_d(k) &\leq g_{d,5}
\end{aligned}
$$

To make a MLD model of the whole system, the matrices can be stacked together in (block diagonal) matrices. The model we get then is as follows:

$$x(k+1) = \underbrace{\begin{bmatrix} A_{b_1} & 0 & 0 \\ 0 & A_{b_2} & 0 \\ 0 & 0 & A_d \end{bmatrix}}_{A} \begin{bmatrix} x_{b_1}(k) \\ x_{b_2}(k) \\ x_d(k) \end{bmatrix} + \underbrace{\begin{bmatrix} B_{b_1,1} & 0 & 0 \\ 0 & B_{b_2,1} & 0 \\ 0 & 0 & B_{d,1} \end{bmatrix}}_{B_1} \begin{bmatrix} u_{b_1}(k) \\ u_{b_2}(k) \\ u_d(k) \end{bmatrix} + \underbrace{\begin{bmatrix} B_{b_1,2} & 0 & 0 \\ 0 & B_{b_2,2} & 0 \\ 0 & 0 & B_{d,2} \end{bmatrix}}_{B_2} \begin{bmatrix} \delta_{b_1}(k) \\ \delta_{b_2}(k) \\ \delta_d(k) \end{bmatrix}$$

$$+ \underbrace{\begin{bmatrix} B_{b_1,3} & 0 & 0 \\ 0 & B_{b_2,3} & 0 \\ 0 & 0 & B_{d,3} \end{bmatrix}}_{B_3} \begin{bmatrix} z_{b_1}(k) \\ z_{b_2}(k) \\ z_d(k) \end{bmatrix} + \underbrace{\begin{bmatrix} B_{b_1,4}(k) \\ B_{b_2,4}(k) \\ B_{d,4}(k) \end{bmatrix}}_{B_4}$$

s.t.

$$\underbrace{\begin{bmatrix} E_{b_1,1} & 0 & 0 \\ 0 & E_{b_2,1} & 0 \\ 0 & 0 & E_{d,1} \end{bmatrix}}_{E_1} \begin{bmatrix} x_{b_1}(k) \\ x_{b_2}(k) \\ x_d(k) \end{bmatrix} + \underbrace{\begin{bmatrix} E_{b_1,2} & 0 & 0 \\ 0 & E_{b_2,2} & 0 \\ 0 & 0 & E_{d,2} \end{bmatrix}}_{E_2} \begin{bmatrix} u_{b_1}(k) \\ u_{b_2}(k) \\ u_d(k) \end{bmatrix} + \underbrace{\begin{bmatrix} E_{b_1,3} & 0 & 0 \\ 0 & E_{b_2,3} & 0 \\ 0 & 0 & E_{d,3} \end{bmatrix}}_{E_3} \begin{bmatrix} \delta_{b_1}(k) \\ \delta_{b_2}(k) \\ \delta_d(k) \end{bmatrix}$$

$$+ \underbrace{\begin{bmatrix} E_{b_1,4} & 0 & 0 \\ 0 & E_{b_2,4} & 0 \\ 0 & 0 & E_{d,4} \end{bmatrix}}_{E_4} \begin{bmatrix} z_{b_1}(k) \\ z_{b_2}(k) \\ z_d(k) \end{bmatrix} \leq \underbrace{\begin{bmatrix} g_{b_1,5} \\ g_{b_2,5} \\ g_{d,5} \end{bmatrix}}_{g_5}$$

This MLD model has also been implemented in Matlab. This implementation can be found in the Appendix.

2.8 Defining the cost function

The cost function is described as given below:

$$J(k) = \sum_{j=0}^{N_p-1} (\sum_{i=1}^{N_b} W_{b,i}|\Delta s_{b,i}(k+j)| + W_d|\Delta s_d(k+j)|) - W_{fuel}(x_d(k+N_p) - x_d(k))$$

$$-W_e \sum_{i=1}^{N_b}(x_{b,i}(k+N_p) - x_{b,i}(k)) + \sum_{j=0}^{N_p-1} P_{imp}(k+j)C_e(k+j)$$

where $P_{imp}(k)$ is the imported power to the microgrid at time step k and $C_e(k)$ is the price (benefit) of importing electricity to (exporting electricity from) the microgrid at time step k.

To determine the optimal Model predictive control (MPC) input sequence for a given sample step k, the MLD needs to be transformed into a mixed-integer linear programming problem (MILP). For this we need the values of $u(k)$, $\delta(k)$ and $z(k)$ up until $k+N_p-1$, where N_p is the prediction horizon. However, we only have the values up until $k+N_c-1$, where N_c is the control horizon. That is why the values from

$k+N_p$ until $k+N_c-1$ are kept equal to the value at $k+N_p-1$. In matrix form it looks like this:

$$\hat{u}(k) = \begin{bmatrix} u(k) \\ \vdots \\ u(k+N_c-1) \\ u(k+N_c) \\ \vdots \\ u(k+N_p-1) \end{bmatrix} = \begin{bmatrix} I_{N_u} & 0 & \cdots & 0 \\ 0 & I_{N_u} & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & I_{N_u} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & I_{N_u} \end{bmatrix} \begin{bmatrix} u(k) \\ \vdots \\ u(k+N_c-1) \end{bmatrix} = K_u \tilde{u}(k)$$

where N_u is the length of $u(k)$. The same can be done for $\hat{\delta}(k) = K_\delta \tilde{\delta}(k)$ and $\hat{z}(k) = K_z \tilde{z}(k)$

The three vectors $\tilde{u}(k)$, $\tilde{\delta}(k)$ and $\tilde{z}(k)$ can also be combined in one matrix as:

$$\tilde{V}(k) = \begin{bmatrix} \tilde{u}(k) & \tilde{\delta}(k) & \tilde{z}(k) \end{bmatrix}$$

To be able to solve the problem, we have to rewrite both the MLD model and the cost function.

2.8.1 Rewriting the MLD model

The MLD model can be rewritten by using successive substitution:

$$x(k+1) = Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) + B_4$$
$$x(k+2) = A(Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) + B_4) + B_1u(k+1) + B_2\delta(k+1) + B_3z(k+1) + B_4$$
$$\vdots$$
$$x(k+j) = A^jx(k) + \sum_{i=0}^{j-1} A^{j-i-1}(B_1u(k+i) + B_2\delta(k+i) + B_3z(k+i) + B_4)$$

This can also be written in matrix format:

$$\begin{bmatrix} x(k+1) \\ x(k+2) \\ \vdots \\ x(k+N_p) \end{bmatrix} = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N_p} \end{bmatrix}}_{M_2} x(k) + \underbrace{\begin{bmatrix} B_1 & 0 & \cdots & 0 \\ AB_1 & B_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N_p-1}B_1 & A^{N_p-2}B_1 & \cdots & B_1 \end{bmatrix}}_{T_1} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p-1) \end{bmatrix} +$$

$$\underbrace{\begin{bmatrix} B_2 & 0 & \cdots & 0 \\ AB_2 & B_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N_p-1}B_2 & A^{N_p-2}B_2 & \cdots & B_2 \end{bmatrix}}_{T_2} \begin{bmatrix} \delta(k) \\ \delta(k+1) \\ \vdots \\ \delta(k+N_p-1) \end{bmatrix} + \underbrace{\begin{bmatrix} B_3 & 0 & \cdots & 0 \\ AB_3 & B_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N_p-1}B_3 & A^{N_p-2}B_3 & \cdots & B_3 \end{bmatrix}}_{T_3} \begin{bmatrix} z(k) \\ z(k+1) \\ \vdots \\ z(k+N_p-1) \end{bmatrix}$$

$$+ \underbrace{\begin{bmatrix} B_4 \\ AB_4 + B_4 \\ \vdots \\ A^{N_p-1}B_4 + A^{N_p-2}B_4 + \cdots + B_4 \end{bmatrix}}_{M_3}$$

So more compactly written this is equal to:

$$\hat{x}(k) = M_2x(k) + T_1\hat{u}(k) + T_2\hat{\delta}(k) + T_3\hat{z}(k) + M_3 = M_2x(k) + T_1K_u\tilde{u}(k) + T_2K_\delta\tilde{\delta}(k) + T_3K_z\tilde{z}(k) + M_3$$

Now by using the $\tilde{V}(k)$ that was previously described:

$$\hat{x}(k) = M_2x(k) + \begin{bmatrix} T_1K_u & T_2K_\delta & T_3K_z \end{bmatrix} \tilde{V}(k) + M_3 = M_2x(k) + M_1\tilde{V}(k) + M_3$$

We need to follow a similar procedure as shown above to generalize the constraint equations and group together the variables in terms of $\tilde{V}(k)$. The constraints are vital and simultaenous to the above derivation since in each constraint step we get the values of $\hat{\delta}$ and $\hat{z}$. The constraint equations follow the pattern:

$$E_1x(k) + E_2u(k) + E_3\hat{\delta}(k) + E_4\hat{z}(k) \leq g_5$$

$$E_1A\hat{x}(k) + E_1B_1u(k) + E_1B_2\hat{\delta}(k) + E_1B_3\hat{z}(k) + E_2u(k+1) + E_3\hat{\delta}(k+1) + E_4\hat{z}(k+1) \leq g_5 - E_1B_4$$

$$\vdots$$

$$\begin{gathered} E_1A^{j-1}x(k) + E_1A^{j-2}B_1u(k) + E_1A^{j-3}B_1u(k+1) + \cdots + E_1B_1u(k+j-2) + E_2u(k+j-1) + \\ E_1A^{j-2}B_2\hat{\delta}(k) + E_1A^{j-3}B_2\hat{\delta}(k+1) + \cdots + E_1B_2\hat{\delta}(k+j-2) + E_3\hat{\delta}(k+j-1) + \\ E_1A^{j-2}B_3\hat{z}(k) + E_1A^{j-3}B_3\hat{z}(k+1) + \cdots + E_1B_3\hat{z}(k+j-2) + E_4\hat{z}(k+j-1) \\ \leq g_5 - E_1B_4 - E_1AB_4 - \cdots - E_1A^{j-2}B_4 \end{gathered}$$

This could also be written in a generalised form as shown below,

$$\underbrace{\begin{bmatrix} E_1 & 0 & \dots & 0 & 0 \\ 0 & E_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & E_1 & 0 \\ 0 & 0 & \dots & 0 & I \\ 0 & 0 & \dots & 0 & -I \end{bmatrix}}_{\hat{E}_1} \begin{bmatrix} x(k) \\ x(k+1) \\ \vdots \\ x\left(k+N_p-1\right) \\ x\left(k+N_p\right) \end{bmatrix} + \underbrace{\begin{bmatrix} E_2 & 0 & \dots & 0 \\ 0 & E_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & E_2 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{\hat{E}_2} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u\left(k+N_p-1\right) \end{bmatrix}$$

$$+ \underbrace{\begin{bmatrix} E_3 & 0 & \dots & 0 \\ 0 & E_3 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & E_3 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{\hat{E}_3} \begin{bmatrix} \delta(k) \\ \delta(k+1) \\ \vdots \\ \delta\left(k+N_p-1\right) \end{bmatrix} + \underbrace{\begin{bmatrix} E_4 & 0 & \dots & 0 \\ 0 & E_4 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & E_4 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}}_{\hat{E}_4} \begin{bmatrix} z(k) \\ z(k+1) \\ \vdots \\ z\left(k+N_p-1\right) \end{bmatrix} \leq \underbrace{\begin{bmatrix} g_5 \\ g_5 \\ \vdots \\ g_5 \\ \bar{x} \\ -\underline{x} \end{bmatrix}}_{\hat{g}_5}$$

$$\hat{E}_1\hat{x}(k-1) + \hat{E}_2\hat{u}(k) + \hat{E}_3\hat{\delta}(k) + \hat{E}_4\hat{z}(k) \leq \hat{g}_5$$

Additionally, we keep in mind that the states are bounded, and that the final state(x_{eq}) must be within the limits of $\underline{x} \leq x_{eq} \leq \bar{x}$. Correspondingly, additional rows are added to the matrices to fit this constraint equation. Although we have written this equation in this manner, we need to find a relation which incorporates the matrix $\tilde{V}_k$ as well as the relation between consecutive states.Now $\hat{x}(k-1)$ is written in terms of $\tilde{V}(k)$(according to the previous definition) and $x(k)$ as follows:

$$\hat{x}(k-1) = \begin{bmatrix} 0 \\ M_1 \end{bmatrix} \tilde{V}(k) + \begin{bmatrix} I \\ M_2 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ M_3 \end{bmatrix}$$

Now, we need to rewrite it in the form $F_1\tilde{V}(k) \leq F_2 + F_3 x(k)$

$$\underbrace{(\hat{E}_1 \begin{bmatrix} 0 \\ M_1 \end{bmatrix} + [\ \hat{E}_2 K_u \quad \hat{E}_3 K_\delta \quad \hat{E}_4 K_z\ ])}_{F_1} \tilde{V}(k) \leq \underbrace{\hat{g}_5 - \hat{E}_1 \begin{bmatrix} 0 \\ M_3 \end{bmatrix}}_{F_2} \underbrace{-\hat{E}_1 \begin{bmatrix} I \\ M_2 \end{bmatrix}}_{F_3} x(k)$$

2.8.2 Rewriting the Cost Function

We have the cost function

$$J(k) = \sum_{j=0}^{N_p-1} (\sum_{i=1}^{N_b} W_{b,i}|\Delta s_{b,i}(k+j)| + W_d|\Delta s_d(k+j)|) - W_{fuel}(x_d(k+N_p) - x_d(k))$$

$$-W_e \sum_{i=1}^{N_b}(x_{b,i}(k+N_p) - x_{b,i}(k)) + \sum_{j=0}^{N_p-1} P_{imp}(k+j)C_e(k+j)$$

where

$$P_{\text{imp}}\ (k+j) = P_{\text{load}}\ (k+j) - u_{\text{d}}(k+j) - \sum_{i=1}^{N_{\text{b}}} u_{\text{b},i}(k+j), \quad \forall j.$$

We need to rewrite this into a MILP form which can be used for optimisation of our MPC problem. Before we substitute the matrices M and F calculated previously, we first need to get our cost function into a linear form. We will do this term by term. The question 2.7 mentions creation of a model for 1 generator and 2 batteries. This code and the following operations will be performed with $N_b = 2$. The first part we address is $\sum_{j=0}^{N_p-1} P_{imp}(k+j)C_e(k+j)$.

$$\sum_{j=0}^{N_p-1} P_{imp}(k+j)C_e(k+j) = \sum_{j=0}^{N_p-1} (P_{\text{load}}\ (k+j) - u_{\text{d}}(k+j) - \sum_{i=1}^{N_{\text{b}}} u_{\text{b},i}(k+j))C_e(k+j)$$

where both $\tilde{P}_{load}$ and $\tilde{C}_e$ values are known. Therefore, their product can be written in a matrix form- $\tilde{C}_e^T \tilde{P}_{load}$. $-u_{\text{d}}(k+j) - \sum_{i=1}^{N_{\text{b}}} u_{\text{b},i}(k+j)$ can be written as $[-1\ \ -1\ \ -1]u(k+j)$ where $u(k+j) = [u_{b,1}(k+j)\ u_{b,2}(k+j)\ u_d(k+j)]^T$. Therefore,

$$\sum_{j=0}^{N_p-1} (P_{\text{load}}\ (k+j) - u_{\text{d}}(k+j) - \sum_{i=1}^{N_{\text{b}}} u_{\text{b},i}(k+j))C_e(k+j)$$

$$= \tilde{C}_e(k)^T \tilde{P}_{load}(k) + \underbrace{\tilde{C}_e(k)^T \begin{bmatrix} -1 & -1 & -1 & 0 & \cdots & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & -1 & -1 & -1 \end{bmatrix}}_{C_u} \hat{u}(k)$$

where $\hat{u}(k) = K_u \tilde{u}(k)$ as previously defined. The next terms we focus on are

$$-W_{fuel}(x_d(k+N_p) - x_d(k)) - W_e \sum_{i=1}^{N_b}(x_{b,i}(k+N_p) - x_{b,i}(k))$$

$$= -[W_e\ W_e\ W_{fuel}]x(k) + \underbrace{[0\ \cdots\ 0\ W_e\ W_e\ W_{fuel}]}_{C_x} \hat{x}(k)$$

where $x(k) = [x_{b,1}(k)\ x_{b,2}(k)\ x_d(k)]^T$ and $\hat{x}(k) = [x(k+1)\ x(k+2)\ \cdots\ x(k+N_p)]^T$ Now we tackle the final 2 terms- the 1 norms.

$$\sum_{j=0}^{N_p-1} (\sum_{i=1}^{N_b} W_{b,i}|\Delta s_{b,i}(k+j)| + W_d|\Delta s_d(k+j)|)$$

$$= \sum_{j=0}^{N_p-1} W_{b,1}|\Delta s_{b,1}(k+j)| + W_{b,2}|\Delta s_{b,2}(k+j)| + W_d|\Delta s_d(k+j)|$$

The trick to linearize these terms and get it into a linear form is using slack variables. Using the previous definitions of δ, we take $s_d(k) = \delta_{d1} + \delta_{d2} + \delta_{d3} + \delta_{d4}$.

$$\Delta s(k) = \begin{bmatrix} \Delta s_1(k) \\ \Delta s_2(k) \\ \Delta s_3(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \delta(k) + \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & -1 \end{bmatrix} \delta(k-1)$$

Using the definition of the 1 norm we can also write

$$= \sum_{j=0}^{N_p-1} W_{b,1}|\Delta s_{b,1}(k+j)| + W_{b,2}|\Delta s_{b,2}(k+j)| + W_d|\Delta s_d(k+j)|$$

as

$$\left\| \underbrace{\begin{bmatrix} W_{b,1} & 0 & 0 & 0 & 0 & 0 \\ 0 & W_{b,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & W_d & W_d & W_d & W_d \end{bmatrix}}_{W_\delta} \delta(k+j) + \begin{bmatrix} -W_{b,1} & 0 & 0 & 0 & 0 & 0 \\ 0 & -W_{b,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & -W_d & -W_d & -W_d & -W_d \end{bmatrix} \delta(k+j-1) \right\|_1$$

=

$$\left\| \underbrace{\begin{bmatrix} W_\delta & 0 & \cdots & \cdots & 0 \\ -W_\delta & W_\delta & 0 & \cdots & 0 \\ 0 & -W_\delta & W_\delta & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & -W_\delta & W_\delta \end{bmatrix}}_{C_{\delta 1}} \hat{\delta}(k) + \underbrace{\begin{bmatrix} -W_\delta \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{C_{\delta 2}} \delta(k-1) \right\|_1$$

where $\hat{\delta}(k) = [\delta(k)\ \cdots\ \delta(k+N_p-1)]^T$. We can now take $C_{\delta 1}\hat{\delta}(k) + C_{\delta 2}\delta(k-1) := \theta$ and our optimization problem can be written as

$$\min_{\rho,\theta \epsilon \mathbb{R}^n} [1\ \cdots 1]p(k)$$

subject to

$$-p \leq C_{\delta 1}\hat{\delta}(k) + C_{\delta 2}\delta(k-1) \leq p$$

in addition to the standard constraints on $C_{\delta 1}\hat{\delta}(k) + C_{\delta 2}\delta(k-1)$. We can thus re-write our cost function as

$$J(k) = [1\ \cdots 1]p(k) + C_x\hat{x}(k) + C_u\hat{u}(k) + [W_e\ W_e\ W_{fuel}]x(k) + \tilde{C}_e(k)^T P_{load}(k)$$

2.8.3 Optimizing the Modified Cost Function using the Modified MLD expressions

Now that we have a linear cost function, we can substitute the expressions for $\hat{x}(k) = M_1 x(k) + M_2 \tilde{V}(k) + M_3$ and $\hat{u}(k) = K_u \tilde{u}(k)$ which have been previously derived. Keep in mind that the objective function has to find optimal values of $\tilde{V}$ and $\tilde{\rho}$.

$$\min\ J(k) = \min\ [1\ \cdots 1]p(k) + C_x M_1 \tilde{V}(k) + C_x M_2 x(k) + C_x M_3 + [C_u K_u\ 0\ 0]\tilde{V}(k) + [W_e\ W_e\ W_{fuel}]x(k) + \tilde{C}_e(k)^T P_{load}(k)$$

The terms $C_x M_2 x(k),\ C_x M_3,\ [W_e\ W_e\ W_{fuel}]x(k)\ and\ \tilde{C}_e(k)^T P_{load}(k)$ can be removed from this optimisation since they are independent of the optimisation variables. Therefore, the final optimization is

$$\min\ [1\ \cdots 1]\tilde{p}(k) + C_x M_1 \tilde{V}(k) + [C_u K_u\ 0\ 0]\tilde{V}(k)$$

$$= \min\ \underbrace{[(1\ \cdots 1)\ (C_x M_1 + [C_u K_u\ 0\ 0])]}_{S_\rho} \underbrace{[\tilde{\rho}(k)\ \tilde{V}(k)]^T}_{\tilde{V}_\rho}$$

Now we need to group the constraints together to complete the definition of our MILP optimization problem. The constraints we have are

$$F_1 \tilde{V}(k) \leq F_2 + F_3 x(k)$$

$$-\rho \leq C_{\delta 1} \hat{\delta}(k) + C_{\delta 2} \delta(k-1) \leq \rho$$

The second constraints can be re-written as

$$-\rho - C_{\delta 1} K_\delta \tilde{\delta}(k) \leq C_{\delta 2} \delta(k-1)$$

$$-\rho + C_{\delta 1} K_\delta \tilde{\delta}(k) \leq -C_{\delta 2} \delta(k-1)$$

Therefore,

$$\underbrace{\begin{bmatrix} 0 & F_1 \\ -I & [0\ -C_{\delta 1} K_\delta\ 0] \\ -I & [0\ C_{\delta 1} K_\delta\ 0] \end{bmatrix}}_{F_{\rho,1}} \tilde{V}_\rho(k) \leq \underbrace{\begin{bmatrix} F_2 + F_3 x(k) \\ C_{\delta 2} \delta(k-1) \\ -C_{\delta 2} \delta(k-1) \end{bmatrix}}_{F_{\rho,2}}$$

Therefore our optimisation problem is

$$\min\ S_\rho \tilde{V}_\rho(k)$$

subject to

$$F_{\rho,1} \tilde{V}_\rho(k) \leq F_{\rho,2}$$