Graduate School Form 30
Updated 1/15/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Cansu Sener

Entitled

MODELING AND SIMULATION OF VEHICLE TO GRID COMMUNICATION USING HYBRID PETRI NETS

For the degree of Master of Science in Electrical and Computer Engineering

Is approved by the final examining committee:

Lingxi Li
Chair

Brian King

Maher Rizkalla

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Lingxi Li

Approved by: Brian King 6/8/2015

Head of the Departmental Graduate Program Date

MODELING AND SIMULATION OF VEHICLE TO GRID COMMUNICATION USING HYBRID PETRI NETS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Cansu Sener

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

August 2015

Purdue University

Indianapolis, Indiana

To my mother and father.

ACKNOWLEDGMENTS

I would like to first thank God for blessing me with such great parents, Sulun Gulsun and Dr. Siddik Sener. They have given me the opportunity to study with their endless support, encouragement, love, and belief. You are the best parents a kid could ever ask for, I truly dedicate all this work to both of you. And a special thanks goes to the rest of my family, I really appreciate all the support they have given me. Thank you Hikmet Duygu, Zeynep, Dr. Kadir Can, Bergil, Arel Emir, and Omar. You guys have supported me in every aspect of life. This would not be possible without your help and continuous moral contributions.

Also, I would like to thank my advisor Dr. Lingxi Li for his guidance, and all the help throughout my undergraduate and graduate education. I would also like to thank my committee members Dr. Maher Rizkalla and Dr. Brian King for their time, advices, and suggestions during the construction of this work. Another thanks goes to Dr. Steven Rovnyak, for pushing me towards great opportunities that are available.

I would like to thank my lab mates, Maryam Alibeik, Aja Anjilivelil for always giving their suggestions and helping out when I really need it. I would like to also give thanks to Sherrie Tucker for taking her time and revising my work, keeping me on track with all the deadlines, and to her never ending moral support. I also would like to thank Amanda Herrera, Mohammad Almutairi, Basak Kalelioglu for their moral support through out my education.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Sener, Cansu MSECE, Purdue University, August 2015. Modeling and Simulation of Vehicle to Grid Communication Using Hybrid Petri Nets. Major Professor: Lingxi Li.

With the rapid growth of technology, scientists are trying to find ways to make the world a more efficient and eco-friendly place. The research and development of electric vehicles suddenly boomed since natural resource are becoming very scarce. The significance of an electric vehicle goes beyond using free energy, it is environmental friendly. The objective of this thesis is to understand what Vehicle to Grid Communication (V2G) for an electric vehicle is, and to implement a model of this highly efficient system into a Hybrid Petri Net. This thesis proposes a Hybrid Petri net modeling of Vehicle to Grid (V2G) Communication topology. Initially, discrete, continuous, and hybrid Petri net's are defined, familiarized, and exemplified. Secondly, the Vehicle and Grid side of the V2G communication system is introduced in detail. The modeling of individual Petri nets, as well as their combination is discussed thoroughly. Thirdly, in order to prove these systems, simulation and programming is used to validate the theoretical studies. A Matlab embedded simulation program known as SimHPN is used to simulate specific scenario's in the system, which uses Depth-first Search (DFS) Algorithm. In addition to SimHPN simulation program, Matlab program is made to output four levels of the reachability tree as well as specifying duplicate and terminate nodes. This code incorporates a technique known as Breadth-first Search (BFS) Algorithm.

1. INTRODUCTION

1.1 Vehicle to Grid (V2G) Communication

A V2G communication topology is used with any type of vehicle that works with a battery, such as Electric Vehicles (EV) or Plug-in Hybrid Electric Vehicles (PHEV). Air pollution, which raised a lot of health issues, forced manufacturers to refocus their research into smarter vehicles with smarter batteries. In [1], the efficiency and energy consumption of an EV and an Internal Combustion Vehicle (ICV) are compared. The results shows that EV consumes eight times more energy then ICV. Also, the tank to wheel efficiency is twice as much in an EV than ICV. However, if the overall efficiency is considered (raw fuel to wheels) ICV has a better percentage. In an electric vehicle, the batteries that are preferred are high technology Lithium-Ion with high energy [2]. A typical Lithium-Ion batteries has the capability to store 1 to 60kWh with the output of 0.2 to 6kW [3]. Thus, with the V2G concept, the efficiency, stability and reliability of the electric grid can drastically improve [4]. Considering the energy sources, the battery of a vehicle is a very small resource that it is possible for us to neglect it's impact to the grid, but even though, its energy can still be useful during peak hours. [5]

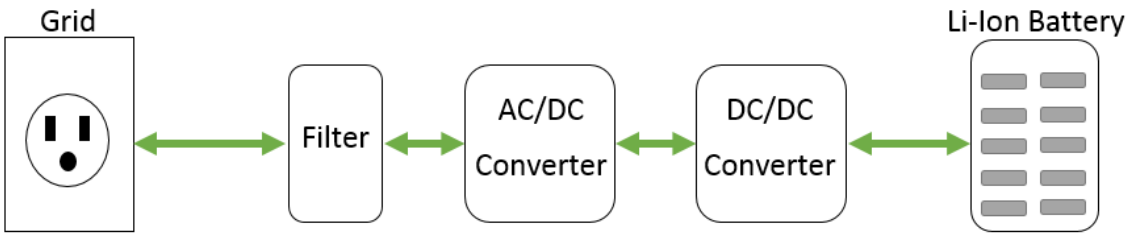


Fig. 1.1. Bidirectional model of V2G topology.

The main idea of charging a battery is to transform alternating current (ac) from the grid into direct current (dc) to a storage device (battery). In [2], the introduction of the charger as a bidirectional power flow operation was studied. In addition to the G2V direction, [2] illustrates the new and opposite concept called Vehicle to Grid (V2G) operations, which helps to return back the energy that has not been used by the vehicle. As observed in Fig. 1.1, a V2G topology that has a bidirectional model of a converter also incorporates a power factor correction to insure proper flow of power from the battery to the grid. This is to insure that the power that will be taken from the vehicle will be in-phase with the grid. In addition to bidirectional power flow to the converter, there is a dc-dc converter to control the battery charging and discharging [2] [6] [7].

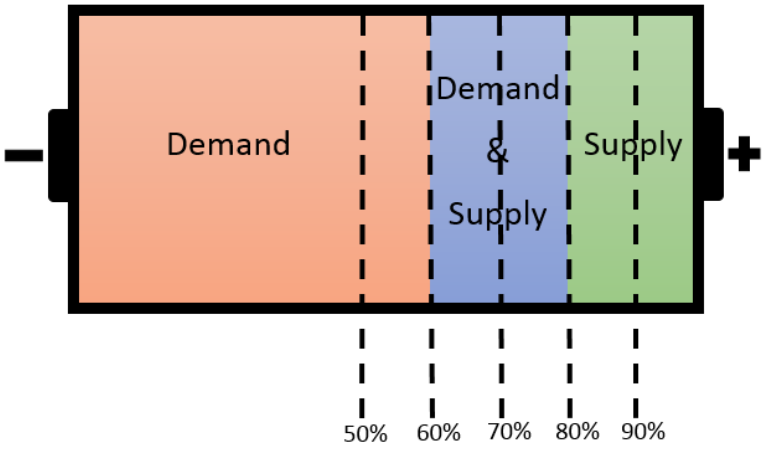


Fig. 1.2. Battery characteristics of it's state of charge.

In [2], three different types of experiments was made to test the behavior of state of charge were shown. For the first experimental type, the author introduced an independent partial differential equation which is cumbersome to determine. In the second part, an analytical simulation was introduced. However, the system ignored voltage and internal resistance, hence the results were not accurate. For the third part of the test, the author used an electric circuit-based model that has been tested

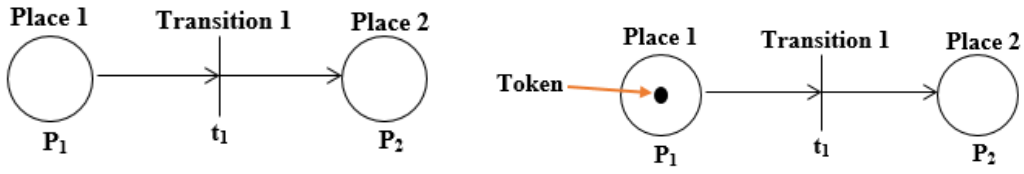
and preferred. Also, in this model was the inclusion of the self-discharge effect, which is an important criterion since cars can be parked for long periods of time. Fig. 1.2 shows the behavior of the state of charge of a typical battery in V2G systems. It is assumed that 60% is the tolerable level, and it can be observed that batteries are more likely to releases energy (supplying resource) when the state of charge is greater than 60%. Otherwise, if the battery has less than 60%, it will be more of the demanding resource.

1.2 Organization

This thesis is organized as follows. Chapter 2, discusses the definitions, behaviors, types, and dynamics of Discrete, Continuous, and Hybrid Petri nets. Chapter 3 illustrates the detailed modeling of the V2G systems, first separately, then combined. Chapter 4 discusses and shows the results of the simulation and Matlab implementation code. Finally, Chapter 5 concludes and presents future work.

2. BACKGROUND INFORMATION ABOUT PETRI NETS

Petri nets are a widely used concept to model and visualize behaviors of a system that has dynamics. These systems can be represented into two different types of nodes: places and transitions. A place is represented with a circle, and a transition is represented with a bar/box. These two nodes are connected with directed arcs (Fig. 2.1(a)). An arc always has the weight of a non-negative integer number. Note that if an arcs weight is not specified its assumed to be one. To indicate the characteristics of the net, the use of tokens or marks can be placed into the places shown in Fig. 2.1(b).



(a) (b)

Fig. 2.1. (a) Complete structure of a Petri Net. (b) Marked Petri Net.

Arcs are only connected with other nodes ($P \cap T = \emptyset$). In other words, no arc can be connected with the same type of node (transition to transition nor place to place) [8]. A mathematical expression that represents the relation between the arc and the nodes is,

$$(N, M_0) = (P, T, A, w) \tag{2.1}$$

where P is a finite set of places and T is a finite set of transition written as,

$$P \rightarrow \{p_1, p_2, ..., p_m\} \qquad T = \{t_1, t_2, ..., t_n\} \tag{2.2}$$

A is a set of arcs from places to transition and visa-versa,

$$A \subseteq (P \times T) \cup (T \times P) \tag{2.3}$$

w is the weight function on arcs,

$$w\colon A \rightarrow \{1, 2, 3, ...\} \tag{2.4}$$

and initial marking,

$$M_0\colon P \rightarrow \{1, 2, 3, ...\} \tag{2.5}$$

2.1 Discrete Petri nets

Discrete Petri nets are used to represent the dynamics of discrete events. It contains finite and non zero numbers for its place, transition, and arcs [9]. An Unmarked Discrete Petri net (UDPN) has four essential elements $UDPN = (P, T, B^+, B^-)$ where B^+ is called the output incident matrix with the dimension of P×T. This matrix captures arc weights from transition to places (if any exists). B^- is called the input incident matrix with the dimension of P×T. This matrix captures arc weights from places to transition (if any exists).

$$B^+(p,t) = \begin{cases} w(p,t), & \text{if } p \in t\bullet \\ 0, & \text{otherwise} \end{cases} \quad B^-(p,t) = \begin{cases} w(p,t), & \text{if } p \in \bullet t \\ 0, & \text{otherwise} \end{cases} \tag{2.6}$$

considering an example to better understand a UDPN. Fig. 2.2 is the representation of a simple UDPN where four places and two transitions exist.

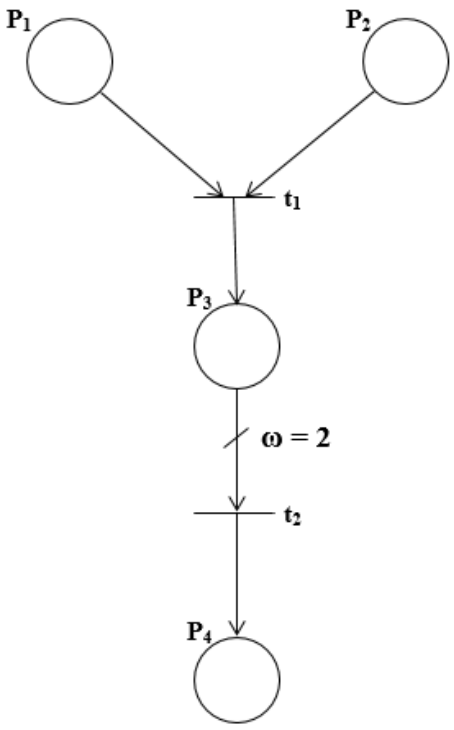


Fig. 2.2. Example of four placed, two transition Petri Net.

Mathematically it is defined as $P = \{P_1, P_2, P_3, P_4\}$, $T = \{t_1, t_2\}$, $A = \{(p_1, t_1)$, $(p_2, t_1), (p_3, t_2), (t_1, p_3), (t_2, p_4)\}$, $w = (p_1, t_1) = 1, w(p_2, t_1) = 1, w(p_3, t_2) = 1$, $w(t_1, p_3) = 1, w(t_2, p_4) = 2$. From the definitions that is obtained previously the input and output incident matrices as follows:

$$B^+ = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.7}$$

$$B^{-} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 2 \\ 0 & 0 \end{bmatrix} \tag{2.8}$$

These matrices have a very important role in analyzing a Petri net. It will help obtain the incident matrix, and therefore through the state equation new markings can be obtained.

2.1.1 Markings of Petri Nets

To create a dynamic in a system, tokens or marks can be added to a place (P). They are graphically represented as black dots, and mathematical represented by $m(P_i)$ or m_i where i is the place number (Fig. 2.3). For discrete petri net systems, tokens can be any non-negative integer number, $m(P_i) \in \mathbb{N}$. In addition to UDPN elements Marked Petri Nets (MPN) has m_0 with the column vector dimension of $P \times 1$. In order to be a MPN, at least one place in the system must have a token which represents in m_0 matrix. According to Eq.2.9, the initial marking of the system is obtained as:

$$m_0 = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 2 & 0 \end{bmatrix}^T \tag{2.9}$$

With marked Petri net the system has became a dynamic system. Capturing the initial marking will help to understand the dynamics of the system and also help to reach the next firing place.

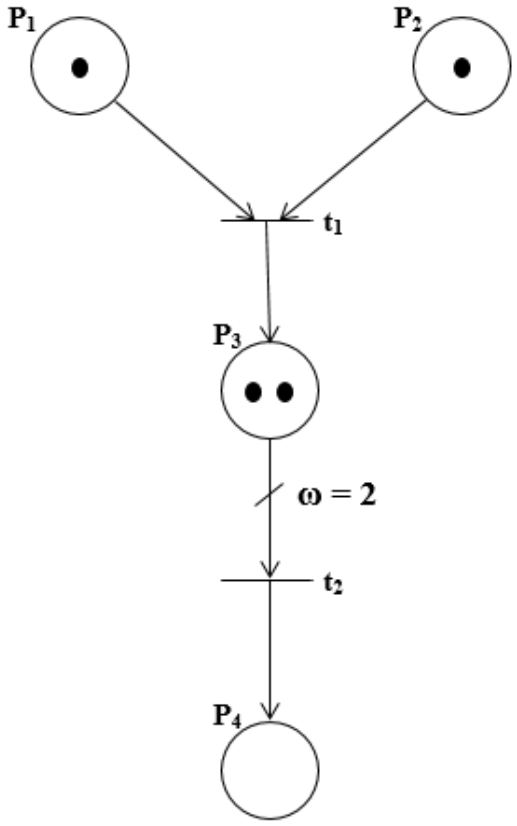


Fig. 2.3. Marked Petri Net.

2.1.2 Dynamics of a Petri Nets

The characteristics of this mechanism can be observed by it's states and changes. A changed state occurs when a transition is enabled (fires token). In an automata system, state transition mechanism is made by the arc that is connected to the place. In Petri net systems, a state transition function is made with the changing states by tokens. A state transition function, $f: \mathbb{N}^n \times T \rightarrow \mathbb{N}^n$, has three main conditions that can make a transition to fire token (change a state) [9]- [12]:

- Transition $t_j \in T$ can be enabled if the $m(p_i) \geq w(p_i, t_j)$ for all $p_i \in I(t_j)$ where, $I(t_j)$ is denoted as set of the input places to transition t_j.

- $m'(p_i) = m(p_i) - w(p_i, t_j) + w(t_j, p_i)$ when $i = 1, 2, ..., n$ if a token fired by the transition t_j, we remove the fired token with the weight of the arc from each input place $w(p_i, t_j)$ and deposit as many token as the weight of the arc into the output place $w(t_j, p_i)$.

- A set of reachable state can only be enabled one at a time. $R[(P, T, A, w, m)] := \{y \in \mathbb{N}^n : \exists s \in T^*(f(m, s) = y)\}$.

Petri nets (PN) are represented in reachability tree's or coverability tree's. It is represented only with markings and transitions with arcs. When illustrating in reachability tree, there are three components to know. The initial marking of a PN is called the root node. If a node appears in the same level or in a previous level, that node will be considered as a duplicate node. And finally, if a node can not fire any more transitions, that node will be labeled as terminate node.

Table 2.1.
Token values for input and output places

Input	P_1	P_2	P_3	P_4
t_1	1	1	0	0
t_2	0	0	2	0

Output	P_1	P_2	P_3	P_4
t_1	0	0	1	0
t_2	0	0	0	1

Upon these conditions, examining the given problem in the previous section will be considered here. Table 2.1 shows the enabled transitions for the input places and output places. In other words, if P_1 and P_2 has at least one token, that will make t_1 enabled ($t_1 : P_1P_2 \rightarrow P_3$). Similarly, if P_3 has at least two tokens, it can enable t_2 ($t_2 : 2P_3 \rightarrow P_4$). According to the example in Fig. 2.3, the initial marking of $m_0 = \begin{bmatrix} 1 & 1 & 2 & 0 \end{bmatrix}^T$, as considering Table 2.1, it can obtained that m_0 can enable the transitions t_1 or t_2. If $m_0 \xrightarrow{t_1} m_{1a}$ fires, the new marking m_{1a}

will be $m_{1a} = \begin{bmatrix} 0 & 0 & 3 & 0 \end{bmatrix}^T$. Or, if $m_0 \xrightarrow{t_2} m_{1b}$ fires, the new marking m_{1b} will be $m_{1b} = \begin{bmatrix} 1 & 1 & 0 & 1 \end{bmatrix}^T$. Following m_{1a} on the reachability tree, only t_2, $m_{1a} \xrightarrow{t_2} m_{2a}$ can fire. And the new marking would be $m_{2a} = \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}^T$. Likewise, if $m_{1b} \xrightarrow{t_1} m_{2b}$, the reached state would be $m_{2b} = \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}^T$. Since m_{2a} and m_{2b} have the same state and can not enable any more transition, this would indicate that all reachable states of this Petri net are achieved. In other words, if considering the firing sequence $S = \{t_1, t_2\}$, it can be seen that by firing transitions t_1 then t_2, the tree reaches the reachable marking of m_2 ($m_0 \xrightarrow{t_1,t_2} m_2$ or $S = t_2, t_1$ $m_0 \xrightarrow{S} m_2$). The reachability tree for this model is shown in Fig. 2.4. Where each node represents markings and arrows indicates the transition that is fired for that node.

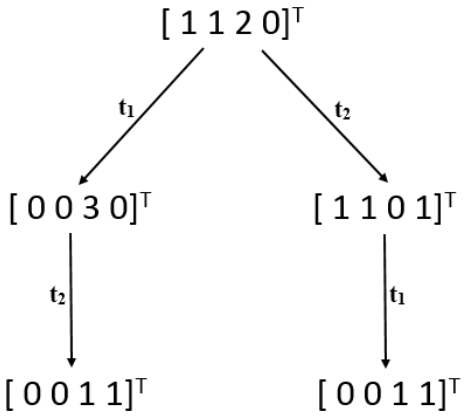


Fig. 2.4. Reachability tree.

Also, in some cases, a Petri net can have an infinite reachable markings. For instance the Petri net in Fig. 2.5(a) can produce infinite number of reachable states as illustrated in Fig.2.5(b).

2.1.3 Equations of a Petri Nets

There are some Petri nets that can have large number of places and transitions, therefore it may be harder to predict the next markings. With the help of linear

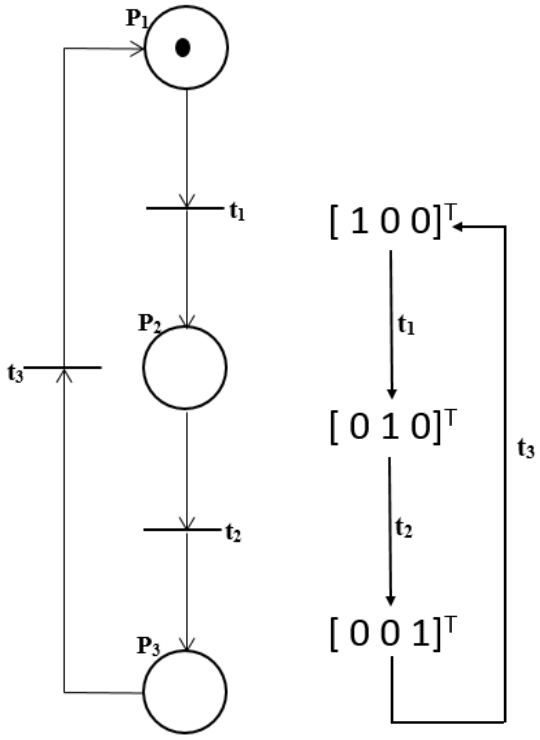


Fig. 2.5. (a) Petri Net. (b) Reachability Tree.

algebra, the construction of the state equation shows the final marking by inputting the initial marking with the firing sequence. The mathematical representation of the state equation is,

$$M_{k+1} = M_k + Bx \tag{2.10}$$

Where M_{k+1} is the new marking, and M_k is the initial/old marking both having dimension of $P \times 1$. B is the incident matrix where the firing vector x was introduced, and it has $T \times 1$ dimensional. B is the incident matrix which is found by,

$$B = B^+ - B^- \tag{2.11}$$

B^+ and B^- matrices have already been discussed Section 2.1. From Equation 2.11, the incident matrix can be found by,

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 2 \\ 0 & 0 \end{bmatrix} = \begin{matrix} \\ P_1 \\ P_2 \\ P_3 \\ P_4 \end{matrix} \begin{matrix} \begin{matrix} t_1 & t_2 \end{matrix} \\ \begin{bmatrix} -1 & 0 \\ -1 & 0 \\ +1 & -2 \\ 0 & +1 \end{bmatrix} \end{matrix} \tag{2.12}$$

An incident matrix shows that when transition t_1 enables, P_1 and P_2 lose 1 token and P_3 gains 1 token. Similarly, when transition t_2 enables, P_3 loses 2 tokens and P_4 gains 1 token.

In Discrete Petri nets (DPN), the firing vector (x) can only have a non-zero integer value of 1. This value indicates the corresponding enabled transition. Taking the example in Fig. 2.3, where $S = \{t_1, t_2\}$, then firing vector of x is equal to $\begin{bmatrix} 1 & 1 \end{bmatrix}^T$. If Equation 2.10 is used, obtaining the reachable state for $m_0 \xrightarrow{S} m_2$ will be possible,

$$M_2 = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ -1 & 0 \\ +1 & -2 \\ 0 & +1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \tag{2.13}$$

Therefore, the state equation is a very useful formula to find the reachable state for complex and large scale Petri net systems, especially when it comes to programming large scale systems.

2.2 Continuous Petri Nets

So far the discussions of discrete sequence events that can be represented by Discrete Petri nets were mentioned. However, there are many examples in real life that use continuous sequence events such as continuous production of a machine.

The rules that have been introduced in Discrete Petri nets are still applied with Continuous Petri nets (CPN). The representation of a continuous place will be with double circles, and the representation of a continuous transition is usually done with a rectangle (Fig. 2.6).

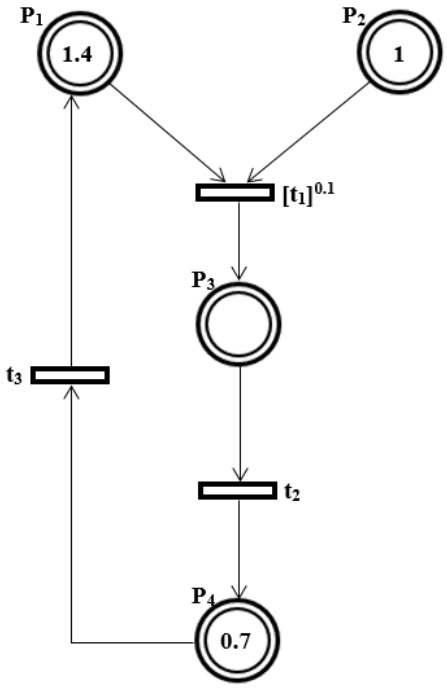


Fig. 2.6. Marked Petri Net.

There are four main differences between DPN and CPN [13],

- The weight of an arc that is connected from a transition to place or visa-versa can have any finite positive real number.

$$w(p_i, t_j)\,\&\,w(t_j, p_i) \rightarrow \forall : \Re^+$$

- Similarly, token values in CPN can be any finite non-negative real number.

- A continuous transition can fire when $t_j \in T$, $m_i > 0$,

$$m_i \iff \forall p_i \in \bullet t_j$$

with a firing degree,

$$f_{deg}(t_j, m) = \min_{p_i \in \bullet t_j} \left(\frac{m_i}{B^-(p_i, t_j)} \right)$$

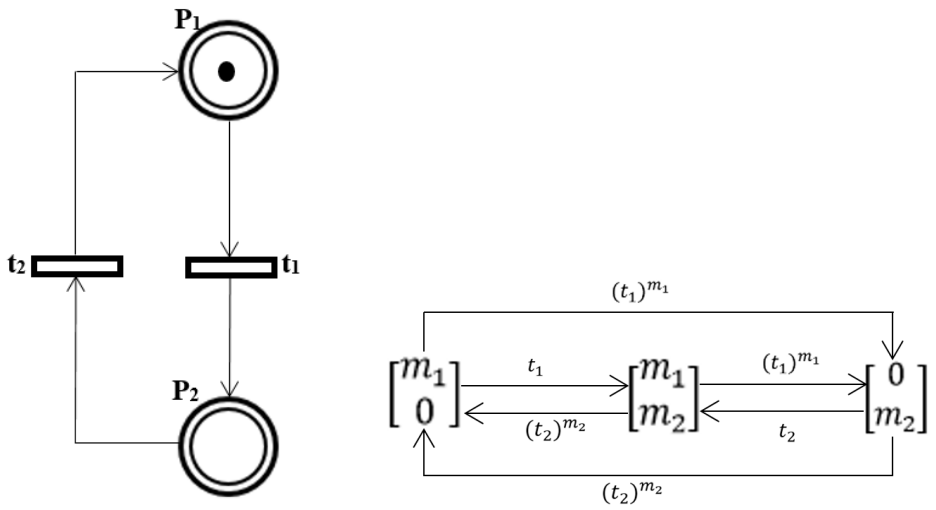
Hence, by observing Fig. 2.6, it can be noted that t_1 is 1-enabled, t_2 is 0-enabled, and t_3 is 0.7-enabled

- In a continuous transition, the firing rate of a transition, $[t]^\alpha$, is introduced where $\alpha \in \Re^+$. For example, in Fig. 2.6, let the firing rate of t_1 be 0.1, $[t_1]^{0.1}$, and knowing that $m_0 = \begin{bmatrix} 1.4 & 1 & 0 & 0.7 \end{bmatrix}^T$ after t_1 fires, $m_0 \xrightarrow{t_1} m_1$, $m_1 = \begin{bmatrix} 1.3 & 0.9 & 0.1 & 0.7 \end{bmatrix}^T$ will be obtained. This means that, when t_1 enables, p_1 and p_2 each drop 0.1 token and deposit 0.1 token to p_3.

Representing a reachability tree of a continuous system would be very challenging, this is because CPN can have infinite number of reachable markings $(m_1, m_2, ..., m_\infty)$ depending on its weight of place, arc, and transitions. Hence, Macro-marking of a reachability tree is introduced. Fig. 2.7(a) examines the concept of reachability, and we can attain a reachability tree of CPN (Fig. 2.7(b)). To retrieve possible reachable states, the $2^{\#ofplace}$ is used. As it's illustrated in Fig. 2.7, there is only two numbers of places. Therefore, $2^{\#ofplace} = 2^2 = 4$, which means that there are 4 possible states that can obtained. These are,

$$m_1^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, m_2^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, m_3^* = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, m_4^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{2.14}$$

However by careful observation, its identified that m_4 state can not be reached by any firing sequence; Fig. 2.6(b) illustrates three reachable states [14].



(a) (b)

Fig. 2.7. (a) Continuous Petri Net. (b) Macro-marking.

2.3 Hybrid Petri Nets

Hybrid Petri nets (HPN) are used for both discrete and continuous systems. These HPN's include transportation, manufacturing, or automation systems. Graphical notations from previous sections describing DPN and CPN will also be used in HPN. To differentiate between continuous places, discrete places, and transitions in mathematical forms, consider,

p_i^d : discrete place $\qquad$ t_j^d : discrete transition

p_i^c : continuous place $\qquad$ t_j^c : continuous transition

All the rules that have been discussed in previous chapters apply to this section. There are some constraints when discrete and continuous places and transitions merge into the same systems. These are,

- A Hybrid Petri net system consists of six important elements which are,

$$P, T, B^+, B^-, m_0, h$$

where h is called a hybrid function and it's responsibility is to indicate the type of the node.

$$h : P \in T \rightarrow \{D, C\}$$

- In a continuous transition (t_j^c), if a discrete place (p_j^d) enters, it has to have an output back to that discrete place with the same weight.

Fig. 2.8 is an example of a HPN system where,

$p_1 \& p_3$: continuous places $\quad$ $t_1 \& t_2$: discrete transitions

$p_2 \& p_4$: discrete places $\quad$ $t_3 \& t_4$: discrete transitions

Let the firing rate be 0.2 ($[t_1^c]^\alpha = [t_1^c]^{0.2}$), It's known that initial marking is $m_0 = \begin{bmatrix} 1.4 & 1 & 0 & 0 \end{bmatrix}^T$ once t_1^c fires ($m_0 \xrightarrow{t_1^c} m_1$) $m_1 = \begin{bmatrix} 1.2 & 1 & 0.2 & 0 \end{bmatrix}^T$ is retrieved.

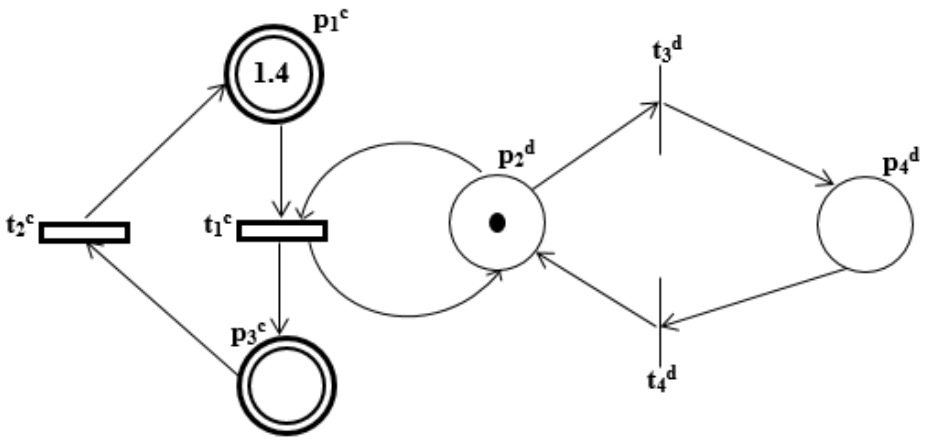


Fig. 2.8. Marked Petri Net.

Notice that when p_1 or p_3 have less token than α, t_1 would not be disabled. In conclusion, In this chapter Discrete, Continuous, and Hybrid petri nets are introduced and illustrated case by case. Now it is the time to use HPN modeling to demonstrate vehicle to grid communication systems.

3. MODELING OF A V2G SYSTEM

To illustrate a Hybrid Petri net model of a V2G system, the grid and electric vehicle systems will be discussed individually. Then, these two systems will be combined into one system with consideration of V2G system characteristics that have been discussed in previous chapters.

3.1 Related Work

Vehicle to grid topology with Petri net modeling is a growing field. In [15], the authors introduced the stochastic Petri net modeling of electric vehicles in transportation system by capturing the discrete events with little traffic flow, and this presents the dynamics that are continuous, which is defined as Hybrid Dynamic Model (HDM). It also introduced the Transportation Electricity Nexus (TEN) concept, which captures the gap between the vehicles along drive portions. Once this concept was combined with HDM, the authors found interesting simulation results. These results showed specific scenarios of five electric vehicles that travel to three different destinations by depicting their state of charge and relative speeds. In [16], the introduction of a Hybrid Petri net model for synchronized and asynchronized dual motored vehicles was obtained. This work has thoroughly exemplified single level three phase inverters for single or dual motor operated vehicles. It used HPN modeling to evaluate and monitor the design capability and performance of the motors. For large power systems, [17] Hybrid Petri nets were used to assess the accuracy of a dynamic system. It investigated the charging stations as a model to analyze dynamic token behavior.

3.2 Model of the Grid side

To develop a HPN model of the grid, analyzing the power flows to the grid was studied. Modeling has been done considering the power rules of the United States. For simplicity, this thesis shows the power flow to the grid through four stages.

- Energy Sources: This can be any type of energy source, usually Nuclear, Solar, or Wind.
- Converters: Depending on the type of energy obtained from the source, it must enter a dc-dc/ac-dc, then to a dc-ac converter.
- Transmission Lines: Transfer high voltages generally between 69 kV to 765 kV for a typical distance of 300 miles [18].
- Distribution Lines: Carries out the lower voltages between 4 kV to 46 kV, and distribute them to commercial, industrial, or residential customers [18].
- Power Grid: The power obtained from energy sources.

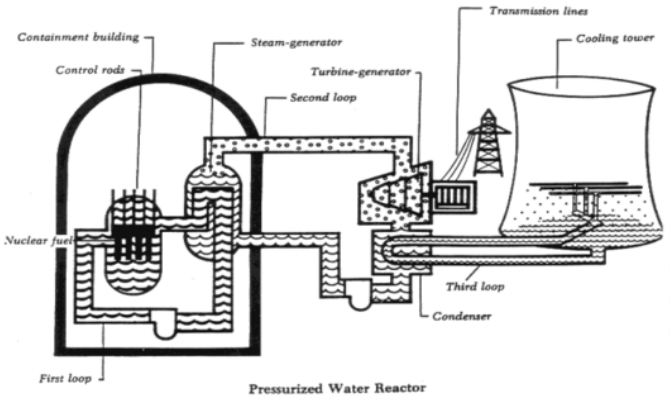


Fig. 3.1. Nuclear Power Plant [19].

Fig. 3.2 is a brief graphical explanation of the voltage flow to the grid. The energy starts from the power plants, regardless of where the energy is coming from. In case of a nuclear power plant, a simple way of explaining Fig. 3.1 is the heat caused by

the radioactive rods that helps making the water steam. The steam then moves the turbines, which generates electricity. A wind turbine works in a similar manner, instead of steam moving the turbine, its just wind. The generator then again converts it to electricity. Solar plants use photovoltaic cells to convert energy from the sun to electric power.

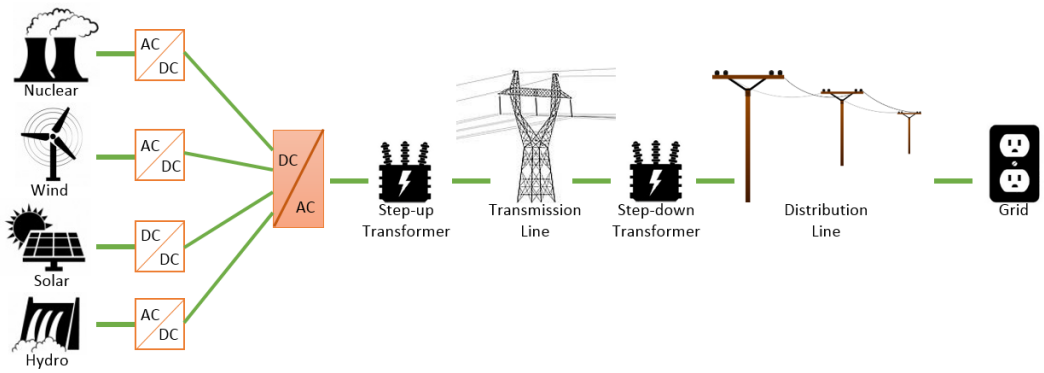


Fig. 3.2. Graphical expression of Grid-side.

Table 3.1.
Corresponding labels of places and transitions.

Places	Denotations
P_1^d	Energy Sources
P_2^d	Converters
P_3^d	Transmission Lines
P_4^d	Distribution Lines
P_5^d	Power Grid

Transitions	Connections
t_1^d	$P_1^d \rightarrow P_2^d$
t_2^d	$P_2^d \rightarrow P_3^d$
t_3^d	$P_3^d \rightarrow P_4^d$
t_4^d	$P_4^d \rightarrow P_5^d$

All the electricity generated by any of these plants is usually gone through a converter and transformer, which converts the obtained power into a three phase ac power, and steps up the voltages to very high levels. The three phase ac power then travels through a transmission line from the plants to the cities. Once it reaches the residential areas, it is then transformed into lower voltages and distributed throughout the neighborhood on distribution lines. Fig. 3.3 gives the PN representation of Fig. 3.2, where places and transitions are defined as discrete characteristics (Table 3.1).

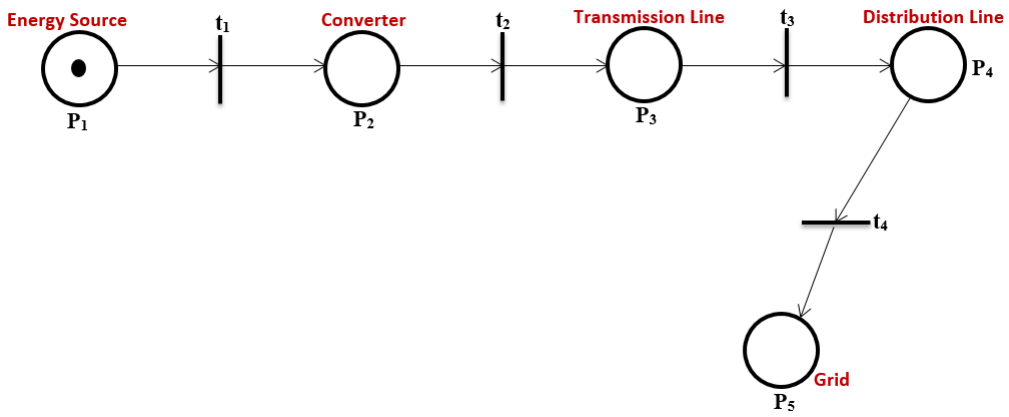


Fig. 3.3. Grid-side Petri Net.

3.3 Model of the Vehicle side

In this thesis, the topology of an electric vehicle will be used for modeling in the V2G communication part. This section of the system is briefly divided into eight component:

- Power Grid: The power obtained from the energy sources which are used to charge the vehicle's battery.
- Battery: Lithium-Ion (Li-Ion) cells are typically used in electric vehicles.
- User Input: The driver, who is operating the vehicle using the acceleration/break commands.
- Controller: Controls the level of voltage needed for propulsion.
- Converter: works as a dc/ac and ac/dc.
- Motor: Converts electrical to mechanical energy.
- Transmission: Converts mechanical energy into propulsion.
- Wheels: Final destination for the energy, easing the movement of the vehicle.

For electric vehicles, the power needed to charge the batteries are available at the grid. The batteries are typically made from Li-Ion cells, and they store all the energy charged from the grid. The batteries also provide power for the vehicle, including propulsion system. According to the graphical expression of an electric vehicle as given in Fig. 3.4, once the driver starts the vehicle, the controller is constantly waiting for commands to see what kind of actions it needs to take. When the car starts

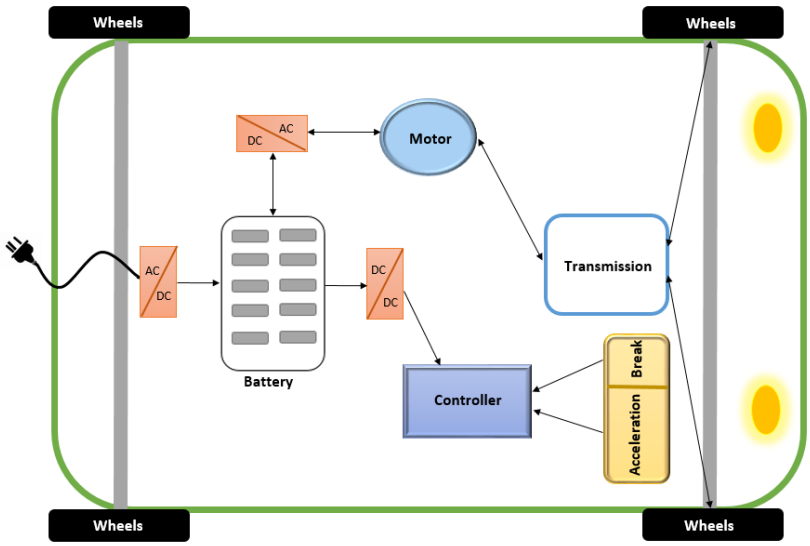


Fig. 3.4. Graphical Representation of Vehicle-side.

to move, and the user starts to push on the gas pedal, the controller is constantly adjusting the signals sent to the converter linearly, which is converting the dc power to ac power. These signals will basically control the power that the motor will be using. Once the power reaches the electric motor, it is then converted into mechanical energy (torque). The transmission then converts the torque into propulsion energy, via driveshaft and differentials, which basically rotate the wheels. When the driver is braking, the controller realizes that there is no more energy needed to move. The motor is then connected directly to the wheels, and the motor acts as a generator so that the energy that was created is not lost completely.

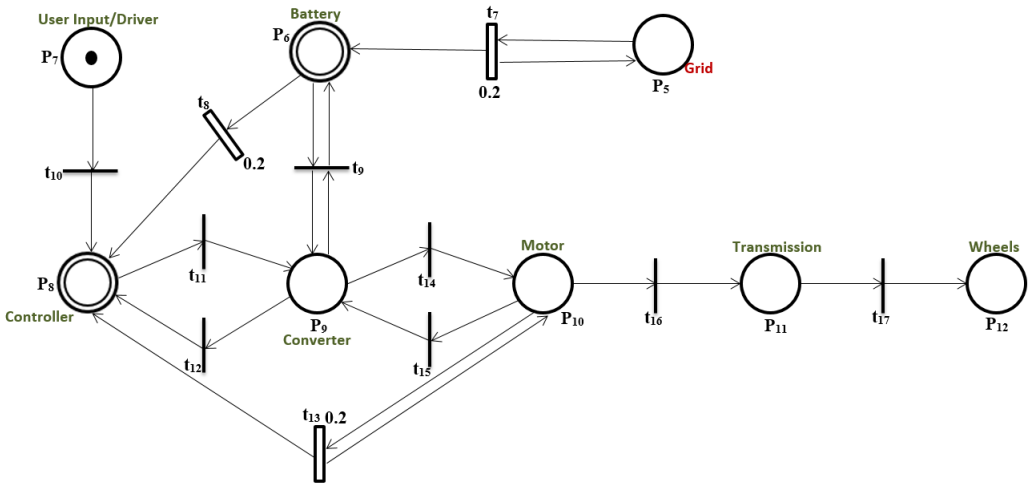


Fig. 3.5. Vehicle-side Petri Net.

Table 3.2.
Labels of places and transitions.

Places	**Denotations**
P_5^d	Power Grid
P_6^c	Battery
P_7^d	User Input/Driver
P_8^c	Controller
P_9^d	Converter
P_{10}^d	Motor
P_{11}^d	Transmission
P_{12}^d	Wheels

Transitions	**Connections**
t_7^c	$P_5^d \rightarrow P_5^d \& P_6^c$
t_8^c	$P_6^c \rightarrow P_8^c$
t_9^d	$P_6^c \& P_9^d \rightarrow P_6^c \& P_9^d$
t_{10}^d	$P_7^d \rightarrow P_8^c$
t_{11}^d	$P_8^c \rightarrow P_9^d$
t_{12}^d	$P_9^d \rightarrow P_8^c$
t_{13}^c	$P_{10}^d \rightarrow P_{10}^d \& P_8^c$
t_{14}^d	$P_9^d \rightarrow P_{10}^d$
t_{15}^d	$P_{10}^d \rightarrow P_9^d$
t_{16}^d	$P_{10}^d \rightarrow P_{11}^d$
t_{17}^d	$P_{11}^d \rightarrow P_{12}^d$

The energy then flows back through the motor (acting as a generator), and goes to an ac to dc converter to charge the battery. Fig. 3.5 illustrates the Petri net representation of the Vehicle-side of the system. Notice that the battery (P_6^c) and controller (P_8^c) have been considered as continuous places. And the transition that connects grid to battery (t_7), battery to controller (t_8) and motor to controller(t_{12})

are considered as continuous transitions with the firing rate $\alpha = 0.2$. The overall the system components are illustrated in Table 3.2.

3.4 Combination of the Models

This section covers the employment of power grid and electric vehicle into the V2G system as given in Fig. 3.4. As it has been deliberated in both structures, there is a common component which is used in both structures, Power grids (P_5^d). Fig. 3.7 illustrates the complete HPN modeling of a V2G communication system. As it has been discussed in Section 1.1, V2G communication system has bidirectional power flow, which means that in addition to the transition (t_4^d) that connects the distribution line to grid ($P_4^d \rightarrow P_5^d$), there should also be a new transition (t_5^d) that would connect grid to distribution line ($P_5^d \rightarrow P_4^d$). Likewise, in addition to the transition (t_7^c) that connects the power grid to the electric vehicle's battery ($P_5^d \rightarrow P_5^d \& P_6^c$), the power that is stored in the battery can be transmitted by a new transition (t_6^c) to the grid ($P_5^d \& P_6^c \rightarrow P_5^d$).

Hence, the new transitions ($t_4^d \& t_7^c$) transmit the stored power inside the electric vehicle's battery to the distribution lines during the peak hours or when there is a big demand of power.

Same constraints for places and transitions still apply in this section. In addition, the transition that connects to grid the battery is considered to have a continuous behavior with a firing rate of 0.2. Detailedly, the affiliation between places and transitions are represented in Table 3.3.

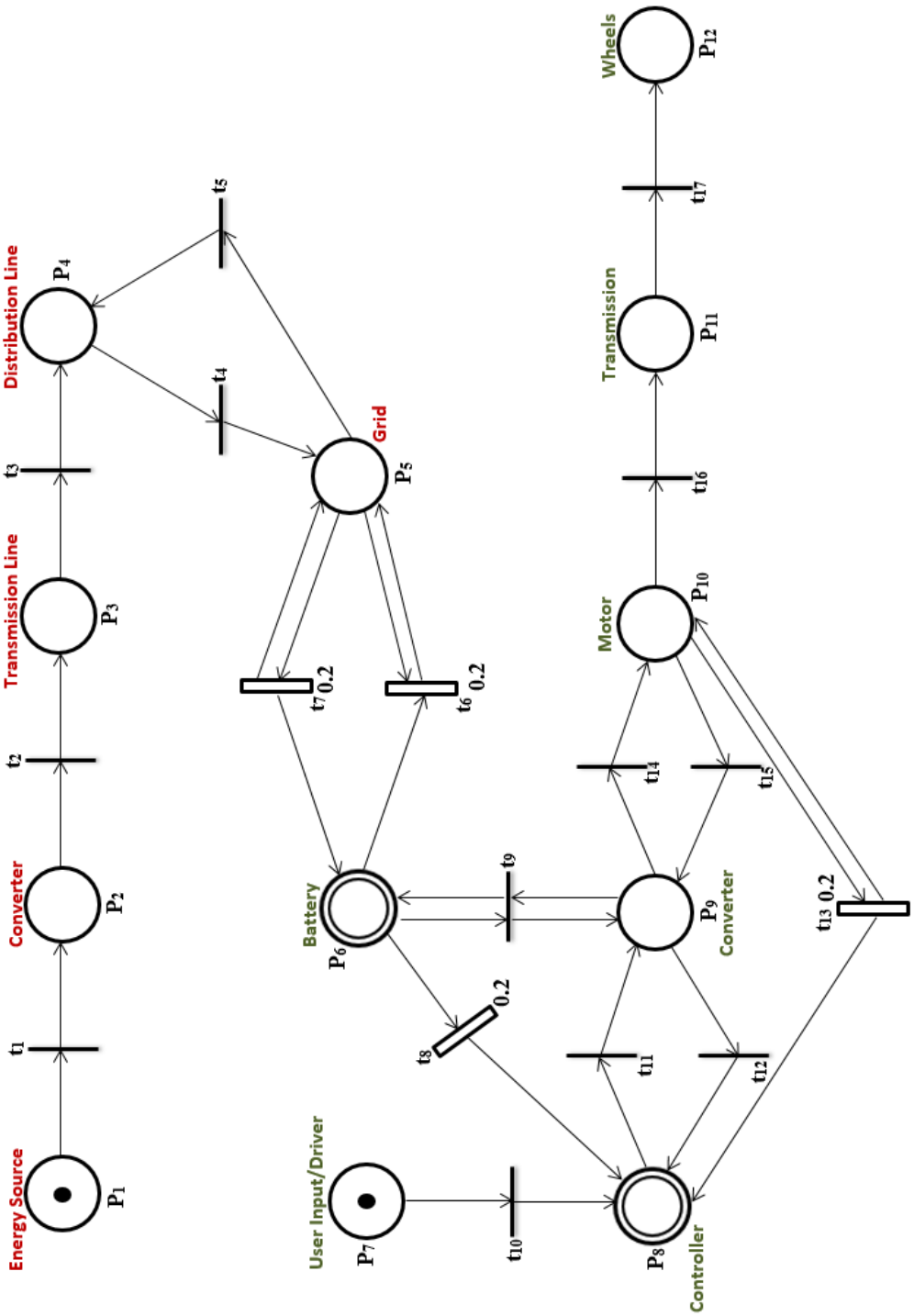


Fig. 3.6. Vehicle to Grid Topology Petri Net.

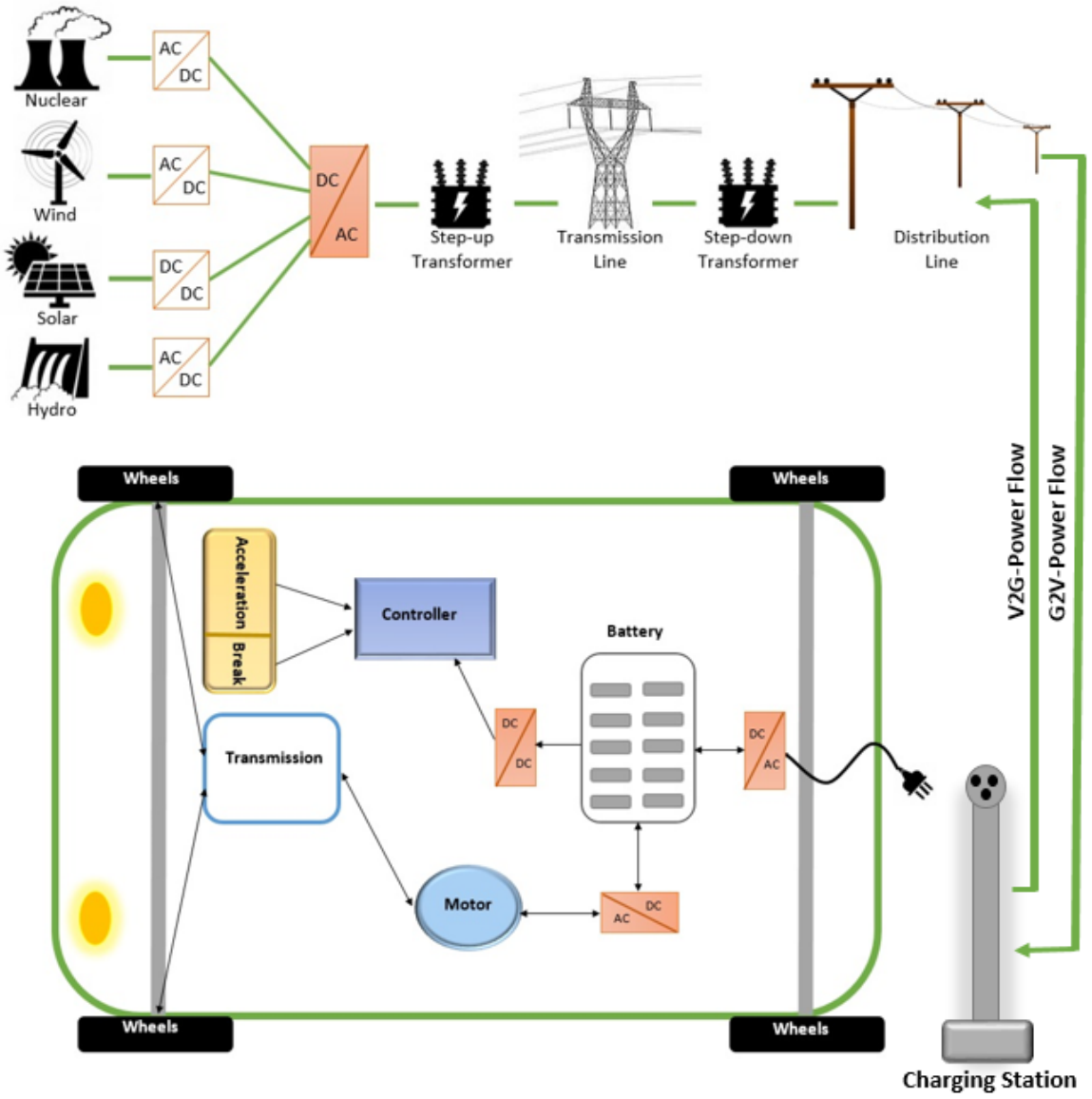


Fig. 3.7. Graphical Representation of Vehicle to Grid Topology.

Table 3.3.
Relationship between places and transitions.

Places	**Denotations**
P_1^d	Energy Sources
P_2^d	Converters
P_3^d	Transmission Lines
P_4^d	Distribution Lines
P_5^d	Power Grid
P_6^c	Battery
P_7^d	User Input/Driver
P_8^c	Controller
P_9^d	Converter
P_{10}^d	Motor
P_{11}^d	Transmission
P_{12}^d	Wheels

Transitions	**Connections**
t_1^d	$P_1^d \rightarrow P_2^d$
t_2^d	$P_2^d \rightarrow P_3^d$
t_3^d	$P_3^d \rightarrow P_4^d$
t_4^d	$P_4^d \rightarrow P_5^d$
t_5^d	$P_5^d \rightarrow P_4^d$
t_6^d	$P_5^d \& P_6^c \rightarrow P_5^d$
t_7^c	$P_5^d \rightarrow P_5^d \& P_6^c$
t_8^c	$P_6^c \rightarrow P_8^c$
t_9^d	$P_6^c \& P_9^d \rightarrow P_6^c \& P_9^d$
t_{10}^d	$P_7^d \rightarrow P_8^c$
t_{11}^d	$P_8^c \rightarrow P_9^d$
t_{12}^d	$P_9^d \rightarrow P_8^c$
t_{13}^c	$P_{10}^d \rightarrow P_{10}^d \& P_8^c$
t_{14}^d	$P_9^d \rightarrow P_{10}^d$
t_{15}^d	$P_{10}^d \rightarrow P_9^d$
t_{16}^d	$P_{10}^d \rightarrow P_{11}^d$
t_{17}^d	$P_{11}^d \rightarrow P_{12}^d$

4. RESULTS

To evaluate and examine the accuracy of the modeling, some testing was performed. These testings helped analyze the response and behavior of the system for a given scenario or possible scenarios. The input incident matrix (4.1) and output incident matrix (4.2) based on V2G modeling given in Section 3.4. are:

$$B^{-} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.1}$$

$$B^{+} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

By using Equation 2.11, the incident matrix B that illustrates the token behavior if a specific transition fires, obtained as:

$$B = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The firing vector x is given as:

$$x = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0.2 & 0.2 & 0.2 & 1 & 1 & 1 & 1 & 0.2 & 1 & 1 & 1 & 1 \end{bmatrix}^T \tag{4.3}$$

The continuous transitions (t_6^c, t_7^c, t_8^c and t_{13}^c) are marked by their firing rate $\alpha = 0.2$, the rest of the discrete transitions has the default firing rate $\alpha = 1$.
According to the V2G communication topology in Section 3.4, there are 12 places, which make the number of possible reachable states to be 4096 (2^{12}), with the initial marking M_0 of:

$$M_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{4.4}$$

Where the system initiates in two different places (P_1^d and P_7^d) with each having one token. The rest of the sections were done by using the findings of Equations 4.1 to 4.4.

4.1 Simulation Results

To simulate the HPN system, a demo version of a Matlab Toolbox embedded simulation program called SimHPN was used [20]. With the input of B^- and B^+ matrices, this simulation program was made to examine the behavior of an individual or multiple places in one specific path in the reachability tree. Fig. 4.1 is the main window of SimHPN simulator, where the x-axis indicates transition number and y-axis indicates the corresponding firing rate of that specified transition. It illustrates t_1^d to t_{17}^d transitions by obtaining the inputs as defined below:

- Pre: Input incident matrix B^-. In addition to the Equation 2.6 representation, it also represents the pre entrance of a transition.
- Post: Output incident matrix B^+. In addition to the Equation 2.6 representation, it also represents the post exiting of a transition.

- M_0: Initial marking as initialized in Equation 4.4.
- Lambda: Firing vector (x) as initialized in Equation 4.3.
- Transition Type: Depending on the type of the transitions classified, if the transition is discrete use "q" and if continuous use "c".

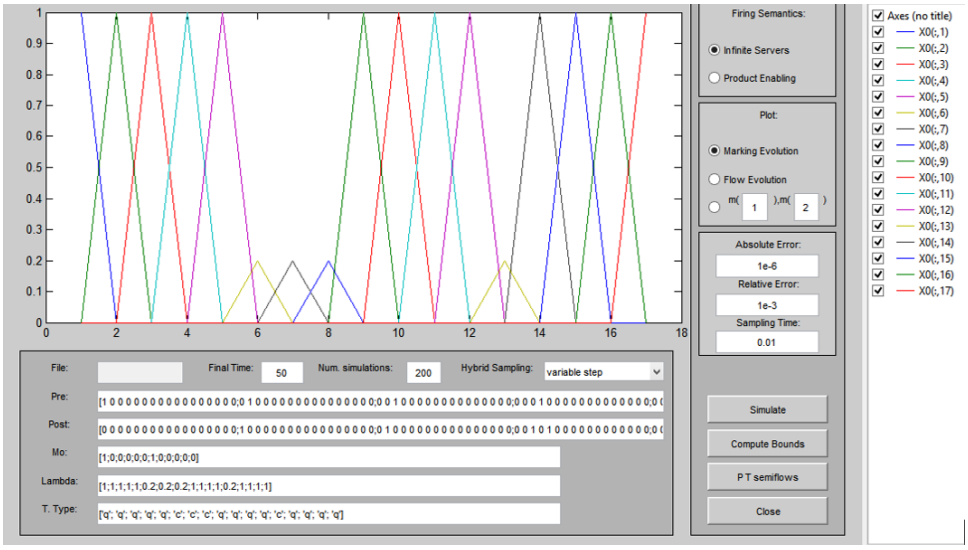


Fig. 4.1. SimHPN Output of transitions

In addition to the graphical representation of transitions of V2G topology in HPN modeling in SimHPN, Fig. 4.2 illustrates the incident matrix. It can be analyzed by Table 4.1.

Another way to understand the data presented in Fig. 4.2 would be by looking at each column of the incident matrix. Recall that each column represents the transitions as well as rows representing places. For example, if $B(:,1)$ is examined, it indicates that when t_1^d fires, P_1^d loses one token but P_2^d gains one token.

If a token starts from P_1^d, the system will flow like in Fig. 4.3. Where blue line represents P_1^d, green line P_2^d, red line P_3^d, light blue line P_4^d, and purple line P_5^d. The x-axis represents the number of steps and y-axis represents the value of tokens. As seen at the begining of the graph, there is one token in Energy sources (P_1^d).

Table 4.1.
Correlation between places and transitions.

Places	**Values**
P_1^d	$(-1)t_1$
P_2^d	$(1)t_1\&(-1)t_2$
P_3^d	$(1)t_2\&(-1)t_3$
P_4^d	$(1)t_3\&(-1)t_4\&(1)t_5$
P_5^d	$(1)t_4\&(-1)t_5$
P_6^d	$(1)t_6\&(-1)t_7\&(-1)t_8\&(-1)t_9$
P_7^c	$(-1)t_{10}$
P_8^c	$(1)t_{10}\&(-1)t_{11}\&(-1)t_{12}\&(-1)t_{13}$
P_9^d	$(1)t_{11}\&(-1)t_{12}\&(-1)t_{14}\&(1)t_{15}$
P_{10}^d	$(1)t_{14}\&(-1)t_{15}\&(-1)t_{16}$
P_{11}^d	$(1)t_{16}\&(-1)t_{17}$
P_{12}^d	$(1)t_{17}$

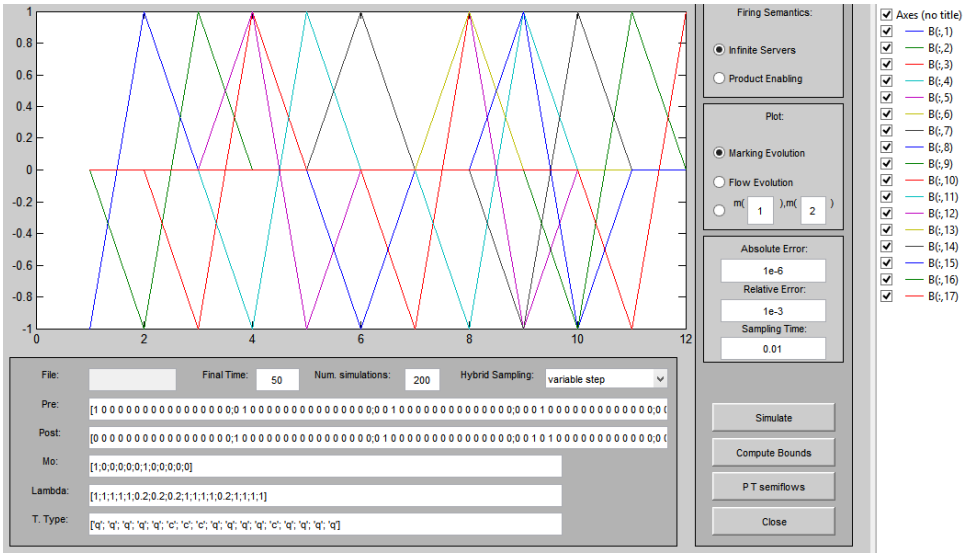


Fig. 4.2. Graphical representation of Incident Matrix

Then by the time the second step occurs, P_1^d loses it's token and the proportionally converter (P_2^d) gains one token. Followed by the transmission line (P_3^d) and the distribution line (P_4^d), when the token reaches the fourth step, it will reach to a loop. Whenever the electric vehicle is in need of power, the grid provides it or vice-versa.

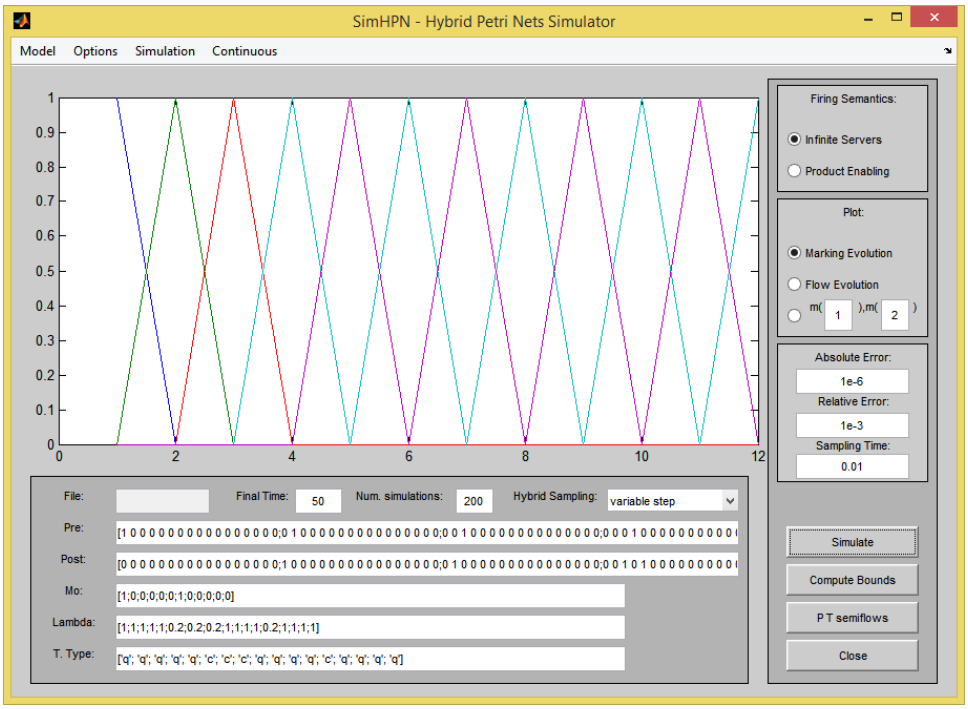


Fig. 4.3. Simulation of Grid Side

Fig. 4.4 illustrates the battery charging case, where the blue line represents P_5^d, green line P_6^c, light blue line P_8^d, purple line P_9^d, yellow line P_{10}^d, and brown line P_{11}^d. The x-axis represents the number of steps and y-axis represents the value of a token. Simulation captures the instance when the grid (P_5^d) has one token which would make t_7^c enabled. Once t_7^c is fired, the grid will have 0.8 tokens left and the battery (P_6^c) gains 0.2. In this case, battery can send the power of 0.2 token amount to the controller by firing t_8^c. These three steps can be repeated until the controller reaches to 1 token, therefore it can enable t_{10}^d,t_{14}^d,t_{15}^d, and t_{14}^d.

Fig. 4.5 illustrates the HPN simulation result of the electric vehicles side of the system. Where the blue line represents P_7^d, green line P_8^c, red line P_9^d, light blue line P_{10}^d, purple line P_{11}^d, and yellow line P_{12}^d. The x-axis represents the number of steps and y-axis represents the value of a token. Fig. 4.5 interprets the case when the driver (P_7^d) brakes or accelerates, therefore enables t_9^d, and as a result P_7^d loses it's token and sends the signal (token) to the controller. From there, controller P_8^c enables and fires

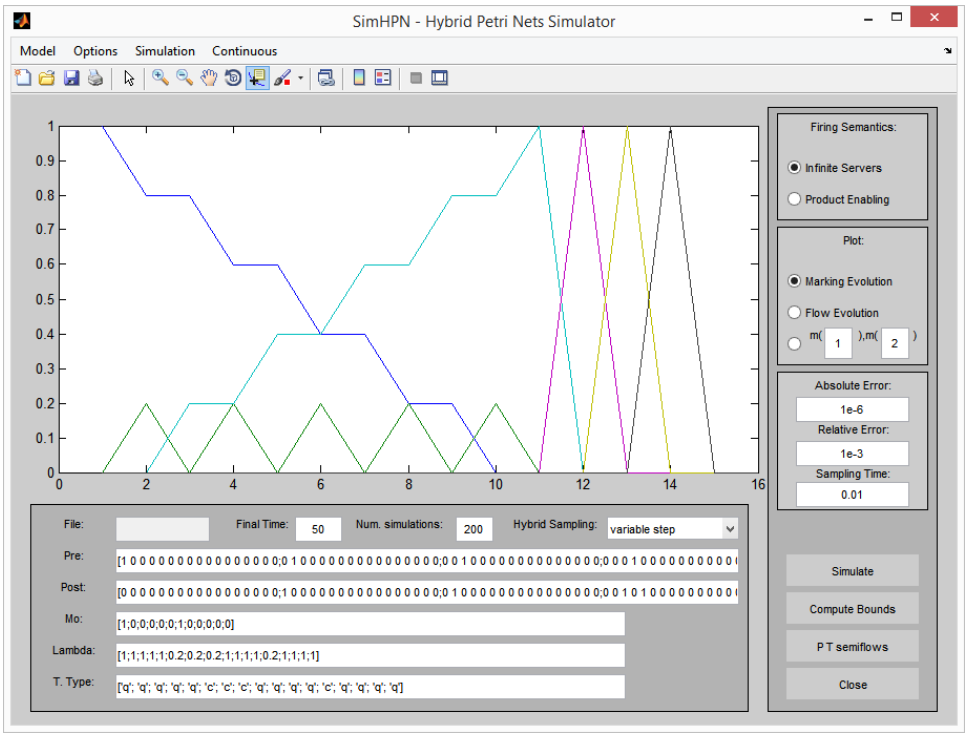


Fig. 4.4. Simulation of grid, battery and converter communication

t^d_{10}. When a token is in the converter (P^d_9), it has 3 possible options to move forward. Fig. 4.5 shows the case when P^d_9 fires t^d_{11}. This will enable P^d_9, therefore it can only fire t^d_{10} again. This time, if P^d_9 fires t^d_{14}, one token will reach to the Motor (P^d_{10}). From there, it has 2 possible firings where it can enables t^c_{12} or t^d_{15}. If t^c_{12} fires, then P^d_{10} will remain with 0.8 token and P^c_8 will gain 0.2 token since t^c_{12} has a firing rate of 0.2. Since P^c_8 has less then 1 token, it can't enable t^d_{10}, similarly the point P^d_{10} can't fire t^d_{13} or t^d_{15}, therefore it will only fire t^c_{12} till P^c_8 reaches to 1. From there P^c_8 can fire t^d_{10},t^d_{14},t^d_{15} to the transmission P^d_{11}. Finally, t^d_{16} fires, and the power can reach the wheels (P^d_{12}) through the driveshaft and splitter. Using SimHPN, simulation of the specific path of the user, can be captured, meaning it provided a visual representation.

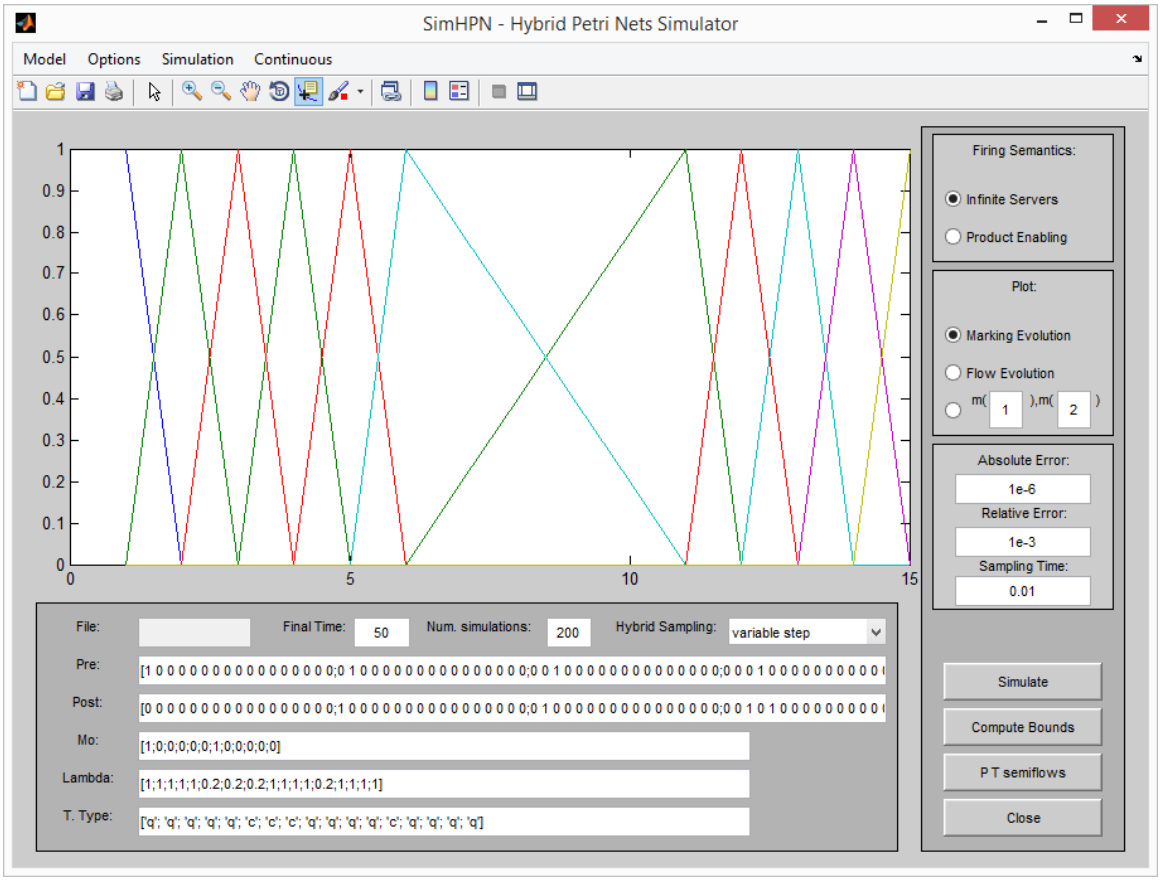


Fig. 4.5. Simulation of Vehicle Side

4.2 Matlab Results

SimHPN helped to provide one specific path of the reachability tree. In addition, some Matlab programming was done to analyze the reachability tree. This code provides several levels with many reached or duplicated nodes. Fig. 4.2.1 illustrates the 4 levels of the reachability tree for this system.

4.2.1 Algorithm

Creating a code for the Petri net, the algorithm concept called graph searching was used. This basically helps find a path from a state node to a target node, or

from all possible paths from a state node to a target node. There are two different graph search algorithms that can help find the reachable nodes. These algorithms are Breadth-first Search (BFS) and Depth-first Search (DFS) (See Fig. 4.6). In Section 4.1, SimHPN uses Depth-first search algorithm where it follows a path of nodes by leaving other nodes without discovering it.

In Fig. 4.2.1, BSF algorithm was used to find the reachable/duplicate nodes. This provides more detail of the tree, since it explores the tree level by level. It can provide the user with the shortest paths, in other words with the fastest way to reach a node form another node. Also, if Fig. 4.2.1 is analyzed, the ease of finding unreachable states from one particular node to another is very evident.

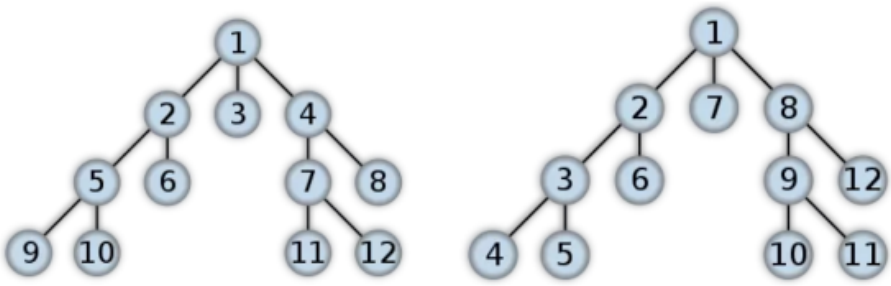


Fig. 4.6. Breadth First Search vs Depth First Search [21].

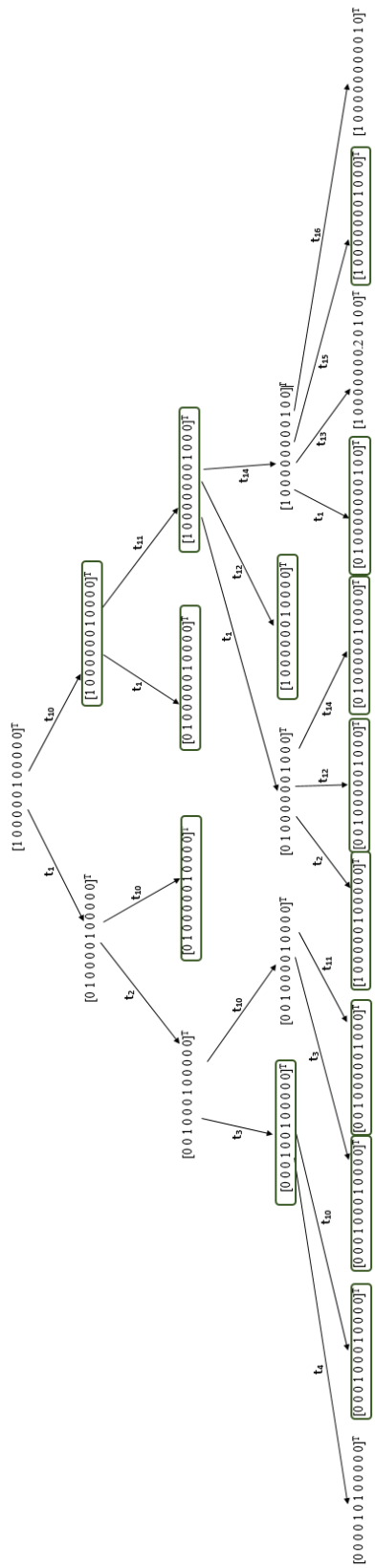


Fig. 4.7. Four Level Reachability tree for Vehicle to Grid Topology.

5. CONCLUSION

In this work vehicle to grid communication system was introduced with examples for discrete, continuous, and hybrid Petri nets. Modelings of these examples include vehicle to grid communication, followed by Petri net modeling of the grid-side, vehicle-side, and combined topologies. SimHPN simulator results, in addition to the developed Matlab algorithm proved the concept of this research work.

Vehicle to grid communication is a newly introduced topology that can be improved over the years. Petri nets are a great modeling tool that can capture the dynamic of the events. The V2G topology Discrete and Continuous Petri nets were approached and the classified places and transitions by their types were successfully described. It was then followed by combining both models that created a Hybrid Petri net model of V2G system. The design of the system was then simulated and proven mathematically through Matlab.

As for future work, it is possible to expand the hybrid Petri net model for the vehicle and grid side. In this thesis, only high levels were captured in the modeling of the Petri nets. This can be expanded to more complex ways by considering not only high levels, but also low level events. Another direction for future work may be considered is by using different types of Petri net models such as Timed Petri nets or Colored Petri nets. While in Timed Petri nets, the net has timed variables so that it can capture the specific dynamic events in real-time nature [22]. In Colored Petri nets, each token has it's unique color which represents a transition or a place [23]. This can help analyze inheritance of the system by observing the colored tokens' location. Last, the vehicle to grid communication topology can be built for plug in hybrid vehicles, where it would have more components for the vehicle side of the system.

REFERENCES

REFERENCES

[1] I. Husain, "Electric and Hybrid Vehicles Design Fundamentals," *CRC Press*, second edition, March 2003.

[2] J. Gallardo-Lozano, M.I. Milanes-Montero, M.A. Guerrero-Martinez, E. Romero-Cadaval, "Three-phase bidirectional battery charger for smart electric vehicles," *Compatibility and Power Electronics (CPE), 2011 7th International Conference-Workshop*, pp. 371 - 376, June 2011.

[3] The V2G Concept: A New Model for Power?. *University of Delaware*. [Online]. Available: http://www.udel.edu/V2G/docs/V2G-PUF-LetendKemp2002.pdf (Last day accessed: June 02, 2015.)

[4] M. Yilmaz, P.T. Krein, "Review of the Impact of Vehicle-to-Grid Technologies on Distribution Systems and Utility Interfaces," *IEEE Transactions on Power Electronics*, vol. 28, issue 12, pp. 5673 - 5689, December 2013.

[5] C. Guille, G. Gross, "A conceptual framework for the vehicle-to-grid (V2G) implementation," *Energy Policy*, vol. 37, pp. 4379 - 4390, 2009.

[6] Y. Du, S. Lukic, B. Jacobson, A. Huang, "Review of high power isolated bi-directional DC-DC converters for PHEV/EV DC charging infrastructure," *Energy Conversion Congress and Exposition (ECCE), IEEE*, pp. 553 - 560, September 2011.

[7] H. Chen, X. Wang, A. Khaligh, "A single stage integrated bidirectional AC/DC and DC/DC converter for plug-in hybrid electric vehicles," *Vehicle Power and Propulsion Conference (VPPC), IEEE*, pp. 1 - 6, September 2011.

[8] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol.77, issue 4, pp. 541 - 580 , Apr 1989.

[9] R. David, H. Alla, "Discrete, Continuous, and Hybrid Petri Nets," *Springer*, first edition, September 2005.

[10] R. David, H. Alla, "On Hybrid Petri Nets," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 11, pp.9 - 40, 2001.

[11] C. G. Cassandras, S. Lafortune, "Introduction to Discrete Event Systems," *Springer*, Second edition, 2008.

[12] R. David, "Modeling of Hybrid Systems using Continuous and Hybrid Petri Nets," *Proceedings of the Seventh International Workshop*, pp. 47-58, June 1997.

[13] S. Haddad, L. Recalde, M. Silva, "Continuous Petri Nets: Expressive Power and Decidability Issues," *Automated Technology for Verification and Analysis*, vol. 4762, pp. 362-377, 2007.

[14] H. Alla, "Continuous and Hybrid Petri Nets," *Journal of circuits, systems, and computers*, Vol: 8 1, 1998.

[15] A. Viswanath, A.M. Farid, "A hybrid dynamic system model for the assessment of transportation electrification," *American Control Conference (ACC)*, pp. 4617 - 4623, 2014.

[16] R. Letchmanan, J.T. Economou, A. Tsourdos, B.A. White, "Hybrid Petri Nets in Conceptual Schema of Dual-Motor Vehicle Synchronization," *Vehicle Power and Propulsion Conference (VPPC), IEEE* , pp. 1-6, 2006.

[17] P. Shamsi, "Applications of non-Markovian hybrid Petri-nets in power engineering," *Industrial Electronics Society, IECON 2014 - 40th Annual Conference of the IEEE*, pp. 84-89, 2014.

[18] Electric Transmission Lines. *Public Service Commission of Wisconsin*. [Online]. Available: https://psc.wi.gov/thelibrary/publications/electric/electric09.pdf (Last day accessed: June 02, 2015.)

[19] Pressurized Water Reactor Schematic. Digital image. [Online]. Available: http://www.nrc.gov/reading-rm/basic-ref/teachers/pwr-schematic.html (Last day accessed: June 02, 2015.)

[20] J. Julves, C. Mahulea, "SimHPN: A Matlab toolbox for Hybrid Petri Nets User Manual," vol. 1, January 2012.

[21] Breadth First Search vs. Depth First Search. Digital image. [Online]. Available: https://ucbcsnotes.wordpress.com/2013/05/08/breadth-first-search-vs-depth-first-search/ (Last day accessed: June 02, 2015.)

[22] J. Wang, "Petri Nets for Dynamic Event-Driven System Modeling," *Handbook of Dynamic System Modeling*, Ed: Paul Fishwick, CRC Press, 2007.

[23] I. Koh, F. DiCesare, "Synthesis rules for colored Petri nets and their applications to automated manufacturing systems," *Proceedings of the 1991 IEEE International Symposium on Intelligent Control*, pp. 152-157, 1991