INTERSECTION COLLISION AVOIDANCE FOR AUTONOMOUS VEHICLES USING PETRI NETS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Valli Sanghami Shankar Kumar

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Science Engineering

August 2019

Purdue University

Indianapolis, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF THESIS APPROVAL

Dr. Lingxi Li, Chair

Department of Electrical and Computer Engineering

Dr. Yaobin Chen

Department of Electrical and Computer Engineering

Dr. Brian King

Department of Electrical and Computer Engineering

Approved by:

Dr. Brian King

Head of the Graduate Program

This research is dedicated to my loved ones who believed in me and stood by me throughout all circumstances.

ACKNOWLEDGMENTS

I would like to dedicate this work to my parents Mr. K. Shankar Kumar (my best friend) and Mrs. M. Uma Soundari (my pillar of support) who believed in my dreams and made me fight for it. Thanks to Dr. Lingxi Li, my thesis advisor, without whom this research could not have been made possible. He imparted his knowledge and motivated me during every phase. I would also like to thank my committee members Dr. Brian King and Dr. Yaobin Chen for their support and time. I owe Sherrie Tucker for the timely reminders and document editing. Special thanks to all my friends for their suggestions and encouragement.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

SYMBOLS

d	distance
v	velocity
mph	miles per hour
m/s	meters per second

ABBREVIATIONS

CC	Cruise Control
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistant Systems
RADAR	Radio Detection And Ranging
ECU	Engine Control Unit
GPS	Global Positioning System
CAN	Control Area Network
ISO	International Organization for Standardization

ABSTRACT

Shankar Kumar,Valli Sanghami. M.S.E.C.E., Purdue University, August 2019. Intersection Collision Avoidance for Autonomous Vehicles Using Petri Nets. Major Professor: Lingxi Li.

Autonomous vehicles currently dominate the automobile field for their impact on humanity and society. Connected and Automated Vehicles (CAV's) are vehicles that use different communication technologies to communicate with other vehicles, infrastructure, the cloud, etc. With the information received from the sensors present, the vehicles analyze and take necessary steps for smooth, collision-free driving. This thesis talks about the cruise control system along with the intersection collision avoidance system based on Petri net models. It consists of two internal controllers for velocity and distance control, respectively, and three external ones for collision avoidance. Fault-tolerant redundant controllers are designed to keep these three controllers in-check. The model is built using a PN toolbox and tested for various scenarios. The model is also validated, and its distinct properties are analyzed.

1. INTRODUCTION

Transportation is one of the fields that has benefited from technological advancements. With inventions in both hardware and software, driving has become less risky and more comfortable. With the advancement of sensing, communication, and control technologies, the number of deaths has dropped in recent years.

Some important safety features are Intersection Collision Avoidance and Cruise Control. The cruise control system helps in maintaining a required safe distance from the vehicle driving in the front, hence tailgating is avoided. Though the cruise control system had been implemented in 1991 by Mitsubishi[19], it was just the beginning with only a warning. The most recent advancements not only notify the driver but also take control and avoid such scenarios by changing the speed via the throttle mechanism or emergency braking systems. Another major obstacle in driving is intersection collisions. Nearly one-third of accidents occur due to intersections in rural and urban areas[27]. With V2X communications and with the information collected from LIDAR (Light Detection and Ranging), RADAR, other sensors, and real-time image detection, the intersection collision avoidance (ICA) system in the vehicle can foresee the collisions and take necessary steps to avoid it via speed control. These changes reduce risk and also improve driving experiences.

1.1 Literature Review

In[1], the authors proposed the priority-based techniques to prevent deadlock situations in intersections. Usually, when talking about the intersection collisions, the vehicle which comes first to the intersection is given a higher priority, and other vehicles wait until this vehicle crosses the intersection. But in the cases where several vehicles arrive at the intersection at the same time and have conflicting signal phases, the problem arises on which vehicle should be given a higher priority. So, in those cases, the authors suggested other priority techniques.

The authors of [2] talked about the vehicle dynamics which get input from systems like the throttle mechanism and the braking system. Depending on the input collected from the throttle and braking mechanisms, the system generates wheel torque. The vehicle model takes throttle and brake inputs and produces longitudinal displacement as output. This system is a cascade of power-train and vehicle models.

The intelligent Intersection Collision Avoidance System was introduced by the authors in[5]. This modular system is flexible and consists of a few intelligent nodes that are embedded inside the desired lane at the mid-point. And there is also a base station that analyzes the data obtained from the nodes. As the vehicles pass, the nodes embedded inside the lane record the time-stamp and vehicle information. This data is sent to the base station which analyzes it and determines if a collision is imminent. If so, then a warning is given to the vehicles at risk.

The authors of[8] mentioned the advantages of Petri nets in manufacturing cells with multiple robots. Here, colored Petri nets are used in modeling and analyzing the behavior of the cells, then the model is validated and simulated in 3D graphic simulation.

The disadvantages of the Petri net, in general, are mentioned by the authors in[9]. The disadvantages mentioned are the construction of Petri nets for a large system and the unavailability of inexpensive software explicitly for the Petri net usage. Also, the reuse of Petri nets is restricted. These are some reasons mentioned for the confined usage of Petri nets in academic institutions and research fields.

The authors of[10] introduced the concepts of the Electronic Stability Program (ESP) and Electromechanical Parking Brake (EPB). The ESP is used for initiating the braking operations by building up pressure by actuating the valves. Then, in the stationary state, the cruise control is deactivated, and the EPB is activated. When the cruise control is activated, the EPB is automatically deactivated.

The Petri net controller design was introduced in[12]. For designing a controller using place invariants, the constraints should be represented in linear inequalities that consist of elements of the Petri net marking vector. If the constraints are written with respect to a firing vector, then the controller will have maximal permissiveness only if the net is safe. But if the constraints are represented in terms of a marking vector, then the maximal permissiveness is guaranteed.

The authors of[16] introduced the first-order hybrid Petri nets and its dynamics. The hybrid net consists of the dynamics that are both time-driven and event-driven. The macro-events that occur in the hybrid Petri nets are those events that occur when a discrete transition is fired or when a continuous transition is either disabled or enabled. Different types of frames in the Controller Area Network were mentioned in[15]. The data frame consists of Arbitration Field, the Data Field, the CRC Field, and the Acknowledgment Field. Another type of frame is called a remote frame which is similar to the data frame but without data. The error frame is a special message that violates the formatting rules of a CAN message. If the message gets delayed, then the overload frame is used: it represents completeness.

The history of Adaptive Cruise Control was mentioned in[19]. Around 1995, the adaptive cruise control system was implemented. Rather than giving a simple warning, it was able to take control and accelerate accordingly, but it was not able to brake when the vehicle traveling in front of the host vehicle slowed down or stopped. Later, the sensors were used, and after the iterative advancements, the cruise control system has become full-fledged.

1.2 Contributions

The contributions of this thesis are summarized as follows:

- Intersection Collision Avoidance System and Cruise Control system are designed.
- Controllers are designed to make sure the intersection collision is avoided.

- Fault Tolerance techniques are implemented using redundant controllers to keep the other controllers in check.
- Both Cruise Control, as well as Intersection Collision Avoidance systems, are verified for different scenarios.
- Properties of the proposed Petri net model are analyzed.

1.3 Organization

This thesis proposes the modeling of Intersection Collision Avoidance and Cruise Control System using Petri nets and analyzes the fault tolerance techniques and the behavior of Petri nets. Chapter 2 deals with Petri net introduction, its types, properties, advantages, and disadvantages. Chapter 3 introduces different types of communications, in-vehicle, vehicle to vehicle, vehicle to pedestrian, and vehicle to infrastructure. Chapter 4 introduces the design of models on intersection collision avoidance and cruise control and the model design. Chapter 5 analyzes the properties of the Petri net model. Chapter 6 concludes and presents future work.

2. INTRODUCTION TO PETRI NET MODELS

The concept of Petri nets was first proposed by Carl Adams Petri in the year 1962. The Petri net model consists of four elements: places, transitions, tokens, and weights. Petri nets are nothing but a marked graph. It is also called a place/transition net(P/T Nets). It has a rigorous mathematical definition that makes them representable. Petri nets also provide a graphical way of modeling step-wise processes. The properties include liveness, boundedness, reachability, coverability, fairness, place invariant, transition invariant, and so on, which helps in the analysis of the system. Among the four elements, the places are represented by a circle that indicates the achievable states of the system. The transitions which are indicated by bars indicate the events or actions that change the state of the system. The arcs connect the places to transitions and vice-versa. The tokens are represented by numerical values that may include the state, signal, condition, or a physical object. The arcs consist of weights, normally the weight of an arc is 1. If the weights are different from 1, it is mentioned explicitly. A Petri net is a 5-tuple, Petri net = (P, T, F, W, MO) where,

- P consists of P1, P2.... -finite places.
- T consists of T1, T2.... - finite transitions.
- F is a subset of (P x T) U (T x P)- set of arcs (flow relations).
- W: weight function.
- M0: initial marking.
- P intersection T is a null matrix and P union T is not equal to a null matrix.

A Petri net structure N = (P, T, F, W) without any specific initial marking is denoted by N. A Petri net with the given initial marking is denoted by (N, MO), where MO denotes the initial marking.

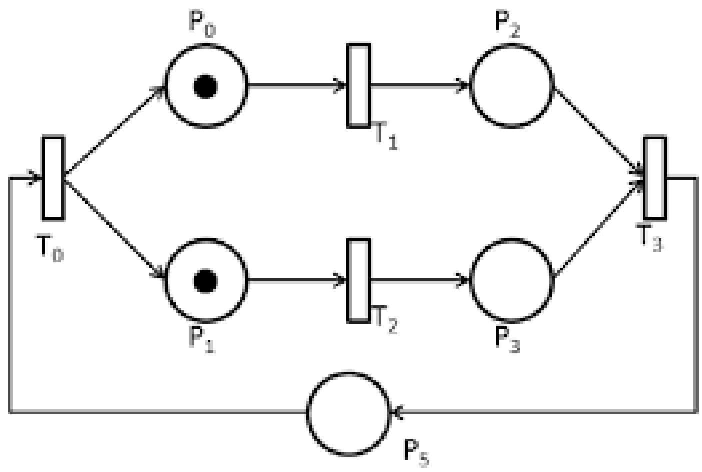

Fig. 2.1. Petri Net Example

In Figure 2.1[24], it can be seen that places are marked by P0, P1...P5. The transitions are marked by T0, T1, T2, T3. We can also see that there are tokens in P0, P1. So, the initial marking can be assumed as M0=[1 1 0 0 0]. The Petri net should have a proper link between places and transitions for the system to work properly. Connections cannot be made between a place to place or a transition to transition. Hence a place is connected to a transition and vice-versa.

2.1 Enabling Transitions

By knowing the states and the way they change, the behavior or the flow of the system could be easily determined. Consider w(P, T)=x: it indicates that the number of tokens is x. A transition T is enabled, only if the place P has a minimum of x tokens. If the number of tokens in place P is less than the arc weight, in this case, x, then the transition is not enabled. And also, the firing of transition not only depends on the arc weights and tokens but also on the fact that conditions, if any, are satisfied or if the event is happening. Also, if the transition is enabled, it takes the tokens from the input place which are equal to arc weight number and gives away tokens to the output places, the number of tokens being the same as the arc weight numbers.

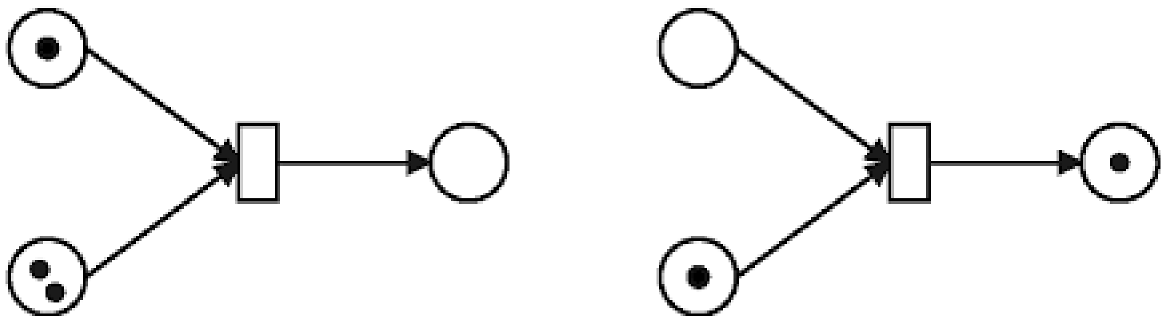

Fig. 2.2. Transition Firing

From Figure 2.2[28], it is seen that there are two tokens in one input place and one token in another input place. Since the weights are not mentioned in the arcs, it is assumed to be 1. Now both the places have a minimum number of tokens for the transition to fire. So, the transition fires and gives one token to the output place, since it also has the arc weight one.

Figure 2.3[4] shows the Petri nets with different arc weights other than the default number 1: the input places are P1 and P2, and the output places are P3, P4, P5. The arc weight of P1 is mentioned as 2. It means that that place should have a minimum of two tokens to make the transition T1 get enabled. So, only when P1 has a minimum of 2 tokens and P2 has 1 token, the transition T1 is fired. Once the transition is fired, the tokens are given to the output places of that transition. It can be seen that the arc weights of output places P3, P4 and P5 are 1, 2, 3 respectively. Hence, when the transition is fired, it gives 1, 2 and 3 tokens to the places P3, P4, and P5 respectively. Hence, the output would look like Figure 2.4[4]. After firing T1, the tokens in the input places will change to 0, so the transition T1 is no longer enabled.

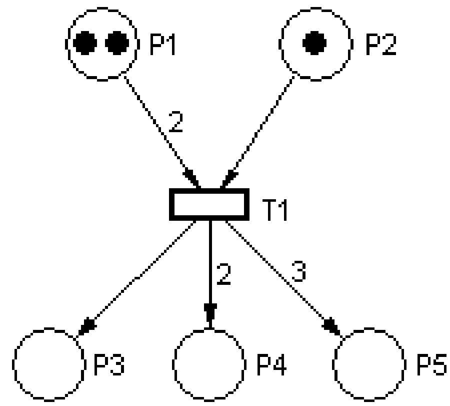

Fig. 2.3. Before Transition Firing

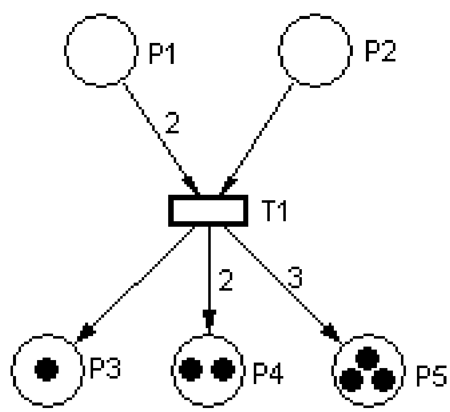

Fig. 2.4. After Transition Firing

When a Petri net has a place that can accommodate an infinite number of tokens, then the Petri net is termed to be Infinite net. If there is a certain limit to the number of tokens it can accommodate, then it is known as a finite net. A transition without any input place is called a source transition, and one without any output place is called a sink transition. This means that the source transitions are enabled indefinitely since the transition does not require any conditions to be satisfied to get enabled. While this is the case for a source transition, the sink transition gets all tokens but does not have any chance to give away those tokens. If a place acts as both input and output to a transition, then the Petri net is supposed to be containing a self-loop. Similarly, if a Petri net does not have any self-loops, then it is called as pure. A Petri net that has all arc-weights as 1 is called an ordinary Petri net.

2.2 Marking

Each place contains an integer (positive number or zero) number of tokens or marks. The number of tokens contained in a place P can be defined by the vector of these markings, i.e., m transpose = (m1, m2, m3, m4, m5). The initial marking of the Petri net in Figure 2.3[4] is thus,

$$m^T = (2, 1, 0, 0, 0)$$

The marking defines the state of the Petri net, or more precisely, the state of the system. The evolution of the state thus corresponds to an evolution of the marking which is caused by the firing of transitions.

2.3 Working of Petri Net

Figure 2.5[22] illustrates the working of a Petri net that is used in daily life. The example taken here is a vending machine. It can be noted that the Petri net looks similar to a state diagram. From the figure, it is understood that the very first transition is fired only when an order arrives. As the transition is fired, it enters the place which contributes to a waiting period after which the next processing work is done. This transition has two outputs. One of them reaches the completion of the order stage, and the other enters the waiting phase before entering the main processing path.

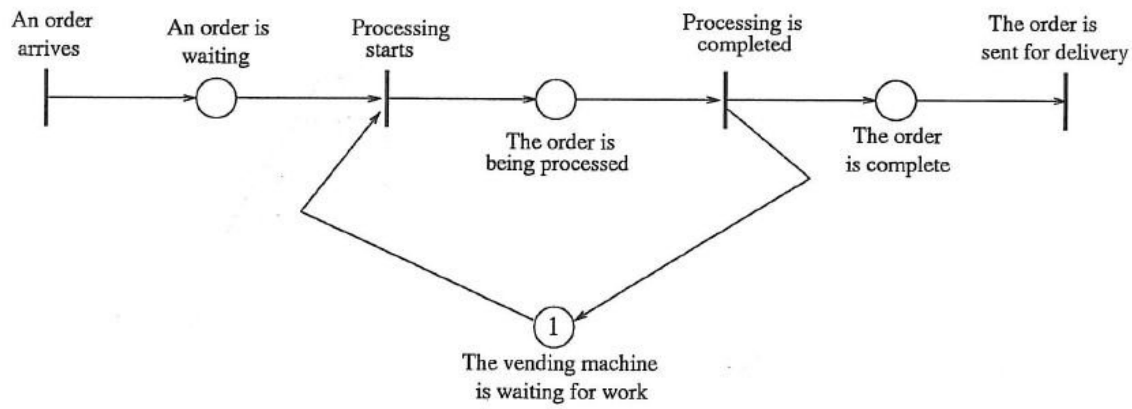

Fig. 2.5. Vending Machine Example

In short, once the order arrives, it enters the waiting stage probably to request resources. Then the processing takes place. Once, the processing is completed, then the order is sent for delivery.

2.4 Properties

The Petri net helps us in analyzing the system and having in-depth knowledge about the system. To discern the model, we need to inspect it by looking into their properties which will help us to apprehend their working, what to expect, and what not to expect from the model and also helps to improve its working. Some of the important properties of Petri net are boundedness, reachability, reversibility, conservation, etc.

2.4.1 Reachability

The reachability tree is one of the important properties of Petri nets. It gives us better knowledge about the states that the model can traverse. For example, let us consider an initial marking m0 and another marking m1. The reachability tree determines if the marking m1 is attainable from the initial marking m0 or not. If it is attainable, the marking m1 is termed to be one of the reachable states. Since it entirely depends on the initial marking, reachability comes under the behavioral property. So, the marking m1 is obtained by firing a transition from initial marking m0.

Figure 2.6[21] is an example of the reachability tree. The first marking [2 0 0] indicates that it is the initial marking. The places are represented by an oval shape. From the initial marking, the new marking [1 1 0] is obtained when the transition t1 is fired. Now as the new marking is obtained, it is possible for transitions t3, t1 and t2 to get enabled from that point. So, when the transition t3 is fired, the marking [1 1 0] becomes [1 0 0]. If transition t1 and t2 fires, the new markings would be [0 2 0] and [1 0 1] respectively.

It is also possible for the reachability tree to include infinite states. Hence, to curtail the tree, the omega factor was included, since an infinite tree leads to complexity. The new tree or graph with omega is called pseudo-infinity, which maintains the tree at a finite state. So, the tree without omega factor is called a reachability tree, but the one with the omega factor is termed as coverability tree. When proceeding with the tree, if no other marking is possible from a certain state, then it is termed as the dead-end. It is understood that the new markings are called Frontier Nodes. If there is a dead-end to a marking, then it is termed as Terminal Node. If a previous marking is obtained, then they are called as Duplicate Nodes.

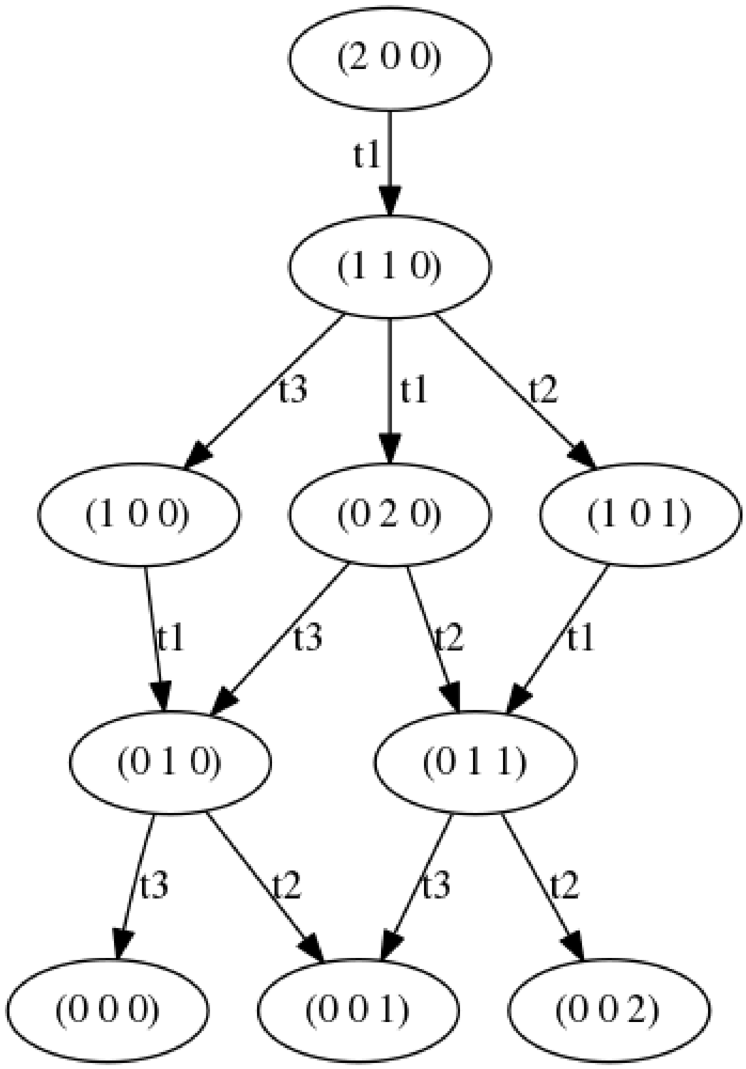

Fig. 2.6. Reachability Graph

2.4.2 Boundedness

A Petri net is called k-bounded if each place gets at most k tokens for all the reachability states with reference to the initial marking m0; here k is an integer.

If k=1, then the net is termed as safe. Safeness is a special attribute to the boundedness property. In a safe net, the maximum number of tokens that any place in a reachability tree can have is 1. So, all places have either zero or one token. If the Petri net is not bounded, then it means that the reachability tree will grow indefinitely. So, the reachability or coverability property is interrelated to the boundedness property.

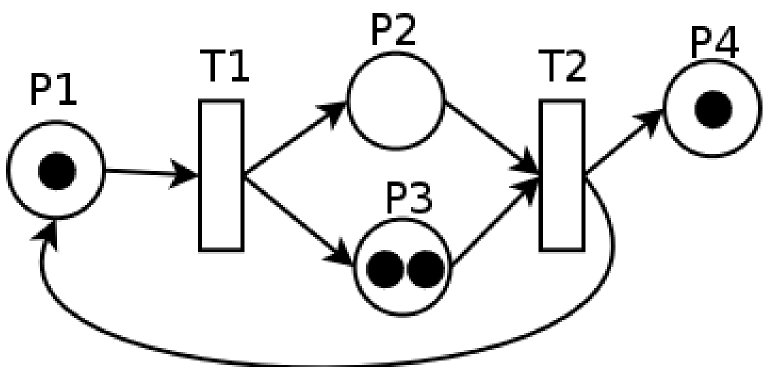

Fig. 2.7. Example to Illustrate Boundedness

It is seen from Figure 2.7[20], there are four places P1, P2, P3, P4, and two transitions T1 and T2. It is noted that the place P4 does not show boundedness. As long as the transition T2 fires, the tokens in P4 will increase. It does not have any way to give away those tokens, so it is unbounded.

2.4.3 Liveness

Liveness is another interesting property of the Petri nets with regards to the initial marking m0. A Petri net is called live if, for every marking that belongs to the reachability set, it is possible to fire all the transitions in the net at least once by some firing sequence. There are different levels of liveness from level 0 to level 4, for the better interpretation of this property. Each levels are explained as follows:

- Level 0: The transition is never fired or in other ways, it means that it is dead.
- Level 1: The transition is fired at-least once. It is termed as Live 1.
- Level 2: If the firing transition t can be fired for a finite integral number of times, then it is called Live 2 Liveness.
- Level 3: The transition t is fired indefinite times at-least once.
- Level 4: It may always fire, i.e. it is L1-live in every reachable marking in R(N, M0).

2.4.4 Reversibility

The reversibility property of the Petri net is also called as Home State. It means that the Net is Reversible if the initial marking is achieved by all the reachable markings. Mathematically, a Petri net is called Reversible or Proper if m0 is reachable from any m. The Net is also termed as Reversible if a marking covered by the initial marking is obtained. In real-time applications, most of the systems can be reached back to their original state or to the state that is close to the initial state unless there has been some energy loss in between the process. So a Petri net is called reversible if every node in the coverability tree contains m0, the initial marking. If it contains some other nodes along with the initial marking, then its Partially Reversible.

2.5 Petri Net Modelling Constructs

2.5.1 Sequential

In the sequential execution of the Petri net model, the transitions are fired in order i.e the transition t2 is fired only after transition t1 is fired. Thus, the firing takes place in a sequence.

2.5.2 Synchronization

Here, in synchronization, the transition is fired if more than one place acts as an input to a single transition. Even if one input place does not satisfy the required condition, then the firing of transition does not take place.

2.5.3 Conflict

In this type of scenario, one single place acts as an input to more than one transition. Hence, a conflict occurs to determine which transition to be fired first, if the place has a sufficient number of tokens. It is solved using a non-deterministic way of an approach by assigning probabilities to the transitions. The sum of probabilities should add up to 1. The transition with higher probability will be fired first, and the flow of the model proceeds in that direction.

2.5.4 Priority and Concurrency

In priority type, there is an inhibit arc apart from the normal one. So, if a place is having an inhibit arc, then the transition connected to that place will be fired if there is no token in that place. In concurrency, more than one place acts as an output to a transition, and each place paves a way to its path. Figure 2.8[23] shows the different structures of Petri net models.

2.6 Incidence Matrix

Let the Petri net PN= (P, T, F, W, MO), then the incident matrix will consist of A: P x T, where P represents the places in the Petri net and T indicates the transitions. They are the rows and columns of a matrix respectively.

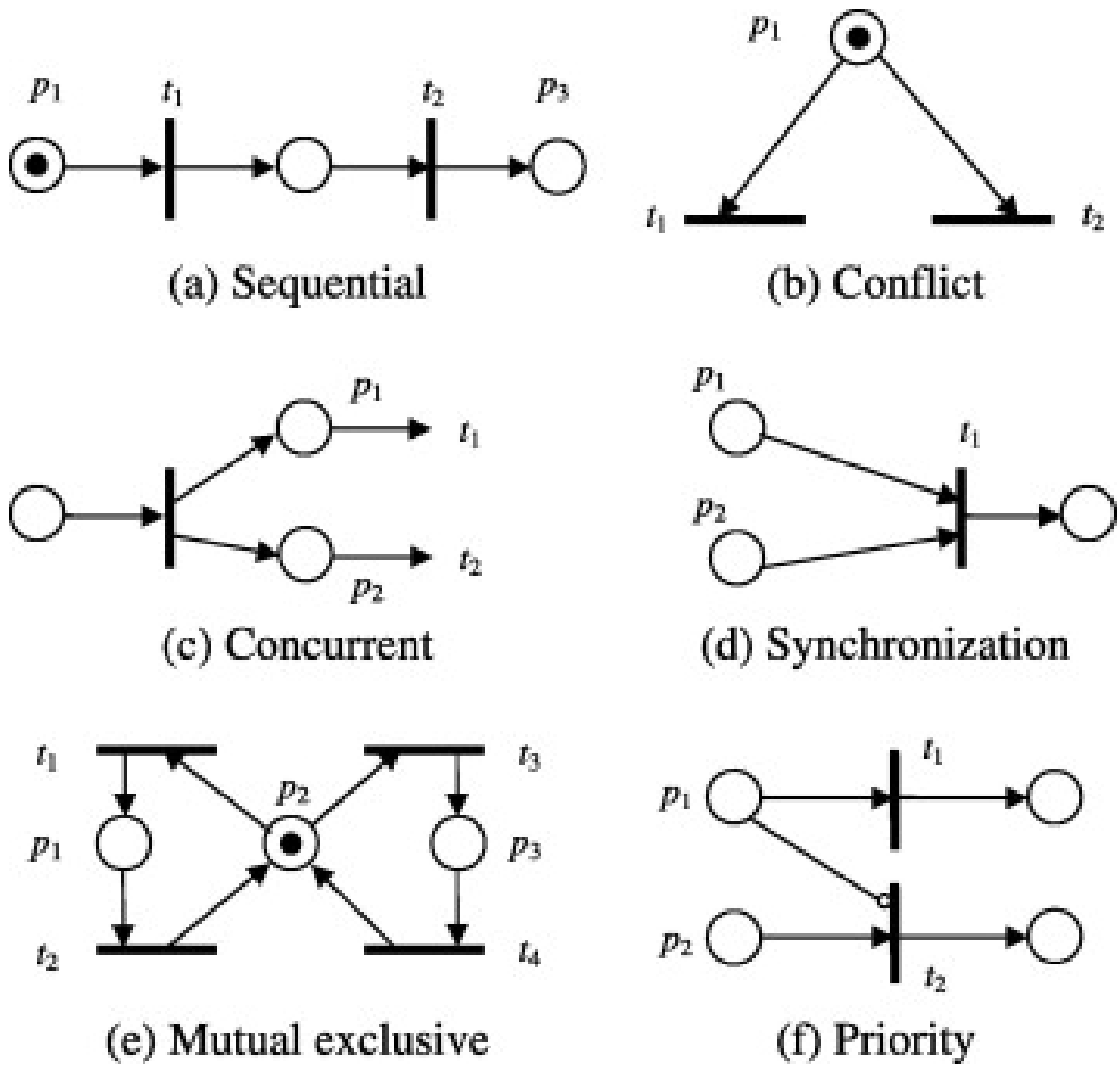

Fig. 2.8. Petri Net Structures

For example: Consider the incident matrix A to be:

$$\begin{vmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \end{vmatrix}$$

Let us assume the initial marking to be m0:

$$\begin{vmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

And the firing vector to be U=

$$\begin{vmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

then the next reachable state with given matrices is calculated using the formula: m1= m0 + A * U. This formula is time efficient in finding the next marking. Here, m1, the next marking would be:

$$\begin{vmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

The above-given equation of finding the next marking m1 helps in determining the reachable states for the complex systems by simply knowing the initial marking and sequence of firing transitions. The incident matrix B is derived from the output matrix and input matrix both having a dimension of P x T where P denotes the places which are represented by rows and T denotes the transitions in the Petri net model which is columns. Considering,

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

$$B^+ = \begin{vmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 & 0 \end{vmatrix}$$

The incident matrix is calculated by the formula

$$B = B^+ - B^-$$

so, B=

$$\begin{vmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

2.7 Autonomous Continuous Petri Nets

Continuous Petri nets are a relaxation from discrete Petri nets. Continuous Petri nets allow users to create a model with a real number of firings as well as tokens. Figure 2.9[13] represents the representation of both continuous and hybrid Petri nets. The inhibitory arc works in such a way that the transition does not fire until the place has tokens.

It is used in the applications where a large number of tokens are used which causes a large number of reachable markings, and it is possible to convert a discrete Petri net to continuous and vice-versa.

Figure 2.10[26] is an example of a continuous Petri net. S1, S2, S3 are places that are represented by thick circles or most of the time by two circles. The r1, r2, r3, and

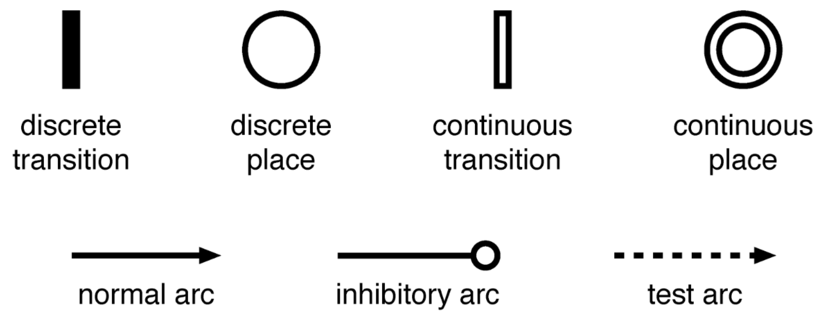

Fig. 2.9. Continuous and Hybrid Petri Net Representation

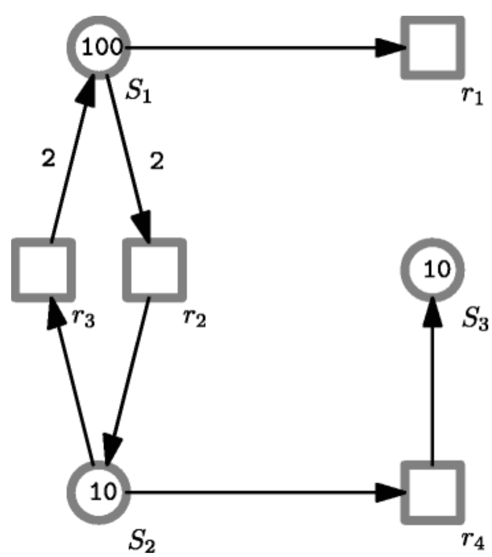

Fig. 2.10. Continuous Petri Net

r4 represents transitions which are also represented by thickening the outline to show the difference between a continuous and discrete Petri net. The number of reachable macro-markings of an n-place continuous Petri net is less than or equal to 2 to the power n.

2.8 Autonomous Hybrid Petri Net

Hybrid Petri net consist of both discrete and continuous Petri nets. To be exact, a hybrid Petri net consists of discrete places and discrete transitions, as well as, continuous places and continuous transitions.

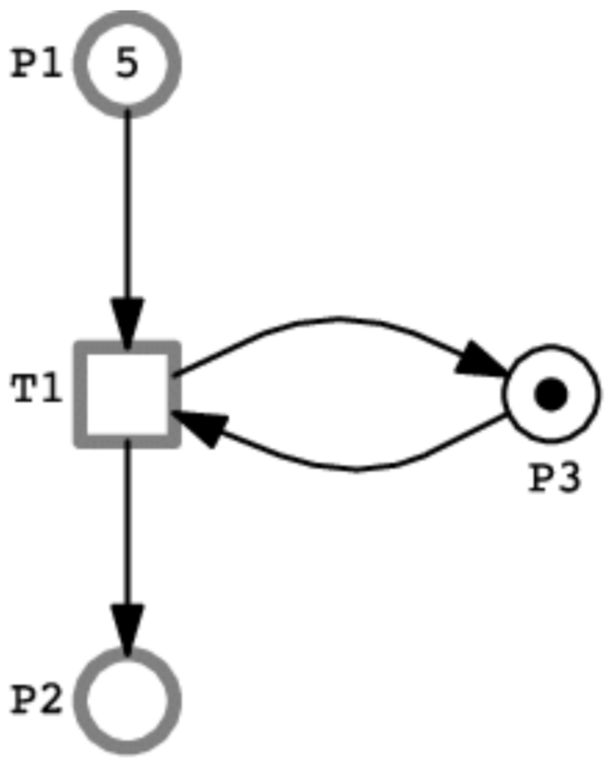

Fig. 2.11. Example of Hybrid Petri Net

Figure 2.11[25] illustrates the hybrid Petri net. The place P3, which is indicated by a single circle is discrete while the other two places which are represented by the thickening of circles are continuous. Similarly, T1 is a continuous transition since it is shaded. The union of both discrete and continuous Petri nets is termed as hybrid Petri net. Another type of hybrid net is a hybrid dynamic net (HDN). It is similar to HPN, but the firing rule for continuous transitions is different here. In the hybrid dynamic net, the outgoing, and incoming arcs of continuous transition should have the same arc weights.

2.9 Applications

Petri nets are easy to express due to their diagrammatic representation; it is similar to state diagrams. Petri nets are used to model the systems in industries. But as industries bloomed, and technologies improved, the simple Petri nets became fruitless in certain aspects, like impossible to trace the messages sent via a communication system. This drove the researchers to build the Petri nets with the basic net structure but differs from the base net by some additional rules and few relaxations. So, different Petri nets are developed and used for specific purposes.

2.9.1 Fuzzy Petri Net

Fuzzy Petri nets are used in the areas where precise control is expected and when there is a need to transfer specific uncertain data[8].

2.9.2 Temporal Petri Net

Temporal Petri nets come into play when the simple Petri nets are not able to deal with the properties like the eventuality, which expects certain places to have tokens and certain transitions to fire eventually. Its applications are in alternating bit protocol, modeling, and analyzing a handshake daisy chain arbiter, etc.

2.9.3 Stochastic Petri Net

Stochastic Petri nets play an important role in modeling, controlling, and analyzing the manufacturing systems. It is also used to analyze a system with regards to the dead-lock. Deadlock is a situation that arises when a process goes into waiting state when the resources it requested is held by another waiting state, which in turn, is waiting for another resource. So, the state of the process remains unchanged indefinitely causing the system to enter into a dead-lock.

The simulation of the Petri net helps in a better understanding of the model. The visual output helps in better perception. Mostly, Petri nets, in general, are used in areas like:

- Business Process Modelling.
- Artificial Intelligence.
- Data Analysis.
- Reliability Engineering.
- Communication networks like Expressnet, Fastnet, D-Net, U-Net, Token Ring, Field buses, such as FIP and ISA-SP50.

3. PROPOSED PETRI NET MODEL FOR INTERSECTION COLLISION AVOIDANCE

3.1 Introduction

Autonomous cars are considered to be influential in the future world. All major companies are invested in the research of autonomous cars. When autonomous cars come into the market, it will not only help in smoothing the traffic flow but will also ensure the accidents due to human error are drastically reduced. The driving experience for the customers will also be manifold.

Autonomous vehicles are vehicles that drive without human guidance. Artificial Intelligence plays a vital role. In short, the computer will get the information needed, will process it, and will decide on the best outcome. Its impact on the future will be astounding, particularly in areas like emergency help, parking, military, and also in making travel easy for the differently-abled persons. And this is one of the reasons for the automakers to jump into the race of building autonomous cars. Though few autonomous cars are used for test-drive purposes, they are not commercially available yet.

But there are challenges to overcome. Since Artificial Intelligence plays a determining role, the programming part becomes tedious. A small mistake may lead to accidents. Other than this, hacking of the vehicle system proves to be a major setback, since it may lead to human fatalities. And also, operating the vehicles during adverse weather conditions is another setback, as the road signs and lines may become obscure.

3.2 Cruise Control System

According to various statistics, at least one person dies in a crash for every second, and one major reason behind this is due to the distraction of human drivers. When traveling long distances, it is required for the drivers to have a complete concentration throughout the journey, which is a great challenge. Hence, the cruise control system is developed to ensure that the computer takes control so that the human drivers can relax, and let the vehicle take control. Adaptive Cruise Control is a feature that allows the cruise control system to take control and manage the desired speed to help maintain smooth traffic flow. It gets the information of other vehicles in the path of the host vehicle. If the vehicle traveling in front of the host vehicle suddenly decreases the speed, the cruise control system comes into play and reduces the speed of the host vehicle to prevent accidents. Thus, a cruise control system in the vehicle ensures that the vehicle is accelerated or decelerated to maintain a safe distance from the target vehicle without the intervention from the humans.

3.2.1 Design

Figure 3.1 indicates the Cruise Control System model. In this model, the velocity, as well as distance controllers, are designed in such a way that the controllers are internal. Desired distance and velocity are obtained by the way the model is designed. The place p1 is the start point of the model. If there is some fault in obtaining the information, then the model proceeds to a stop. If all goes well, the model enters the next stage of the velocity controller.

Both p7 and p6 are the places that store the velocity of the host vehicle and the target vehicle. The place ”display” indicates the velocity obtained after velocity controller acts. If the velocity of the target vehicle is less than the host vehicle, then the velocity of the host vehicle is reduced immediately. With the obtained velocity, the place p12 stores the information of the distance at which the host vehicle should follow the target vehicle i.e safe distance.

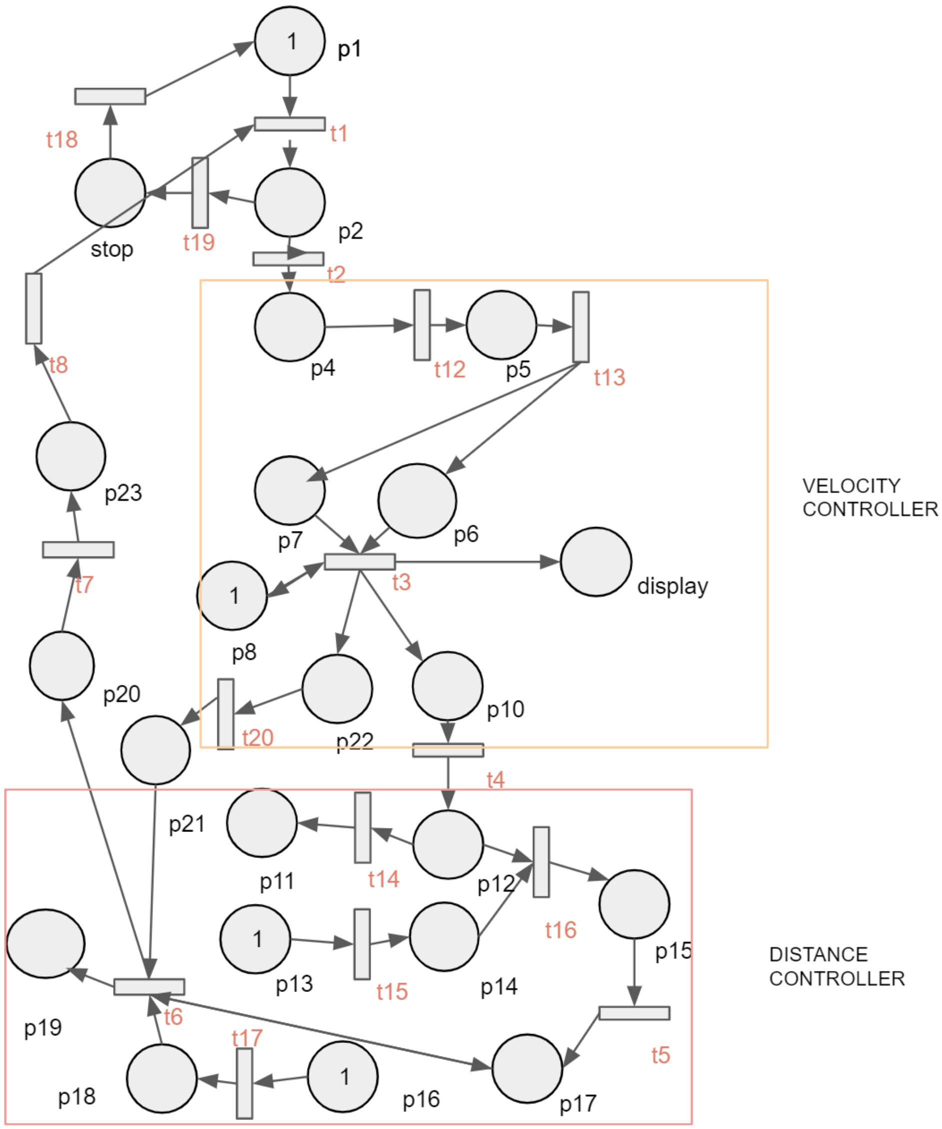

Fig. 3.1. Cruise Control System

In normal cases, a safe distance would be traveling 2 seconds behind the target vehicle. During adverse weather conditions, it would be 4 seconds. Taking the difference in the safe distance that is to be maintained and currently maintained by the vehicle into account, the velocity is reduced again, which is indicated by p19. Thus, p19 is re-changed velocity. If the distance controller is not necessary, then the model enters p11. This maintains the final velocity, which is the velocity of the host vehicle before entering the distance controller. When a conflict arises, then the transition with a higher priority is fired.

3.3 Intersection Collision Avoidance

A collision avoidance system, also known as a pre-crash system, forward-collision warning system, or collision mitigating system, is an automobile safety system designed to prevent or reduce the severity of a collision. It uses radar and sometimes LIDAR and camera to detect an imminent crash. Intersection Collision Avoidance prevents the collision at the intersection by taking control to reduce fatality rates and property damage. According to the National Highway Traffic safety Administration(NHTSA), intersection collision is one of the top causes of accidents taking place which is around 40 percent of vehicle collision i.e around 2 million crashes occur at the intersections. The crashes at intersections occur mostly due to:

- Insufficient surveillance of the surrounding area (44.1 percent).
- Wrong assumptions about how other drivers would react (8.4 percent).
- Obstructed view when turning (7.8 percent).
- Misjudgment about the other drivers speed or the gap between vehicles (5.5 percent).
- Distracted driving (5.7 percent).

Accidents at intersections can occur easily even if there is a slight distraction or uncontrolled speed. Some normally occurring crashes are due to rear-end collision, which occurs if a vehicle is closely following other vehicles. The percentage of such crashes is higher than for the same scenario in a straight road. The other normal scenario is turning without noticing a pedestrian or cyclist in the path. And there are side-impact crashes, which occurs when one driver runs a red signal and T-bones another car. Certain definitions involved in the communication between Vehicle to Vehicles, Network, or Pedestrian are:

Vehicle ID: Each vehicle has a unique identification number.

Current Road Segment: Identifies the current road that the vehicle is using to get to the intersection.

Arrival-Time: The time at which the vehicle gets to the intersection.

Exit-Time: The time at which the vehicle will exit the intersection.

Current Lane: Identifies the lane being used.

Using C-V2X communication all the details for the vehicles are collected and shared, so any vehicle is aware of its surroundings and in a position to determine future actions.

3.3.1 Collision Detection

Based on the location of the vehicle, moving direction, and velocity obtained, the system predicts the probability of a potential collision and warns the driver accordingly. An intersection collision avoidance system can be designed to manage the intersection traffic and to detect the collision. If a collision is detected, then the controllers will act to reduce the speed of the host vehicle to avoid a collision. There are certain conditions to be satisfied for a collision to occur in an intersection. They are as follows:

- The vehicles traveling in the same intersection or one vehicle turning left.
- Time conflict.
- Space conflict.

Figure 3.2[1] and 3.3[1] indicates the scenarios that cause space conflict and scenarios that do not cause conflict. The green and blue lines indicate the path of two vehicles, and the red color indicates the chances of overlapping paths of both vehicles. The black lines indicate the intersection.

Fig. 3.2. Scenarios without space conflict

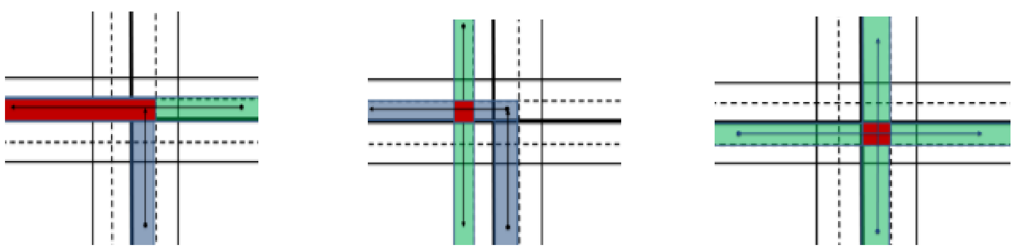

Fig. 3.3. Scenarios with space conflict

The intersection point is considered to be a square that is present at the heart of the diagram. So, if two vehicles are assumed to be traveling at the same intersection with the same space, same entry time as well as exit time, then the collision becomes imminent. So, there should be no two vehicles entering and exiting the intersection at the same time. If such a scenario exists, then the velocity of the vehicle with a lower priority is reduced.

In Figure 3.4, the Intersection Collision Avoidance System along with the cruise control system is shown. It can be noted that the system is without the controllers.

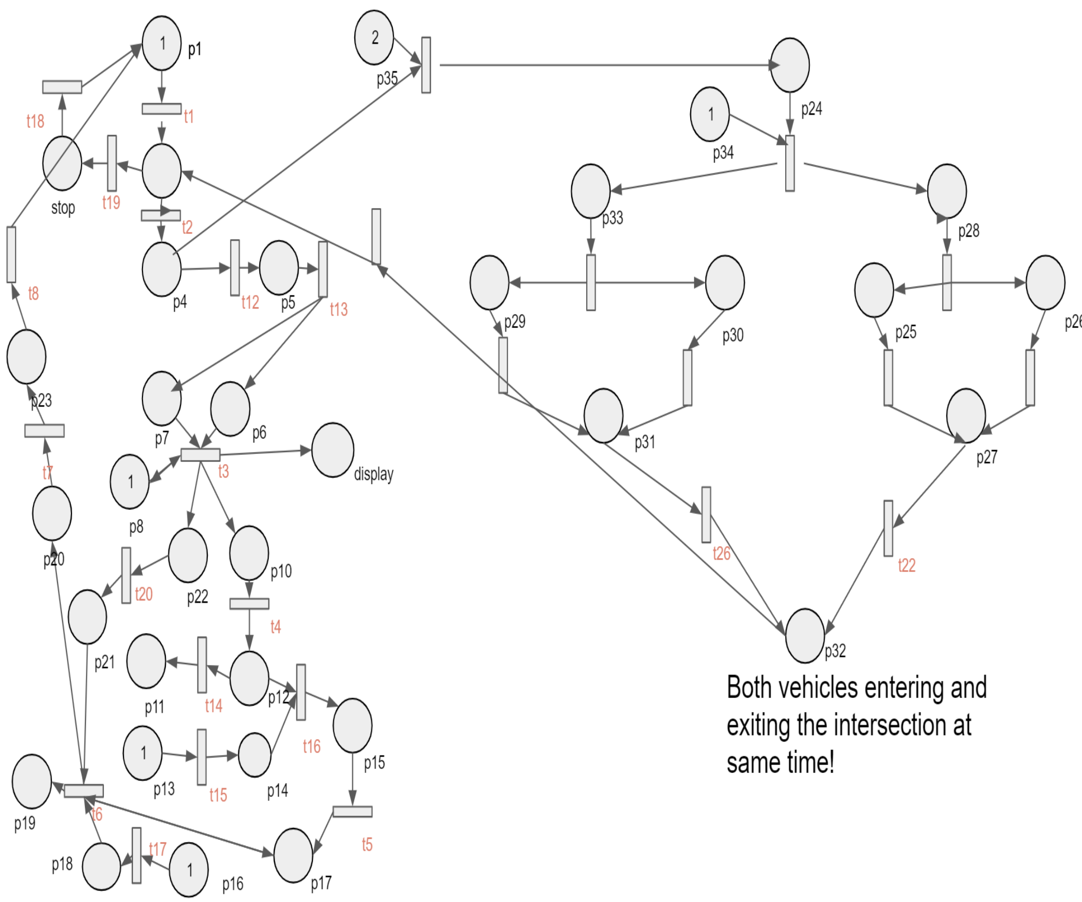

Fig. 3.4. Intersection Collision Avoidance System

The place p35 stores information about both vehicles obtained via V2X communication. The V2X communication sends data using a 5.9 GHz frequency band or using the mobile network. The in-vehicle information is obtained using the Controller Area Network (CAN). In the intersection collision avoidance system without controllers, places p31 and p27 can contain two tokens. These places represent the entry as well as the exit place in an intersection. So, if these places have two tokens, it means that the exit and entry time of the two vehicles are the same, causing time conflict and a higher probability of collision. In this model Figure 3.5, three controllers are designed to make sure that the entry, as well as exit place, has exactly one vehicle at a given time, to avoid intersection collision. The controllers make sure to follow this rule by the speed change. Speed change would be made via the braking or throttle mechanism. Once the intersection has been crossed, the vehicle enters the cruise control stage.

Table 3.1.: Functions of Places

Places	Definitions
p1	Start of the Model
p2	Checks for disruptions
p3	stop
p4	Checks for intersection
p5	Enters the velocity controller
p6	The Velocity of the host vehicle
p7	The Velocity of the target vehicle
p8	Ensures the velocity controller is in process
p9	Final velocity
p10	Input to distance controller
p11	Ensures that safe distance is followed
p12	Enters distance controller

continued on next page

Table 3.1.: *continued*

Places	Definitions
p13	Information about distance followed
p14	Distance between both vehicles
p15	Distance followed
p16	Starts the process of re-changing velocity
p17	Ensures that final velocity is re-changed
p18	Difference between final and re-changed velocity
p19, p20	Re-changed Velocity
p21, p22	Final velocity
p23	Indicates the end of the cycle
p24	Start of Intersection Collision Avoidance system
p25	Exit information of host vehicle
p26	Exit information of another vehicle
p27	Exit place
p29	Entry information of host vehicle
p30	Entry information of another vehicle
p31	Entry place
p32	End of Intersection Collision Avoidance System
p34	Indicates the state before Intersection crossing
p35	Information on both vehicles
p36	Entry place Controller
p37	Exit place Controller
p38	Controller to indicate the intersection crossing
p39,p40,p41	Redundant Controllers

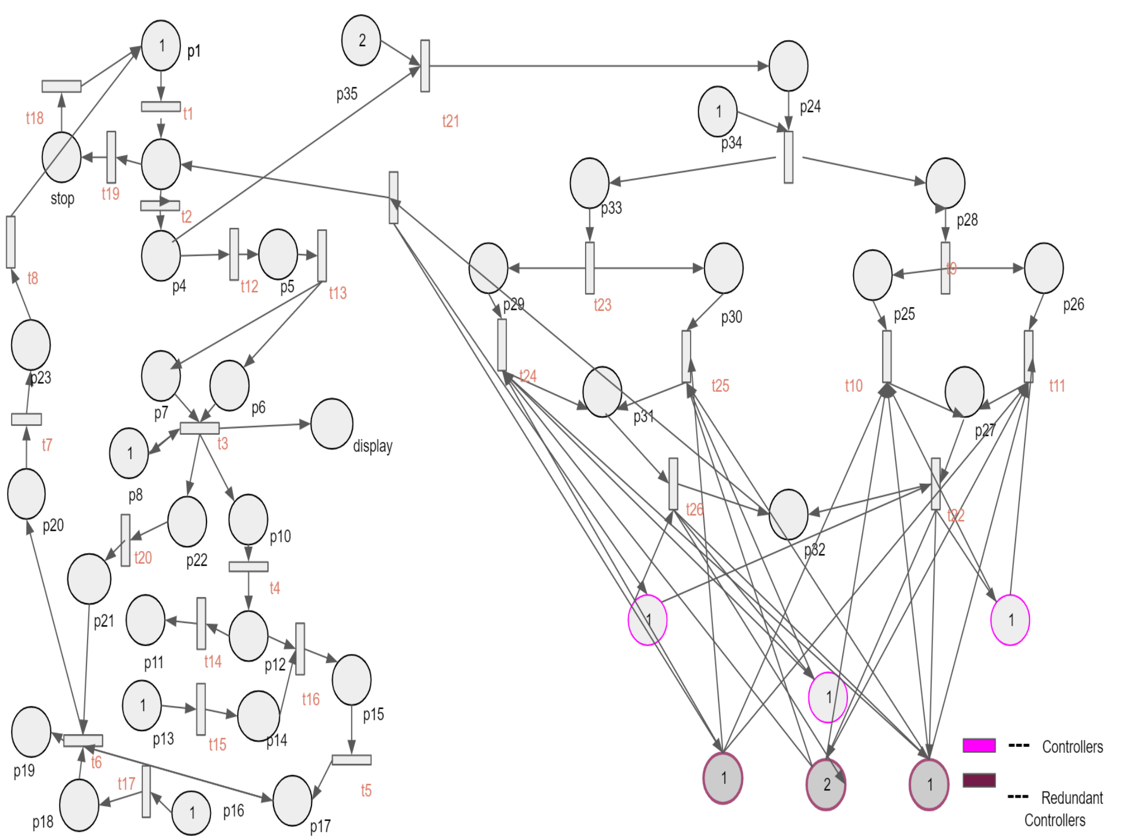

Fig. 3.5. Intersection Collision Avoidance System with Controllers

Figure 3.5 shows the Intersection Collision Avoidance System with controllers. The places p25, p26, p29, p30 consists of exit and entry information of the host and the target vehicle. Places p27 and p31 are exit and entry points.

The controllers are designed such that those two places do not contain more than one token. There is another controller for the place p32. Place p32 indicates that the intersection has been crossed by the vehicle. So, in total, there are three controllers to ensure that the model is working fine. And three redundant controllers to check the working of those normal controllers.

Limitations

The model has been designed and implemented in such a way that it is used for simple intersections. But when there is a polygon intersection, the model fails. And also, the space of two vehicles is considered to be the same, and one vehicle per lane is considered here.

3.4 Fault Tolerance

The fault tolerance technique was introduced for Petri net controllers. It provides tolerance in case of fault occurrences in the net. Any faults in the controller in the net may cause the net to function abnormally, hence the fault tolerance technique is used to detect place and transition faults. Faults in real-time can be defined as conditions or scenarios that cause a device or a component to fail to perform in a required manner. A fault may lead to the termination of the ability of an item to perform a function. If the failure can be detected early, then changes can be made, or places may be controlled to ensure that the functionality remains the same.

A place fault results in an incorrect token-load of a place i.e it changes the number of tokens to be given to a place. A transition fault arises when the token-load of either the input or output place - set of transitions is not appropriately updated following the firing of a transition. We say that transition tj has a post-condition fault if no tokens are deposited at its output places, even though the tokens from its input places are consumed. Similarly, we say that transition tj has a pre-condition fault if the tokens that are supposed to be removed from the input places are not removed, even though tokens are deposited at the corresponding output places. In this system, the redundant controllers are designed to avoid faults. The use of d redundant places enables the detection and identification up to:

$$d - 1$$

transition faults and up to d/2 place faults. Thus, the redundant controller detects the faults at controllers.

For each controller, one redundant controller is designed to keep that controller in check. By performing linear parity checks on the combined marking of the original controller places and the additional redundant places, our methodology can detect and identify faults in the redundant Petri net controller in a systematic manner. The redundant controller is constructed in such a way that it does not interfere with the already designed model, such as disrupting the transitions.

3.4.1 Design of Controllers and Redundant Controllers

The controllers are designed in such a way that the constraints are enforced using the formula:

$$L * q <= b$$

where the rows in L matrix deals with the number of constraints, q is the marking vector. For designing the controllers, the formulas taken into considerations are:

$$L * q_0 + q_{c0} = b$$

where, q_0 is the initial marking and q_{c0} is the initial marking of the Petri net controller.

$$-L * B = B_c$$

where, B indicates the incident matrix of the Petri net model and B_c indicates the incident matrix of the controllers. The constraints taken into account are such that the number of tokens in places p31, p32 and p27 should be equal or less than 1.

```
M(p31)<=1

M(p27)<=1

M(p32)<=1
```

The place fault is given by:

$$q_f(t) = q(t) + e_p$$

where q(t) is the state that would have been reached if there are no faults and

$$e_p \textit{ is a place error vector}$$

While designing the redundant controllers, extra places and tokens are added to the original Petri net model. The controller's state is changed according to the transitions. These redundant controllers will obtain information related to faults.

The initial marking of the redundant controllers are given by,

$$q_{c0} = \begin{pmatrix} b - L * q_0 \\ c * b - c * L * q_0 \end{pmatrix}$$

The incident matrix of the redundant controllers is given by,

$$B_c^{rT} = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & -1 & -1 \\ -1 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & -2 & -1 \\ 0 & -1 & 0 & -1 & -2 & -1 \\ 0 & 1 & -1 & 1 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

The parity check matrix is given by P=[-c I] where c matrix ensures that the columns of the parity check P are not rational multiples of each other. The incident matrix of the redundant controller is:

$$B_c^r = \begin{pmatrix} B_c \\ c * B_c \end{pmatrix}$$

Here, the c matrix is considered to be:

$$c = \begin{vmatrix} 1 & 1 & 0 \\ 1 & 2 & 0 \\ 1 & 1 & 1 \end{vmatrix}$$

D matrix D=[0]. To allow the redundant controllers to admit all the firing transitions that are allowed in normal controllers, the D matrix should only have non-negative integers.

$$P = \begin{vmatrix} -1 & -1 & 0 & 1 & 0 & 0 \\ -1 & -2 & 0 & 0 & 1 & 0 \\ -1 & -1 & -1 & 0 & 0 & 1 \end{vmatrix}$$

If the obtained result is multiples of these arrays, then there is place faults in p1, p2, p3, p4, p5, p6 respectively.

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Hence, in this model, the place faults were determined and the transition faults were assumed to be absent.

4. ANALYSIS OF THE PETRI NET MODELS

4.1 Place Invariants

Invariants are one of the properties of the Petri net which does not depend on the initial marking. It talks about the topographical structure of the net. The invariant analysis is an important property of the Petri net since it describes the dynamic process of the system. Place invariants are a set of places that contain constant tokens for all possible markings.

P invariants can be used for determining the reachability states. It is a pre-processing step. The weighted sum of the tokens in the invariant remains constant at all markings, and the sum is determined by the initial marking of the Petri net. The place invariants of a net can be determined using the formula,

$$X * A = 0$$

where A is the N x M incidence matrix of the Petri net, with N being the number of places and M the number of transitions of the net.

Multiplying a place invariant with a constant or adding two invariants will give rise to another invariant. A P-invariant in which all entries are either 0 or 1 indicates a set of places in which the number of tokens remains unchanged in all reachable markings.

Figure 4.1 shows some P invariants associated with this model. When multiplied with the Incident matrix, it gives a zero matrix. Each column represents one p-invariant vector. In the first p-invariant vector the place p8 has 1 token. It can be seen from the Cruise Control System, that this place always remains 1. Thus, the p-invariant condition is satisfied.

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	0	0
0	0	0
0	0	0
0	0	0
0	0	50
0	0	1
0	0	1
0	0	0
0	0	50
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	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	0
0	0	0
0	0	0
0	0	0
0	0	0

Fig. 4.1. P Invariants

4.2 Transition Invariants

A property that does not vary and remain constant even after the firing of transitions is termed to be invariant. When the markings in the net return to its original state, it is termed to be transition invariant. The order of the transition firing is not important here. Whatever be the order a transition is fired, the cycle will be complete. It is the same as the place invariant which can be used for reachability analysis. A T-invariant is a vector with one entry for each transition. A T-invariant indicates a possible loop in the net, that is, a sequence of transitions whose total effect is null, i.e. which leads back to the initial state from which it started.

Multiplying an invariant with a scalar number gives back an invariant. If a Petri net is defined by (P, T, F) each transition is assigned a non-negative number z as a weight.

So, if a particular transition is fired z times, then the marking is not changed. Let T includes a set of transitions t1, t2 till tn. Then, a transition invariant is defined as the summation of z1*t1, z2*t2 till zn*tn. The transition invariant indicates a loop or cycle. The formula for transition invariant is,

$$A * T = 0$$

with A being Incident matrix and T invariant vector. In this model, the transient invariant obtained is,

$$T = \left| \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right|$$

5. CONCLUSIONS AND FUTURE WORK

5.1 Conclusion

In this thesis, the intersection collision warning system is designed and analyzed along with the cruise control system in Petri net models. The Petri net controllers are designed to make sure that the model is working fine. Redundant controllers are formulated using the fault-tolerance technique to ensure proper working of the controllers. These redundant controllers detect the place faults and help in preserving the functionality of the controllers. The model is validated for different scenarios using MATLAB PN toolbox.

5.2 Future Work

The given Petri net model can be upgraded by adding certain features that are relevant to safety and data security. The features to improvise are related to,

- Adverse weather conditions.
- Cybersecurity.
- V2V communication.

5.2.1 Adverse Weather Conditions

In this thesis, the Petri net model was designed for typical weather elements. So, in the case of adverse weather, this model is not suitable for implementation. Hence, additional conditions should be implemented in such a way that the vehicle works fine even during harsh weather conditions like snow, fog, rain, ice, etc.

5.2.2 Cybersecurity

Since this model was developed for autonomous vehicles, the protection of the data becomes significant. If the vehicle is vulnerable to any cyber-attack, then it would lead to complete havoc. Depending on the circumstances, it would lead to many deaths. For the autonomous vehicles, the vehicle-to-vehicle communication transmits information like velocity, the distance of a target vehicle, updates in infrastructure like traffic signals, etc. Since a vehicle predicts its next action using this information, the probability of this data being attacked is not to be taken easily. So, incorporating certain features for the vehicle's data security would be an extension of this model.

5.2.3 V2V Communication

This model barely covers the communication issues between two vehicles. V2X communication is used to get real-time information in this model, but the model itself does not deal with the communication part. Hence, the model can be improvised by including a communication block along with other safety features.

REFERENCES

REFERENCES

[1] Reza Azimi, Gaurav Bhatia and Ragunathan (Raj) Rajkumar "Vehicular Networks for Collision Avoidance at Intersections" *SAE International Journal of Passenger Cars-Mechanical System*, 4(1):406-416, 2011.

[2] Michael R. Hafner, Drew Cunningham, Lorenzo Caminiti, and Domitilla Del Vecchio "Cooperative Collision Avoidance at Intersections: Algorithms and Experiments" *IEEE Transactions on Intelligent Transportation Systems*, no. 3:11621175, 2013.

[3] Petri Nets, [Online] Available: `https://www.techfak.uni-bielefeld.de/~mchen/BioPNML/Intro/ pnfaq.html` (Last Accessed: 07/02/2019)

[4] Ren David, Hassane Alla, Discrete, Continuous, and Hybrid Petri Nets, *Springer-Verlag Berlin Heidelberg*, 2010 edition.

[5] Asaad Kaadan, Hazem Refai "iICAS: Intelligent intersection collision avoidance system" *IEEE Conference on Intelligent Transportation Systems* 2012, 1184-1190. 10.1109/ITSC.2012.6338750.

[6] How Cruise Control Systems Work, [Online] Available: `https://auto.howstuffworks.com/cruise-control4.htm` (Last Accessed: 07/02/2019)

[7] Cruise Control, [Online] Available: `http://www.cs.cmu.edu/~ModProb/CC.html` (Last Accessed: 07/02/2019)

[8] Husam Kaid, Abdulaziz M. El-Tamimi, Emad Abouel Nasr, and Abdulrahman Al-Ahmari "Applications of Petri nets Based Models in Manufacturing Systems: a review" *International Conference on Operations Excellence and Service Engineering Orlando, Florida, USA,2015.*

[9] Richard Zurawski and MengChu Zhou Petri Nets and Industrial Applications: A Tutorial" *IEEE Transactions on Industrial Electronics*, Vol. 41, No. 6, December 1994.

[10] Rohan Kumar, Rajan Pathak "Adaptive Cruise Control - Towards a Safer Driving Experience" *International Journal of Scientific and Engineering Research* Volume 3, Issue 8, August-2012, ISSN 2229-5518.

[11] Petri Nets, [Online] Available: `http://www.labri.fr/perso/anca/FDS/Pn-ESTII.pdf` (Last Accessed: 07/02/2019)

[12] Katherina Yamalidou, John Moody, Michael Lemmont and Panos Antsaklist "Feedback Control of Petri Nets Based on Place Invariants" *Automatica*, Vol. 32, No. 1, pp. 15-28, 1996.

[13] Hybrid functional Petri net, [Online] Available: http://genome.ib.sci.yamaguchi-u.ac.jp/gon/HFPNincludes.htm (Last Accessed: 07/02/2019)

[14] Vehicle-to-Vehicle Communication, [Online] Available: https://www.nhtsa.gov/technology-innovation/vehicle- vehicle-communication (Last Accessed: 07/02/2019)

[15] Introduction to the Controller Area Network, [Online] Available: http://www.ti.com/lit/an/sloa101b/sloa101b.pdf (Last Accessed: 07/02/2019)

[16] Mariagrazia Dotoli, Maria Pia Fanti, Agostino Mangini "Operational Management of Supply Chains: A Hybrid Petri Net Approach" *ResearchGate*,10.5772/6657,2009.

[17] Lingxi Li, Jiaxiang Yan "Fault-tolerant controller design for automated guided vehicle systems based on Petri nets" 15th*International IEEE Conference on Intelligent Transportation Systems*,2012.

[18] Lingxi Li, Christoforos N. Hadjicostis and R. S. Sreenivas "Fault Detection and Identification in Petri Net Controllers" 43rd *IEEE Conference on Decision and Control December* 14-17, 2004.

[19] The history of cruise control, [Online] Available: https://www.autonomousvehicleinternational.com/features/adas-3.html (Last Accessed: 07/02/2019)

[20] Petri Nets, [Online] Available: https://en.wikipedia.org/wiki/Petrinet (Last Accessed: 07/02/2019)

[21] Petri Net boundedness, [Online] Available: https://stackoverflow.com/questions/33598007/petri-net-boundedness (Last Accessed: 07/09/2019)

[22] Modelling with Petri Nets, [Online] Available: http://jklp.org/profession/books/pn/3.html (Last Accessed: 07/09/2019)

[23] M.M. Mansoura, Mohamed A.A. Wahabb, Wael M. Soliman "Petri nets for fault diagnosis of large power generation station" *Ain SHams Engineering JOurnal* Volume 4, Issue 4,December 2013, Pages 831-842.

[24] Gustavo Callou, Paulo Maciel, Dietmar Tutsch, Julian Arajo, Joo Ferreira and Rafael Souza "A Petri Net-Based Approach to the Quantification of Data Center Dependability" *IntechOpen* August 29th 2012, DOI: 10.5772/47829.

[25] Mostafa Herajy, Fei Liu, Monika Heiner "Efficient modelling of yeast cell cycles based on multisite phosphorylation using coloured hybrid Petri nets with marking-dependent arc weights. Nonlinear Analysis: Hybrid Systems" *IEEE Access* 27. 191-212. 10.1016/j.nahs.2017.09.002, 2018.

[26] Mostafa Herajy, Fei Liu, Monika Heiner "Modeling biological systems with uncertain kinetic data using fuzzy continuous Petri nets" *BMC Systems BiologyBMC series* 27. 12 (Suppl 4) :42, April 2018.

[27] Fadi Basma, Yehia Tachwali and Hazem H Refai "Intersection Collision Avoidance System Using Infrastructure Communication" *14th International IEEE Conference on Intelligent Transportation Systems* 2011.

[28] Chanon Dechsupa, Wiwat Vatanawood, Arthit Thongtak ”Transformation of the BPMN Design Model into a Colored Petri Net Using the Partitioning Approach” *IEEE Access* PP. 1-1. 10.1109/ACCESS.2018.2853669.