

DEVELOPMENT OF MOBILE SURFACE WATER FILTRATION SYSTEM THROUGH SIMULATION USING COLOURED PETRI NET (CPN)

Aashish Vatsyayan¹, B. Hemalatha², A Vimala Juliet³, Ishwarya Venkatram⁴

AP(Sr.G), HOD, SRM University, SRM University, SRM University, SRM University

Aim

Testing the working of the mobile surface water filtration system through the design and simulation of a CPN generated schema

Methodology

For understanding the required action for different conditions the programming is developed using CPN tools for analysis of different conditions and obtaining a holistic picture of the system.

Result

The simulation model has demonstrated how the working of the mobile system is intended and has the actions of a microcontroller clearly cut out for the programmer to understand and write the code. Each module was looked upon in the different scenarios and the changes in the system after simulation were shown, providing a holistic understanding of the entire system. The sensors, microcontrollers and actuators have been discussed to bridge the gap between completely theoretical concepts to an implementable entity.

Conclusion

Impending release of tool for direct conversion of Coloured Petri Net simulation into hexadecimal format opens up infinite avenues for system modelling in Coloured Petri Net environment. The model design can be improved upon by adding time dependency property of CPN to simulate more dynamic and time dependant processes, this aspect has not been explored as a holistic illustration, but the system for mobile surface water filtration device has been thoroughly and holistically explained and explored.

Key words: Coloured Petri Nets, simulation, modelling, filtration system, mobile, navigation, electro coagulation.

1. INTRODUCTION

The simulation of CPN is based on triggering of inputs and demonstrating the resultant action impact on the system, it is parallel process simulation software where the triggering of one input has response from different sub systems, all these processes can be observed to code a microcontroller for performing similar actions in real time situations.

Before actual fabrication of an electrical system involving microcontrollers, sensors, actuators it is necessary to develop a logic that dictates when which actuator should react to given input, here CPN is used to display a blueprint of the logic that will be implemented into an actual functioning hardware prototype.

C P nets have various characteristics and some of the major characteristics touched upon while modeling the design were the liveness, boundedness and stochastic nature of the net. The design is that of a discrete, continuous and hybrid net. We also have introduced a dimension of high-level Petri nets by modeling using a CPN interface. It allows us to generate high - level nets with token values and hierarchy that allow

the transitions on various levels to fire. The problem of non-determinism has been tackled and the net has been incorporated with concurrent behavior in selected simulations. The Bayesian nature of the Petri nets has been explored and incorporated in the net schema.

The entire paper revolves around creating a simulation model to understand the working of the developed system and display the interconnectivity amongst the different parameters. Coloured Petri Nets is chosen as the platform for creating a simulation model as the token transfer in CPN is clearly understandable along with it treating all values in 0 or 1, which creates a direct colliery with the working of a microcontroller. CPN adds the concept of coding, graphical interface and Petri nets to produce a system which has powerful primitives and has a vast potential. The obtaining of a hexadecimal code from a CPN simulation is in its path of a commercial release and it will improve current modelling schemes as CPN will greatly reduce the processing time in microcontrollers. The simulation displays the working of the system under different situations and what response will be triggered in each scenario.

The case by case analysis is done to explain the working of the mobile surface water filtration device. The modules discussed are: navigation, electro coagulation and an overall working and output of the system. Before and after screen shots are used to explain the reaction of the output actuators on the basis of triggering of the input sensors.

MATERIALS AND METHODS

The CPN model was designed using the CPN tools interface window. A new page was created in the workspace and was saved using .cpn extension. The places, transitions required for the model were created and arcs connecting the places and transitions were designed. The places and the transitions were labelled with unique names and assigned data types. The places then were designated with a set number of tokens (i.e. 0 or 1). Next the variable declarations were made by creating an empty declaration and then modified according to the variable names and their respective data types. The function declarations were defined according to the various input parameters and an optimal code for generation of suitable output was written, for the respective functions. The variables and functions were then labelled onto the arc inscriptions, thus developing a proper place -arc-transition schema defining the structure of the autonomous mobile surface water filtration system. Subsequently, hierarchy and other parameters were added to increase the parallelism and complexity in the net and the net was colour coded. The non-determinism was verified and the schema was completed. The model generated was simulated with various input parameters and the optimal output was generated by the net.

The sensors are hydrocarbon sensors which can differentiate between crude oil and water, depending upon the range of their output a suitable analogue to digital converter can be used to give a high (i.e. 1) value when oil is detected and a low (i.e. 0) to the microcontroller. The sensors are connected on the body of the mobile device at 5 locations in a star format, 5 sensors are considered only to show how the discrimination of direction is to be made however more sensors can be added to increase the accuracy of system, it would result in more data to process though. One hydrocarbon sensor is connected on the input pipe to determine is oil is present in immediate proximity of the input site; it plays a nominal role in navigation.

Level sensors are used inside processing chamber to determine the level of the process liquid, floating switches type can be used.

Microcontroller of choice is 8051 is due to it being familiar and easy to use with it and as the mobile device is still in development stage 8051 provides with the processing requirements at this stage.

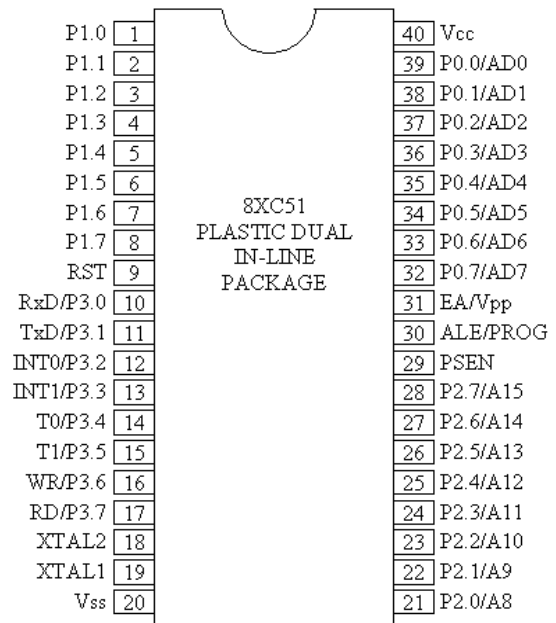


Figure 1 A pin diagram shows the number of available inputs and outputs.

Output actuators

The output actuators in this model are primarily valves and motors and propellers

Valves: for open and close operation at the input site. A simple solenoid valve can be used for this operation.

Motors: the motor being used is a simple DC motor having high torque and low rpm to facilitate the movement of input pipe. The motor is enclosed inside the processing tank and is replaceable in case the motor burns out; a high torque has been chosen to move the input pipe despite the viscosity offered by water.

Propellers: the propellers are high torque and high rpm to facilitate the movement of the mobile device.

The battery backup is not discussed in this paper, plausible avenues for research into supply of electricity or energy are: crude oil internal combustion engines, solar cells, microbial fuel cells and wind energy.

The process of electro coagulation mentioned in the paper can be looked up in relevance in the paper 'mobile surface water filtration system'.

Operation:

The navigation and electro coagulation units work in 6 different modes. The before and after simulation screen shots are explained.

CASE 1: When all the sensors test positive for oil:

1.1 BEFORE SIMULATION:

1.1.1 WHOLE MODEL SCHEMA:

Sensor 1 : 1
 Sensor 2 : 1
 Sensor 3 : 1
 Sensor 4 : 1
 Sensor 5 : 1

Navigation(Propeller main,Propeller left, Propeller right) : 0
 Electrocoagulation(Input valve,Level sensor1,Levensensor2) : 0

Output : 0

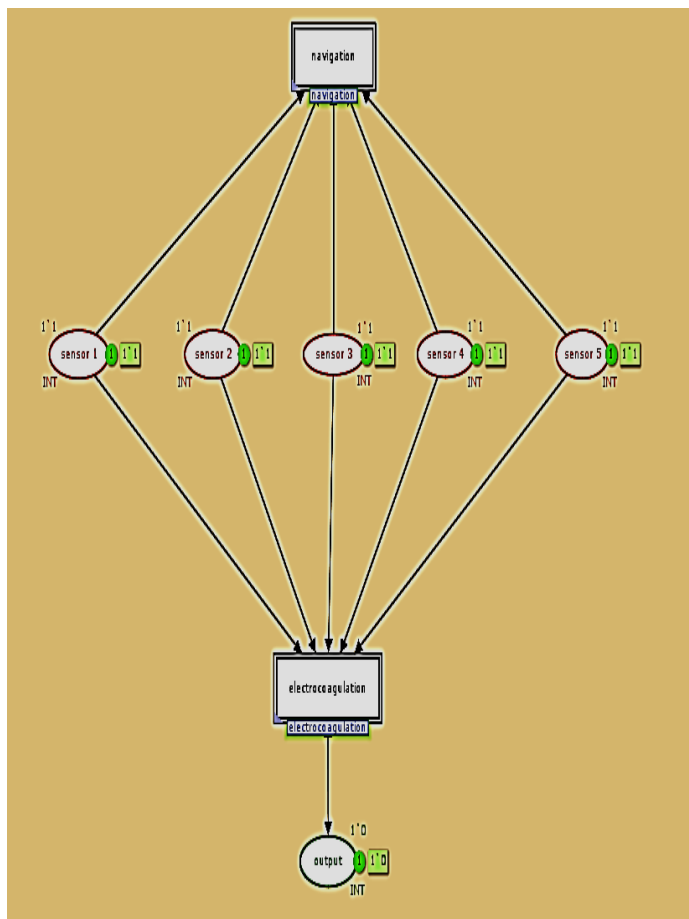


Figure 1.1.1 Model before simulation when all sensors are 1

1.1.2 NAVIGATION SCHEMA:

Sensor 1 : 1
 Sensor 2 : 1
 Sensor 3 : 1
 Sensor 4 : 1
 Sensor 5 : 1

Navigation(Propeller main,Propeller left, Propeller right) : 0

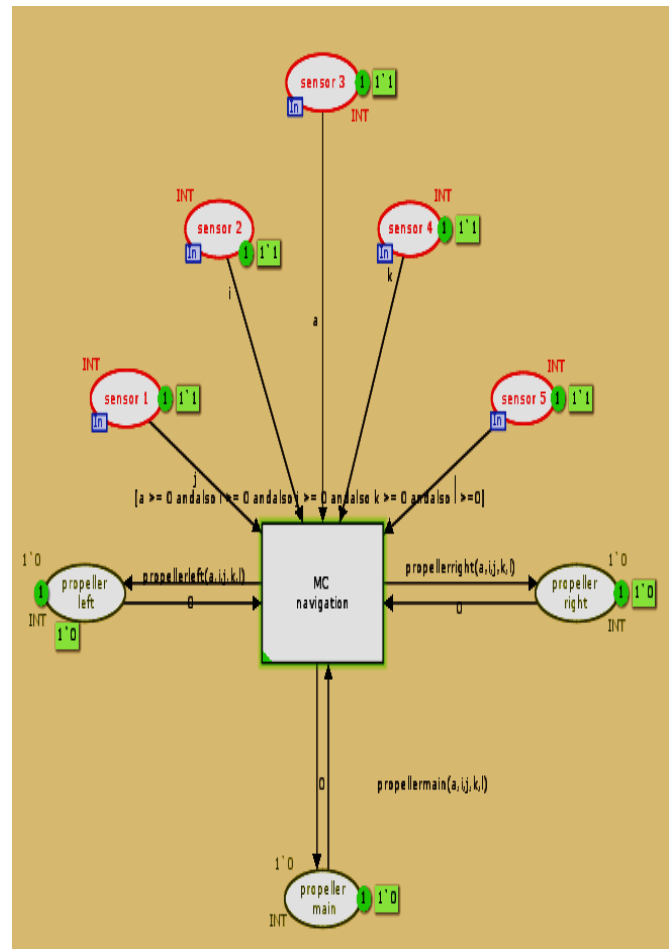


Figure 1.1.2 Navigation model before simulation when all sensors are 1 and propellers are 0

1.1.3 ELECTRO COAGULATION SCHEMA:

Sensor 1 : 1
 Sensor 2 : 1
 Sensor 3 : 1
 Sensor 4 : 1
 Sensor 5 : 1
 Electrocoagulation(Input valve,Level sensor1,Levensensor2) : 0
 Output : 0

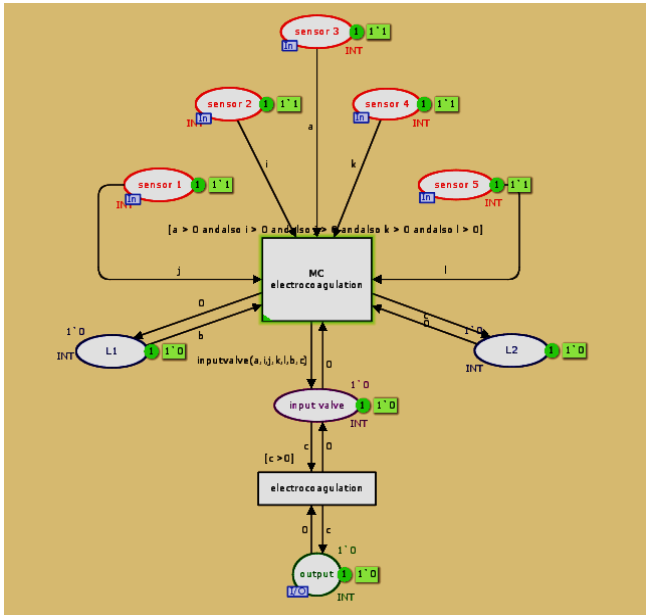


Figure 1.1.3 Electro coagulation before simulation when all sensors are 1, level sensors are 0 and output is 0

1.2. AFTER SIMULATION:

1.2.1 WHOLE MODEL SCHEMA:

Navigation(Propeller main, Propeller left, Propeller right) : 0
 Electrocoagulation(Input valve, Level sensor1, Level sensor2) : 0
 Output : 1

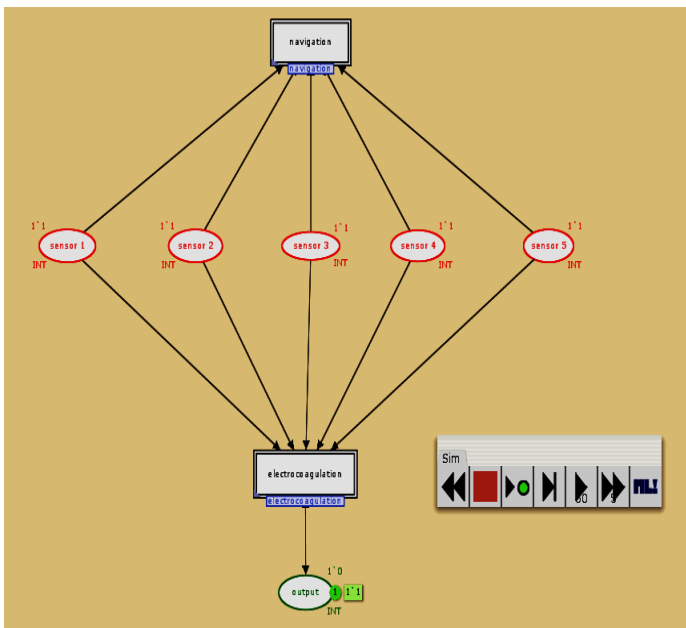


Figure 1.2.1 Model after simulation yields output 1

1.2.2 NAVIGATION SCHEMA:

Propeller main : 0
 Propeller left : 0
 Propeller right : 0

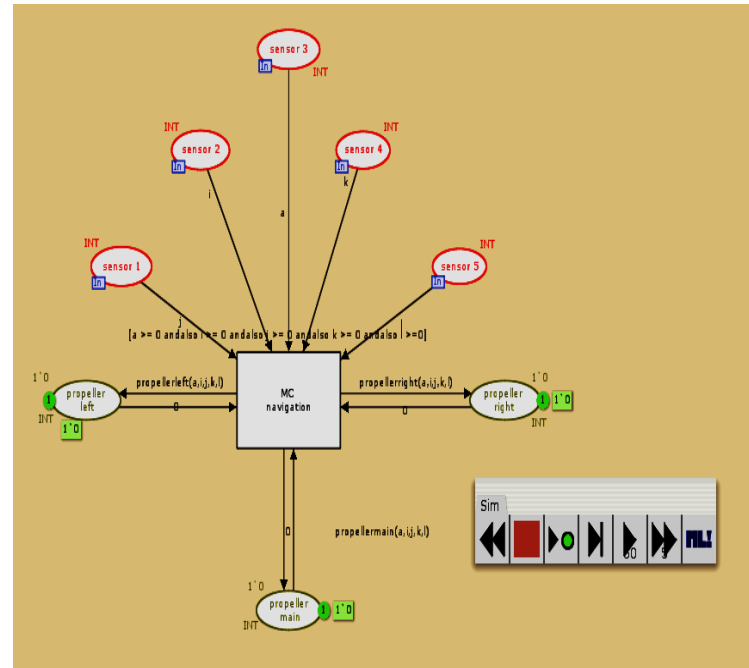


Figure 1.2.2 Navigation model after simulation yields propeller outputs as 0

1.2.3 ELECTRO COAGULATION SCHEMA:

INPUT VALVE : 1
 OUTPUT : 1

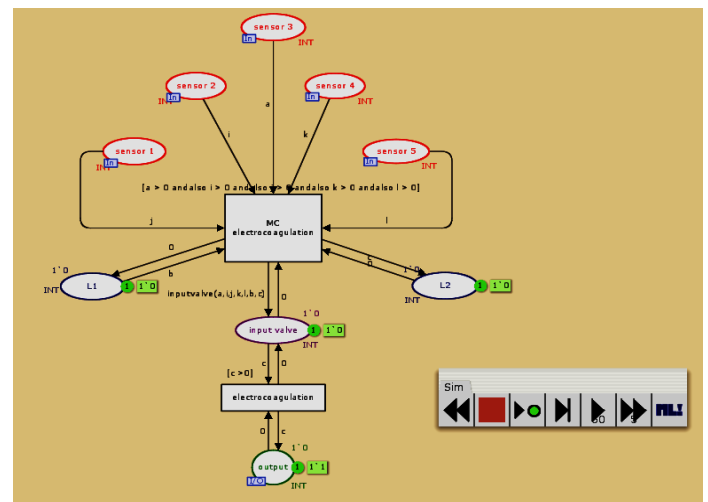


Figure 1.2.3 Electro coagulation after simulation yields output as 1

CASE 2: When only three sensors test positive for oil(i.e. sensor1,sensor2,sensor3):

2.1 BEFORE SIMULATION :

2.1.1 WHOLE MODEL SCHEMA:

Sensor 1 : 1
 Sensor 2 : 1
 Sensor 3 : 1
 Sensor 4 : 0
 Sensor 5 : 0
 Navigation(Propeller main,Propeller left, Propeller right) : 0
 Electrocoagulation(Input valve,Level sensor1,Leveldsensor2) : 0
 0

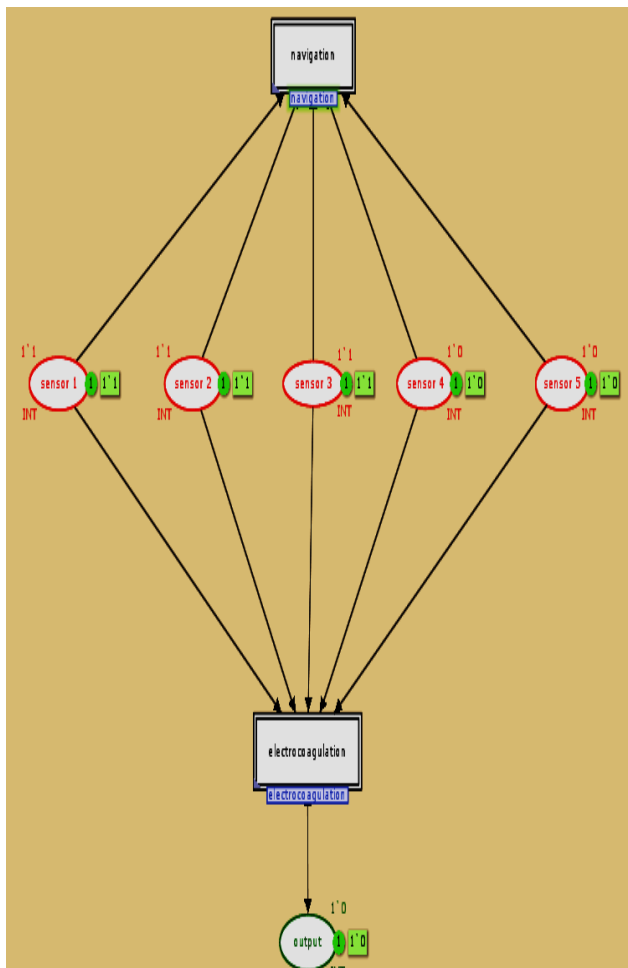


Figure 2.1.1 Model before simulation when all sensor1, sensor2, sensor3 are 1

2.1.2 NAVIGATION SCHEMA:

Sensor 1 : 1
 Sensor 2 : 1
 Sensor 3 : 1

Sensor 4 : 0
 Sensor 5 : 0
 Navigation(Propeller main,Propeller left, Propeller right) : 0

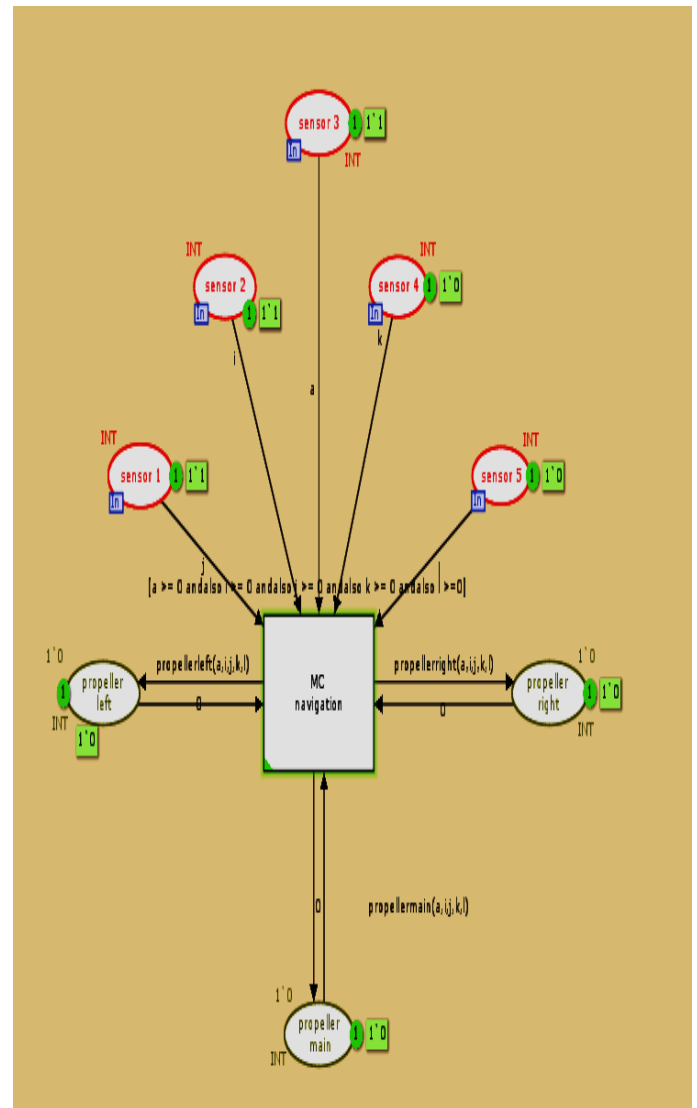


Figure 2.1.2 Navigation model before simulation when all sensor1, sensor2, sensor3 are 1 and propellers are 0

2.1.3 ELECTRO COAGULATION SCHEMA:

Sensor 1 : 1
 Sensor 2 : 1
 Sensor 3 : 1
 Sensor 4 : 0
 Sensor 5 : 0
 Electrocoagulation(Input valve,Level sensor1,Leveldsensor2) : 0
 0
 Output : 0

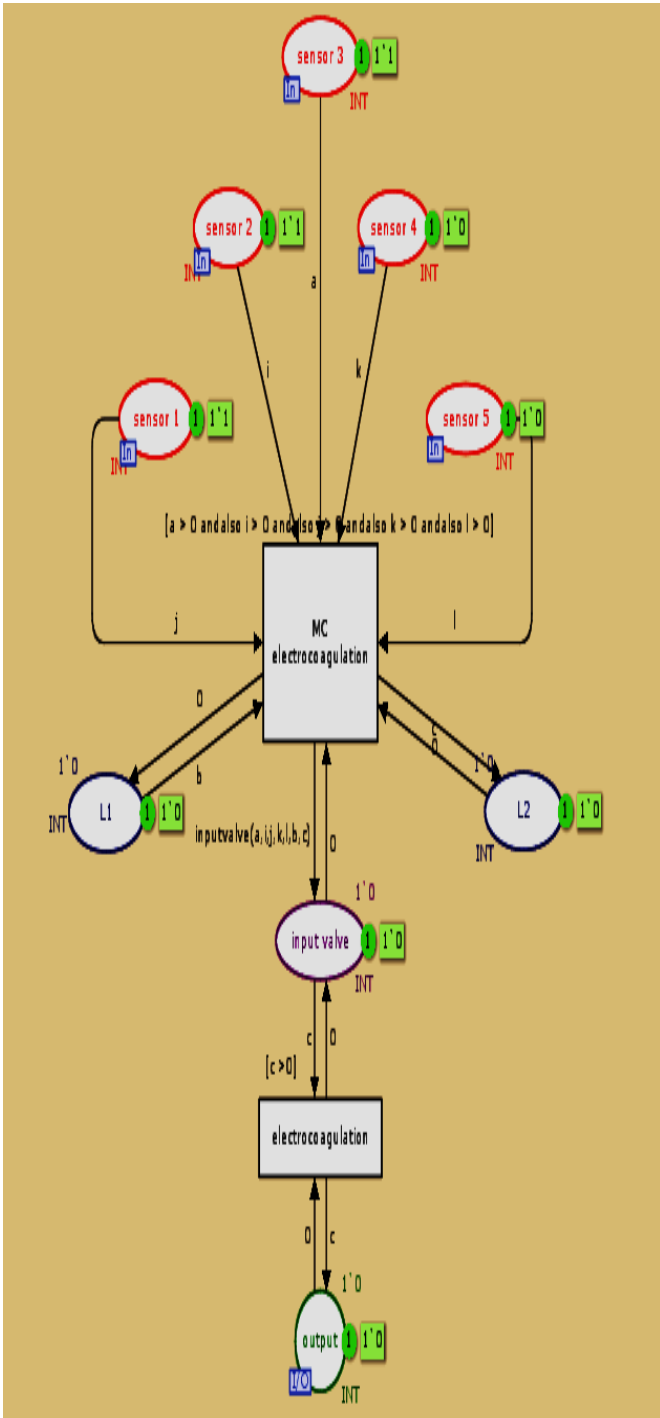


Figure 2.1.3 Electro coagulation before simulation when all sensor1, sensor2, sensor3 are 1, level-sensors are 0 and output is 0

2.2 AFTER SIMULATION :

2.2.1 WHOLE MODEL SCHEMA:

Output : 0

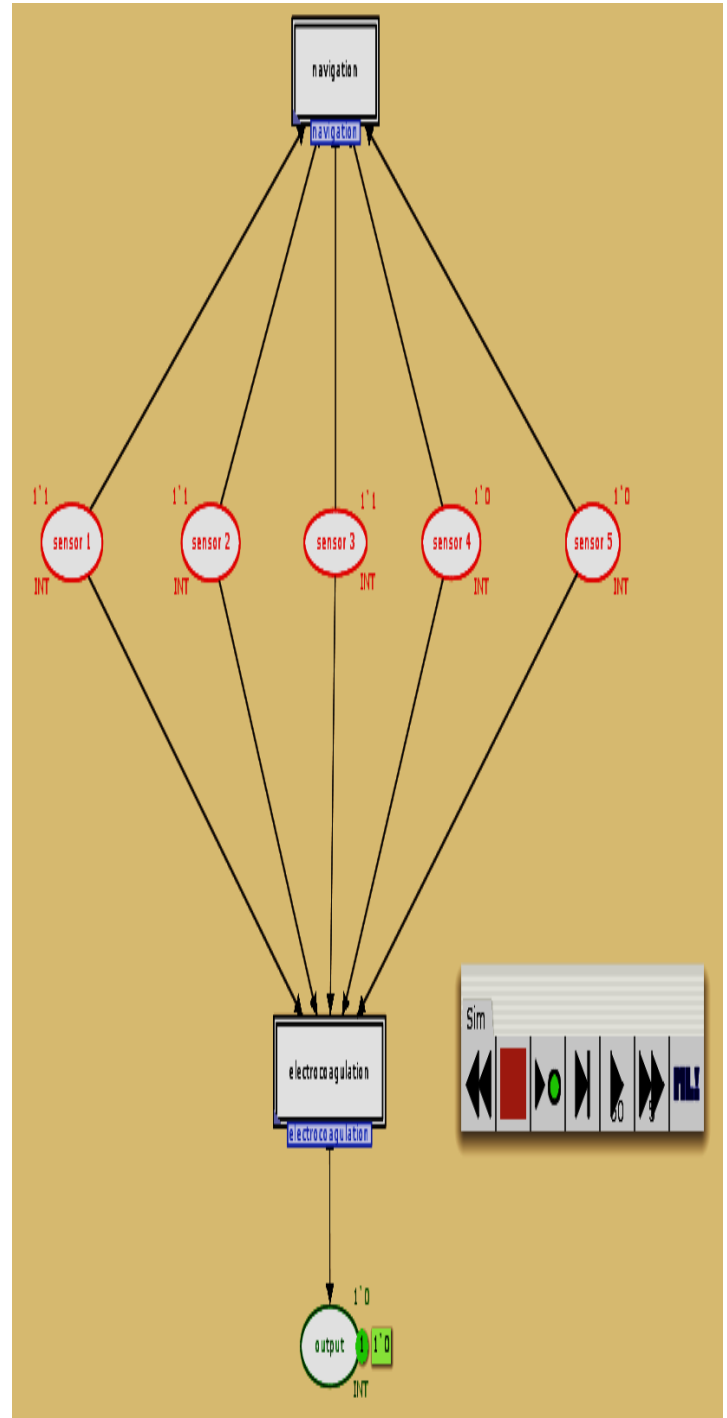


Figure 2.2.1 The simulation yields no output

2.2.2 NAVIGATION SCHEMA:

Propeller main : 500
 Propeller left : 0
 Propeller right : 500
 Inference: Left turn is made

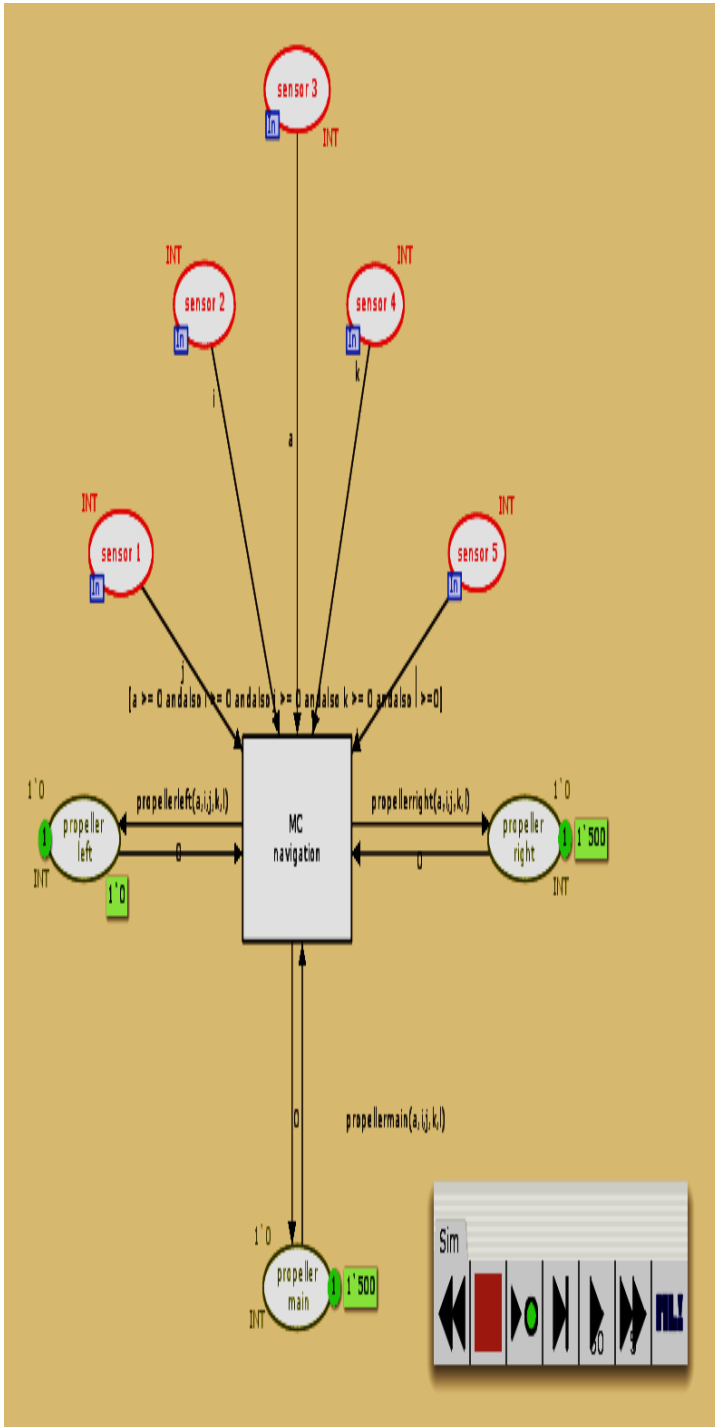


Figure 2.2.2 The simulation yields the system to turn left

2.2.3 ELECTRO COAGULATION SCHEMA:

INPUT VALVE : 0
 OUTPUT : 0

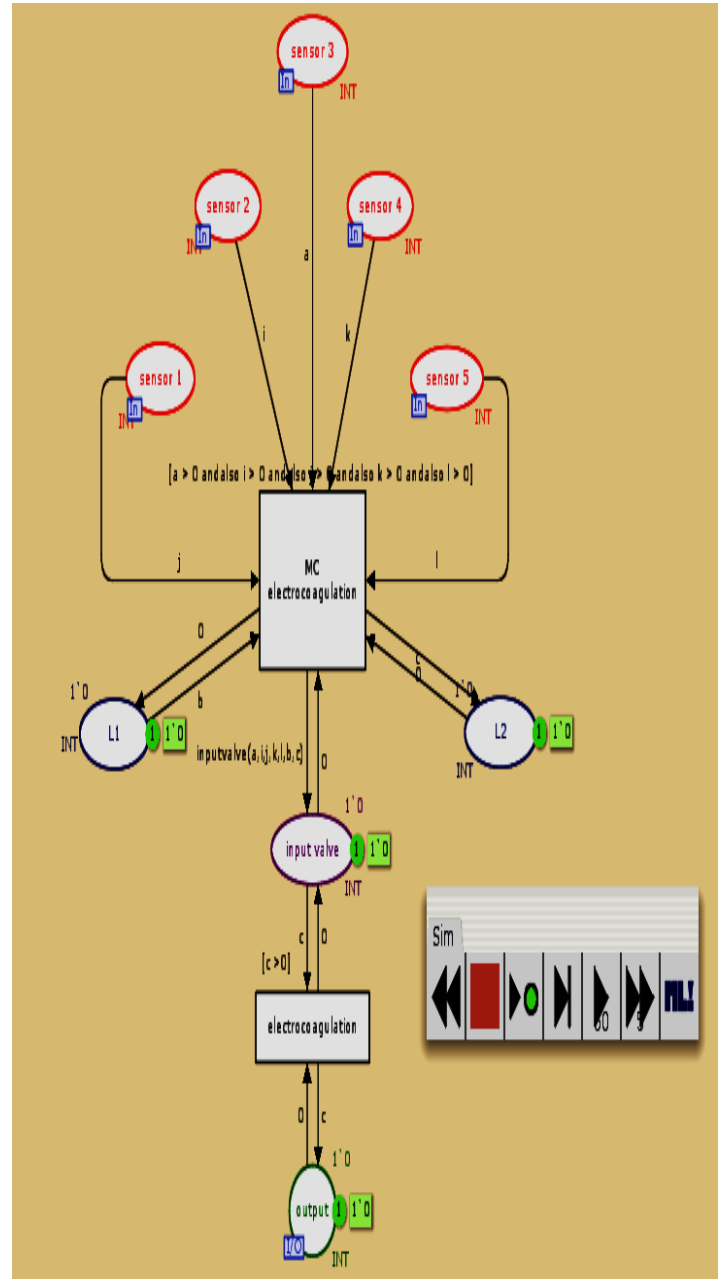


Figure 2.2.3 Electro coagulation after simulation yields no output

CASE 3: When only three sensors test positive for oil(i.e. sensor3,sensor4,sensor5):

3.1 BEFORE SIMULATION :

3.1.1 WHOLE MODEL SCHEMA:

Sensor 1 : 0
 Sensor 2 : 0
 Sensor 3 : 1
 Sensor 4 : 1

Sensor 5 : 1
 Navigation(Propeller main,Propeller left, Propeller right) : 0
 Electrocoagulation(Input valve,Level sensor1,Levensensor2) : 0

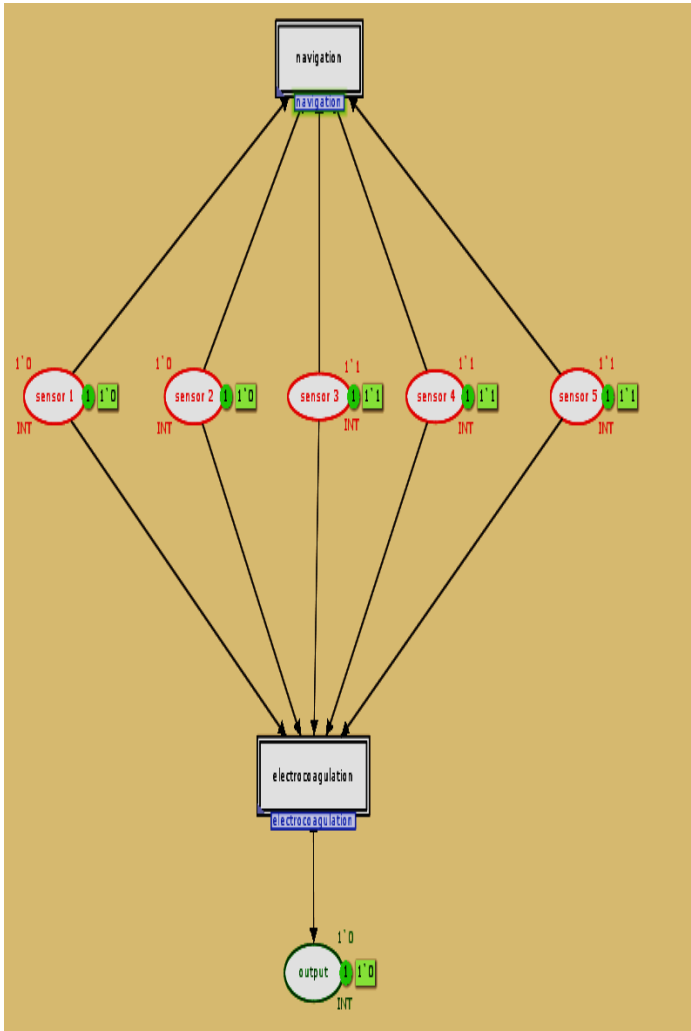


Figure 3.1.1 Model before simulation when all sensor3, sensor4, sensor5 are 1

3.1.2 NAVIGATION SCHEMA:

Sensor 1 : 0
 Sensor 2 : 0
 Sensor 3 : 1
 Sensor 4 : 1
 Sensor 5 : 1
 Navigation(Propeller main,Propeller left, Propeller right) : 0

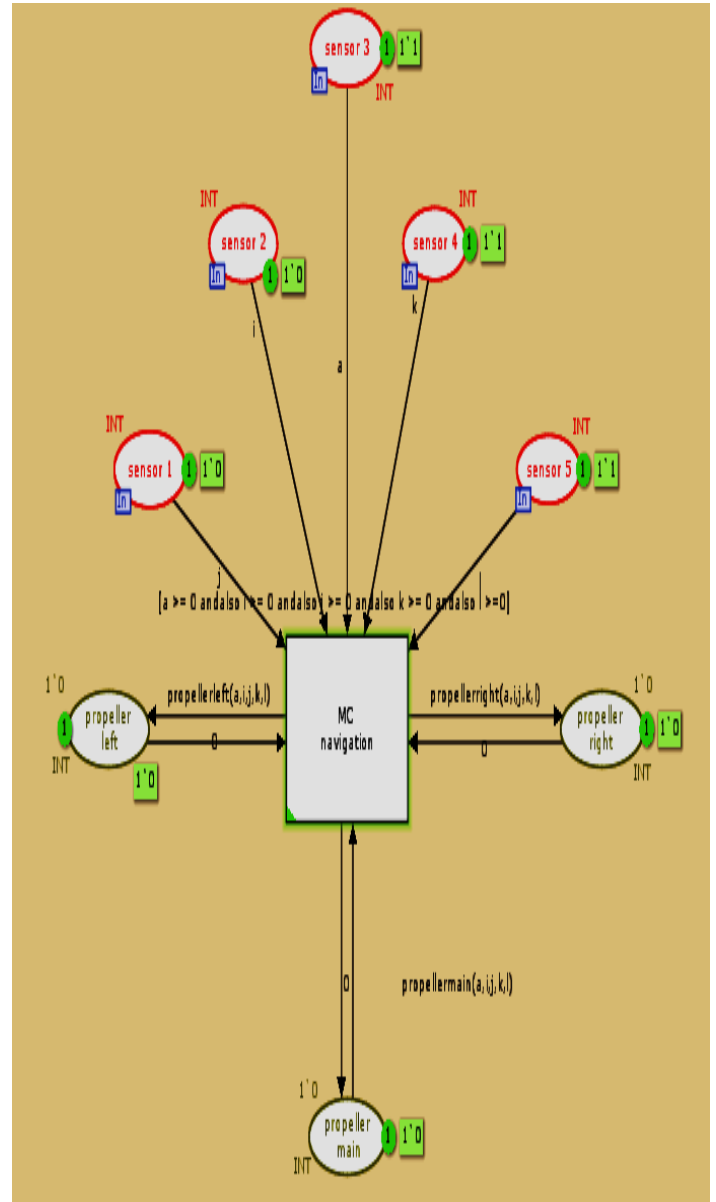


Figure 3.1.2 Navigation model before simulation when all sensor3, sensor4, sensor5 are 1 and propellers are 0

3.1.3 ELECTRO COAGULATION SCHEMA:

Sensor 1 : 0
 Sensor 2 : 0
 Sensor 3 : 1
 Sensor 4 : 1
 Sensor 5 : 1
 Electrocoagulation(Input valve,Level sensor1,Levensensor2) : 0
 Output : 0

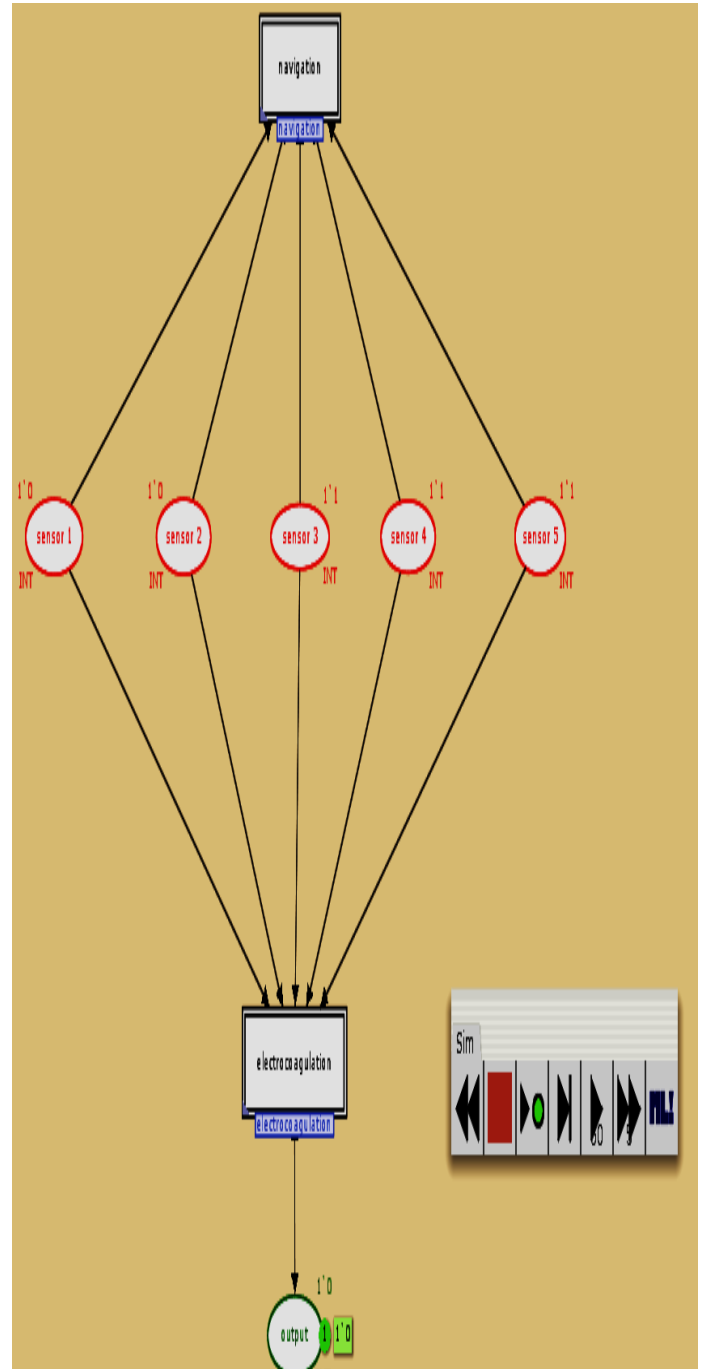
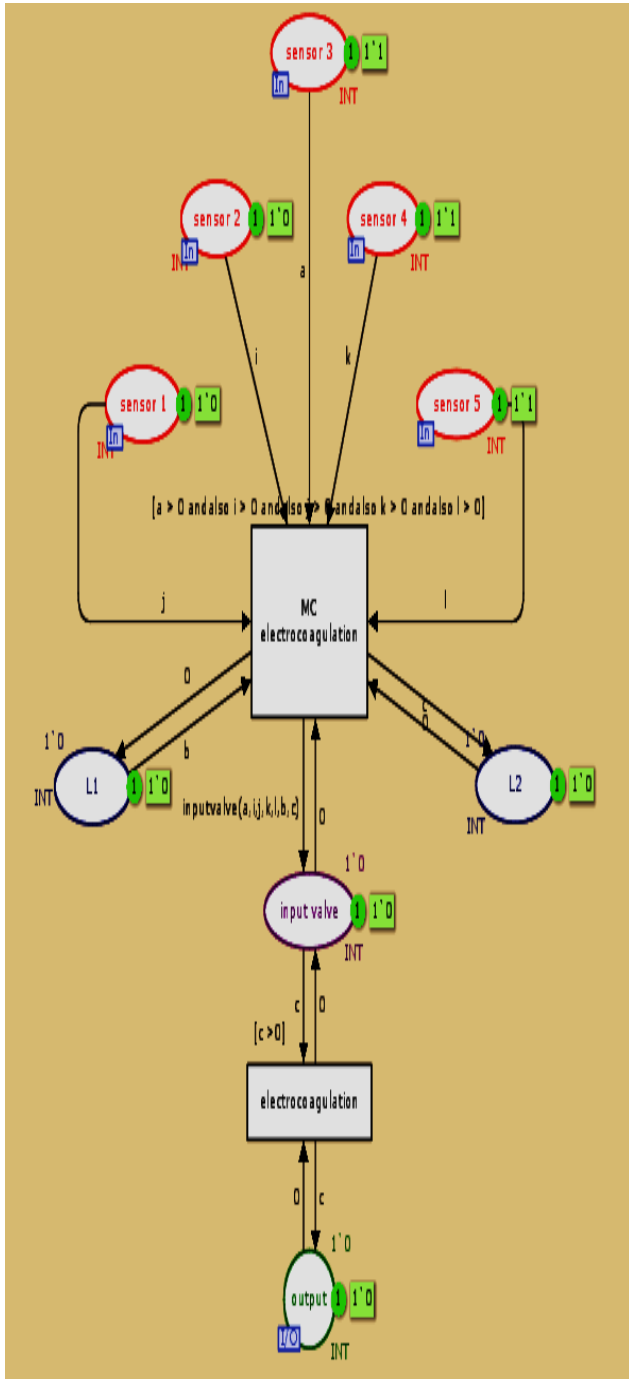


Figure 3.1.3 Electro coagulation before simulation when all sensor3, sensor4, sensor5 are 1, level-sensors are 0 and output is 0

Figure 3.2.1 The simulation yields no output

3.2 AFTER SIMULATION :

3.2.1 WHOLE MODEL SCHEMA:

Output : 0

3.2.2 NAVIGATION SCHEMA:

Propeller main : 500
 Propeller left : 500
 Propeller right : 0
 Inference: Right turn is made

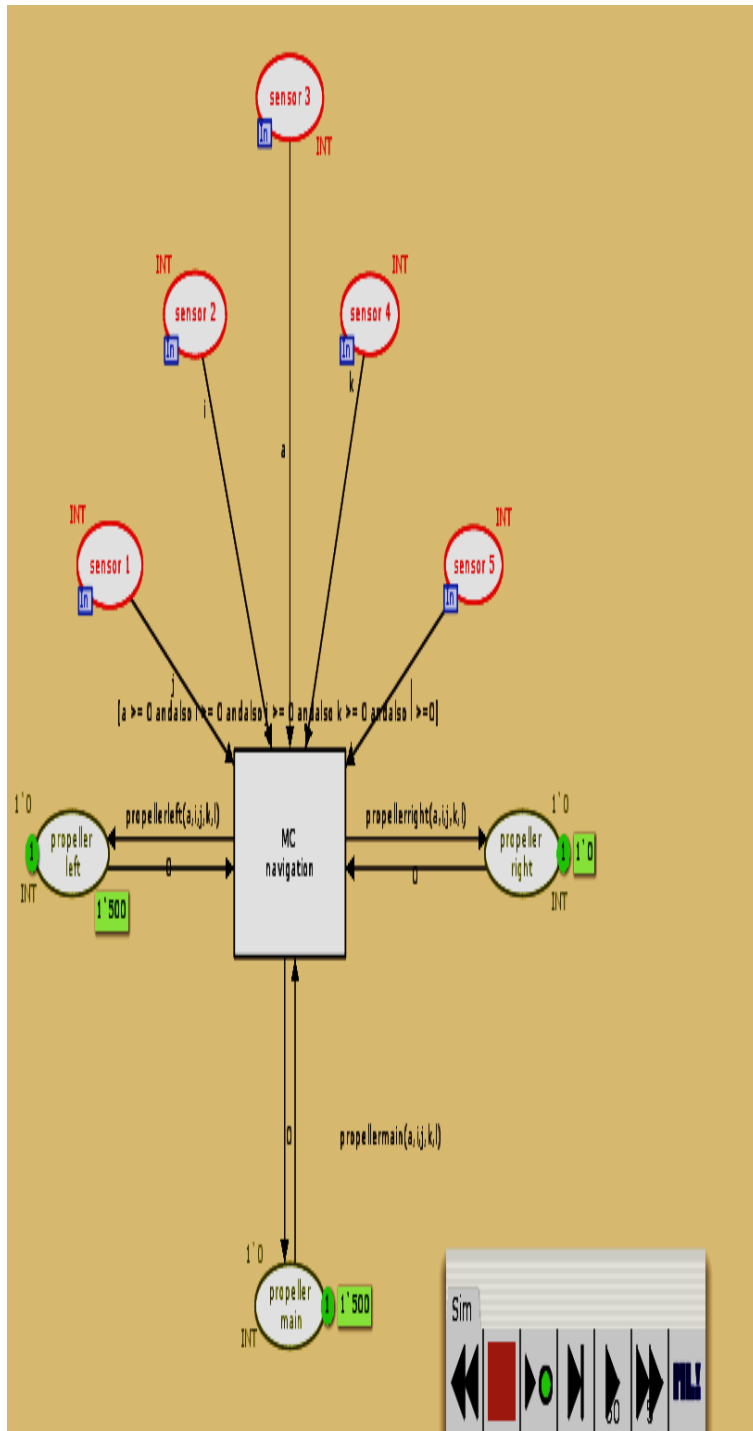


Figure 3.2.2 The simulation yields the system to turn right

3.2.3 ELECTRO COAGULATION SCHEMA:

INPUT VALVE : 0
 OUTPUT : 0

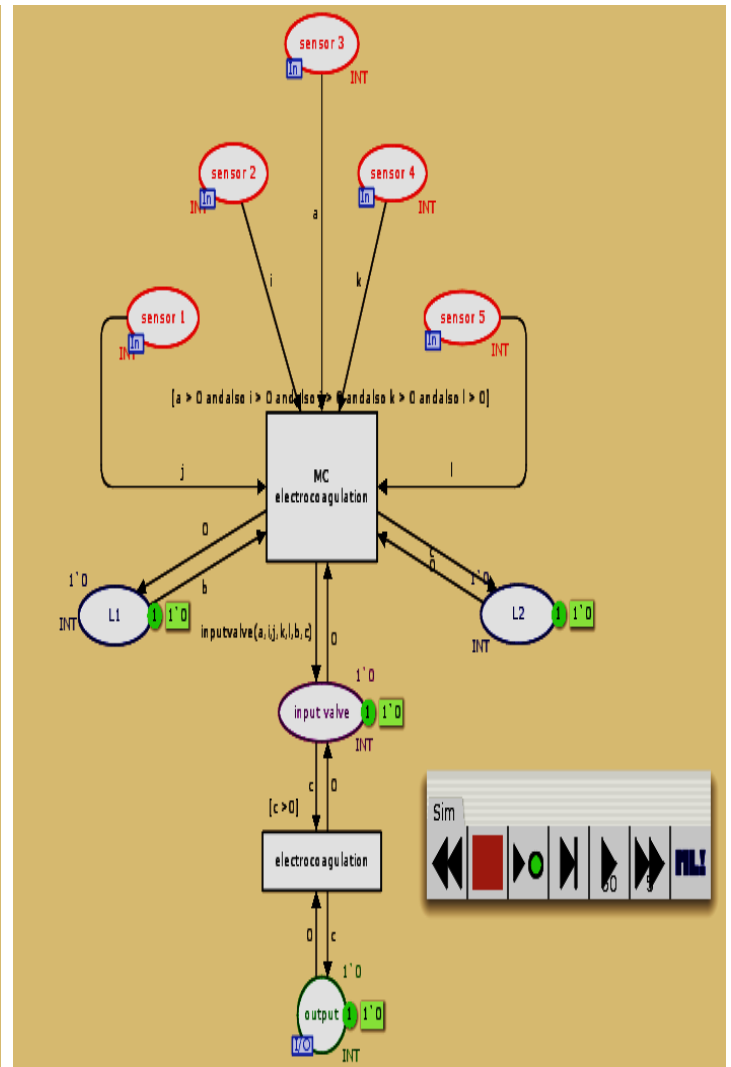


Figure 3.2.3 Electro coagulation after simulation yields no output

CASE 4: When all 5 primary sensors and 1 shaft sensor test positive for oil:

4.1 BEFORE SIMULATION :

4.1.1 MODEL SCHEMA:

- Sensor 1 : 1
- Sensor 2 : 1
- Sensor 3 : 1
- Sensor 4 : 1
- Sensor 5 : 1
- Sensor 6 : 1
- Shaft position : 0
- Navigation(Propeller main,Propeller left, Propeller right) : 0
- Electrocoagulation(Input valve,Level sensor1,Levensensor2) : 0

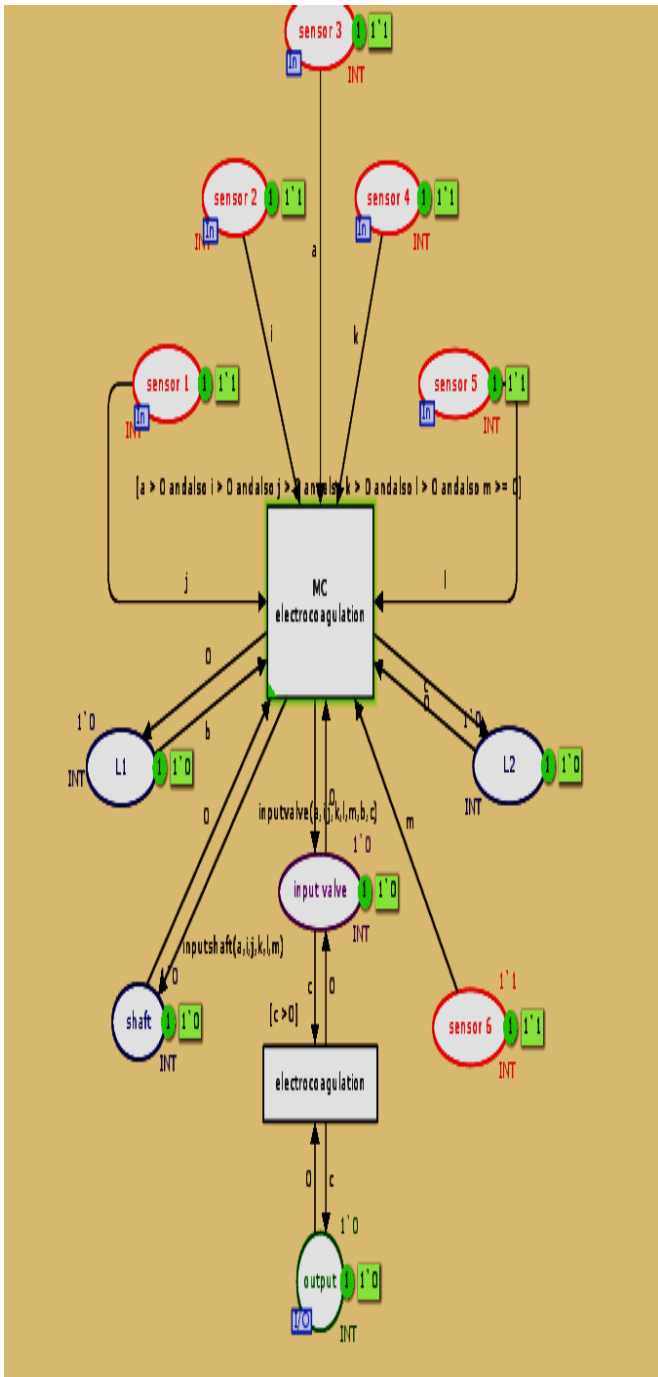


Figure 4.1.1 All 6 sensors test positive for oil and shaft position is 0

4.2 AFTER SIMULATION :

4.2.1 MODEL SCHEMA:

The inlet shaft is connected to a stepper motor of step angle 900 and hence it has 4 positions in its circular path at 00, 900, 1800, 2700 and then back to 00.
 Shaft Position : 0

Inference: Shaft position is unchanged and input valve is opened and electro coagulation happens

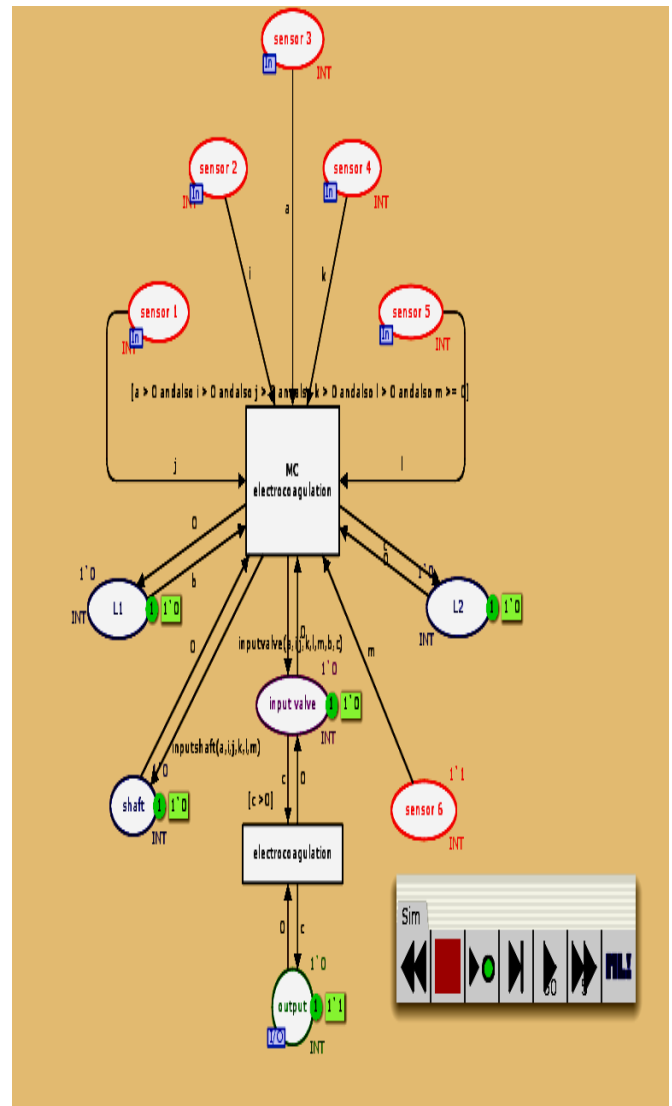


Figure 4.2.1 The simulation yields no change in position of the shaft

CASE 5: When all 5 primary sensors except 1 shaft sensor test positive for oil:

5.1 BEFORE SIMULATION :

5.1.1 MODEL SCHEMA:

- Sensor 1 : 1
- Sensor 2 : 1
- Sensor 3 : 1
- Sensor 4 : 1
- Sensor 5 : 1
- Sensor 6 : 0
- Shaft position : 0

Navigation(Propeller main,Propeller left, Propeller right) : 0
 Electrocoagulation(Input valve,Level sensor1,Levensensor2) : 0
 Output : 0

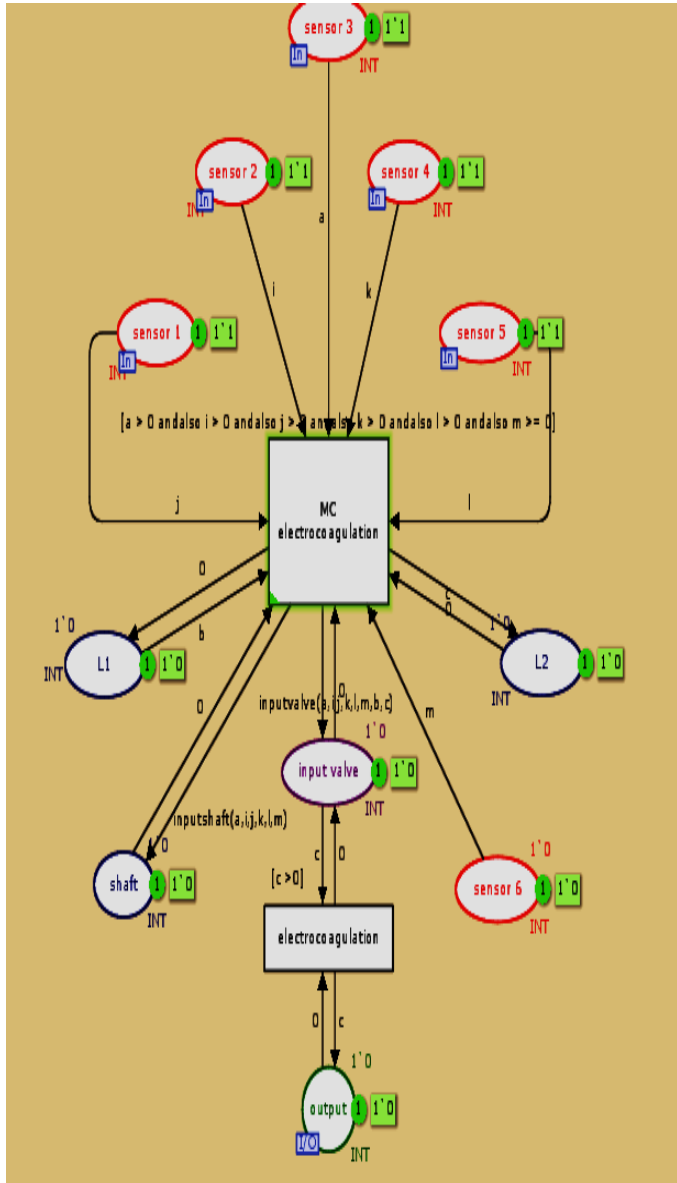


Figure 5.1.1 All 5 sensors test positive for oil except sensor 6 and shaft position is 0

5.2 AFTER SIMULATION :

5.2.1 MODEL SCHEMA:

Shaft Position : 1
 Inference: Shaft position changes from 0 to 1 and new position is assigned value 0 again. Though the shaft position changes to 1, till sensor 6 sends a new input as positive the input valve does not open.

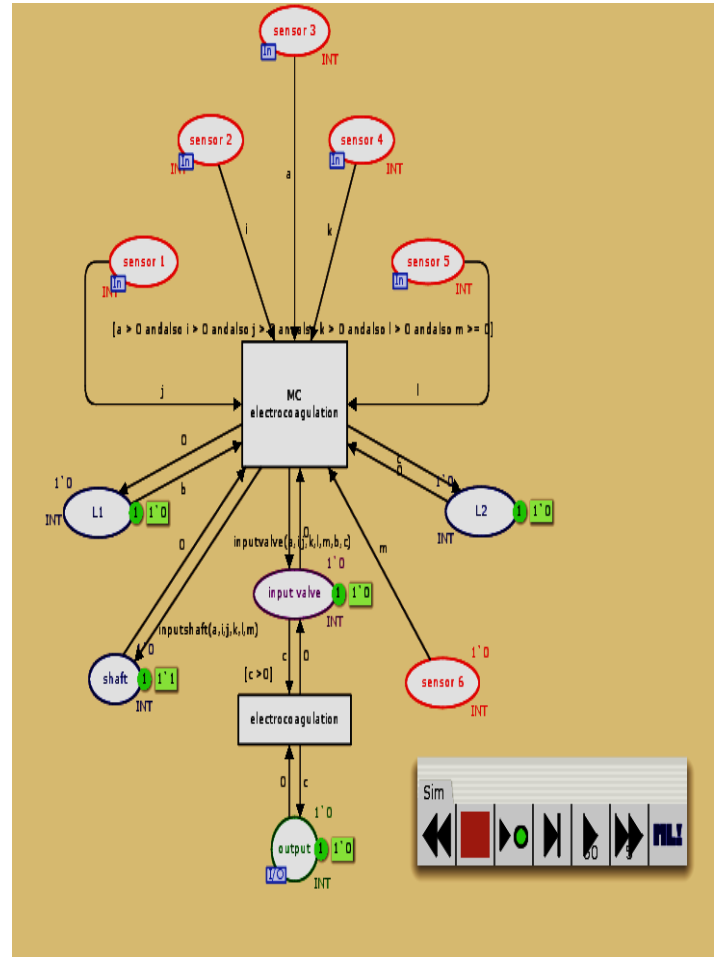


Figure 5.2.1 The simulation yields the shaft position to change

CASE 6: When no oil sensors detect oil :

In this case there is no oil in the immediate vicinity of the mobile device. There are 2 approaches to aid the device in such a situation.

One approach is to make the mobile device to trace circles of increasing diameters until it interacts with oil after which the movement via sensors is guided. The movement will be achieved by switching off one propeller (taking the left propeller) and keeping the main propeller ON. This mode can also be understood as ‘scouting’ for oil.

An alternate method is the use of thermal imaging to determine the concentrated oil puddles and directing the mobile device in that direction. Thermal imaging can be utilized due to the unique property of oil that it heats up rapidly during the day and cools off in the night. Hence at all points of time oil will have a different heat signature from water. By setting up a threshold algorithm to differentiate the pixels in greyscale the position of oil can be determined. The

thermal camera can be installed at a height at an inclination so that it is facing the water body. The thermal camera will operate within a range, the camera will be installed on a base connected to a stepper motor having a step angle of 900, such a step angle is kept due to the high amount of data that needs to be processed, we can have a smaller step angle and operate at a higher accuracy but it would need higher computation and data storage. The implementation of thermal imaging with relevant data on setting threshold has not been discussed in the paper.

Arc inscription code

Navigation:

```
Propeller-main
fun propellermain(a,i,j,k,l) =
if(a > 0 andalso i > 0 andalso j > 0 andalso k > 0 andalso l > 0 )
then l`0
else if(a > 0 andalso i > 0 andalso j > 0 andalso k < 1 andalso l
< 1) then l`500
else if (a > 0 andalso i < 1 andalso j < 1 andalso k > 0 andalso
l > 0 ) then l`500
else l`1000
Propeller -left
fun propellerleft(a,i,j,k,l) =
if(a > 0 andalso i > 0 andalso j > 0 andalso k > 0 andalso l > 0 )
then l`0
else if(a > 0 andalso i > 0 andalso j > 0 andalso k < 1 andalso l
< 1) then l`0
else if (a > 0 andalso i < 1 andalso j < 1 andalso k > 0 andalso
l > 0 ) then l`500
else l`1000
Propeller- right
fun propellerright(a,i,j,k,l) =
if(a > 0 andalso i > 0 andalso j > 0 andalso k > 0 andalso l > 0 )
then l`0
else if(a > 0 andalso k > 0 andalso l > 0 andalso i < 1 andalso j
< 1) then l`0
else if (a > 0 andalso i < 1 andalso j < 1 andalso k > 0 andalso
l > 0 ) then l`500
else l`1000
```

In the ever changing trend of modern day society, there develops a need to bridge the gap between user interface and direct implementation into hard drive. CPN thus offers a much simpler, more dynamic interface with powerful primitives that enable parallel processing and high throughputness. The system that has been modelled is aimed at cleaning oil-spills but can also be used to clean still water bodies against eutrophication and other pollutants. The paper describes the structure and concepts of mobile surface water filtration system but not that of process detail and power management, as they are not necessary for simulation. The impending release of a compiler that can generate hexadecimal code from CPN xml code can facilitate easier code generation from a dynamic and user friendly tool such as CPN. Also the future

design of the paper could incorporate time dependency to make the simulation more real-time oriented.

ACKNOWLEDGEMENTS

The authors would like to thank the departments of Instrumentation and Control Engineering and Bioinformatics Engineering of SRM University, Chennai for their support and aid in making this paper possible.

REFERENCES

- [1]. 'Mobile surface water filtration system'- Aashish Vatsyayan, B. Hemalatha, IJET
- [2]. OIL SPILL SENSOR USING MULTISPECTRAL INFRARED IMAGING VIA `1 MINIMIZATION, Yingying Liy_, Wei-Chuan Shihy, Zhu Hany, and Wotao Yin_yElectrical & Computer Engineering Department, University of Houston_Computational and Applied Mathematics, Rice University
- [3]. An Introduction to the Practical Use of Coloured Petri Nets, Kurt Jensen, Department of Computer Science, University of Aarhus, Ny Munkegade, Bldg. 540, DK-8000 Aarhus C, Denmark