Annals of DAAAM for 2012 & Proceedings of the 23rd International DAAAM Symposium, Volume 23, No.1, ISSN 2304-1382
ISBN 978-3-901509-91-9, CDROM version, Ed. B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2012
Make Harmony between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals & Proceedings of DAAAM International 2012

MODELING OF MOBILE ROBOT BEHAVIOR FOR LINE-LESS BIONIC ASSEMBLY SYSTEM

**KUKUSHKIN, I[lya] K[onstantinovich]; KATALINIC, B[ranko]; CESAREC, P[aulina];
ZDYB, D[ariusz] & KETTLER, R[oman]**

Abstract: *This paper presents a technical report from the continuous research of IMS Group from Vienna University of Technology, in a field of the development and implementation of the next generation of assembling systems. Main focus of this paper is a presentation of working scenarios and mobile robot working algorithms in Bionic Assembly System and their realization in AnyLogic software.*

Keywords: *modeling, mobile robot, Bionic Assembly System, AnyLogic software, self-organization*

1. INTRODUCTION

Paper presents an example of modeling a lineless assembly system with a self-organizing behavior, called Bionic Assembly System (BAS). The description of working scenarios and strategies of the system and its control structure is shown in [2]; possible reconfigurations within the system in [3] and basic layout and functions and are described in [5] Current paper is focused on a results of continuous research of IMS Group from Vienna University of Technology, in a field of the development and implementation of the next generation of assembling systems.

The purpose of modeling of BAS is to get understanding the behavior of the system and/or evaluate various strategies for the operation of the system. Using modeling, it is possible to observe the system, which is not implemented for some reasons (technological, financial, etc.) and compare it with an existing ones. The existing parameters of model are shown in Tab.1.

Criteria	Parameters
Time influence	Static model
	Dynamic model
Changes in state of the system	Discrete model
	Continuous model
Input data	Deterministic model
	Stochastic model
Modeling approach	System Dynamic
	Discrete event modeling
	Agent based modeling
	Dynamic systems

Tab. 1. Parameters of the model

Dynamic models depend on time and static models are the ones, where the time plays no role. State of the continuous models changes over time and in discrete ones, only at separated points of time. From the input date view, there are deterministic models, which use only pre-defined data, and stochastic ones, operating with a random data. Modeling approach is defined according to the input parameters of the system, number of elements within this system, and number of interconnections between them.

2. BAS WORKING SCENARIO

The main aim of planning BAS work is to achieve the highest possible productivity of BAS during the assembly of one unlimited sequence of orders. Maximal productivity means maximal number of assembled product during one particular period of time, taking into consideration the external priority of BAS orders, system's bottle-necks, limitations in the number of production facilities, and the limited capacity of each essential production unit. It is only possible to achieve the above-mentioned aim through the carrying out of all activities which are placed on the critical path in as short a time as possible. The work of assembly stations, mobile robots and operators has to be simultaneous and synchronized, based on the chosen BAS working scenario.

Each single assembly module or assembly station has two-communication channels one vertical to BAS central computer and other horizontal to the mobile robots. Main tasks of central computer of BAS are to plane the global production of BAS, synchronize the part supply and setups, bring the demands from factory level, and organize the BAS as an integral part of factory. The horizontal communication between assembly station and mobile robot with the assembly pallet which carry one particular product from one to other assembly station in the search for the assembly station which can complete the next assembly operation is kernel part of self-organization of systems.

The assembly pallets are transported through the assembly system by lineless mobile robots. After each assembly operation the assembly station makes the quality check to find out, was the assembly operation completed successfully if yes the assembly station give this information to the mobile robot which carry the product on the assembly pallet during the process of assembly.

This information has key role in the search for the station that can carry out the next assembly operation on the product of this type.

The horizontal communication between the control system of one assembly unit and the mobile robot includes following information: pallets type, palettes status, product type, assembly stage of product - which is next assembly operation on that product which has to be done, quality status of product - was the last assembly operation completed successfully. If the last assembly operation was not successful the quality status of product will be negative, and all assembly units will tell that they are not responsible for next operation. For such cases is in the system organized special repairing station. At this place the robots/pallets are waiting on the shop operators which will try to correct the part. If he cannot correct the mistake, he will move the product from the pallet and reset the pallet and send it to the system as new pallet free to take the first part of next product. After the product successfully completed all assembly operations and tests he will be removed from the pallet and packed for transport. The robot/pallet will be reset and send as the free robot/pallet back to the system.

4. BAS CYCLE

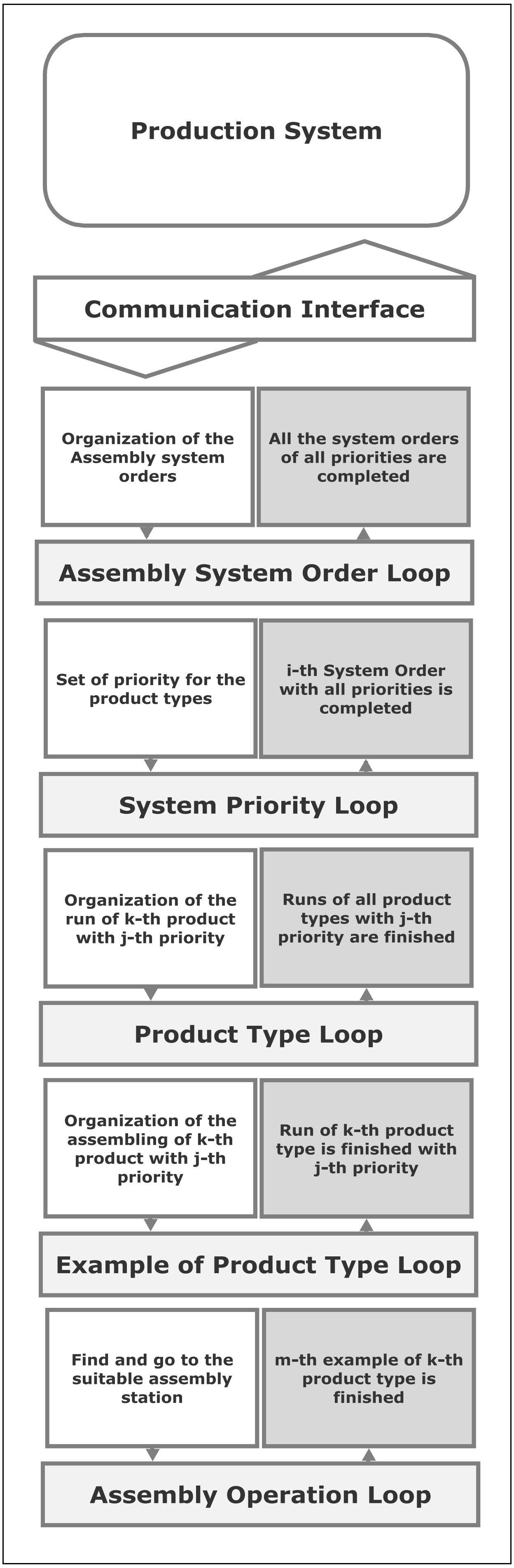
BAS is a part of the production system, so there is a communication between it and other parts of the system. It is organized through the communication interface. Orders from an upper part are injected through this interface to BAS.

Logic in BAS is organized within a five different loops, shown at Fig.1. In the first loop, Assembly System Loop, all coming orders are rearranged to a local assembly orders, to get a higher assembly stations utilization. Each assembly system order includes following data:

- number of types of the products
- priority for each of the product types
- number of the pieces of the products within the product type
- Number of assembly operations within the assembling of one example of one product

In the second, System Priority Loop, orders get a priority. There are 3 priorities within the system.

- High (1). Products with this priority will be assembled in the shortest period of time
- Normal (2). Products with this priority will be assembled in case, when the there are no products with the higher priority in front of the assembly stations at the same time
- Low (3). Products with this priority would be assembled only when there are no products with the high or normal priority in front of the assembly stations at the same time



Production System
Communication Interface
Organization of the Assembly system orders
All the system orders of all priorities are completed
Assembly System Order Loop
Set of priority for the product types
i-th System Order with all priorities is completed
System Priority Loop
Organization of the run of k-th product with j-th priority
Runs of all product types with j-th priority are finished
Product Type Loop
Organization of the assembling of k-th product with j-th priority
Run of k-th product type is finished with j-th priority
Example of Product Type Loop
Find and go to the suitable assembly station
m-th example of k-th product type is finished
Assembly Operation Loop

Fig. 1. Working cycle of BAS System

In the third loop, Product Type Loop is given the amount of the pieces of the products for each product type. And in the loop of Assembly Operation for the mobile robot is given a task with an amount of assembly operations to fulfill one example of product type with the priority within the assembly system order.

From i=1 till the last system order BAS starts to fulfill the products with the j-th priority till j=3, the lowest priority.

For each priority system starts to make the product runs for the k-th type of product, where k varies from 1 till the last set product. For run of each product system assembles the l-th example, where the first example till l_{RUN} example of k-th type of product.

Assembling of l-th product consists of a repetition of m_{LAST} assembly operations. When all the assembly operations on the l-th product of k-th type with j-th priority within the i-th system order are finished, system starts to assemble the next product till the end of the product run of k-th type. After run of k-th product is finished system is completing the other runs, till all the runs with j-th priority are finished within one system order.

4. MODELING OF MOBILE ROBOT IN BAS

First step in the creation of a model for dynamic system is a determination of a model type. In case of a BAS it is a dynamic continuous model using a stochastic data. For modeling of self-organization there is a need of a combination of model approaches. From one side it will be agent-based model, because of the big number of acting elements inside the system (mobile robots, assembly stations, system orders, etc.). From another side to set the algorithms for the working elements there is a need to use discrete event modeling approach. AnyLogic also used as tool for a supply chain management [2], which allows easier connectivity with a logistics organization and integration it to a production system.

To combine more modeling approaches inside one model, was used AnyLogic software from XjTech company. Main part of the model is created in AnyLogic, but some elements are written in Java and integrated to a model. There are 3 agent classes realized for a BAS modeling in AnyLogic: *robot, assembly station* and *customer order*. In this paper will be described *robot* class.

Mobiles robots are core elements in BAS. Their behavior represents self-organization inside the system. Mobile robots are following one scenario but according to the situation in the system could change it.

The main idea inside the robot algorithm is time minimization of one assembly step. For that, robot must find all the suitable working stations for the assembly step and make the decision to get the shortest time of the assembly operation according to (1)

$$T_{as} = T_{op}+T_w+T_m \qquad (1)$$

Where T_{as} is total assembling time, T_{op} –time of assembling operation on the assembling station, T_w –time of the waiting in the queue in front of this station and *Tm* is a time of movement to this station. When the station is found, robot moves there and makes the assembly operation.

Robot class in AnyLogic represents mobile robots behavior. The realization of robot class is based on the state charts of robots and the transitions between them. The logic of a robot states is based on the algorithms described in [4], shown in a Fig. 3. There are following rules for mobile robot behavior exist:

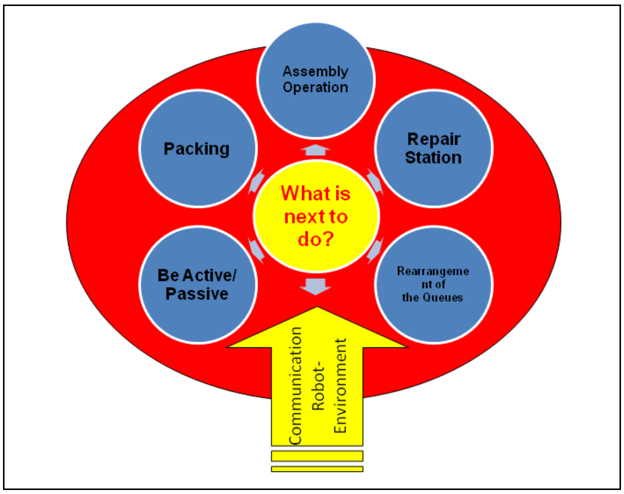


Fig. 3. Logic of mobile robot behavior

- If the quality state of product is positive and the next operation is assembly operation, mobile robot checks which assembly station(s) can perform this operation. If there are more than one, robot will count the best option according to existing queues and priorities
- If the quality state of product is negative, then the robot has to go the repair station and wait for the shop floor operator. If it will be not possible to repair the product, it will be removed from the system and robot will be reset manually.
- If the mobile robot is present at the assembly station and the assembly station is free and there are no robot(s) with higher priority in the queue, it starts docking, executes assembly operation, checks the quality of results of the assembly operation, writes the new state of assembly and the quality state of product, makes undocking and goes to the state of a next operation.
- If the mobile robot is present at the assembly station and the assembly station is busy or there are waiting robot(s) with equal or higher priority or there are robot(s) of equal priority which are waiting for longer time, the mobile robot has to wait in the queue of the assembly station for the next operation.
- If an assembly station within the system becomes active or passive, mobile robot checks the possibility

to shorter its order time by changing the station. If the time to finish the order at another station will shorter than current, then robot will start to move to the alternative station. This procedure is called the rearrangement of the queues of alternative assembly stations.

- If the next step of assembly is packing, mobile robot has to go to the loading/unloading station, where it will get the new assembly order.

Mobile robot state chart realization is presented at Fig. 3. and its dynamical representation according to the priority of its order is shown at Fig. 4.

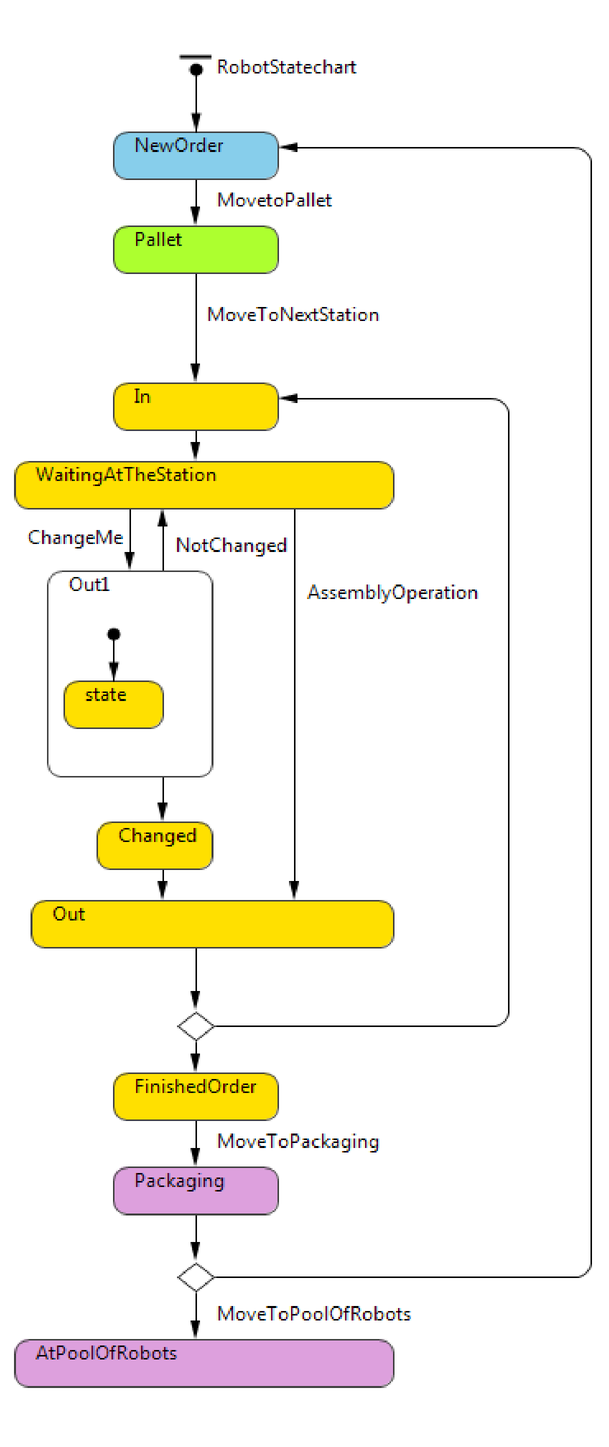


Fig. 3. State chart of a mobile robot (AnyLogic)

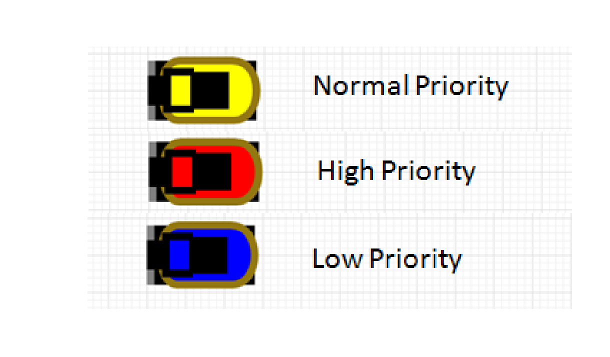
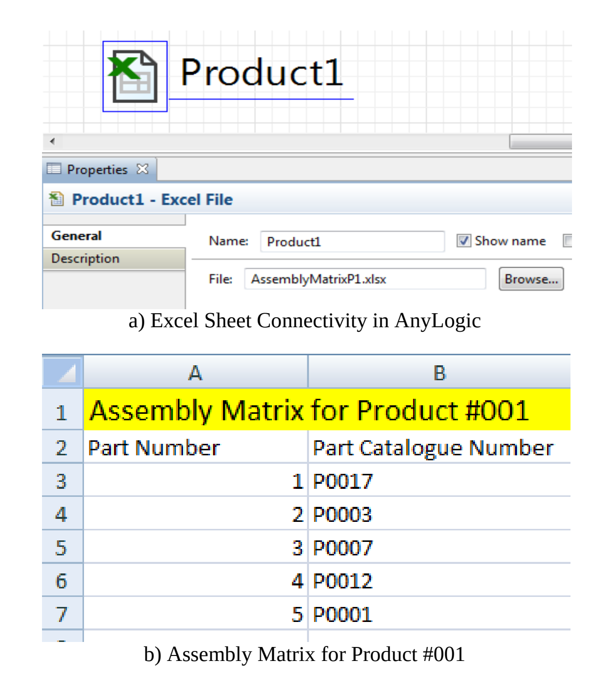


Fig. 4. Mobile Robot Representation in AnyLogic

Each mobile robot after switch on has to take an assembly order for one piece of product. Information about this order is stored in Assembly Matrix. In Anylogic model it is realized using the possibility to read data from the .xlsx sheet. The example of Assembly Matrix is shown in Fig. 5.



Product1 - Excel File

Name: Product1 ☑ Show name

File: AssemblyMatrixP1.xlsx

a) Excel Sheet Connectivity in AnyLogic

	A	B
1	Assembly Matrix for Product #001	
2	Part Number	Part Catalogue Number
3	1	P0017
4	2	P0003
5	3	P0007
6	4	P0012
7	5	P0001

b) Assembly Matrix for Product #001

Fig.5. Assembly Matrix in Excel and AnyLogic Connectivity

When Assembly Matrix is chosen, mobile robot takes a suitable pallet in the pool of pallets and starts to search the parts from 1 till the last part into the assembly matrix. At first, it looks for the stations which are able to assemble this part, and then, chooses the station with a minimal *Tas (see (1))* and drives to this station. Anylogic realization of a *GetStation* function is shown in a Tab. 2.

When robot is at the station, and there are robots with the same or higher priority in front, it has to wait in the queue (WaitingAtTheStation state in Fig.3). It has the possibility to optimize his assembly step with choosing another station in case of a new opened station, or a problem on chosen one (*ChangeMe* transition in Fig.3).

Then, function *GetStation* is repeated, and if the station with the lower *Tas* is found, follows a *Changed* state, if not the transition *NotChanged* is executed. Algorithm of a queues reconfiguration is described in [3].

<table>
<tr><td colspan="2">GetStation Arguments</td></tr>
<tr><td>Name</td><td>Type</td></tr>
<tr><td>Type
Prior
Robot-Name</td><td>String
String
int</td></tr>
<tr><td colspan="2">GetStation Body</td></tr>
<tr><td>Return Type</td><td>int</td></tr>
<tr><td>FunctionBody</td><td>int k=0
double time=1000000;
for (int i=0;i<stations.size();i++)
{
if ((Type.equals(stations.get(i).partA))||
  (Type.equals(stations.get(i).partB))||
  (Type.equals(stations.get(i).partC)))
  {if ((stations.get(i).timetofinishA!=infinity)||
    (stations.get(i).timetofinishB!=infinity)||
    (stations.get(i).timetofinishC!=infinity))
    {if (stations.get(i).TimeToOut
      (Prior,RobotName)<time)
      {time = stations.get(i).TimeToOut
      (Prior,RobotName);
      k=i;
      }
    }
  }
}
robot.get(GetNumberFromPar(RobotName,robot))
.StTimeToFinish = time+time();
return k;</td></tr>
</table>

Tab. 2. Description and Java realization of a GetStation function of mobile robot in AnyLogic

In the case of introduce of new stations it is necessary to rearrange the queue from the robots, waiting in front of the other station, as shown in (3). Java realization in AnyLogic is shown in Fig. 6.

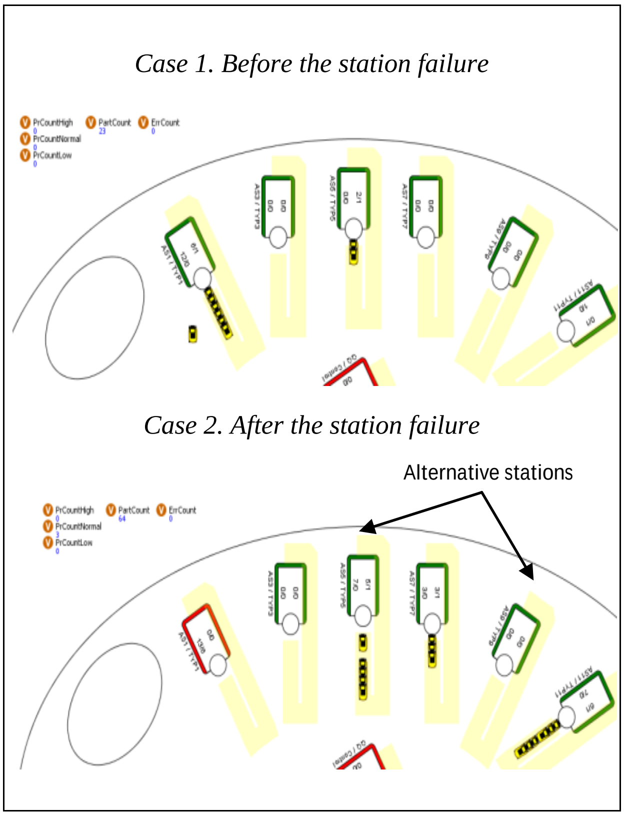


Fig. 6. System Reconfiguration after a station failure

After all the assembly steps are made and there is no more stations in assembly matrix, mobile robot drives to packaging. After package is completed, robot checks, if there more orders available. If yes, robot returns to a state new order. If there are no more available orders, robot returns to a pool of robots and hibernates.

5. CASE STUDY

Bionic Assembly System has a higher flexibility as an traditional assembly systems. Its design allows the higher utilization of assembly stations. There are different ways to increase the utilization and reduce the assembling time of one run of products.

One of this ways is to increase the number of mobile robots in the system. This increase will give effect till certain point called saturation point. After that point, time of one run will stay on a following point and will start to increase again.

AnyLogic model is used to find the saturation point for a defined product run time minimization. An experiment was set with 14 assembly stations, assembling one run of 500 products consisting from 5 parts. There are following results obtained:

On interval 1-5 robots time decreases almost linearly, as shown in Fig. 7. The ground for this is that *Tw* on this interval is incomparably less than production time and system orders are fulfilled in parallel on the alternative stations.

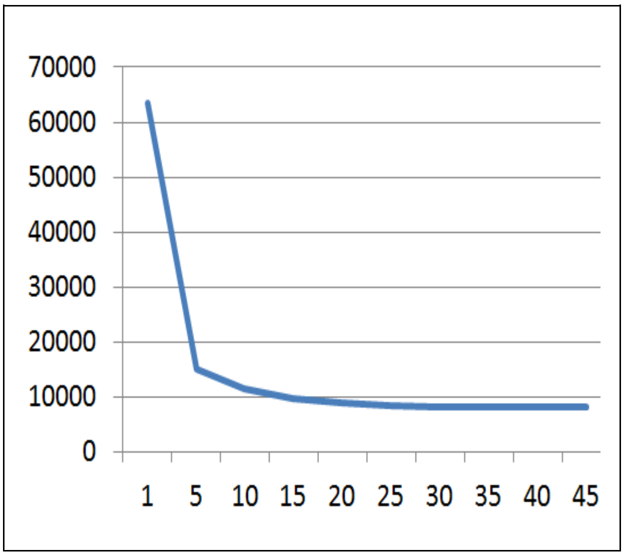


Fig. 7. Diagram of a assembling time, depending on the number of mobile robots within the system

On the second interval (5-35) the time of assembling one run of products decreases exponentially till the saturation point of 35 robots. Number of robots comes to a saturation point and assembling time for one product run reaches minimum of 8190 seconds as shown in Fig. 8.

At following point there is no more flexibility of change the alternative station, because of the equal and comparable to an assembling time *Tw*

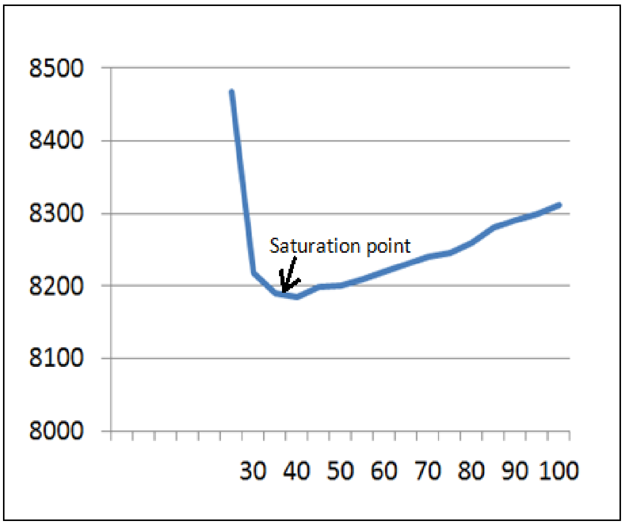


Fig. 8. Diagram of a assembling time, depending on the number of mobile robots within the system

On the third interval, after a saturation point of 35 robots the time of one run production starts to increase slowly (Fig. 8). This phenomena grounds on an increase of a waiting time (*Tw*) in the queues in front of assembly stations. For each run of products a number of robots should be optimized according to the parameters of the system.

Results, shown by simulation show that time reduction is possible with a mobile robots quantity increase, but only until the saturation point. Saturation point for a number of mobile robots in BAS depends on following factors:

- Number of parts in the assembled product
- Number of stations in the system
- Number of products within one run
- Productivity of one assembly stations

These factors can be defined and changed within modeling of the system. Influence of those factors will be studied in the following research.

4. CONCLUSION

Current paper is focused on a results of continuous research of IMS Group from Vienna University of Technology, in a field of the development and implementation of the next generation of assembling systems. Paper presents a description, behavior rules and state chart algorithm for modeling of a mobile robots in Bionic Assembly System (BAS).

AnyLogic software is a most suitable product for a modeling of self-organizing assembly systems, because of the possibilities to combine a agent-based and discrete event modeling approaches inside of one model. Flexibility of used software allows to model either robots on a lower level, or control system for a order on a higher ones. Java engine gives the possibility to export the model and run from any browser without the software installation. This gives the flexibility to get a faster calculation using parallel computers.

Increase of a number of mobile robots is one of the different methods of decrease of assembly time within BAS. This method is applicable only till saturation point, which depend on a number of system parameters. This point is different for each case, and optimal number of mobile robots could be found by system simulation.

As a future for a research is considered to cover an absence of a actual data about real mobile robots and assembly stations parameters. After the comparison with a real flexible assembling system this problem will be solved. In simulation it is not possible to model all the factors, influencing on the real system. This means, that the results from the simulation model will be acceptable only till some level and not 100% comparable with a real system.

The next step of a research will be focused on a description of assembly stations algorithms, modeling of a control structure and presentation of modeling results, comparing to an existing assembly systems.

5. ACKNOWLEDGEMENTS

This paper is supported by the Erasmus Mundus Action 2 Programme of the European Union.

6. REFERENCES

[1] Ivanov, D., Sokolov B., Kaeschel J. (2010) A multi-structural framework for adaptive supply chain planning and operations control with structure dynamics considerations, in: European Journal of Operational Research, 200(2), 2010, pp. 409-420

[2] *** (2012) Weigl, K.H., Lecture Notes to a University Course 307.446 Simulation of Transport Systems SS2012, Vienna University of Technology, Vienna, 2012

[3] Kukushkin, I. K.; Katalinic, B.; Cesarec, P. & Kettler, R.: Reconfiguration in Self-Organizing Systems, Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium, ISBN 978-3-901509-83-4, ISSN 1726-9679, pp 0641-0642, Editor B[ranko] Katalinic, Published by DAAAM International, Vienna, Austria 2011

[4] Katalinic, B., Kukushkin, I. K., Cesarec, P & Kettler, R., Hybrid Control Structure and Scheduling of Bionic Assembly System, Proceedings of the 8th International Conference of DAAAM Baltic, Industrial Engineering, 19-21 April 2012, pp 483 – 489, ed. T. Otto, Tallinn, Estonia, 2012

[5] Katalinic, B., Kordic, V. (2002). Concept of Design and Scheduling of Self-Organizing Complex Flexible Assembly System, Proceedings of the 4th International Workshop on Emergent Synthesis – IWES 02, ISBN 961-4238-49-3, pp. 89–96, March 12-13, 2002, Kobe, Japan

[6] V. Pyanichnikov, A. Platonov and B. Katalinic, Supervisory Group Control of Mobile Technological Robots, Report at the 1st Russian-German Seminar on Space Robotics, Stuttgart, 2012-02-20, Karlsruhe Institute of Technology and German Airspace Academy, Stuttgart, 2012

[7] Tomomi Kito and Kanji Ueda, *Introducing Bounded Rationality into Self-Organization-Based Semiconductor Manufacturing*, 2008, Part 2, 65-73, DOI: 10.1007/978-3-540-76862-3_5, Dynamics in Logistics, Springer, 2008

[8] *** (2012) www.xjtech.com, official website of AnyLogic software, *accessed on 2012-09-10*