Proposal control of manipulators controlled via technological network DeviceNet.

[1]Matej ČOPÍK, [2]Ján Ilkovič

[1]Department of Cybernetics and Artificial Intelligence, FEI TU of Košice, Slovak Republic
[2]Department of Cybernetics and Artificial Intelligence, FEI TU of Košice, Slovak Republic

[1]matej.copik@tuke.sk, [2]jan.ilkovic@tuke.sk

***Abstract* — this paper deals with the proposal for the control of manipulators whose drives are connected to the technological network DeviceNet. These manipulators are integrated in the school model of the flexible manufacturing system. The main objective is to explain the employment of technological networks in the control and implement this knowledge in teaching process.**

***Keywords* — Flexible manufacturing system, DeviceNet, Programmable Logic Controller, Manipulator, Drive**

I. Introduction

In practice, often used interconnection of individual parts of the production through communication networks, because controlled processes are often large and use the network is cheaper and easier. It is the same for manipulators that controlling deals this paper. Manipulators whose controlling will address this paper are integrated in a flexible manufacturing system (FMS), which was created for the purpose of teaching at the Department of Cybernetics and Artificial Intelligence at the Technical University in Košice. This model project was funded by the Centre of Excellence. FMS shown in Fig. 1 is a production system, which includes machinery and equipment. In which are integrated sensors and actuators, some of them communicate using technology network with the Programmable Logic Controller (PLC). One of such network is the technological network DeviceNet. Through this network are connected to each drive manipulator in FMS. FMS consists of two tri-axial manipulators whose two axes are controlled by drives attached to the DeviceNet network. For more detailed description of this model, see [1] and [2].

The first part briefly describes the activities of the FMP which are integrated tri-axial manipulators. This part contains a description of the manipulators drives connected to the DeviceNet network.

The second part deals with the proposal control and configuration technological network DeviceNet, which is connected to the control PLC.

The third section describes the actual proposal of manipulator drive control.

Fig. 1. Model flexible manufacturing system

II. Description of control problems

A. Description FMS and manipulators

Flexible manufacturing system consists of six posts, which are carried out each step of the manufacturing process. The final product is a color image stack in template.

The first post is post implementation templates. This post is a first manipulator with indication MAXP12R-H41BR-C41BR0300. The selected color squares from the small cube trays and store them according to the. The model is entered from the information system (IS), into the template of the size of 5x5 blocks. Movement in X and Z axis provide drives, which have three-phase stepper motors, gearboxes and integrated electronics. These drivers are controlled via technology network DeviceNet. Movement in the Y axis is performed with a pneumatic piston drive.

On the second post is camera control which checks the accuracy stacked picture. After camera control the template for this position moves from one conveyor to the second conveyor.

On the third post is the second tri-axial manipulator with indication MAXR12R-S41BR1000-P41BR0600, which saves the template to rack or in their withdrawal from collecting from the rack. To rack can store up to 28 templates. Drives moving in X and Y axes. The drivers include the three-phase stepper motors, gearboxes and electronics. Drive in the Y axis is equipped with a brake that arm does not fall into the down position when power failure. Also, these drives are connected via technology network DeviceNet. Movement in the Z axis carries pneumatic piston.

On the fourth post is only the transfer between second conveyor and first conveyor.

On the fifth post the templates that are intended for removal, or the wrong stacking, emptied. Templates are emptied by the tipping manipulator, which tilts the template over the cubes container and clear the template.

Sixth post, post sorting cubes are located outside the production cycle. Cubes come from the vibrating container and are sorted into individual small storage cubes, where are then using a manipulator in the position one stored into the template.

Deployment of posts FMS is shown in Fig.2.

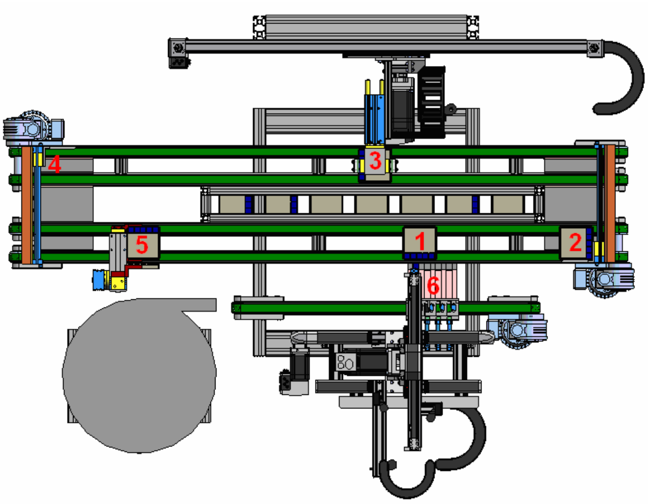

Fig. 2. Deployment of individual posts flexible manufacturing system

B. *Description of the connection drive manipulators*

Individual drives in both the manipulators in terms of control, administrative autonomy, even if they perform parallel operations. Each drive on the DeviceNet network is a one node of network, which communicates via a master module with control PLC. Master module is a scanning module with the type designation 1769-SDN, which is located in the PLC as an expansion card. This arrangement of drives on DeviceNet network is shown in Fig. 3 [3].

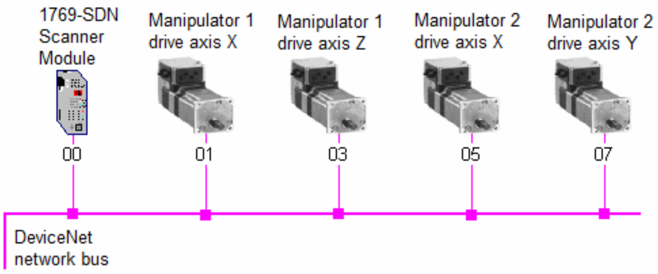

Fig. 3. Connection drivers to the network DeviceNet

III. Control Proposal

A. *DeviceNet network configuration*

Network nodes, thus particular drives, can have arbitrary addresses. The first node of network is DeviceNet scanner module, which also has the first address on the network - 00. Particular network nodes can be numbered from 00 to 63, so it constitutes a possibility to connect 64 nodes to one network. Chosen addresses of particular drives nodes are mentioned in table 1.

TABLE I
TABLE OF ADDRESSES FOR NETWORK NODES

Node number	Description
01	Motor of axis X for manipulator 1
03	Motor of axis Y for manipulator 1
05	Motor of axis X for manipulator 2
07	Motor of axis Y for manipulator 2

For possibility to add new nodes to the network in cases when PVS was modified addresses were defined in such a way as it is shown in Tab. 1. It is also possible to add a new network node in such a way that it is clear which nodes belong together with this numbering. For instance, if it was needed to add a new node into the network on the first post, thus for manipulator 1, it would be possible to use address 02 or 04. All nodes of the first post would be side by side in this case.

Also, it is possible to make network nodes reconfiguration by assignment addresses from already connected nodes. However, this procedure is rather difficult, because it is necessary to set addresses for particular nodes in hardware profile. Concretely for manipulators motors, it can be done by two rotary switches S1 and S2 on each motor. A decimal number of nodes address can be set by switch S1 (MSD) and a unit number of nodes address can be set by switch S2 (LSD). Switches for setting addresses of motor are depicted in Fig. 4. More detailed settings are listed in [4].

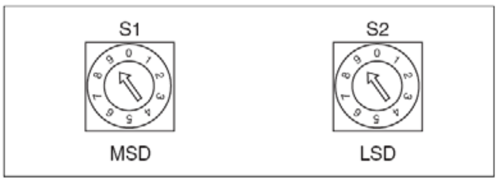

Fig. 4. Setting addresses of motors

B. Basic functions for motors control

Additional functions, which are directly created for development environment RSLogix 5000 by Rockwell Automation Company, are used for this kind of motors control by DeviceNet network. Overall there are 24 created functions, which carry out different functions from single data sending and receiving from and to DeviceNet network (*MC_UpdateInputData_DVN_ILx2D, MC_UpdateOutputData_DVN_ILx2D*), through functions for motors control (*MC_Power_DVN_ILx2D, MC_MoveAbsolute_DVN_ILx2D*), to function for motors diagnostics (*MC_ReadStatus_DVN_ILx2D, MC_ReadParameter_DVN_ILx2D*). All functions for control of drives are listed in [4]. The function MC_*UpdateInputData_DVN_ILx2D* must be carried out as first at control cycle. This function provides messages acquisition, which control manipulators motors from DeviceNet network. Message sending to network is carried out by function *MC_UpdateOutputData_DVN_ILx2D*. It is necessary to put this function as last from added functions at the cycle. Different functions for control or diagnostics can be added between these functions. Every motor has accurately defined an array of variables, which it uses for communication and control. It is necessary to use data structures appertaining to particular motor for its control. All motors have included control electronics at themselves, which makes possible communication by DeviceNet network. It also provides direct connection of some signals (for example: from sensors at and positions).

C. Proposal for the control

The main problem in controlling these types of drives is that when a power failure can not remember your last position. Therefore it is necessary each time you start control program make the initialization at zero or the desired position. In this case, the initialization is made so that both drives the manipulators pass in its extreme positions, where they stop at the end position sensors from that position pass into the desired position. After reaching the desired position, this position set as the default for other movements, to what serves the function *MC_SetPosition_DVN_ILx2D*. After completion of the initialization it can be included in the production process. To initialize the drive position is also possible to use already created function *MC_Home_DVN_ILx2D*, but to use this feature, you must have an end position sensors attached directly to the control unit to drives.

To carry out operations themselves is the creation of appropriate control sequence, which is in the program call in any control cycle and at the beginning of the cycle only changes the parameters of position, since movements drives are made with regard to the starting position.

D. Proposal for the control of manipulator 2

The first manipulator that stores blocks of small storage cubes to the template on the conveyor is shown in Fig. 5.

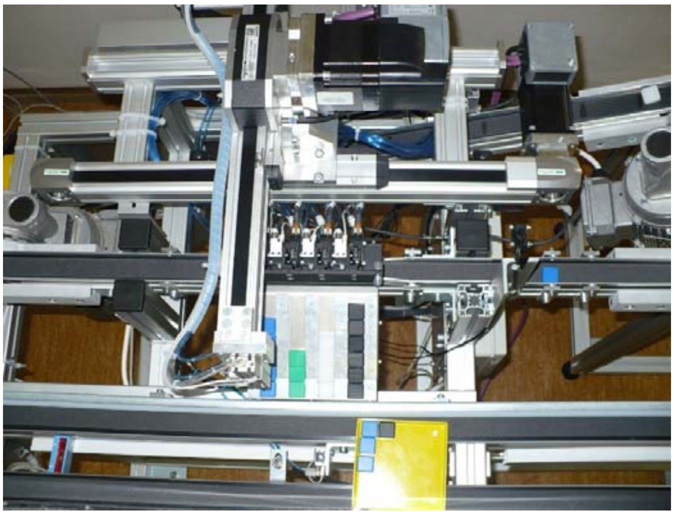

Fig. 5. First manipulator

Information on color picture, which has to be stack in template, is given a request for stack of the information system. Upon arrival requirements for stack resets the counter cubes. Subsequently, progressively from the requirements for stack gradually fetched serial numbers in a palette of cubes with their corresponding color. Progressive loading and storage cubes are made on the basis of two counters. The first is counter selected blocks, to be storage and the second counter is stored cubes those already stored on the template. If the values of these counters are the same, it is a sign of that is to grab a cube from the tray. In that case, the first color is loaded cubes as required. Based on the color of the loaded dice will send the drive in the X axis value on which to move in order to grasp the cube colors. Drive in the Y axis is sent in this case always the same value independent of the color cube, since all the containers have the same position in the axis Y. If the cube is grasped in the manipulator, the counter increments the selected blocks. A different value of the counters is a sign that has a cube that is stored in templates. Counter to the value stored blocks are sent driven in X and Y axis values, which corresponding to the location in the template. After saving the cube into the template increments the counter stored cubes. Cycle of grab and storage cubes is repeated, but with other values, which correspond to the next cube. After stacking all of the cubes changed template status to stack and is sent to the next post.

E. Proposal for the control of manipulator 2

The second manipulator within the production mode takes out empty and compiles full patterns into the rack during production. On the contrary, within discharge mode it takes out full and compiles empty patterns. The second manipulator is shown in Fig.6.

Fig. 6. Second manipulator

During production of new patterns the function for browsing the store is called and it determines where in the store an empty template is located. Based on the position in the store, values to the manipulator drives are sent. After transfer to a given position a template is taken and the manipulator moves it to a precisely defined place on the conveyor. While manipulator is moving towards the conveyor, same positions are sent with the drive. When a production process is completed and the template is supposed to be put in store, the opposite procedure begins. Function for searching an empty place in the store is called and at the same time manipulator moves to a fixed defined position where it takes a template from the conveyor. After taking the template based on a found empty position in the store, appropriate values are sent to drives. These values represent empty place in the store. After putting the full template in the store, this place is assigned a number that corresponds with the bar code of manufactured template.

For discharging, the control has a similar procedure where cubes are emptied from the full template in the store into the vibrating container. Based on the bar code, the template that should be discharged, template position is determined. This position is sent to manipulator drives that are transferred to a given position within a store. It takes a template in given position a puts it on a defined place on the conveyor. It is necessary to place an empty template in the store after discharging. Function for searching the empty place in the store is called and at the same time is the template taken. Values are sent to manipulator drives that correspond with empty place in the store. Subsequently, the template is put in the empty place in the store.

Functions for searching an empty place are called only in cases when we do not know to determine the position otherwise. If a full template is in the store and has a bar code assigned, position in the store can be determined.

Each position in the store corresponds with drives to which corresponding drives should move so that they are at a desired position. These values are kept in the chart and it is from here that corresponding values are sent to drives according to the position in the rack. For example, positions 1 to 7 has the same position in the axis X and only axis Y changes. Similarly, positions 1,8,15 and 21 have the same axis Y and only the axis X changes. It means the corresponding position is made up of seven positions in axis X and four in axis Y according to desired position in the rack.

IV. Conclusion

Realized control does not have to be finished. Further requirements may occur with time and they will have to be implemented in the control process. Various functions for diagnostics or for detection of error conditions can be implemented in the control. Modeling and diagnosis of various control options as well as error conditions are also involved in our dissertations. We want to simulate these conditions and eliminate their weaknesses before the implementation of the control. The aim is to use the given model in teaching process and realizing various proposals for the control of these manipulators.

Acknowledgment

This work is the result of the project implementation: Development of the Center of Information and Communication Technologies for Knowledge Systems (ITMS project code: 26220120030) supported by the Research & Development Operational Program funded by the ERDF.

References

[1] Jadlovský, J. - Laciňák, S. - Chovaňák, J. - Ilkovič, J.: Proposal of distributed control system of flexible production line (in Slovak), In: Cybernetics and Informatics: International Conference: Vyšná Boča, Slovak Republic, 10th-13th February 2010. - Bratislava: STU, 2010. - ISBN 978-80-227-3241-3. - S. 1-6.

[2] Čopík, Matej: Proposal and implementation of flexible production system control. Diploma thesis. Košice: Technical University of Košice, Faculty of Electrical Engineering and Informatics, 2010. 74 s.

[3] Catalog: Elsys, s.r.o., DeviceNet – Omron basic industrial network, AT&P journal, 8/2004.

[4] Catalog: ILS2D DeviceNet, Lexium Integrated Drive, Product manual, V2.01, 11.2008