DEDICATED ROBOT-ROBOT COOPERATION

GRIGORESCU, S.; VATAU, S. & DOBRA, A.

Abstract: The paper presents applications on collaborative robots. The paper objective is position control in the slave robot motion, synchronized to the master robot motion. The cooperation of two heterogeneous industrial robots, working together on parallel trajectories (linear and circular) in the same time implies kinematic control, Cartesian coordinate's transformation and communication between robot controllers. Dedicated software made in Visual Basic environment achieves command on robot-robot cooperation.

Key words: cooperation, synchronization, robots, matrix, task

Authors´ data: Lecturer. Dipl.-Ing. Dr. **Grigorescu**, S[anda]; Assistent. Dipl.-Ing. Dr. **Vatau**, S[teliana]; Lecturer Dipl.-Ing. Dr. **Dobra**, A[ndreea], Politechnica University of Timisoara, bl. M. Viteazul 1, Timisoara, Romania, sgrig@mec.upt.ro, steliana.vatau@mec.upt.ro, andreea.dobra@mec.upt.ro

1. Introduction

Cooperation has been the key to success of most human endeavour; the similar incorporation of cooperation in robotic systems is critical to realize the next generation of systems and applications. Interest in cooperating systems arises when the tasks may be too complex for a single system to accomplish, or when building and using several simple systems can be more flexible, fault-tolerant or cheaper than using a single large system. In recent years, cooperative robots have continued to receive a great deal of attention from both the robotics research community and the robotics industry. The cooperation of multiple manipulators can extend the fields where robots can be used effectively. Synchronization, coordination, and cooperation are intimately linked subjects, and sometimes it used as synonymous to describe the same kind of behaviour, mainly in mechanical systems. Typically robot coordination and cooperation of manipulators (Liu et al., 1997) form important illustrations of the same goal. It is desired that two or more mechanical systems, either identical or different, are asked to work in synchrony. In robot coordination the basic problem is to ascertain synchronous motion of two (or more) robotic systems. This is obviously a control problem where, at least for one of the robots, a suitable feedback controller has to be designed such that this robot (slave) follows the other robot (master). This problem is further complicated by the fact that frequently only position measurements of both master and slave robots are available. This partial access to the state of the system has been the reason for developing model-based observers which are integrated in the feedback control loop. In practice, robot manipulators are equipped with high precision position sensors, such as encoders. Meanwhile new technologies have been design for measuring velocities, e.g. brushless AC motors with digital servo-drivers. Nevertheless such technologies are not yet common in applications. Therefore, velocity measurements are often obtained by means of tachometers which are contaminated by noise. Moreover, velocity sensor equipment is frequently omitted due to the savings in cost, volume and weight that can be obtained. For these reasons, a number of model-based robot control methods have been proposed (Nicosia & Tomei, 1990). In these methods a velocity observer is integrated in the control loop, although exact knowledge of the non-linear robot dynamics is assumed, which in practice is generally not available. To overcome this drawback, robust tracking controllers only based on position measurements have been proposed (Rodriguez-Angeles & Nijmeijer 2001). However, all the mentioned before papers deal with the tracking control problem and not with the robot coordination problem. The problem of coordinating (synchronizing) physical systems can be seen as tracking between two physical systems. Although it seems to be a straightforward extension of classical tracking controllers, this problem implies challenges that are not considered in the design of tracking controllers. Most of the tracking controllers are based on full knowledge of the desired reference to be tracked, and no one predicts what would happen in the case of partial knowledge of the reference signal, or how to deal with it. Cooperative manipulation is an important capability for extending the domain of robotic applications. It allows multiple robots to work together in such way that results a significant increase in their overall effective

workload and workspace. For many tasks, the use of heterogeneous robots is indicated because of the difficulties of constructing a single robot that has the needed size, strength, dexterity, etc. One such application domain is assembly of large-scale structures, such as terrestrial buildings, planetary habitats, or space solar power structures. Such domains need both heavy lifting capabilities, as well as precise, dexterous manipulation to connect parts together. Another application is in industrial field, where is necessary to handling different kind of object (see figure 1).

Fig. 1. Cooperation of two angular robots in the flat glass handling (Novotny & Horak, 2008)

2. System and cooperation task description

The robot cooperation was studied on the two "Eshed Robotec" robots; these are components of a small scale Flexible Manufacturing System (FMS). The structure of the system consists of three stations: the ASRS, the Milling and the Assembly Workstations. The systems layout is presented in figure 2. Notations used in figure 2:1-The Storage Carousel; 2-ER VII Robot; 3-ER VII Controller; 4-Conveyor Pallet; 5-Pallet Stop Stations; 6-Conveyor; 7-Conveyor PLC; 8-Milling Machine; 9-Milling Machine CNC; 10-ER V+ Robot; 11-ER V+ Controller; 12-Station Buffers; 13-PC for Station 2; 14- Central PC; 15- Central Controller; 16-PC for Station 3; 17-Scora ER 14 Robot; 18-Assembly Table; 19-XY Motions Table. The Central and Station PC's carry out different functions, among them is Human-Machine Interface for on-line robotic system visualization, programming and set-up. The control architecture of FMS is hierarchical, the Central Controller achieves the sequences of whole production (sends commands and receives rapports to/from Robot Controller), via RS 232 ports. The two robots have RRRRR joints, but different mechanical structure and different workspace dimensions. The collaboration possibilities between the two robots ER VII and ER V+ include motion synchronisation (the same motion in the same time), cooperation (different motions in the same time for a common goal) and coordination master-slave (master robot's motion followed up by slave robot's

motion). Due to the relative robot installation, the cooperation refers to the large dimensions pieces manipulation.
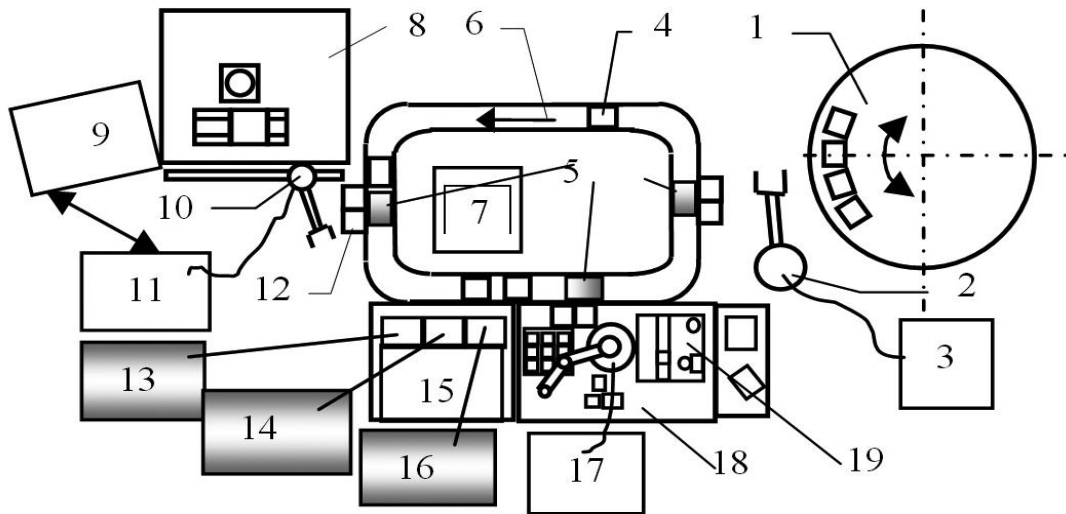

Fig. 2. The Eshed Robotec Systems Layout

The paper's proposed tasks are synchronous motions of ER VII and ER V+ robots, on linear and full circle trajectories.

The ER VII is considered the master robot, with programmed poses, which are transformed in slave robot ERV+ position coordinates. The motion speed on both robots is constant and the programmed values are determined experimentally. The motion instructions used for this application is MOVED <pos> <time>. The motion speeds are constant and are determined in internal routine, as well as the time value for each motion is the same, on both robots.

3. Application algorithms and results

The application problems were: on-line robot programming for getting necessary poses coordinates, acquiring the mathematical model of transformation matrix and developing a dedicate software for robots motions synchronisation.

3.1 Measure the tool's length

Measure the base reference system attached to a piece in the workspace of both robots: P1 for ER VII ($^{P1}T_{R1}$) and P2 for ER V+ ($^{R2}T_{P2}$). For this purpose, the precision poses on the pieces P_1 and P_2 are reached with a tip device fixed in the gripper robot, therefore the TCP will move from point T_1 (attached to gripper) in point T_5 (figure 3). The poses coordinates are modified accordingly to the pin length (vectors $\overline{U_i}$ with different directions, same lenght). For the calculation of the unknown tool's length, the new robot's TCP is brought four times to the same point in workspace and robots coordinates are determined and named Poz_i, i=1÷4.

Figure 3 shows the situation of two position vectors of TCP with/without the pin, relative to the tip (fixed point T_5).

636

Given the cartesian coordinates of the robot positions Poz_i, $(i=1 \div 4)$: x_i, y_i, z_i, from Robot Controller, relative to the point T_5, the position vectors of the robots TCP, without the pin, are named $\overline{r_i}$, $i = 1 \div 4$. The vectors $\overline{U_i}$ are the $T_i T_5$, vectors of constant length and different directions, around the point T_5.
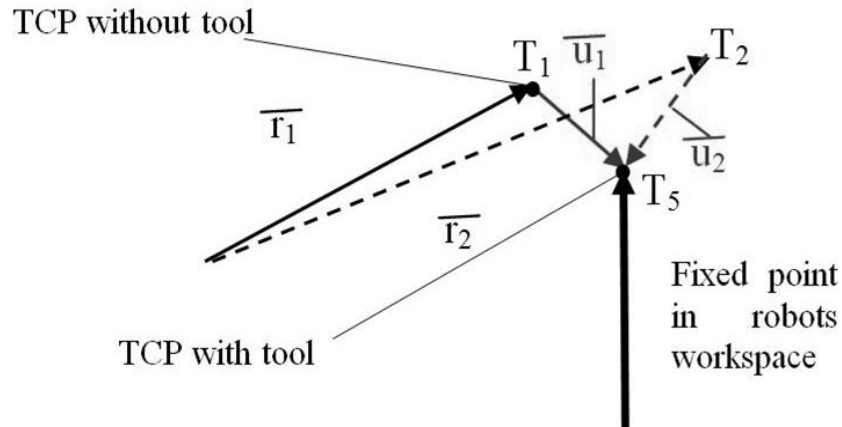

Fig. 3. Vectors scheme for the tool measurement

From figure 3 results:

$$\overline{r_1} + \overline{U_1} = \overline{r_2} + \overline{U_2} = \overline{r_3} + \overline{U_3} = \overline{r_4} + \overline{U_4} \tag{1}$$

$$\left|\overline{U_1}\right| = \left|\overline{U_2}\right| = \left|\overline{U_3}\right| = \left|\overline{U_4}\right| \tag{2}$$

where

$$\overline{r_i} = x_i \overline{i} + y_i \overline{j} + z_i \overline{k}; \overline{U_i} = u_i \overline{i} + v_i \overline{j} + t_i \overline{k}; i = \overline{1 \div 4} \tag{2'}$$

From equations (1) is obtained:

$$(x_1 + u_1)\overline{i} + (y_1 + v_1)\overline{j} + (z_1 + t_1)\overline{k} = (x_2 + u_2)\overline{i} + (y_2 + v_2)\overline{j} + (z_2 + t_2)\overline{k}$$
$$(x_1 + u_1)\overline{i} + (y_1 + v_1)\overline{j} + (z_1 + t_1)\overline{k} = (x_3 + u_3)\overline{i} + (y_3 + v_3)\overline{j} + (z_3 + t_3)\overline{k} \tag{3}$$
$$(x_1 + u_1)\overline{i} + (y_1 + v_1)\overline{j} + (z_1 + t_1)\overline{k} = (x_4 + u_4)\overline{i} + (y_4 + v_4)\overline{j} + (z_4 + t_4)\overline{k}$$

From equations (2) is achieved :

$$u_1^2 + v_1^2 + t_1^2 = u_2^2 + v_2^2 + t_2^2$$
$$u_1^2 + v_1^2 + t_1^2 = u_3^2 + v_3^2 + t_3^2 \tag{4}$$
$$u_1^2 + v_1^2 + t_1^2 = u_4^2 + v_4^2 + t_4^2$$

In conclusion, there are 12 unknowns (u_i, v_i, t_i) $i=1 \div 4$ and 12 ecuations (3 equations (4) and 9 equations (3)).
From vector equations (3) results:

$$x_1 + u_1 = x_2 + u_2 = x_3 + u_3 = x_4 + u_4$$
$$y_1 + v_1 = y_2 + v_2 = y_3 + v_3 = y_4 + v_4 \qquad (5)$$
$$z_1 + t_1 = z_2 + t_2 = z_3 + t_3 = z_4 + t_4$$

or

$$u_2 = a_1 + u_1; v_2 = b_1 + v_1; t_2 = c_1 + t_1$$
$$u_3 = a_2 + u_1; v_3 = b_2 + v_1; t_3 = c_2 + t_1 \qquad (6)$$
$$u_4 = a_3 + u_1; v_4 = b_3 + v_1; t_4 = c_3 + t_1$$

where

$$\begin{cases} a_1 = x_1 - x_2; a_2 = x_1 - x_3; a_3 = x_1 - x_4 \\ b_1 = y_1 - y_2; b_2 = y_1 - y_3; b_3 = y_1 - y_4 \\ c_1 = z_1 - z_2; c_2 = z_1 - z_3; c_3 = z_1 - z_4 \end{cases} \qquad (7)$$

replacing the unknowns u_i, v_i, t_i ($i = 2 \div 4$) of equations (6) in equations (4):

$$u_1^2 + v_1^2 + t_1^2 = (a_1 + u_1)^2 + (b_1 + v_1)^2 + (c_1 + t_1)^2$$
$$u_1^2 + v_1^2 + t_1^2 = (a_2 + u_1)^2 + (b_2 + v_1)^2 + (c_2 + t_1)^2 \qquad (7')$$
$$u_1^2 + v_1^2 + t_1^2 = (a_3 + u_1)^2 + (b_3 + v_1)^2 + (c_3 + t_1)^2$$

which is equivalent with:

$$a_1 u_1 + b_1 v_1 + c_1 t_1 = -\frac{1}{2} \cdot (a_1^2 + b_1^2 + c_1^2)$$

$$a_2 u_1 + b_2 v_1 + c_2 t_1 = -\frac{1}{2} \cdot (a_2^2 + b_2^2 + c_2^2) \qquad (8)$$

$$a_3 u_1 + b_3 v_1 + c_3 t_1 = -\frac{1}{2} \cdot (a_3^2 + b_3^2 + c_3^2)$$

Relations (8) form a linear system of 3 equations with 3 unknowns, the system is determinated and it is possible to calculate $\overline{U_i}$ and the length of tool, $|\overline{U_i}|, i = 1 \div 4$.

3.2 Measure a piece reference system

Measure the reference system attached to a piece in the workspace of each robot, with respect to the robot base reference system: P1 for ER VII ($^{P1}T_{R1}$) and P2 for ER V+ ($^{R2}T_{P2}$), see figure 5.

The transformation matrix calculations from reference system 2 to reference system 1, is:

$$\underline{^{1}T_{2}} = \begin{bmatrix} \cos(x_{1},x_{2}) & \cos(x_{1},y_{2}) & \cos(x_{1},z_{2}) & ^{1}x_{O2} \\ \cos(y_{1},x_{2}) & \cos(y_{1},y_{2}) & \cos(y_{1},z_{2}) & ^{1}y_{O2} \\ \cos(z_{1},x_{2}) & \cos(z_{1},y_{2}) & \cos(z_{1},z_{2}) & ^{1}z_{O2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

where, for instance, $\cos(x_1, y_2)$ is the director cosine of the angle between the axis x_1 of reference system (1) and axis y_2 of reference system (2), and $^{1}z_{O2}$ is the z coordinate of origin of the reference system (2) versus reference system (1).

Three points: O_2, A and B are necessary and sufficient to determine a tri-orthogonal reference system, as follows:

- O_2 is the origin of the reference system $O_2x_2y_2z_2$,
- the vector $\overline{O_2A}$ determines the direction and sense of axis O_2x_2,
- $\overline{O_2B}$ is a vector in the $O_2x_2y_2$ plane (see figure 4).

Are known the position vectors:

$$\overline{O_1O_2}: \ ^{1}r_{O_2} = X_1 \cdot \bar{i} + Y_1 \cdot \bar{j} + Z_1 \cdot \bar{k}$$

$$\overline{O_1A}: \ ^{1}r_A = X_2 \cdot \bar{i} + Y_2 \cdot \bar{j} + Z_2 \cdot \bar{k}$$

$$\overline{O_1B}: \ ^{1}r_B = X_3 \cdot \bar{i} + Y_3 \cdot \bar{j} + Z_3 \cdot \bar{k}$$

with respect to the reference system $O_1x_1y_1z_1$.
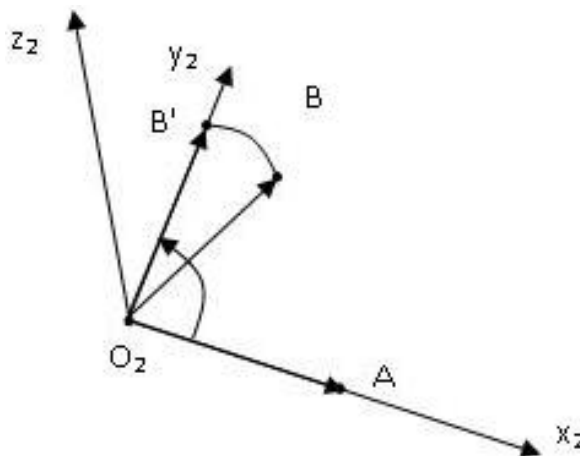

Fig. 4. Three points determination of the reference system $O_2x_2y_2z_2$ (O_2, A, B)

In order to determine the transformation matrix, it was used vectorial calculus, with the following steps:

- calculating the vectors $\overline{O_2A}$ and $\overline{O_2B}$:

$$^{1}\overline{O_2A} = (X_2 - X_1) \cdot \bar{i} + (Y_2 - Y_1) \cdot \bar{j} + (Z_2 - Z_1) \cdot \bar{k} = A_1 \cdot \bar{i} + B_1 \cdot \bar{j} + C_1 \cdot \bar{k} \quad 10)$$

$$^{1}\overline{O_2B} = (X_3 - X_1) \cdot \bar{i} + (Y_3 - Y_1) \cdot \bar{j} + (Z_3 - Z_1) \cdot \bar{k} = A_2 \cdot \bar{i} + B_2 \cdot \bar{j} + C_2 \cdot \bar{k} \quad (11)$$

- the vector $\overline{O_2B}'$ is determined to be perpendicular to the vector $\overline{O_2A}$:

$$^{1}\overline{O_2B'} = A_3 \cdot \bar{i} + B_3 \cdot \bar{j} + C_3 \cdot \bar{k} = (A_2 + \alpha \cdot A_1) \cdot \bar{i} + (B_2 + \alpha \cdot B_1) \cdot \bar{j} + (C_2 + \alpha \cdot C_1) \cdot \bar{k} \tag{12}$$

where

$$\alpha = \frac{-(A_1 \cdot A_2 + B_1 \cdot B_2 + C_1 \cdot C_2)}{A_1^2 + B_1^2 + C_1^2} \tag{13}$$

- check the perpendicular alignment of vectors $\overline{O_2A}$ and $\overline{O_2B}'$:

$$A_1 \cdot A_3 + B_1 \cdot B_3 + C_1 \cdot C_3 = 0 \tag{14}$$

- the determination of vector $\overline{O_2z_2}$:

$${}^1\overline{O_2z_2} = {}^1\overline{O_2A} \times {}^1\overline{O_2B}' = \begin{bmatrix} i & j & k \\ A_1 & B_1 & C_1 \\ A_3 & B_3 & C_3 \end{bmatrix} \tag{15}$$

which is equivalent with:

$$\begin{aligned}
{}^1\overline{O_2z_2} &= A_4 \cdot \overline{i} + B_4 \cdot \overline{j} + C_4 \cdot \overline{k} = \\
&= (B_1 \cdot C_3 - B_3 \cdot C_1) \cdot \overline{i} + (-A_1 \cdot C_3 + A_3 \cdot C_1) \cdot \overline{j} + (A_1 \cdot B_3 - A_3 \cdot B_1) \cdot \overline{k}
\end{aligned} \tag{16}$$

- check the perpendicular alignment of vectors $\overline{O_2A}$ and $\overline{O_2z_2}$:

$$A_1 \cdot A_4 + B_1 \cdot B_4 + C_1 \cdot C_4 = 0 \tag{17}$$

and of vectors $\overline{O_2B}'$ and $\overline{O_2z_2}$:

$$A_3 \cdot A_4 + B_3 \cdot B_4 + C_3 \cdot C_4 = 0 \tag{18}$$

- the director cosine (required in relation (9)) are calculated:

$$\cos(Par_1, x_2) = \cos(O_1Par_1, \overline{O_2A}) = \frac{k_1}{\sqrt{A_1^2 + B_1^2 + C_1^2}}$$

$$\cos(Par_1, y_2) = \cos(O_1Par_1, \overline{O_2B}') = \frac{k_3}{\sqrt{A_3^2 + B_3^2 + C_3^2}}$$

$$\cos(Par_1, z_2) = \cos(O_1Par_1, \overline{O_2z_2}) = \frac{k_4}{\sqrt{A_4^2 + B_4^2 + C_4^2}} \tag{19}$$

where $Par_1 = \{x_1, y_1, z_1\}$, $k = \{a, b, c\}$

3.3 Calculation of the TCP's Cartesian Coordinates

The robot position direct problem suppose the knowledge the motion variables in the robot axes (θ_i), the dimensional parameters of a mechanical robot structure (elements length and offset: offx, offy, lung2, lung3, lung4) and the calculation of the TCP's Cartesian coordinates (x_M, y_M, z_M):

$$x_M = (offx + lung2 \cdot \cos(\theta_2) + lung3 \cdot \cos(\theta_3) + lung4 \cdot \cos(\theta_4)) \cdot \cos(\theta_1)$$

$$y_M = (offx + lung2 \cdot \cos(\theta_2) + lung3 \cdot \cos(\theta_3) + lung4 \cdot \cos(\theta_4)) \cdot \sin(\theta_1) \qquad (20)$$

$$z_M = offz + lung2 \cdot \sin(\theta_2) + lung3 \cdot \sin(\theta_3) + lung4 \cdot \sin(\theta_4)$$

3.4 Finding out the transformation matrix $^{R2}T_{R1}$

Finding out the transformation matrix $^{R2}T_{R1}$ follows the steps:
- measure the tool length (equations $1 \div 8$)
- measure the reference systems P_1 and P_2 (robot poses O_2, A, B)
- calculate the TCP coordinates x_M, y_M, z_M for each of robot poses (with equations 20), where lung4 becomes lung4+$|U_1|$
- calculate the transformation matrix $^{R2}T_{R1} = {}^{R2}T_{P2} \cdot {}^{P2}T_{P1} \cdot {}^{P1}T_{R1}$.

3.5 Finding out the corresponding poses of the slave robot

$$\underline{{}^{R2}r} = \underline{{}^{R2}T_{P2}} \cdot \underline{{}^{P2}T_{P1}} \cdot \underline{{}^{P1}T_{R1}} \cdot \underline{{}^{R1}r} \qquad (21)$$

The relative positions of the "R_1", "P_1", "P_2", "R_2" reference systems are presented in figure 5.
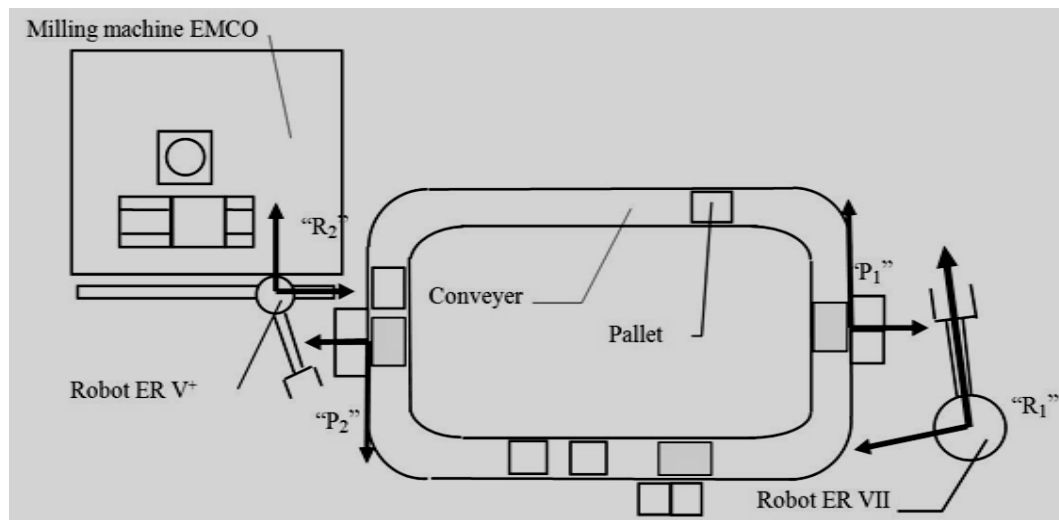

Fig. 5. The relative positions of coordinates systems

3.6 Conceiving a program (Robot-Robot Cooperation) in Visual Basic environment

The program "Robot-Robot Cooperation" in Visual Basic environment (Schneider, 1998) has three main program modules: the first for the determination of the two robots relative positions (steps presented above and equations $1 \div 20$), the second module for computing the transformation coordinates (equation 21) and the third module for serial communication with robot controller (receiving the data from master controller and downloading the data to the slave controller). The input data window of the program is presented in figure 6.

The control system for robot cooperation consists in two robot controllers and a cooperation PC (figure 7). Between the PC and the controllers, the serial communication is realized for sending and receiving data.

3.7 Programming in ACL Language

ACL (robot language) motion programs contain instructions such as MOVED <pos> <time> for PTP motion, MOVELD <pos> <time> for linear and MOVECD <pos> <time> for circular movement. The communication programmes in controllers include the instructions:

> PRLNCOM <n> <arg1> for sending on the COM n the string <arg1>;
> READCOM <n> <var[j]> for receiving on the COM n the string and setting the variable <var[j]> with this value, where j=1÷5.

For the Cartesian Coordinates of slave robots poses, the received values from the cooperation PC are assigned to the robots positions via the instruction SETPVC <pos> x <var[j]>.
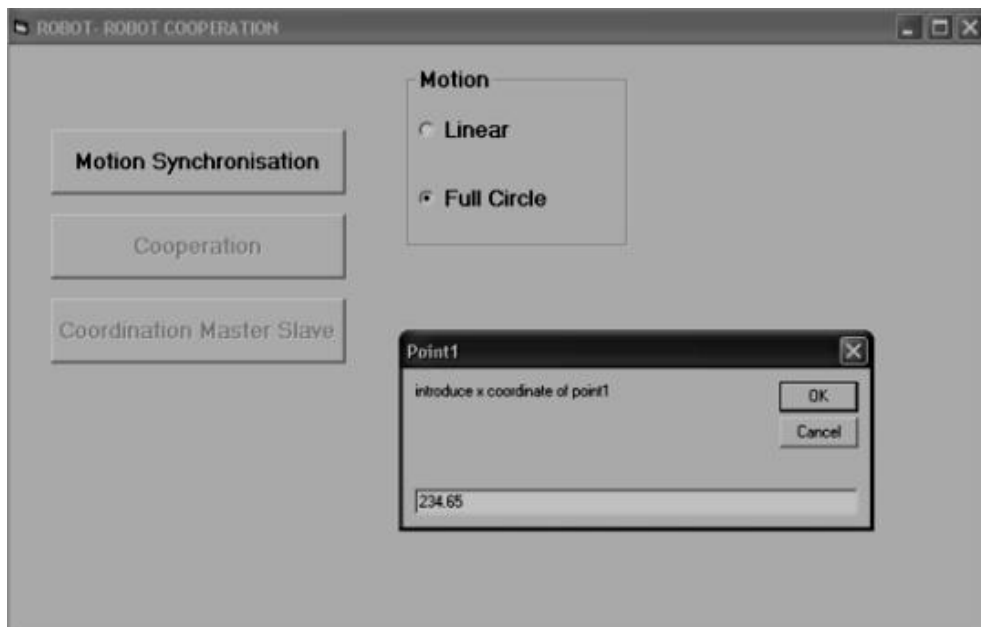

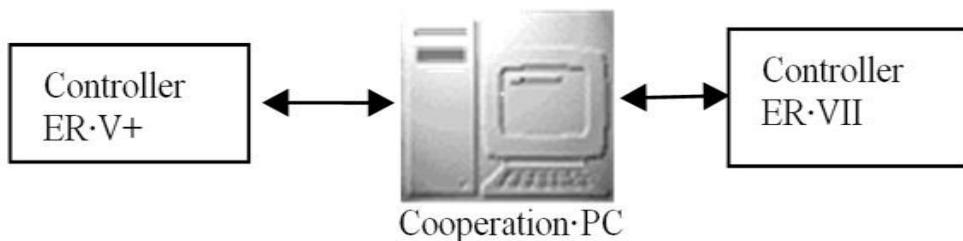Fig. 6. The input data window of Robot-Robot Cooperation program

Fig. 7. The control system of robot cooperation

3.8 Testing the algorithm for different robot master positions and motions and testing the ACL programs

For the full circle motion, there were three teach-in poses, with respect to the master robot world coordinates. The fourth point was calculated in following steps:

- Three points C_k (k=3) determine a plane. A reference system $O_C x_C y_C z_C$ was attached to this plane, in the manner presented in figure 8, the Cartesian coordinates of the points are with respect to the robot reference system.
- The transformation matrix $^C T_{R1}$ is determined with the same algorithm as the measure of the piece (equations 9÷19), where O_2 is $^{R1}C_1$, A is $^{R1}C_2$, and B is $^{R1}C_3$.
- The calculation of the C_k points coordinates with respect to the "C" reference system:

$$\underline{^C r_{C2}} = \underline{^C T_{R1}} \cdot \underline{^{R1} r_{C2}} \; ; \; \underline{^C r_{C3}} = \underline{^C T_{R1}} \cdot \underline{^{R1} r_{C3}} \; ; \; ^C r_{C1} = 0i + 0j + 0k \tag{22}$$
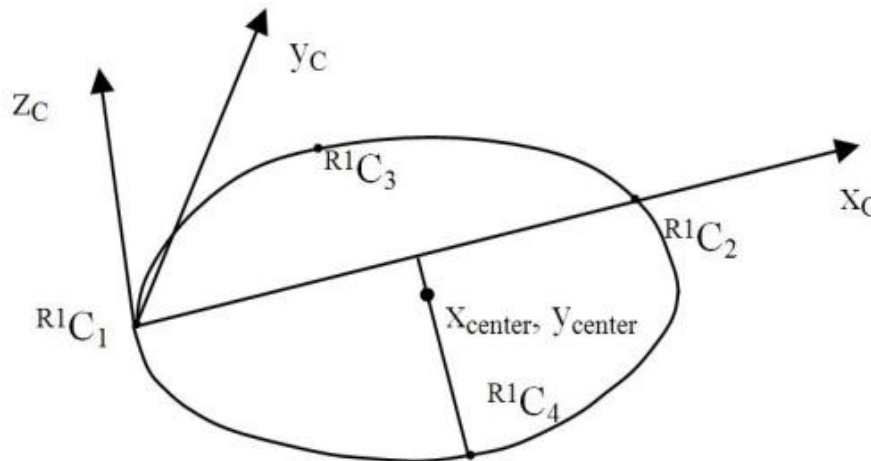

Fig. 8. The plane and the four points on the full circle

- The determination of the circles center point coordinates:

$$x_{center} = \frac{^C x_{C2}}{2}, y_{center} = \frac{1}{2} \cdot \left(^C y_{C2} - \frac{(^C x_{C3} - x_{center})^2 - (^C x_{C2} - x_{center})^2}{^C y_{C2} - ^C y_{C3}} \right) \tag{23}$$

- The calculation of the fourth points coordinates on the circle (figure 8):

$$^C x_{C4} = x_{center}, ^C y_{C4} = y_{center} + r_{circle}, r_{circle} = \sqrt{x^2_{center} + y^2_{center}} \tag{24}$$

- Check the fourth point on circle condition:

$$\begin{vmatrix} (^C x_{C1})^2 + (^C y_{C1})^2 & ^C x_{C1} & ^C y_{C1} & 1 \\ (^C x_{C2})^2 + (^C y_{C2})^2 & ^C x_{C2} & ^C y_{C2} & 1 \\ (^C x_{C3})^2 + (^C y_{C3})^2 & ^C x_{C3} & ^C y_{C3} & 1 \\ (^C x_{C4})^2 + (^C y_{C4})^2 & ^C x_{C4} & ^C y_{C4} & 1 \end{vmatrix} = 0 \tag{25}$$

- The four points coordinates transformation with respect to the R1 reference system and verification that the four points are on the same plane:

$$
\underline{{}^{R1}r_{C4}} = \underline{{}^{R1}T_C} \cdot \underline{{}^{C}r_{C4}} \; ; \;
\begin{vmatrix}
{}^{R1}x_{C1} & {}^{R1}y_{C1} & {}^{R1}z_{C1} & 1 \\
{}^{R1}x_{C2} & {}^{R1}y_{C2} & {}^{R1}z_{C2} & 1 \\
{}^{R1}x_{C3} & {}^{R1}y_{C3} & {}^{R1}z_{C3} & 1 \\
{}^{R1}x_{C4} & {}^{R1}y_{C4} & {}^{R1}z_{C4} & 1
\end{vmatrix} = 0
\tag{26}
$$

4. Conclusion

The program was experimented on the two robots ER VII and ER V+. Firstly, the robot ER VII was on-line programmed for different linear and circular motions. Then the poses coordinates were transferred to the ER V+ robot and the motions synchronisation was reached. The robot ER VII was selected to be the master robot, because the ER V+ is more versatile and the checking on mechanical blockages was already done during the on-line programming of the master robot.

A further development on collaborative robots may include a "slave robot track movement" for the manual movement command of master robot and the force and speed control in both master and slave robot controllers.

5. References

Liu, Y. H.; Arimoto, S.; Parra-Vega, V. & Kitagaki, K. (1997). Decentralized adaptive control of multiple manipulators in cooperations. *International Journal of Control, Volume 67, Number 5, 20 July 1997, pp. 649-674(26)*

Nicosia, S. & Tomei, P. (1990). Robot control by using only joint position measurements. *Automatic Control, IEEE Transactions on Volume 35, Issue 9, Sep 1990, pp. 1058 - 1061*

Novotny, F. & Horak, M. (2008). Parallel Cooperation of Robots during Handling with Jumbo Glass Sheets. *Available from* http://www.scientific.net/0-87849-387-5/465/ *Accessed:* 2008-04-15

Rodriguez-Angeles, A. & Nijmeijer, H. (2001). Coordination of two robot manipulators based on position measurements only. *International Journal of Control, Volume 74, Number 13, pp. 1311 — 1323*

Schneider, D (1998). *An introduction to programming using Visual Basic 6.0,* Prentice Hall, ISBN 3-13-936428-5, USA