Multi-arm robot control system for manipulation of flexible materials in sewing operation

Makoto Kudo\textsuperscript{a}, Yasuo Nasu\textsuperscript{a,}\textsuperscript{*}, Kazuhisa Mitobe\textsuperscript{a}, Branislav Borovac\textsuperscript{b}

\textsuperscript{a}Department of Mechanical Systems Engineering, Yamagata University, Jonan 4-3-16, Yamagata, 992-8510, Japan
\textsuperscript{b}Faculty of Technical Sciences, University of Novi Sad, 21000-Novi Sad, Trg Dositeja Obradovica 6, Yugoslavia

Received 12 January 1999; received in revised form 7 July 1999; accepted 12 July 1999

Abstract

A new automated sewing system is described, consisting of two robots handling the fabric on the table in a similar manner as does a human operator during sewing. To enable user-friendly operation of the system operation, particularly in the phase of preparing new tasks, the original Multi-arm Robot Control (MRC) system has been developed. It incorporates a task-oriented robot language and graphical user interface to enable easy programming of complex motion such as hands coordination during task execution, fabric tension control, and synchronization with the sewing machine speed. To avoid possible collisions, simulation of the already programmed task can be easily performed. The control of hand coordination and the fabric tension has also been developed and implemented. To control seam path and its deviation from the desired trajectory, visual feedback was adopted. Complete system functioning was verified experimentally by sewing. © 2000 Elsevier Science Ltd. All rights reserved.

* Corresponding author. Tel.: +81-238-26-3237; fax: +81-238-26-3205.
E-mail address: nasu@mnsu2.yz.yamagata-u.ac.jp (Y. Nasu).
| Nomenclature |
|--------------|---------------------------------------------------|
| Pi           | point of the sewing trajectory comprising the gripper position and orientation |
| if           | output vector of PID controller                   |
| if_ref       | reference force vector                            |
| f_sensor     | force sensed at the robot wrist, expressed in the reference coordinate frame |
| f'           | force sensed at the robot wrist, expressed in the coordinate frame |
| ip_ref       | reference position vector                         |
| iq           | actual joint angle vector                         |
| iq_ref       | reference joint angle vector                      |
| F_{tension}  | reference fabric tension force                    |
| F_{tension}  | estimated tension force                           |
| iF_{friction}| estimated friction force                          |
| iV           | reference point velocity in the direction of x-axis of coordinate system fixed at the ref. point |
| G_{tension}  | fabric tension feedback gain                      |
| K_1          | rise-time feedback gain                           |
| K_2          | damping gain                                      |
| V_{feed}     | actual sewing speed                               |
| V_{ref}      | reference sewing speed                            |
| d            | estimated seam width error                        |
| z            | angle between the trajectory tangent and the direction of sewing |
| \omega       | angular speed                                     |
| \mu          | friction coefficient                              |
| \Delta x     | compensation of virtual stick length              |
| \Delta t     | sampling period                                   |
| \Delta \theta| angular compensation from visual feedback         |
| iS           | selection matrix                                  |
| iR_{virtual}, iB, iT, iH and iN | homogeneous transformation matrices involved in ip_{ref} evaluation |
| Trans(x, y, z) | translation matrix of the ip_{ref} with respect to the task coordinate system |
| Rpy(a, b, c)  | orientation matrix of the ip_{ref} with respect to the task coordinate system |
| Rot(z, \theta) | rotational transformation matrix about the z-axis |
| \Lambda^{-1} | inverse kinematics function                       |
1. Introduction

Use of limp materials in robotic tasks is difficult because its dynamic behavior is very inconvenient for technical applications. During handling, the fabric can buckle, curl up or fold over, while in sewing, even the appearance of wrinkles, distortion or improper fabric tension can cause serious problems. To overcome these problems, an automated sewing system requires either highly specialized mechanical devices like properly designed grippers for textile handling or complex systems involving multi-arm manipulation with very sophisticated task execution. Most of the papers addressing this issue are dealing with grasping and manipulation of textile ply. Some of the popular fabric grippers based on the use of incisive needles, or combination of air foil, needles and suction has been described in Ref. [1], two-fingered gripper with force-torque sensing in Ref. [2], use of water washable adhesive and electrostatic attraction in Refs. [3–5], vacuum suction in Ref. [6]. Studies of fabric properties have been reported in Refs. [7,8]. In Ref. [9], the robotized handling of flat textile materials (a special gripper with two distant fingers is mounted on a robot) has been reported. A force sensor at the wrist was used to detect the contact of the fingers with the table, maintaining a desired finger pressure against the table, and to restrict force applied during sweeping motion. Also, vision has been extensively used for identification of the material position (edge) on the table and possible material surface deformations (wrinkles and folds), and tracking of the material during handling operations of dragging, folding or sweeping. An automated sewing system has been described in Ref. [10] in which the robot hand was, in fact, a small sewing machine. Robotic sewing using a single-arm robot has been reported in Refs. [11,12] and an edge seam task was performed on the different fabric types. The system incorporated the end-effector with two spring-loaded, rubber-tipped fingers and two cameras. The material tension control was based on the sewing machine shaft encoder signal and the fabric tension was measured by an instrumented finger.

Manipulation of the limp fabric during sewing essentially requires a multi arm robotic system. Such systems have already been the subject of extensive studies for the handling of rigid objects but the results are not directly applicable in this domain. The only result we found to be applicable in this area, i.e. for controlling of internal forces in bilateral manipulation, has been reported in Refs. [13,14].

Use of multi-arm robotic systems in tasks where the work piece is a limp material, imply increased importance of proper realization of various subtasks encapsulated within the same realization, and each of them may be essential for task success. In the first place one should mention hands coordination during the manipulation of limp material, including switching from the phase where a single arm is performing to the phase where both arms are involved in the task realization. Each arm presses the fabric against the sewing table surface by fingertip and pushes it forward with the proper feed speed as a rigid panel, enabling sewing along a desired path. During sewing, in addition to the proper coordination, synchronization of hands motion with sewing speed must be simultaneously satisfied. It should also be mentioned the problem of maintaining
desired fabric tension, which is directly controlled by the ‘on-line’ change of the relative distance between the hands while performing the task. Also, when two robots are working in a single working volume (with possibly some other hardware interfered, as for example a sewing machine) they represent obstacles to each other and the paths of the robots hands should be carefully defined. Hence, it is not easy, either to program a complex task to be performed without collision or to avoid collision while the robots are performing actions that were not programmed in advance.

In this paper we described a new system for automated sewing. The aim was to give a complete insight into the realization of a bilateral robotized sewing task by describing the whole system: its structure, programming and algorithms applied, as well as all the ‘accompanying’ problems. The most important difference compared to previous results is that instead of a single robot we propose a bilateral robot system to execute the task in a similar manner as does the human operator. This decision has had significant consequences. To enable programming of such a complex system in a reasonably convenient way, we developed the original Multi-arm Robot Control (MRC) system incorporating a task-oriented robot language, a graphical user interface, and having the capability to perform simulation of the already programmed task sequences prior to its execution, particularly, to check for the possible collisions during task realization. Also, there had to be developed and implemented, an approach to control the hands coordination during motion and the fabric tension based on the concept of virtual rigid stick [13]. To control seam path and its deviation from the desired trajectory, visual feedback was adopted without any subsidiary device. The proposed approach as well as the functioning of the whole system were verified experimentally by sewing.

This paper is organized as follows. The basic description of the robotized sewing task, its setup and programming are described in Section 2. The MRC control system is presented in Section 3, experimental results in Section 4, and the conclusion of the paper is summarized in Section 5.

2. The task of robotic sewing

2.1. Basic task description and programming

To introduce some basic problems associated with the task we are dealing with, and with all phases to be performed with different conditions to be satisfied in each phase, let us focus first on the simple example of the robotized sewing task.

The sewing setup consists of a sewing machine and two robots for fabric manipulation on the table, as schematically shown in Fig. 1. To explain all phases (sequences) constituting the sewing task a simple example is sketched in Fig. 2. For the sake of simplicity, only one robot with a piece of fabric and sewing machine is shown. The sewing is performed in such a way that the robots press the fabric against the sewing table and then, by a coordinated motion of both
hands, a desired motion of the fabric as a rigid panel is performed. The task is defined by a sequence of points (P0–P7) corresponding to the position and orientation of the robot end-effector. Each point is represented in the task (reference) coordinate system placed at the point where the sewing needle penetrates the table. Each robot starts from its home position (point P0), then moves in a free space to the point P1 and down to the point P2. At the point P2,
the robot touches the fabric and ensures the pressing force (contact force pressing the fabric against the table) attains the predefined intensity \( F_{z_2} = F_z \). Then, keeping the pressing force constant, and performing constrained motion, the robot moves to the point P3 \( (F_{z_3} = F_z) \), where sewing starts. During the motion from the point P3 to point P4 the additional conditions of motion synchronization of both hands ensuring the fabric motion with the desired sewing speed are employed while the predefined vertical pressing force \( F_z \) should be preserved. After that, the sewing machine stops and the robot moves through the point P5 to the point P6, keeping constantly the predefined intensity of the pressing force \( F_z \). Then, the robot moves up to the point P7 and returns to its home position P0. During this task, different constraints are imposed to robot’s motion. From P0 to P2 both robots perform motion in a free space, from P2 to P3 a coordinated and constrained motion, from P3 to P4 a coordinated, constrained and synchronized motion. The motion from P4 to P6 is coordinated and constrained (but not synchronized), while from P6 back to their home position P0 the robots move again in a free space.

To cope with the increased complexity of programming all the various requirements imposed, to the realization of such a class of tasks by conventional industrial robots, a dedicated language for programming robots in the MRC system was developed. The C-surface theory proposed by M.T. Mason [15] was used as a theoretical basis for constrained tasks description. The most important feature of the system is that the robot motion fulfilling artificial constraints, can be easily programmed. For example, the command ‘mvs Pi’ comprises a straight line movement of the robot tip to the point Pi while preserving the desired position during motion, specified by the motion velocity. The command ‘untilforce F’ means that force intensity should be observed and command executed until the force attains a predefined intensity. The command ‘force(F)’ means that artificial constraints of the desired force intensity should be preserved during the motion execution. The system allows the commands ‘mvs Pi’ and ‘untilforce F’ or ‘force(F)’ to be merged together and executed simultaneously to meet more complex task specifications. For example, the command ‘mvs Pi untilforce F’ means the robot will move from its present position to the point Pi, where the contact with the environment is to be established. The command execution will be continued until force F reaches a predefined intensity. When both requirements (point Pi position and force intensity F) are fulfilled, the subsequent command will be executed. In a similar manner, the command ‘mvs Pj force (. . . F)’ will execute the robot motion along a straight line from the present position to the point Pj while preserving the attained force F intensity.

A serious problem in the programming and execution of a two-arm system is the synchronization of ensuring a simultaneous start of both the robots involved in coordinated motion. This problem is resolved using the ‘SYNC’ function. If timing during the free-space motion is not perfect, when this command is applied, one arm will wait for the other before coordinated motion starts, this is illustrated in Fig. 3. If the timing is perfect, this command will not be effective and will not affect the program execution.
One of the very important conditions for successful and accurate seam realization is the proper fabric tension. A high tension causes an unacceptable fabric deformation, too low tension may affect fabric manipulation on the sewing table as a rigid panel. To achieve a desired tension of the fabric during sewing, the command 'inforce F' should be used. To cancel it, the command 'inforce' should be placed in the program.

It is also important to emphasise that an independent robot motion (a motion which is neither synchronized nor coordinated) should be defined in the program with respect to the running robot, while the coordinated motion should be defined with respect to the master robot, denoted here as robot 0. An example of the simplified program for the task sketched in Fig. 2 is presented in Table 1. The first column shows the movement of each robot, the second column shows the corresponding artificial constraints, and the third column shows the guarded moves. The corresponding robot language commands are shown in the last column, named MRC System Commands. The example of the complete program for the real sewing task experiment is shown in Fig. 21 and a list of corresponding MRC system commands with comments is given in the Appendix.

2.2. Setting nominal trajectories and basic motion parameters

In a new task programming, the first step is to define nominal trajectories for both robots by defining all the points, $P_i$ are called reference points and are defined by six Cartesian coordinates. The placement of points $P_i$ depends on the type of the robot’s motion. In case of a free-space motion, reference points are fixed at the tip of each robot. Then, the coordinates of the point $P_1 =
have the following meanings: \(i\), \(j\), \(k\), \(a\), \(b\), \(c\) define the tip position, \(i\), \(j\), \(k\) the gripper orientation (Roll, Pitch, and Yaw angles). In the example given in the Appendix, the starting points, \(P_0\) for both robots, are defined as follows:

- point \(P_0\) for the robot 0 (5DOF): \(P_{000} = (-40.2, 90.0, 50.0, -125.0, 0, 180)\) means the following; 
  \(x = -40.2\) [mm], \(y = 90.0\) [mm], \(z = 50.0\) [mm], \(a = -125\) [deg], \(b\) meaningless, \(c = 180.0\) [deg];
- point \(P_0\) for the robot 1 (4DOF): \(P_{101} = (-97.5, -19.3, 10.0, 55.0, 0, 0)\) means the following; 
  \(x = -97.5\) [mm], \(y = -19.3\) [mm], \(z = 10.0\) [mm], \(a = 55.0\) [deg], \(b\) meaningless, \(c\) meaningless.

### Table 1

<table>
<thead>
<tr>
<th>Sewing task motion</th>
<th>Artificial constraints</th>
<th>Guarded move</th>
<th>MRC System Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>(V_x = 0, V_y = 0, V_z = 0,) (\omega_x = 0, \omega_y = 0, \omega_z = 0)</td>
<td>MVS P1</td>
<td></td>
</tr>
<tr>
<td>Move to P1</td>
<td>(V_x = V_{x1}, V_y = V_{y1},) (V_z = V_{z1}, \omega_x = \omega_{x1},) (\omega_y = \omega_{y1}, \omega_z = \omega_{z1})</td>
<td>MVS P2</td>
<td>If (F_z = F_{z1}) then Stop, and execute the next.</td>
</tr>
<tr>
<td>Press fabric</td>
<td>(V_x = 0, V_y = 0, V_z = V_{z2},) (\omega_x = 0, \omega_y = 0, \omega_z = 0)</td>
<td>MVS P3</td>
<td>MVS P3 force ((\ldots, F_{z1}))</td>
</tr>
<tr>
<td>Move to P3</td>
<td>(V_x = V_{x3}, V_y = 0, F_z = F_{z3}^b,) (\omega_x = 0, \omega_y = 0, \omega_z = 0)</td>
<td>MVS P4</td>
<td>Inforce (F_{z1\text{b}},) sew on, MVS P4, inforce</td>
</tr>
<tr>
<td>Fabric tension control ON</td>
<td>(V_x = V_{x4}, F_z = F_{z4},) (F_z = F_{z4}^b,)</td>
<td>MVS P5</td>
<td>Sew off, MVS P5</td>
</tr>
<tr>
<td>Sew Off, Move to P5</td>
<td>(V_x = V_{x5}, V_y = 0, F_z = F_{z5}^b,) (\omega_x = 0, \omega_y = 0, \omega_z = 0)</td>
<td>MVS P6</td>
<td></td>
</tr>
<tr>
<td>Release fabric</td>
<td>(V_x = 0, V_y = V_{y5}, V_z = V_{z5},) (\omega_x = 0, \omega_y = 0, \omega_z = 0)</td>
<td>MVS P7</td>
<td>MVS P7 force ((\ldots))</td>
</tr>
<tr>
<td>Move to P0</td>
<td>(V_x = V_{x6}, V_y = V_{y6}, V_z = V_{z6},) (\omega_x = \omega_{x6}, \omega_y = \omega_{y6},) (\omega_z = \omega_{z6})</td>
<td>MVS P0</td>
<td></td>
</tr>
</tbody>
</table>

*a* \(V_{x}, V_{y}, V_{z}\) are reference speed of end-effector and \(\omega_{x}, \omega_{y}, \omega_{z}\) are reference rotational speed of end-effector for task coordinate system.

*b* \(F_{z1} = F_{z1\text{b}} = F_{z1\text{c}}\) (reference force for task coordinate system).

*c* \(F_{z1}\) is reference force of fabric tension.
But, when the robot reaches the fabric, instead of two different reference points for two robots, a unique reference point for the whole system (two robots and the fabric) has to be selected (see points P004 and P005 in the example given in the Appendix). It is advisable that the reference point is defined on the fabric because it is more convenient for the operator to deal with the fabric trajectory rather than with the robot’s end-effectors trajectories during the programming of a coordinated motion. The choice of a particular reference point position depends on the type of sewing task, but in general, it has to be selected at a convenient position between the fingers. Then, the vector from the origin of the task coordinate system (the coordinate system placed at the point where the needle of the sewing machine penetrates the table, see Figs. 1 and 7) to the reference point is named the reference position vector $^0p_{\text{ref}}$. Now, the six coordinates defining point $P_i$ are the position and orientation of the $^0p_{\text{ref}}$, instead of the gripper as in a previous case.

To completely determine the motion along a straight line path, velocity profile should also be defined. For the MRC system three quantities should be specified: motion speed; acceleration time; and deceleration time. These parameters should be specified in the beginning of the program and are valid for the whole program (see the first three lines in the program: labels 1010; 1020; and 1030) and cannot be changed later. The specified motion speed defines the gripper velocity between two points for which a constant speed is supposed. Acceleration time is the time for which velocity should reach a constant value starting from 0 [mm/s]. Deceleration time is the needed time to decrease the gripper velocity to 0 (see Fig. 4(a)). Acceleration and deceleration intensities are obtained by direct calculation from the requested motion speed and specified acceleration time. Maximal acceleration and deceleration intensities are limited by specifications of joint motors. If the distance between two points is too short to reach the specified gripper velocity, acceleration and deceleration times will be recalculated automatically based on specified motion parameters and the distance between two points. In such a case the robot motion will become the motion with no constant speed, as sketched in Fig. 4(b).

In addition to the straight line motion it is also possible to program the motion along a curved (i.e. circular) path. For each curved path (corresponding command is ‘MVC’) three points should be defined: path starting point ($P_1$); passage point ($P_2$) somewhere on the path; and the end point ($P_3$), as shown in Fig. 5. To realize a specified trajectory two commands have to be defined sequentially:

- MVS $P_1$ positioning the gripper at the starting point $P_1$;
- MVC $P_2$ $P_3$ curved path execution passing through the point $P_2$, ending at $P_3$.

When execution is started, the path trajectory is calculated automatically by the MRC Trajectory Generator and then the motion is executed. During motion the command ‘SPD’ (see Appendix) for setting a desired velocity along the path is effective, too. At this moment only a circular path can be programmed.
Fig. 4. Velocity profile along straight line trajectory.
Non-regular paths as well as circular path can be achieved using vision and ‘MVS’ command simultaneously, as shown in Section 4.

2.3. The experimental sewing setup

The experimental two-arm robot system for sewing, consisting of one robot arm having five Degrees of Freedom (5DOF), another arm having 4DOF, and an industrial sewing machine, is shown in Fig. 1. The robots were chosen to exactly match the task complexity (number of necessary DOF) and space limitations relating to the sewing machine. In fact, to perform a desired fabric motion one of the robots has to operate below the sewing machine structure (see Fig. 1), and a smaller 4DOF SCARA type robot was selected for this purpose. All joints of the 5DOF arm are driven by DC servomotors with harmonic drives, joints of the 4DOF arm by the ultrasonic motors using PWM method. Each robot arm has two plate spring-loaded, rubber-tipped fingers. To execute pressing force control and the tension control, each robot arm was equipped with a force sensor at its wrist. The sewing machine used was a JUKI-DDL-5570 type, which is an ordinary industrial sewing machine. It can receive several commands such as start, stop, cutting thread, and the control signal from the host computer for adjusting motor speed. A CCD camera was mounted on the sewing machine, to monitor the sewing task execution. The observed area was of 512 × 512 pixels size, each of which having 255 levels of gray scale.

The structure and interconnection of the whole system hardware is shown in Fig. 6. All the units are connected to two microcomputers (Intel 80486 DX2-66 MHz + 5X86-100 MHz and Pentium 166 MHz). In this figure, A/Ds are A/D converters, D/As are D/A converters, and D I/O is a digital parallel interface.

![Curved (circular) path specification](image-url)
3. Control system

3.1. Structure and operation of the control system

The MRC system consists of the three basic modules: the operation module; the display module; and module for servo control. The block diagram of the control system is shown in Fig. 7. The operation module comprises the system operation, editing the programs and interpretation of the robot language, trajectory generator, hybrid position/force control, and the two-arm coordinated motion control. The display module enables interaction between the operator and the system. The servo control module implements the position feedback during motion of each of the robots’s joints. All modules are integrated in our original kernel. The kernel’s role is the task scheduling and the communication between each module of the control system. The task of the scheduling function is the adjustment of the execution of timing of the software servo module, operation module, and the display module.

The basic module for the operator’s interaction with the MRC system is the display module. Its operation is based on a desired menu selection. Each menu offers all functions available at the moment (Fig. 8). The menu selected can be: the type of operating mode; display; or functions. The available types of operating modes are: return to home position; simulation; robot on-line mode; and the editor for the robot program. Return to home position mode is necessary in order to restore initial joint angles after power is turned off because incremental encoders are used for joint angles measurement. Simulation mode is very useful in
the development of a new program because it enables the simulation of the programmed motion of the robots to be displayed on the monitor. Fig. 8 shows the simulation of the manipulation of a rectangular piece of fabric from its initial position to the position where its center is translated by 40 mm and rotated for an angle of 45°. It is particularly useful when a program written for a new sewing task should be verified. By performing simulation, all possible collisions and singularities can be detected and the path corrected before the task execution.

In the robot on-line mode, while the robots are running, the robot motion is shown on display. In the editor operating mode, writing of a new or editing the already existing program, is enabled. The function menu enables the following functions:

- ‘PROGRAM LOAD’ — loads the program;
- ‘PROGRAM SAVE’ — saves the program;
- ‘COPY’ — copies the program line for editing;
- ‘DELETE’ — deletes the program line;
- ‘PROGRAM/POSITION’ — selects the robot program and position/data for editing;

![Fig. 7. Block diagram of the control system.](image-url)
• ‘ROBOT NO’ — selects the Robot Number [0 or 1] for editing;
• ‘HOME’ — moves to home position;
• ‘START’ — starts the loaded program;
• ‘STOP’ — stops the running program;
• ‘JOINT JOG’ — enables the robot joints to be manually controlled;
• ‘CARTESIAN JOG’ — enables the gripper position in the Cartesian coordinates to be manually controlled.

3.2. Control strategy

3.2.1. Hybrid position/force control, coordinated position and tension control

Manipulation of a limp fabric is difficult because of its flexibility and tendency to curl up, distort, or wrinkle. To manipulate such an object on the table like a rigid panel and accomplish sewing by a bilateral robot system, it is necessary to develop an appropriate hybrid position/force control. Its purpose is to meet
artificial constraints imposed by the motion of each hand separately, and coordinated position and tension control to be applied to both hands simultaneously to ensure desired fabric dynamic characteristics and its behavior during the task execution.

To ensure a desired fabric path, each arm of the robot must simultaneously exert force in the direction orthogonal to the sewing table (force pressing the fabric) and perform the motion parallel to the sewing table. To achieve this, a hybrid position/force control \[16\] was adopted. Then, the nominal trajectories for each joint angle \( ^{iq}_{ref} \in \mathbb{R}^m \) (\( m \) is the number of the robot joints) can be obtained from:

\[
^{iq}_{ref} = \Lambda^{-1}(^{ip}_{ref} + ^{iS}^{if})
\]

Here, \( \Lambda^{-1} \) is the inverse kinematics function, \( ^{ip}_{ref} \in \mathbb{R}^6 \) is the specified reference position vector, and \( ^{iS} \in \mathbb{R}^{6 \times 6} \) is the selection matrix. The vector \( ^{if} \in \mathbb{R}^6 \) is the output from the PID controller based on the difference (\( \Lambda^{if} = ^{if} - ^{if}_{ref} \)) between the force acting on the end-effector \( ^{if}_{sensor} \) and the reference force vector \( ^{if}_{ref} \). (In Fig. 10 \( ^{if}_{sensor} \) denotes the force sensed by the sensor at the robot wrist, expressed in the reference coordinate frame, \( ^{if}_{sensor} \) is the same force but expressed in the task coordinate frame. Thus, ‘coordinate transform’ block in Fig. 10. means transformation from \( ^{if}_{sensor} \) to \( ^{if} \).

The \( ^{iq}_{ref} \) quantities are the nominal joints trajectories to be performed during task realization and, accordingly, the reference values for the position feedback loops for joint actuators by the manufacturer and it is not possible to change them.

By controlling the position of each robot arm during a coordinated motion in the plane parallel to the sewing table, two tasks have to be accomplished simultaneously, a desired motion of the fabric on the table according to the desired sewing path ensured by proper coordination of the hands position, and proper fabric tension, ensured by managing the appropriate distance between the hands.

To implement coordinated position control, we have to define first the virtual rigid sticks from the center of each end-effector to a reference point between the end-effectors on the fabric as shown in Fig. 9(a). These virtual rigid sticks are defined by the transformation matrix \( R_{virtual} \in \mathbb{R}^{4 \times 4} \). The transformation matrix from the needle point to the end of the stick is denoted as \( ^{iN} \), while the same distance is represented by the vector \( ^{ip}_{ref} \). If trajectory of the reference point during the task execution is specified (i.e. if \( ^{ip}_{ref}(t) \) is defined) by keeping the virtual rigid stick unchanged, the fabric will be manipulated on the table as a rigid panel. The relationship between each transformation matrix for the whole system and \( ^{ip}_{ref} \) is obvious from Fig. 9(a), and from the transform graph \[17\], shown in Fig. 9(b). The \( \text{Trans}(x, y, z) \) denotes a translation transformation, \( (x, y, z) \) are the Cartesian coordinates of the reference point, i.e. of \( ^{ip}_{ref} \) relating the task coordinate system, while \( \text{Rpy}(a, b, c) \) is representing orientation \( (a, b, \text{ and } c \text{ are the angles (Roll Pitch, and Yaw) of } \theta_{ref}) \) relating to the same coordinate
Fig. 9. Relationship between (a) Transformation Matrices, and (b) Transform Graph.
Fig. 10. Coordinated motion and visual trajectory tracking control.
system. While the coordinated position control is executed, the two robots are performing the motion along the predefined $p_i^{\text{ref}}$ trajectory, as shown in Fig. 10. In such a case the transformation matrices are $^0N_i=^1N$.

The fabric tension is controlled by a simultaneous coordinated motion of both hands by maintaining proper distance between them. To change the distance between the hands, virtual rigid sticks have to be changed on the basis of the measurement of forces acting on the robots tips. The total force sensed by each sensor $iF_{\text{sensor}}$ ($i = 0, 1$) can be decomposed into two components, horizontal component $iF_{\text{horizontal}}$ involved in the fabric tension control and normal component (pressing force) $iF_{\text{press}}$ ensuring adequate friction between the robot finger and the fabric, as shown in Fig. 11. The fabric tension control is accomplished by the internal force control method [14]. The measured force component $iF_{\text{horizontal}}$ comprises the table friction force and the actual fabric tension force. Because those two forces cannot be resolved by measurement, the friction force $iF_{\text{friction}} \in R^1$ has to be estimated under the assumption that the force between the table and the fabric is regarded as a Coulomb friction. Thus, it follows:

$$iF_{\text{friction}} = \mu iF_{\text{press}} \operatorname{sgn}(d'v/dt)$$  \hspace{1cm} (2)

Here, the pressing force $iF_{\text{press}}$ is measured by a force sensor, the friction coefficient $\mu$ is estimated from the experiment, while direction of the velocity of the robot finger in contact with fabric $d'v$ is the direction of $x$-axis in the coordinate system fixed to the reference point. Then, the tension force $\hat{F}_{\text{tension}}$ can be estimated as

$$\hat{F}_{\text{tension}} = (\hat{0}{F}_{\text{horizontal}} - \hat{0}{F}_{\text{friction}})/2 - (\hat{1}{F}_{\text{horizontal}} - \hat{1}{F}_{\text{friction}})/2.$$  \hspace{1cm} (3)

The difference between the estimated $\hat{F}_{\text{tension}}$ and the reference (desired) fabric tension force $F_{\text{tension}}^{\text{ref}}$ is then used to determine the change of the actual distance between the robot hands $\Delta x$ to be applied for the fabric tension control.

Fig. 11. The forces acting on the fabric.
\[ \Delta x = G_{\text{tension}}(F_{\text{ref}}^{\text{tension}} - \hat{F}_{\text{tension}}) \]  

where \( G_{\text{tension}} \in \mathbb{R}^1 \) is the gain, also determined by experiment. By adding the \( \Delta x \) calculated from (4) to the actual length of the virtual stick \( i^R_{\text{virtual}} \) in the direction of fabric tension, the proper adjustment of the fabric tension without affecting the coordinated position control can be achieved. This procedure is described in the Hybrid Position/Force Control block (see Fig. 10). From the homogenous transformation matrix defining \( i^R_{\text{virtual}} \), \( x \)-component of the translational vector \( i^p_x \) is taken (recall that sewing is performed in the \( x \)-direction of the coordinate system fixed to the reference point), then corrected by adding \( \Delta x \) taken from the ‘fabric tension controller’, and the new value \( i^{\text{new}}_p_x = i^p_x + \Delta x \) is feed back into \( i^R_{\text{virtual}} \). By properly updating the virtual stick length (i.e. \( i^{\text{new}}_p_x \)), the desired fabric tension can be maintained without affecting the coordinated motion of the robot hands.

3.2.2. Visual trajectory tracking

Despite a coordinated position and tension control being employed, the tracking error may become unacceptably large and not possible to correct using the already proposed control strategy. This is particularly obvious in some tasks when the radius of the sewing trajectory is smaller than 80 mm. To overcome these difficulties and extend the system’s capability, an additional visual servoing is added. Two types of sewing trajectories were tested with visual servoing system: (1) drawing a sewing line on the fabric workpiece; and (2) attaching a sewing paper pattern to the fabric workpiece. Both were experimentally tested.

![Fig. 12. Sensing zone of the CCD camera and visual servoing.](image-url)
In the actual sewing, the region about the sewing needle is hidden by the presser
foot of the sewing machine, and it cannot be observed by the CCD camera. In
this system, the edge contour of the pattern or the sewing line in the region (40 ×
160 pixels) in front of the needle was observed to measure the sewing tracking
error \( d \in R^1 \) and \( z \in R^1 \), as shown in Fig. 12.

In the sewing task performed by a human operator, the fabric is translated
(transported) by the sewing machine feed mechanism and rotated about the needle
by the operator. The same is valid for the robotized sewing. The feeding
mechanism determines the direction of sewing, and to compensate for the
trajectory error we can only additionally rotate the fabric around the needle.
Thus, in the system with coordinated motion control applied, the internal
trajectory generator is followed during the sewing task. In the case of the applied
coordinated motion control with a visual servoing system, the internally generated
trajectory is followed with the added angular compensation \( \Delta \theta \) from the visual
controller.

To achieve rotation of the fabric without affecting the coordinated position and
tension control, both virtual rigid sticks should be rotated simultaneously for the
compensation angle \( \Delta \theta \).

\[
iR_{\text{virtual}} = (iBTiH)^{-1} \text{Rot}(z, \Delta \theta)IN
\]

where \( iR_{\text{virtual}}, iB, iT, iH, \) and \( iN \) are the transformation matrices which have
already been explained (see Fig. 9(b)), while \( \text{Rot}(z, \Delta \theta) \) is the transformation
corresponding to rotation about the \( z \)-axis by an angle \( \Delta \theta \). The intensity of
angular compensation \( \Delta \theta \) to be applied at one sampling interval can be
determined recalling the fact that the relationship between the angular
compensation \( \Delta \theta \) and angular speed \( \omega \) is given by

\[
\Delta \theta \approx \omega \Delta t
\]

where \( \Delta t \) is the sampling period. Thus, the angular speed by which the fabric
should be rotated about the needle has to be determined using vision feedback.
This was adopted from [12] in the form:

\[
\omega = K_1d + K_2z
\]

where \( \omega \in R^1 \) is the angular speed at which the fabric should be rotated about the
needle, \( d \) is the estimated seam width error, \( z \) is the angle between the trajectory
tangent and the direction of sewing, as shown in Fig. 12. \( K_1 \in R^1 \) is the gain
which is related to the rise-time, and \( K_2 \in R^1 \) is the gain introduced to damp the
system, so as to remove overshoots inherent in the control system. On the basis of
the experiments we adopted for \( K_1 \) and \( K_2 \) 0.025 rad/pixel\( \cdot \)s and 1.5625 s\(^{-1}\),
respectively, while the sampling period was \( \Delta t = 16.0 \) ms.

The block diagram of the visual servoing system is presented in Fig. 10.
4. Experimental results

To verify the proposed approach and evaluate the system’s capability to perform the desired sewing task, a number of experiments were undertaken. The system setup consisting of two robots and an industrial sewing machine has already been described in Section 2.3 and shown in Fig. 1. In Fig. 13 a detail of the setup consisting of the sewing machine and two robots pressing a piece of the fabric is shown. The fabric material used in all experiments is Kanebo 7206 kanelion. This was a cotton fabric with a 55% polynosic mix.

In Fig. 14 the result of sewing along an octagonal trajectory in the synchronized sewing task is shown. The task of fabric moving in desired directions was divided between the robot 0 and the robot 1. The robot 0 takes charge of moving the fabric workpiece to the forward direction and the robot 1 rotates it about the sewing needle. This means that there was no coordinated motion of the robots in this experiment. Actually, when only one robot was in contact with the fabric, i.e. while the robot 0 was moving the fabric workpiece forward, the hand of the other robot was raised, and while the robot 1 rotated the fabric about the sewing needle to ensure correct position for a new straight line octagon edge sewing, the hand of the robot 0 was raised. There was a certain discrepancy between the starting and
ending point of the sewing trajectory. This was mainly due to deformations of the fabric caused by insufficient constraints conditions imposed to the fabric manipulation (it was not requested to ensure the tension force) during the sewing. To cope with this problem we performed the following coordinated motion control experiment. Sewing along a straight-line trajectory was performed with the applied pressing force control, the fabric tension control, and the fabric position manipulation control. The results are shown in Fig. 15. The measured intensities of pressing forces for both robots and the positions of their hands are shown in Fig. 15(a,b), respectively. It can be seen that the deviations of pressing force intensities and the position errors are within an acceptable margin. Small amplitude oscillations in the pressing forces and the fabric tension can be observed. The cause of these oscillations in the pressing force is in the character of the feed mechanism of the sewing machine, which alternately pulls and releases the fabric. The main cause of the difference in the quality of force control for the

Fig. 14. Experimental result of synchronized sewing.
Fig. 15. Experimental results of coordinated sewing along a straight-line trajectory.
Fig. 16. Variation of arm-to-arm distance ($D_R$).

(a) Experimental Results

(b) Distance Change $\Delta R$.
two robots is due to the difference in the controllable speeds of their joint actuators. The fabric tension and the sewing speed are illustrated in Fig. 15(c,d), respectively. The two quantities were also kept close to the desired levels of 1.0 N and 30 mm/s, respectively.

In Fig. 16 the arm-to-arm distance change $\Delta R$ during sewing is shown. It is clear that if tension force exceeds 1 N, $\Delta R$ increases rapidly. So, to prevent excessive deviation of $R$ from its desired value the applied tension force must be less than 1 N.

Fig. 17 shows some examples of experimental results obtained in the curved-path sewing task with coordinated position/force control. These experimental results, concerning the tracking error, were compared with the curve gauge by an overlapping method. It was found that the trajectory tracking error is within $\pm 0.5$ mm. However, the tracking error becomes too large if the radius of the

\[ R = 120 \text{[mm]} \quad R = 100 \text{[mm]} \quad R = 80 \text{[mm]} \]

Fig. 17. Experimental results of curved path-sewing task.
The sewing trajectory is less than 80 mm. The cause of the problem may be either the flexibility of the fabric or the slippage between the fingertips and fabric.

In all previous experiments, only tension control and pressing force control were used with no visual feedback employed. To improve tracking accuracy, a

Fig. 18. Experimental results of sewing along a straight line drawn on the fabric with visual servoing.

Fig. 19. Angular speed $\omega$ of the fabric around the needle.
control method with the feedback of tracking error is needed. In Figs. 18 and 19 are shown the experimental results of sewing along a straight line with visual servoing, Fig. 18 being a photograph of the seam result. In this case, the system follows the sewing line drawn on the fabric, where the initial offset was 2 mm. In

Fig. 20. Experimental results of sewing along a curved line drawn on the fabric with visual servoing.

Fig. 21. Experimental results of sewing along curved paper pattern using only visual servoing.
Fig. 19 the angular speed $\omega$ of the fabric around the needle during sewing is shown. Fig. 20 shows the experimental results of sewing along a curved line with the coordinated motion control and visual servoing. In this case, the system follows the sewing line drawn on the fabric with no initial offset. The sewing trajectory follows the desired line smoothly and with a little error.

Fig. 21 shows the experimental result of sewing along a curved line. In this case, the sewing pattern paper was fixed on a two-ply fabric workpiece because a good quality seam was obtained with the two-ply panel. In this experiment the circular path was followed not using the MVC command. Instead, we have used the straight-line motion command ‘MVS’ and the visual servo command ‘VISUAL ON’. The fabric was translated in the sewing direction (‘MVS’ command) synchronized with the sewing machine feeding speed, and rotated about the needle (‘VISUAL ON’ command) according to the visual feedback information. These two actions were carried out simultaneously. Thus, the commanded trajectory was a straight line, but the visual servoing system followed a paper pattern and curved sewing was performed without using the ‘MVC’ command.

It was found that the trajectory error was within $\pm 0.5$ mm. Thus, the developed MRC system with visual servoing is also very effective for the coordinated task of sewing along both the straight and curved trajectory. The actual program executed in this task is given in the Appendix.

5. Conclusion

A system for robotic sewing employing two robots for handling limp fabric work-piece has been designed, built, and tested. To enable convenient programming and task realization the Multi-arm Robot Control system incorporating dedicated task-oriented robot language for fabric manipulation and simulation of programmed motion was developed. To manipulate the limp fabric on the table and accomplish sewing by a bilateral robot system desired fabric trajectory and tension should be permanently ensured. To achieve this, the appropriate hybrid position/force control effecting the motion of each hand separately and the coordinated position and tension control applied to both hands simultaneously were developed. Thus, the MRC system also includes cooperation and synchronization control of robots hands using position/force control and position/fabric tension control. The system additionally incorporates visual tracking control during the task execution to allow for compensation of the sewing trajectory tracking error caused by the fabric distortion and slippage between the fingertips and the fabric. Experiments with different sewing trajectories were carried out. The results illustrate the effectiveness of both the developed system and applied approach.
Appendix A

A.1. The trajectory of the actual sewing task

Each robot hand moves independently from the initial position until the beginning of the coordinated motion. Hand 0 moves from point P000 to P001. The reference point is changed from the point P001 on the hand to the point P002 on the fabric. Then, to press the fabric on the table hand 0 moves to the point P003, until force reach 0.8 [N]. At the same time, hand 1 moves to P101. The reference point change from the point P101 on the hand to the point P102 on the fabric, and hand 1 moves to the point P103 until the force, reaches also 0.8 [N]. From this moment till the point P004, to ensure sewing, both hands should operate in a synchronized manner. There are simultaneously enforced: position cooperated control, fabric tension control with incorporated synchronization with sewing machine speed, ensuring pressing force control (1.5 [N]), and visual feedback control. Then, both hands move up to the point P005 and return to the point P002. The reference point of hand 0 changes to P001 from the point P002, the reference point of hand 1, changes to P101 from the point P002. Finally, the hand 0 moves to P006 and hand 1 moves to P106.

A.2. List of commands used in the Program

THE COMMANDS FOR SETTING MOTION PARAMETERS

SPD v — setting motion speed v [mm/s].
ACL $a$ — setting acceleration time $a$ [s].
DACL $da$ — setting deceleration time $da$ [s].

THE COMMAND FOR MOTION ALONG A STRAIGHT LINE
MVS $P_i$ — motion from the present position to point $P_i$ along straight line.

THE COMMAND FOR SYNCHRONIZATION
SYNC — waiting command for the other robot.

THE FORCE CONTROL COMMANDS
MVS $P_i$ UNTILFORCE 1.5 — motion to point $P_i$ along a straight line until the pressing force is 1.5 [N].
MVS $P_i$ FORCE ($F_x$, $F_y$, $F_z$) — straight line motion to $P_i$ while maintaining the reference force $F=(F_x, F_y, F_z)$.

THE COMMANDS FOR SETTING REFERENCE POINT
REF $P_i$ — setting new reference point $P_i$ (rewriting over old data) without motion command.
REF — clearing (reset) of reference the point without motion.

THE COMMANDS FOR COORDINATION CONTROL
COOPERATE — start and end of coordinated position control.
INFORCE $F$ — activation of the internal force control with the reference tension force $F$ [N].
INFORCE — cancellation of the internal force control.

THE COMMANDS FOR SEWING MACHINE
DOWN — lowering the presser foot.
UP — lifting the presser foot.
SEW ON — starting the fabric feeding.
SEW OFF — canceling the fabric feeding.
CUT — cutting the sewing thread.

THE COMMANDS FOR VISUAL TRACKING CONTROL
VISUAL ON — activation of visual servo control.
VISUAL OFF — deactivation of visual servo control.

END — end of the program

The Program for Coordinated Sewing Task for Experiment Shown in Fig. 21.
### A.3. Sewing Along Curved Paper Pattern using Visual Servoing without MVC Command

<table>
<thead>
<tr>
<th>Program for Robot 0</th>
<th>Program for Robot 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1010 SPD 50</td>
<td>2010 SPD 50</td>
</tr>
<tr>
<td>moving speed = 50[mm/sec]</td>
<td>moving speed = 50[mm/sec]</td>
</tr>
<tr>
<td>1020 ACL 0.3</td>
<td>2020 ACL 0.3</td>
</tr>
<tr>
<td>acceleration time = 0.3[sec]</td>
<td>acceleration time = 0.3[sec]</td>
</tr>
<tr>
<td>1030 DACL 1.0</td>
<td>2030 DACL 1.0</td>
</tr>
<tr>
<td>deceleration time = 1.0[sec]</td>
<td>deceleration time = 1.0[sec]</td>
</tr>
<tr>
<td>1040 MVS P000</td>
<td>2040 MVS P101</td>
</tr>
<tr>
<td>moving to P000</td>
<td>moving to P101</td>
</tr>
<tr>
<td>1050 SPD 10</td>
<td>2050 SPD 10</td>
</tr>
<tr>
<td>moving speed = 10[mm/sec]</td>
<td>moving speed = 10[mm/sec]</td>
</tr>
<tr>
<td>1060 MVS P001</td>
<td>2060 MVS P001</td>
</tr>
<tr>
<td>moving to P001</td>
<td>moving to P001</td>
</tr>
<tr>
<td>1070 REF P002</td>
<td>2070 REF P102</td>
</tr>
<tr>
<td>setting the reference point P002</td>
<td>setting the reference point P102</td>
</tr>
<tr>
<td>1080 MVS P003 UNTILFORCE 0.8</td>
<td>2080 MVS P103 UNTILFORCE 0.8</td>
</tr>
<tr>
<td>moving to P003 until pressing force is 0.8[N]</td>
<td>moving to P103 until pressing force is 0.8[N]</td>
</tr>
<tr>
<td>1090 SYNC</td>
<td>2090 SYNC</td>
</tr>
<tr>
<td>waiting for robot 1</td>
<td>waiting for robot 0</td>
</tr>
<tr>
<td>1100 COOPERATE</td>
<td>coordinated position control</td>
</tr>
<tr>
<td>1110 DOWN</td>
<td>lowering the presser foot</td>
</tr>
<tr>
<td>1120 SEW ON</td>
<td>start of fabric feeding</td>
</tr>
<tr>
<td>1130 INFORCE 0.8</td>
<td>start of internal force control (reference force = 0.8[N])</td>
</tr>
<tr>
<td>1140 VISUAL ON</td>
<td>start of visual servo control</td>
</tr>
<tr>
<td>1150 MVS P004 FORCE(1,1.5)</td>
<td>moving to P004, start of pressing force control (reference force = 1.5[N])</td>
</tr>
<tr>
<td>1160 VISUAL OFF</td>
<td>end visual servo control</td>
</tr>
<tr>
<td>1170 INFORCE</td>
<td>end internal force control</td>
</tr>
<tr>
<td>1180 SEW OFF</td>
<td>end of fabric feeding</td>
</tr>
<tr>
<td>1190 UP</td>
<td>lifting the presser foot</td>
</tr>
<tr>
<td>1200 CUT</td>
<td>cutting the sewing thread</td>
</tr>
<tr>
<td>1210 MVS P005 FORCE(,)</td>
<td>moving to P005 with end of force control</td>
</tr>
<tr>
<td>1220 SPD 10</td>
<td>moving speed = 10[mm/sec]</td>
</tr>
<tr>
<td>1230 MVS P002</td>
<td>moving to P002</td>
</tr>
<tr>
<td>1240 COOPERATE</td>
<td>end the coordinated position control</td>
</tr>
<tr>
<td>1250 SYNC</td>
<td>2250 SYNC</td>
</tr>
<tr>
<td>waiting for robot 1</td>
<td>waiting for robot 0</td>
</tr>
<tr>
<td>1260 REF P001</td>
<td>2260 REF P101</td>
</tr>
<tr>
<td>setting the reference point P001</td>
<td>setting of reference point P101</td>
</tr>
<tr>
<td>1270 MVS P006</td>
<td>2270 MVS P106</td>
</tr>
<tr>
<td>moving to P006</td>
<td>moving to P106</td>
</tr>
<tr>
<td>1280 REF</td>
<td>2280 REF</td>
</tr>
<tr>
<td>clearing the reference point data</td>
<td>clearing the reference point data</td>
</tr>
<tr>
<td>1290 END</td>
<td>2290 END</td>
</tr>
<tr>
<td>end of this program</td>
<td>end of this program</td>
</tr>
</tbody>
</table>

#### Position-data

- **P000=\((-0.2, 90.0, 50.0, -125.0, 0, 180.0)\)**
- **P001=\((-0.2, 62.7, 50.0, -125.0, 0, 180.0)\)**
- **P002=(0.0, 0.0, 0.0, 0.0)**
- **P003=(0.0, 0.0, 0.0, 0.0)**
- **P004=(40.0, 0.0, 0.0, 0.0)**
- **P005=(40.0, 0.0, 0.0, 0.0)**
- **P006=(-100.0, 285.0, 390.0, 0.0, 180.0)**

- **P101=(-97.5, -19.3, 10.0, 55.0, 0.0)**
- **P102=(0.0, 0.0, 0.0, 0.0)**
- **P103=(0.0, 0.0, -4.0, 0.0, 0.0)**
- **P104=(40.0, 0.0, -5.0, 0.0, 0.0)**
- **P105=(40.0, 0.0, 10.0, 0.0, 0.0)**
- **P106=(-79.4, -73.0, 10.0, 90.0, 0.0)**
References