Indira Gandhi Institute of Technology

Guru Gobind Singh Indraprastha University

Kashmere Gate, Delhi

Lab Manual

For

ROBOTICS LAB

(ETME-404)

CONTENTS

About Robotics Lab

Robotics lab of IGIT, MAE Dept. was established to cater the practical learning needs of the students of Final year undergoing Robotics theory course (ETME-404)

Learning Objective

This course unit introduces undergraduate students to robotic systems covering multi-link robotic systems, mobile robotic systems, actuators, sensors, biologically inspired robotics and machine learning techniques. The main aim is to give students an introduction to the field, historic background, development and current cutting edge research points, as well as a practical introduction how to move and control robots. The course unit is practical, and students will be given access to robots for exercises.

Learning Outcomes

At the end of the course unit students will be able to

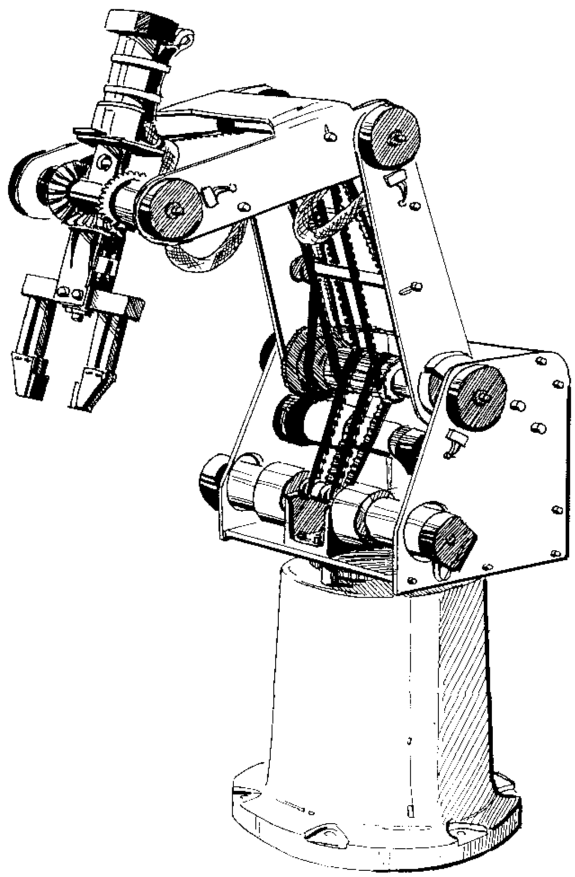
- Undertake kinematics analysis of robot manipulators
- Understand the importance of robot dynamics
- Have an understanding of the functionality and limitations of robot actuators and sensors
- Understand and be able to apply a variety of techniques to solve problems in areas such as robot control and navigation
- Describe different mechanical configurations of robot manipulators
- To be able to program a robot to perform a specified task (e.g obstacle avoidance or wall following) in a target environment.
- Understand how simulations of robots work, where they can be useful and where they can break down.
- Appreciate the current state and potential for robotics in new application areas.

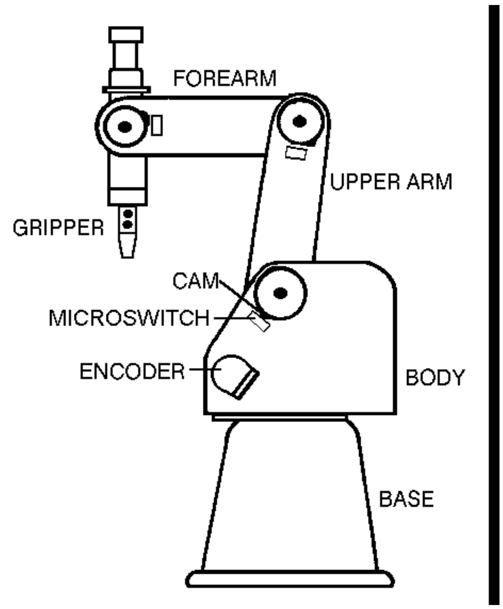
Contribution to Programme Learning Outcomes:

The course contributes towards knowledge and understanding of industrial robotic systems through its practical orientation towards programming robots, real time control and workcell layout design. Intellectual skills are trained through the analysis of control problems, identification of ways of solving them and implementation of the solution. Successes or failures are immediately evident through the resulting robot behaviour. Practical skills are trained through the practical sessions of the course. Finally transferable skills are trained by having to work tight (lab session) deadlines working in groups during practical sessions, understanding task statements, analyzing them and solving problems.

This section illustrates the description of Equipment in Robotics Lab

SCORBOT-ER 4u (ROBOTIC ARM)



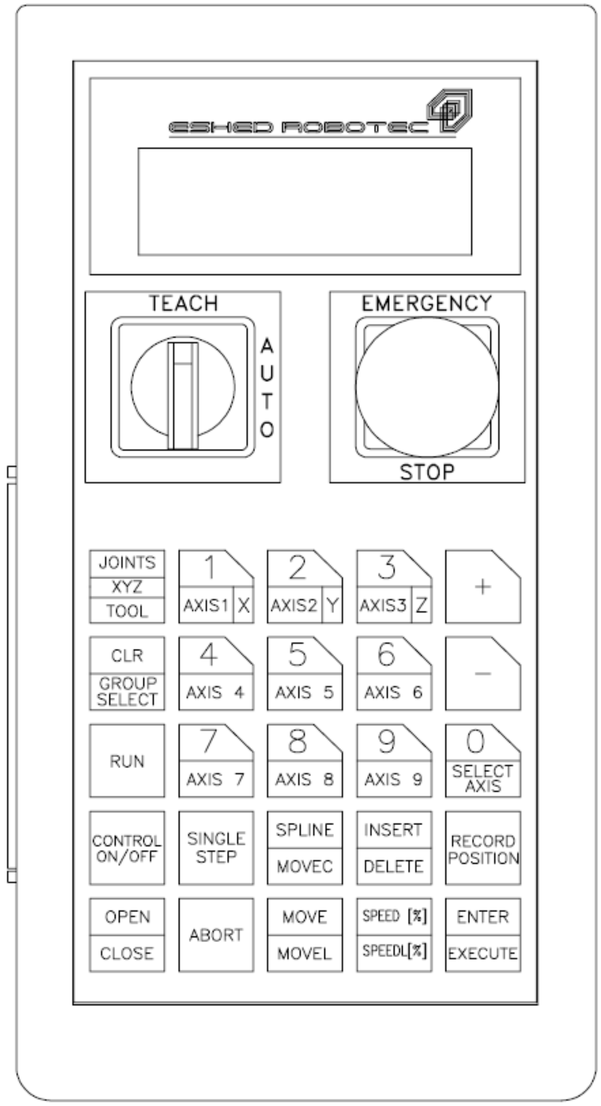


Specifications

This chapter includes the specifications the **SCORBOT-ER 4u** robot arm and descriptions of its components.

SCORBOT-ER 4u Specifications	
Mechanical Structure	Vertical articulated
Number of Axes	5 axes plus servo gripper
Axis Movement Axis 1: Base rotation Axis 2: Shoulder rotation Axis 3: Elbow rotation Axis 4: Wrist pitch Axis 5: Wrist roll	 310° +130° / –35° ±130° ±130° Unlimited (mechanically); ±570° (electrically)
Maximum Operating Radius	610 mm (24.4")
End Effector	DC servo gripper, with optical encoder, parallel finger motion; Measurement of object's size by means of gripper sensor and software.
Maximum Gripper Opening	75 mm (3") without rubber pads 65 mm (2.6") with rubber pads
Homing	Fixed position on each axis, found by means of microswitches
Feedback	Optical encoder on each axis
Actuators	12 VDC servo motors
Motor Capacity (axes 1–6)	15 oz. in Peak Torque (stall) 70 W Power for Peak Torque
Gear Ratios	Motors 1, 2, 3: 127.1:1 Motors 4, 5: 65.5:1 Motor 6 (gripper) 19.5:1
Transmission	Gears, timing belts, lead screw
Maximum Payload	1 kg (2.2 lb), including gripper
Position Repeatability	±0.18 mm (0.007") at TCP (tip of gripper)
Weight	10.8 kg (23.8 lb)
Maximum Path Velocity	600 mm/sec (23.6"/sec)
Ambient Operating Temperature	2°–40°C (36°–104°F)

TEACH PENDANT



This teach pendant (TP) is an industrial quality teach pendant which has been tailored for use in an educational environment.

The teach pendant is a sophisticated portable terminal for operating and controlling the axes connected to the controller This teach pendant is equipped with an EMERGENCY STOP push button, an AUTO/TEACH selector switch, and a DEADMAN switch. The teach pendant can be either hand-held or mounted in a special fixture outside the robot's working envelope.

CONTROLLER – USB

The computer is connected to Controller-USB via a USB module. The robot is connected to Controller-USB by a proprietary interface cable. See block diagram in Figure 1-2.

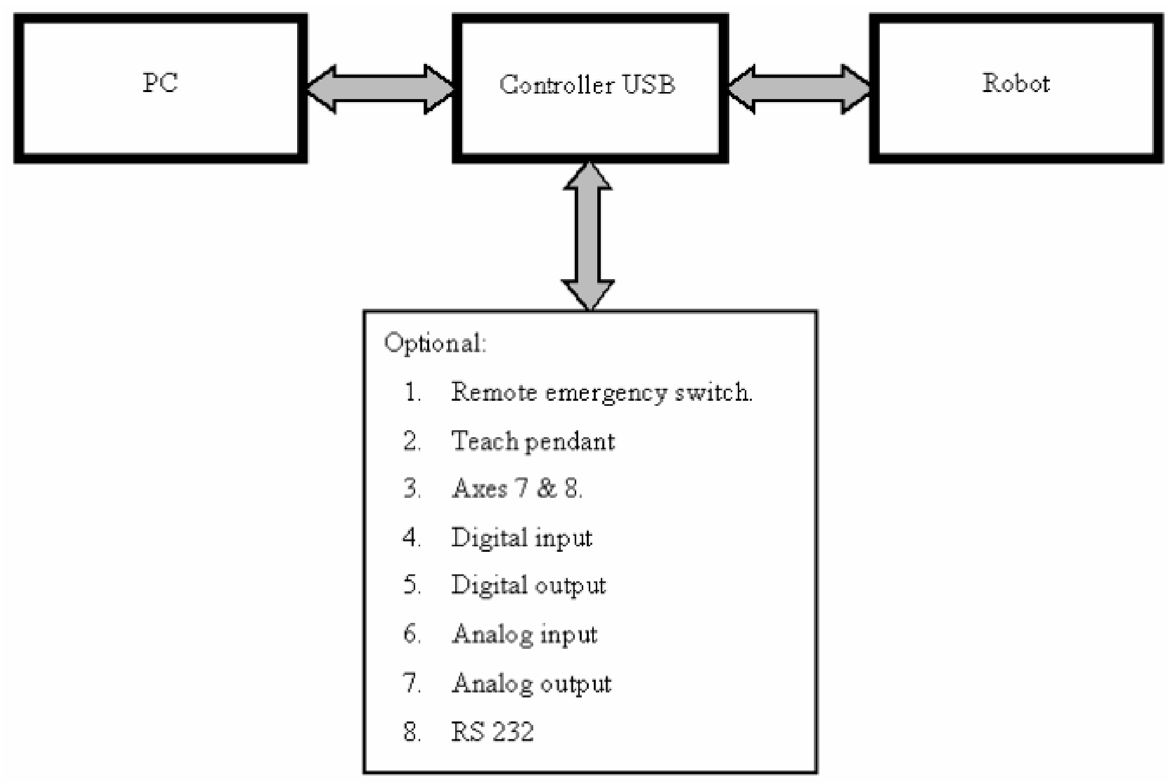
Controller-USB controls the 24V power supply to six robot motors and to two motors of (optional) peripheral devices (axes 7 and 8).

Controller-USB contains the circuits that control the motors (by means of PWM signals), and reads the encoder and microswitch signals.

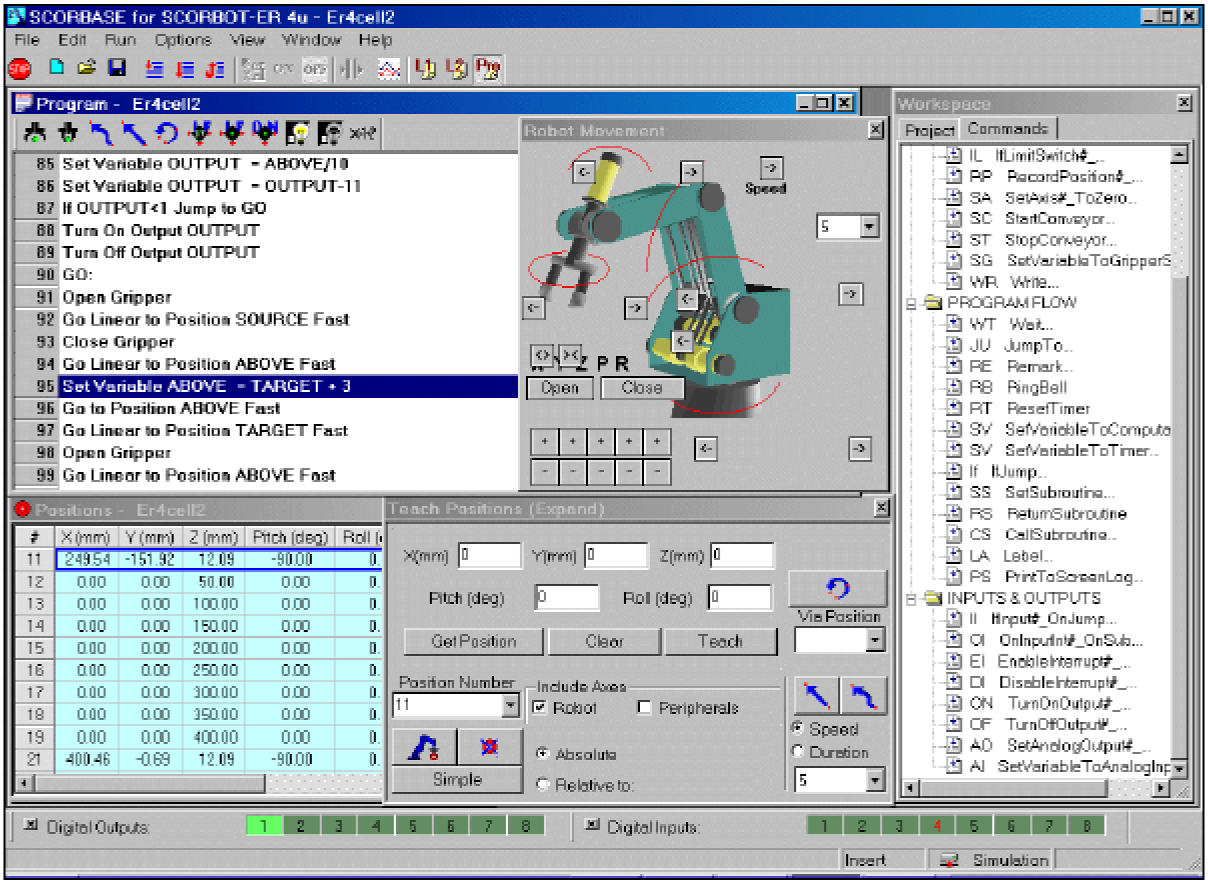
Analog and digital I/O devices can also be interfaced with Controller-USB via the digital and analog I/O ports.

Controller-USB can be connected to:

- Teach pendant (enables direct manual control of axes)
- Two motorized peripheral devices
- Analog and digital I/O devices
- RS232 communication port (for future use).


PC
Controller USB
Robot
Optional:
1. Remote emergency switch.
2. Teach pendant
3. Axes 7 & 8.
4. Digital input
5. Digital output
6. Analog input
7. Analog output
8. RS 232

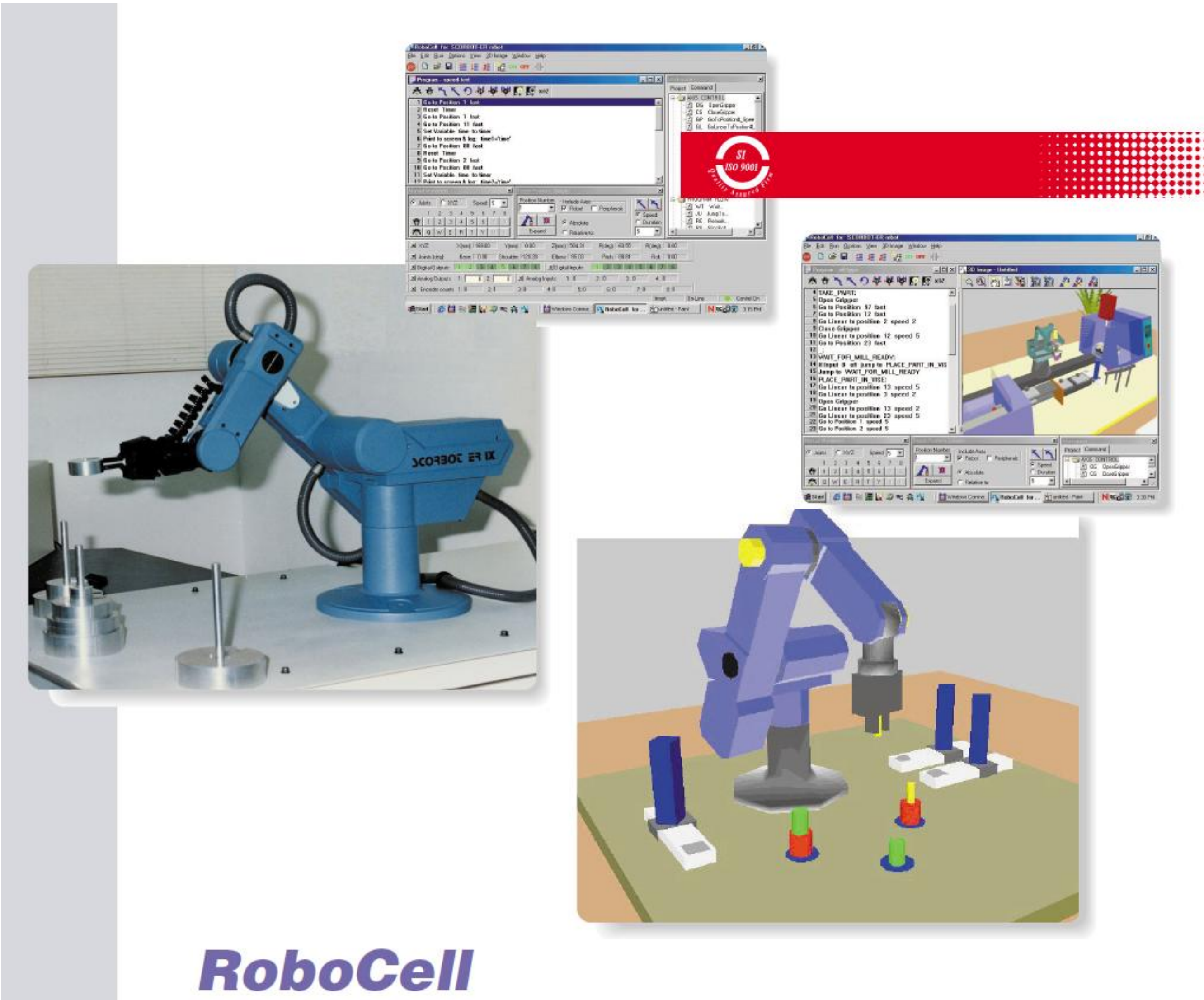
SCORBASE SOFTWARE



SCORBASE for SCORBOT-ER 2u/4u is a robotics control software package for robot programming and operation. SCORBASE for SCORBOT-ER 2u/4u provides numerous capabilities:

- Communication with the robot controller over USB channel.
- Control and real-time status display of five robot axes, gripper and two peripheral axes.
- Full support and real-time status display of eight digital inputs, eight digital outputs, two analog outputs and four analog inputs.
- Position definition and display as well as manual robot movement in joint coordinates (encoder units).
- Cartesian coordinates (X,Y,Z Pitch and Roll) are also available.
- Robot movement definition as Joint, Linear, or Circular, with ten active speed settings. (Availability depends on level.)
- Default setting of 1000 positions and 1000 active program lines.
- Interrupt programming for handling responses to changes in digital input status.
- Variable programming, in three levels of complexity, to moderate the learning curve. This makes it possible for beginners to start at a lower level, and advance through the levels, as they become more skilled in robotics programming.
- Saving and loading projects.
- SCORBASE can be installed as part of RoboCell, an interactive graphic software package, which provides simulation of the robot and other devices in the workcell.

ROBOCELL SOFTWARE



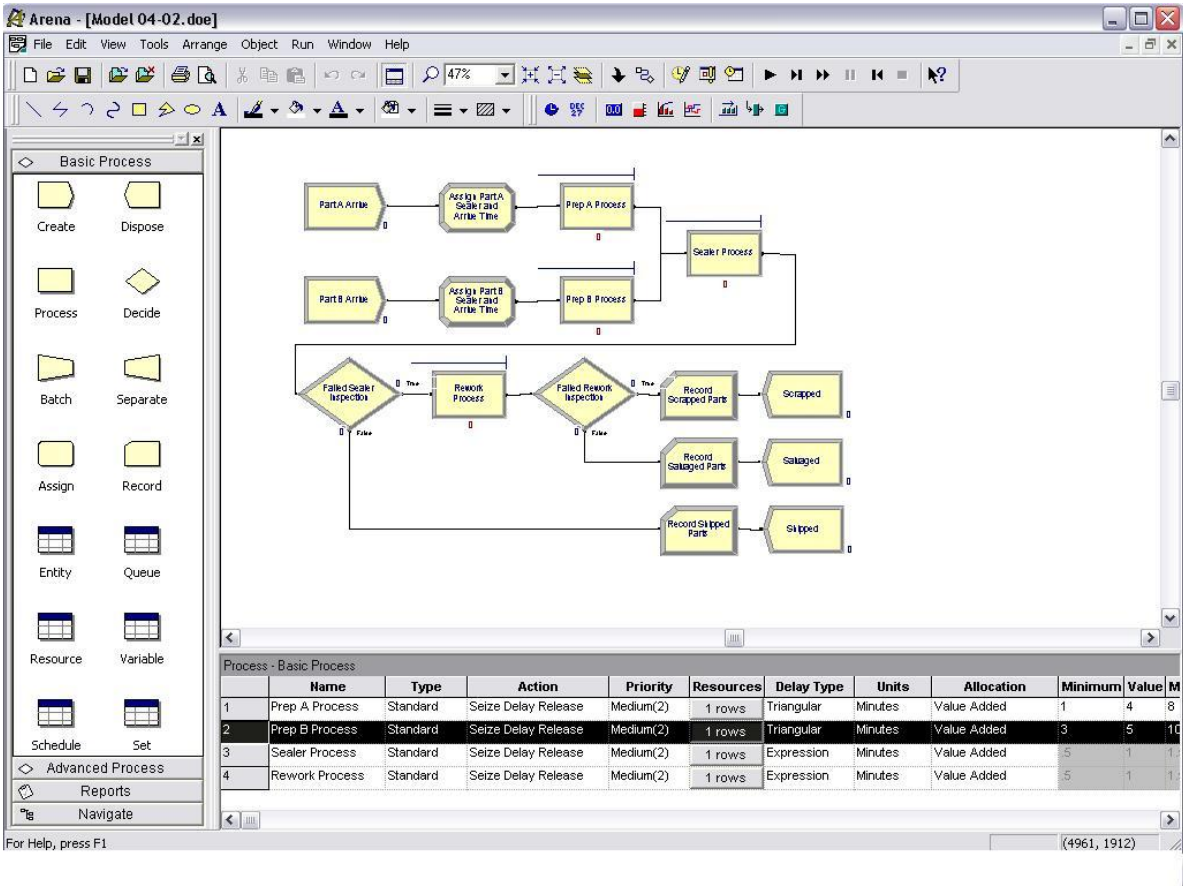
RoboCell lets students create, program, simulate and control the entire operation of robotic workcells and flexible manufacturing systems (FMS).

RoboCell integrates SCORBASE robotic control software with interactive 3D solid modeling simulation software. RoboCell's virtual robots and devices accurately replicate the actual dimensions and functions of the equipment in the robotic workcell. Students can teach positions, write programs and debug robotic applications offline before executing them in an actual workcell.

RoboCell allows students to experiment with a variety of simulated workcells, even if the actual workcells do not exist in the lab. Advanced students can even design 3D objects and import them into RoboCell for use in virtual workcells.

Fully integrated with other intelitek systems and products, RoboCell takes your students to even higher levels of technology learning and skills.

ROCKWELL® ARENA 11.0 SOFTWARE



Arena Professional simulation software is most effective when analyzing complex, medium to large-scale projects involving highly sensitive changes related to supply chain, manufacturing, processes, logistics, distribution, warehousing, and service systems. In addition, Arena PE is used to create customized simulation modeling products; that is, templates focused on specific applications or industries. With Arena Professional, customers develop custom templates that consist of "libraries" of modeling objects that make it significantly easier and faster to develop models that require repeat logic. Arena includes templates such as, Basic Process, Advanced Process, Advanced Transfer, and Flow. Arena Professional is targeted toward industrial or manufacturing systems engineers, business analysts, consulting firms providing services to a specific industry/application, or corporations with a dedicated simulation team.

Robotics lab – Class- 1

Activity -1

In this activity you will accomplish the following:
♦ Measure your knowledge of robotics and materials handling.
♦ Learn the safety measures for robotics and materials handling systems.
♦ Identify hardware components and software.

Activity – 2 (Homing and Moving the Robot)

In this activity you will accomplish the following:
♦ Activate and use SCORBASE robotic control software.
♦ Home the robot.
♦ Move the robot joints.
♦ Operate the gripper.
♦ Set and use different speeds of motion.

Apparatus required

Robotic Arm with Controller, Computer with SCORBASE software

What is a Robot?

The most widely accepted definition for an industrial robot was given by the Robotics Institute of America: "A robot is a re-programmable, multifunctional manipulator designed to move material, parts, tools, or specialized devices, through variable programmed motions for the performance of a variety of tasks."
This carefully composed definition contains several key words:
♦ *Manipulator*: The robot must have a mechanical arm.
♦ *Re-programmable*: The robot must have a computer-based controller that can be programmed through software.
♦ *Multifunctional*: Robots must be versatile, capable of performing different tasks.

Robotic System Components

Following Figure shows elements of a typical robot system. The robot system includes a manipulator arm, an end-effector (the gripper or tool connected to the end of the arm), a controller and a computer. Other devices such as a hand-held control pendant or camera may also be included.

Some robot systems contain internal feedback devices (such as encoders), and may include external sensing devices (such as a vision system).

1. End Effector

A robot must be properly equipped for the kind of task it is to perform. The end effector of the ER 4u is a gripper that can be used for grasping and picking up objects. In other robotic systems the end effector may be a tool, such as a welding gun, a spray gun for painting, or a grinding tool for milling.

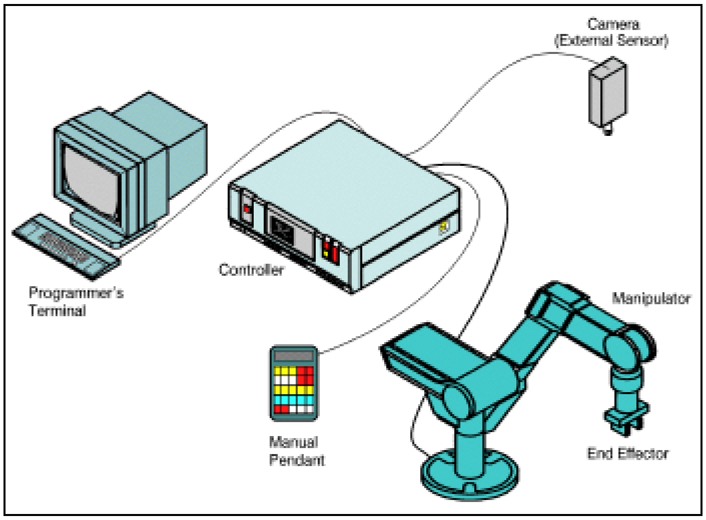


Figure 1-1

2. Manipulator (arm)

A manipulator is a mechanical arm that moves (manipulates) the end effector to the required positions and orientation. The manipulator arm must be strong enough to carry the specified payload and flexible enough to perform the required movements with the required level of accuracy.

3. Controller

The controller controls the power supplied to the motors that drive the arm. The controller's microprocessors simultaneously perform many functions during robot program execution:

♦ The controller calculates the motion required of each axis to produce the proper arm movement.

♦ The controller monitors the motion of the axes and sends commands to the motors in order to control the speed and correct positioning errors.

♦ The controller reads input data from sensors and sends output signals that activate grippers, tools and other devices.

♦ The controller receives commands from the computer or the control pendant and transmits status messages to the operator.

4. Sensors

Most robotic arms have internal sensors that provide data on the position or motion of the arm joints. These sensors help the controller to position the arm accurately. Some robotic systems include external sensors, which provide data on objects in the robot's vicinity. Without these sensors the robot might not know when and where objects are to be handled or avoided.

5. SCORBASE Robotic Software

SCORBASE is a robotics control software package, which provides a user-friendly tool for robot programming and operation.

Safety

Safety precautions in the robotic work environment serve to protect the human operators as well as the robotic equipment. Although smaller and slower than an industrial robot, the SCORBOT robot is potentially dangerous. ***You must use caution when working with the system to avoid personal injury and damage to equipment.*** All necessary hardware installation and wiring connections are to be performed by the laboratory instructor or system manager.

Students should not tamper with wiring or connectors or any of the devices in the robotic cell.

Be sure to heed the following safety guidelines:

♦ Make sure loose hair and clothing are tied back when you work with the robot.

♦ Make sure the robot arm has ample space in which to operate freely.

♦ Do not enter the robot's safety range or touch the robot when the system is in operation. Do not put your fingers into the gripper or any other moving part.

♦ Do not overload the robot arm. The combined weight of the workload and gripper may not exceed 1kg (2.2 lb).

♦ Do not leave a loaded arm extended for more than a few minutes.

♦ Do not use physical force to move or stop any part of the robot arm.

♦ Do not drive the robot arm into any object or physical obstacle.

♦ Never leave a system unattended while it is in operation.

Tasks To be Done in this Lab Class

Task 1-1: Introduction to Robotics and Materials Handling

> **Q** *What are the three key elements in the definition of a robot?*
>
> **Q** *What are the four most important components in a robotic system?*
>
> **Q** *What allows a flexible automation system to perform different operations?*

Task 1-2: Safety Guidelines

> **Q** *Does your robotic workcell conform to the safety guidelines?*
>
> **Q** *List points in the system (both the robot and the controller) that are the most dangerous to touch.*

Task 1-3: Identifying Robot Components

1. Refer to the actual SCORBOT-ER 4u robot. In the figure on the worksheet, mark the following parts of the robot:
   - Robot joints: base, shoulder, elbow, wrist pitch, wrist roll
   - Gripper
   - Motors and encoders

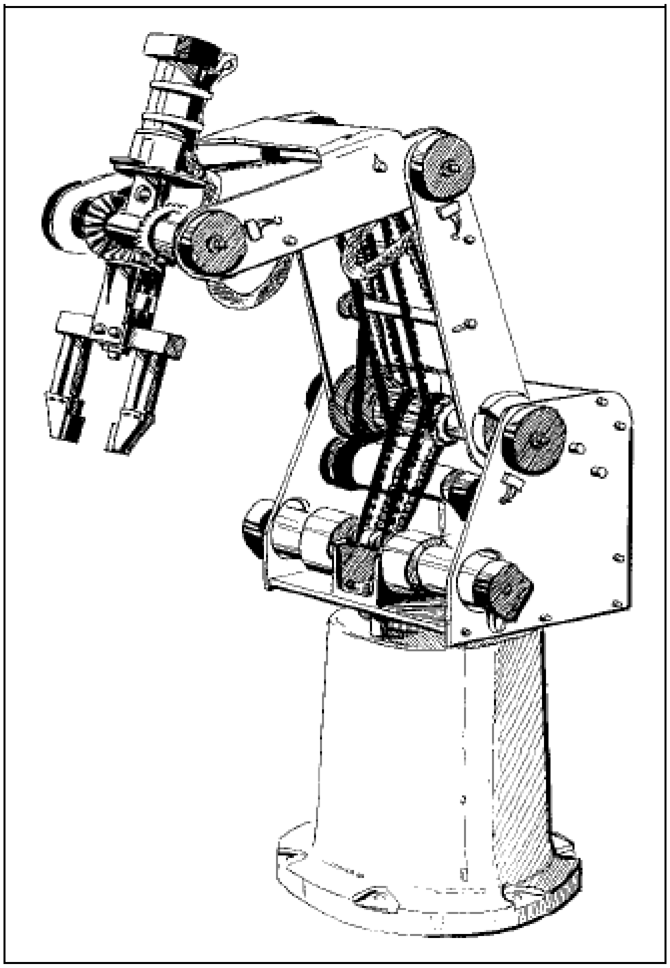


Figure 1-6

Task 1-4: Identifying Controller Components

1. Refer to the actual USB controller. In the figures on the worksheet, mark the following parts of the controller:
    - On/off power switch
    - Input and output terminals and LEDs
    - Connectors for peripheral axes
    - Emergency button

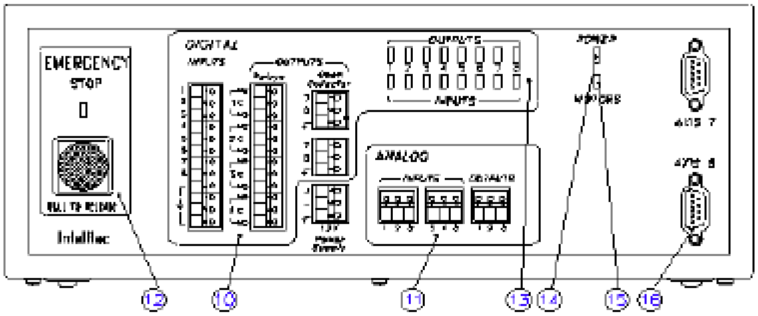


Figure 1-7

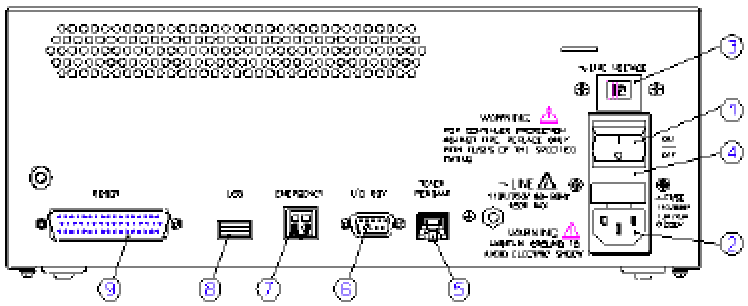


Figure 1-8

Task 2-3: Homing the Robot

1. To start the homing routine, do either of the following:
   - Click the Search Home button.
   - Select Run | Search Home.
2. *Describe the homing routine. Note the order in which the axes moved, fast or slow motion dialog boxes, and messages on the screen.*

Task 2-4: Moving the Robot

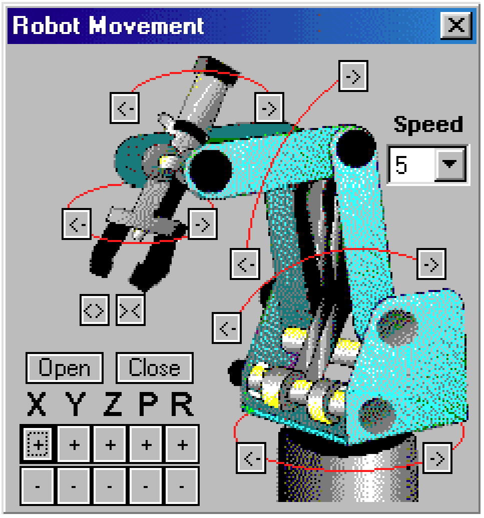
1. Select View | Manual Movement.

The following chart explains how the buttons/keys move the robot's joints.

Keys	Joint Motion
Open / Close	Fully open/closes the gripper.
1 / Q	Rotates the BASE to the right and left
2 / W	Moves the SHOULDER up and down.
3 / E	Moves the ELBOW up and down.
4 / R	Moves the wrist (PITCH) up and down.
5 / T	Rotates the wrist (ROLL) to the right and left.
6 / Y	Opens and closes the gripper incrementally.
7 / U	Moves peripheral axis (if connected)
8 / I	Moves conveyor (if connected)

Movement of an axis will continue as long as the button or key is pressed, or until a software or hardware limit is reached.

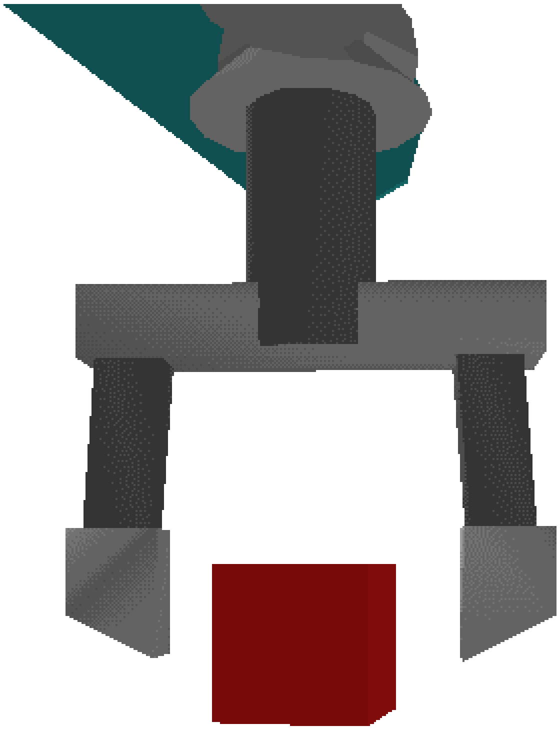
2. Using the buttons/keys listed above, practice moving the robot joints.
3. Click View | Robot movement to open the Robot Movement dialog box



Clicking the arrows in this image of a robot moves the corresponding robot joints.

Task 2-6: Operating the Gripper

1. From the Manual Movement dialog box do the following:
- Click the Open icon. The gripper opens. (If the gripper is already open, it will not move).
- Click the Close icon to close the gripper.
- Click the 6 and y icons (or keys) to open and close the gripper in incremental movements.
2. Open the gripper. Take and carefully hold a round or square block between the gripper fingers as shown in Figure 2-7. Close the gripper on the block.



3. Gently extract the block from the gripper. Click Close Gripper again.

Task 2-7: Changing Speed Settings

1. Look at the Speed box in the Manual Movement dialog box. By default, the speed setting is 5. Speed 10 is the fastest setting; speed 1 is the slowest.
2. Select 10. With the speed set at the fastest rate, try moving the robot joints.
3. Select 1. With the speed set at the slowest rate, try moving the robot joints.
4. Send the robot to its home position.
5. Place one of the blocks on the worktable.
6. In the Speed box, select 10. With the speed set at the fastest rate, bring the gripper as close as possible to the block.
7. Send the robot to its home position. Select 3. With the speed set at a slow rate, again bring the gripper as close as possible to the block.

Robotics lab – Class- 2

Activity – 3 (Recording Robot Positions)

In this activity you will accomplish the following:
♦ Record absolute positions.
♦ List and delete positions.
♦ Save and load positions.
♦ Move the robot to recorded positions

Recording Positions

Once you have moved the robot to a location in space, you can record this *position*. Once a position is recorded, you can give the robot a command to go to it. A recorded position in joint mode is a set of five to eight numbers that defines the angle of each axis. These angles are measured by means of encoders that are attached to the motor on each robot axis. The encoder monitors the rotations of the motor, and sends a corresponding number of signals, or encoder counts, to the controller. Before you exit SCORBASE, recorded positions should be saved as a SCORBASE project. The positions can then be reloaded with the project and used later.

Procedure

1. *Preparing a Folder (Directory) for Project Files*
2. *Recording Positions*
   - **a)** Activate the system, load SCORBASE and **home** the robot.
   - **b)** To open a new untitled SCORBASE project do one of the following:
     Click the **new** icon or Select File | New project or Press the CTRL+N keys.
   - **c)** Using the Manual Movement dialog box, bring the robot arm to the position illustrated in Fig.3-1

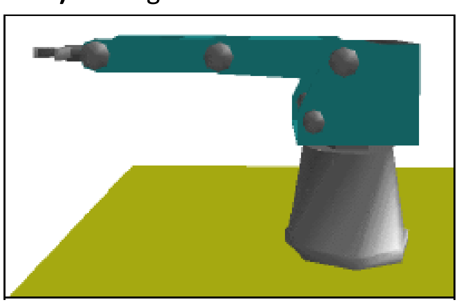


Figure 3-1: Position #1

   - **d)** Select the Teach Positions dialog box. In the Position Number box enter 1, as shown in Fig. 3-2.

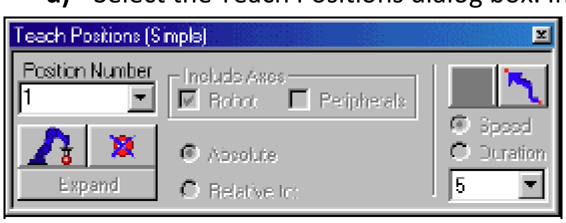


Figure 3-2: Teach Positions, teaching position #1

   - **e)** Click the Record button .

You have just recorded position #1. This number now represents this specific position (that is, the coordinates of the robot when the position was recorded).

   - **f)** Turn the base axis about 45°, as shown in Figure 3-3, and record this as position #2.

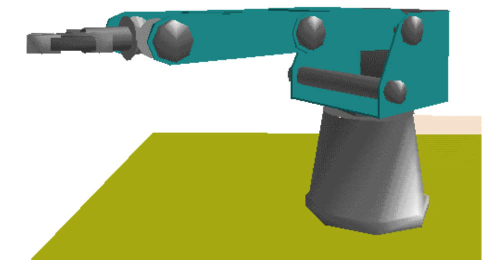


Figure 3-3: Position #2

g) Moving only the shoulder axis, bring the robot arm to the position shown in Figure 3-4, and record this as position #3.

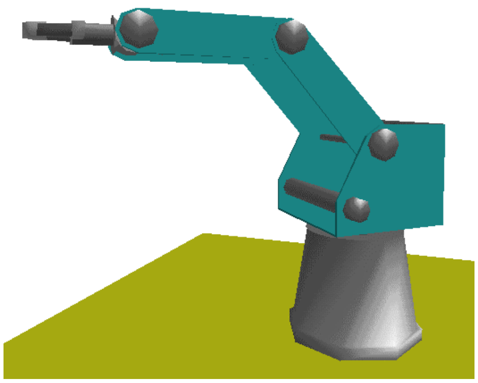


Figure 3-4: Position #3

h) Moving only the elbow axis, bring the robot arm to the position shown in Figure 3-5, and record this as position #4.

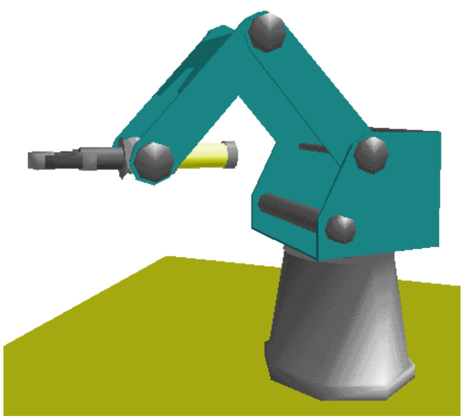


Figure 3-5: Position #4

i) Moving only the wrist pitch axis, bring the arm to the position shown in Figure 3-6, and record this as position #5.

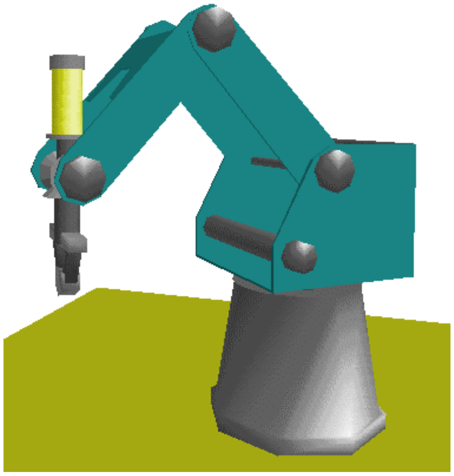


Figure 3-6: Position #5

j) Moving only the wrist roll axis, bring the arm to the position shown in Figure 3-7, and record this as position #6.

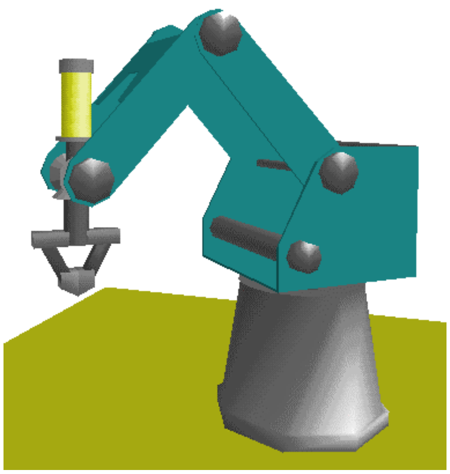


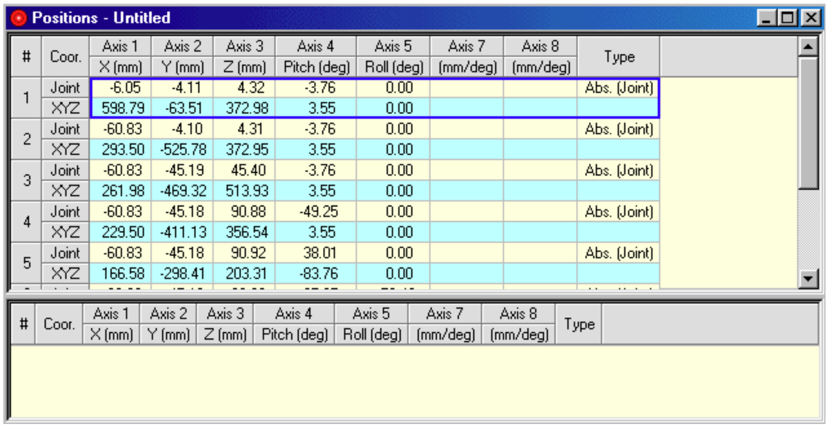
Figure 3-7: Position #6

k) Move the robot one more time, to a position of your choice, and record this as position #7.
l) Select Run | Go Home. The robot will move to the home position.
m) Record this as position #99. There are now eight different positions (1-7 & 99) in the computer's memory.

3. Positions Database

a) Click the **Project** tab in the workspace window.
b) Open untitled folder.
c) Open the positions – untitled file

The List Positions window appears showing the positions that you recorded in the previous task.


Positions - Untitled

#	Coor.	Axis 1 X (mm)	Axis 2 Y (mm)	Axis 3 Z (mm)	Axis 4 Pitch (deg)	Axis 5 Roll (deg)	Axis 7 (mm/deg)	Axis 8 (mm/deg)	Type
1	Joint	-6.05	-4.11	4.32	-3.76	0.00			Abs. (Joint)
	XYZ	598.79	-63.51	372.98	3.55	0.00			
2	Joint	-60.83	-4.10	4.31	-3.76	0.00			Abs. (Joint)
	XYZ	293.50	-525.78	372.95	3.55	0.00			
3	Joint	-60.83	-45.19	45.40	-3.76	0.00			Abs. (Joint)
	XYZ	261.98	-469.32	513.93	3.55	0.00			
4	Joint	-60.83	-45.18	90.88	-49.25	0.00			Abs. (Joint)
	XYZ	229.50	-411.13	356.54	3.55	0.00			
5	Joint	-60.83	-45.18	90.92	38.01	0.00			Abs. (Joint)
	XYZ	166.58	-298.41	203.31	-83.76	0.00			

#	Coor.	Axis 1 X (mm)	Axis 2 Y (mm)	Axis 3 Z (mm)	Axis 4 Pitch (deg)	Axis 5 Roll (deg)	Axis 7 (mm/deg)	Axis 8 (mm/deg)	Type

Figure 3-8: Positions database

This window displays the positions database.

- The numbers in the left column are the position numbers.
- The numbers in the columns Axis 1, Axis 2…Axis 8 display the axis position data.

d) Close the List Positions window.

4. Saving the Positions and Exiting SCORBASE

a) To save the project data (including the positions you recorded) do one of the following:

- Click the save icon.
- Select File | Save project.

5. Moving the Robot to Recorded Positions

You are now going to move the robot to the positions you recorded in Task 2.

a) In the Teach Positions dialog box, click the arrow next to the Position Number field to see the list of positions recorded. You should see a listing of 1 through 7 and 99.

b) Select 6. Then click the Go Position icon in the Teach Position window. The robot (now at its home position) will move to position #6.

c) Send the robot to position #5.

d) Send the robot to position #3.

e) Change to Speed 9 and send the robot to position #4.

f) Change to Speed 2 and send the robot to position #2.

g) Continue changing speeds settings and moving the robot to different positions.

Robotics lab – Class- 3

Activity

(Writing and Running Robot Programs - In this activity you will perform a common robotic task named Pick and Place. In a pick and place task the robot picks an object from a position (known as PICK) and places it in another position (known as PLACE) as shown in Figure 4-1.)

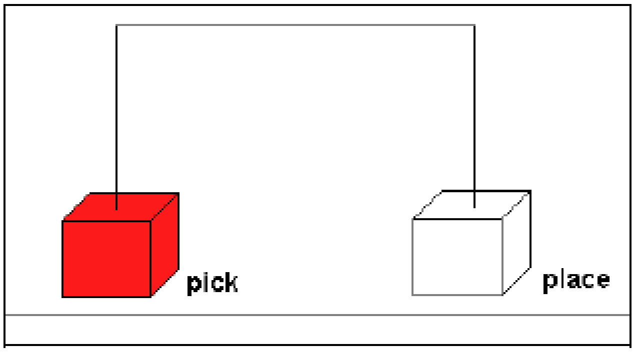


Figure 4-1: Pick and place positions

To perform this task the robot should perform the following operations:

a) Move (from its initial position) to a position above the **pick** position. *The robot is sent to that position since it should reach the pick position from above*.
b) Open the gripper.
c) Move down to the **pick** position. *After moving down, the object is between the opened gripper jaws.*
d) Close the gripper *and grip the object.*
e) Move up to the position above the **pick** position. *Now the robot grips the object. The robot moves up to ensure the object does not touch the table surface when moving to the place position.*
f) Move to the position above the **place** position. *This movement is done above the table plane.*
g) Move down to the **place** position. *The object is now in the place position but is still gripped by the robot.*
h) Open the gripper *and release the object.*
i) Move up to the position above the **place** position *to clear the place position area.*

<u>**Procedure**</u>

1. *<u>Activating SCORBASE and Recording the Positions</u>*

a) Activate the system load SCORBASE and **home** the system.
b) Open a new project.
c) Place an object on the table (make sure the object is within the robot's reach). It is recommended that you mark the pick position to assist you in placing the object at exactly the same location every time.
d) Open the gripper (use the **open gripper** icon in the manual movement dialog box).
e) Using the manual movement dialog box keys, manipulate the robot so the object is between the open gripper jaws.
f) When the object is between the open jaws of the gripper: Type 1 in the position number field in the *teach position* dialog box and click record to record the pick position as position #1.
g) Close the gripper. Make sure the object is firmly gripped by the robot.
h) Move the robot up (about 20 centimeters).
i) Type 11 in the position number field in the *teach position* dialog box and click record. The number 11 serves as a reminder that the position is *above* position #1.
j) Move the robot (and the object) to a position above the *place* position and record it as position #12.
k) Lower the robot so that the object just touches the table and record this as position #2. You have recorded the four positions.

2. *<u>Writing a Robot Program</u>*

a) Click on the command tab in the workspace window.
b) Click and open the axis control folder.
c) Double click the command *GP GotoPosition# speed...* The Go to Position dialog box appears:

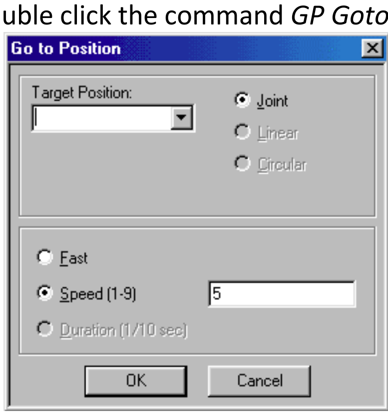


Figure 4-2: Go to Position dialog box

- Type or select 11 in the Target Position field.
- Select Fast (speed setting).
- Click OK to end.

A command line now appears in the Program window:

1 Go to Position 11 fast
|

You have just written the first program line. It will send the robot to position #11 at the fastest speed.

*If line numbers do not appear in the program, select **Options | Line Number**.*

d) Repeat the previous step, and enter a second command to open the gripper. Now your program window reads:

1 Go to Position 11 fast
2 Open Gripper

e) Add a command that will send the robot to position #1 at the slowest speed (1). Now your program reads:

1 Go to Position 11 fast
2 Open Gripper
3 Go to Position 1 speed 1

In cases when the object is moved from its original Pick position, and the gripper moves down, it may hit the object. Therefore a slow speed, which reduces impact and possible damage, is used for this movement.

f) Add the following commands:

4 Close Gripper
5 Go to Position #11 fast
6 Go to Position #12 fast
7 Go to Position #2 speed 1
8 Open Gripper
9 Go to Position # 12 fast.

You have finished writing the program!!

3. *Editing Your Program*

SCORBASE programs are edited by means of standard Windows text editing tools. If you made a mistake or you want to modify your program you can:

a) Edit a command using the command parameters (click on the command to re-open the window).

b) Delete a command line by selecting it and pressing the delete key on the keyboard.

c) Cut (Ctrl+X), Copy (Ctrl+C) and Paste (Ctrl+V) any selected program sections.

d) Insert commands into an existing program by placing the cursor on the line **below** the position of the new line and then add the required lines. Make sure SCORBASE is in insert mode (default). The mode is displayed on the status bar. In overwrite mode the new command replaces the selected command. Use the **insert** key on the keyboard to toggle between the insert and overwrite modes.

4. *Saving the Program*

5. *Running the Program*

A SCORBASE program can run in three modes:

♦ *Single line* mode that executes only the selected line.

♦ *Single cycle* mode that executes the program from the selected line to the end of the program (the last program line).

♦ *Run continuously* mode that executes the program from the selected line to the end and then continues from the first line.

6. *Aborting a Program*

There are three methods of stopping the program when it is running.

♦ Pressing the stop icon or pressing F9 immediately stops program execution and robot movements. The system remains online.

♦ Pressing the pause icon or pressing F10 stops SCORBASE from executing the next command.

♦ In case of emergency, press the **emergency** stop switch on the controller front panel. The emergency stop turns the motor and the control off.

7. *Using Wait Commands*

The wait command halts program execution. The units of duration are tenths of a second. To insert a *Wait* command set for 3 seconds at the end of the program:

a) Select the empty line at the end of program.
b) In the command tree open the Program flow folder.
c) Double click the command Wait or type WT. A dialog box appears:

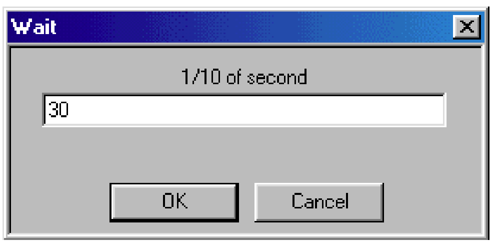


Figure 4-3: Wait command dialog box

d) Type 30 (30 tenths of a second) and choose OK.
The command Wait 30 causes the program (and robot) to pause for three seconds after executing the command in line #9 and before moving on to line #11 or, if Run Continuously is selected, to the beginning of the program.
e) Save the program as *USER4A*.
f) Run the program. Note the effect of the Wait command.

Robotics lab – Class- 4

Activity

(Writing and Running Robot Programs – write down IGIT & INTELLITEK word using welding torch).

To perform this task the robot should perform the following operations:

j) Move (from its initial position) to a position above the **pick** position. *The robot is sent to that position since it should reach the pick position from above*.

k) Open the gripper.

l) Move down to the **pick** position. *After moving down, the object is between the opened gripper jaws.*

m) Close the gripper *and grip the object.*

n) Move up to the position above the **pick** position. *Now the robot grips the object. The robot moves up to ensure the object does not touch the table surface when moving to the place position.*

o) Move to the position above the **place** position. *This movement is done above the table plane.*

p) Move down to the **place** position. *The object is now in the place position but is still gripped by the robot.*

q) Open the gripper *and release the object.*

r) Move up to the position above the **place** position *to clear the place position area.*

Procedure

8. RECORD the Positions

l) Activate the system load SCORBASE and **home** the system.

m) Open a new project.

n) Place an object on the table (make sure the object is within the robot's reach). It is recommended that you mark the pick position to assist you in placing the object at exactly the same location every time.

o) Open the gripper (use the **open gripper** icon in the manual movement dialog box).

p) Using the manual movement dialog box keys, manipulate the robot so the object is between the open gripper jaws.

q) When the object is between the open jaws of the gripper: Type 1 in the position number field in the *teach position* dialog box and click record to record the pick position as position #1.

r) Close the gripper. Make sure the object is firmly gripped by the robot.

s) Move the robot up (about 20 centimeters).

t) Type 11 in the position number field in the *teach position* dialog box and click record. The number 11 serves as a reminder that the position is *above* position #1.

u) Move the robot (and the object) to a position above the *place* position and record it as position #12.

v) Lower the robot so that the object just touches the table and record this as position #2. You have recorded the four positions.

9. *Write the Robot Program*

g) Click on the command tab in the workspace window.

h) Click and open the axis control folder.

i) Double click the command *GP GotoPosition# speed...* The Go to Position dialog box appears:

Figure 4-2: Go to Position dialog box

- Type or select 11 in the Target Position field.
- Select Fast (speed setting).
- Click OK to end.

A command line now appears in the Program window:

1 Go to Position 11 fast
|

You have just written the first program line. It will send the robot to position #11 at the fastest speed.

*If line numbers do not appear in the program, select **Options | Line Number**.*

j) Repeat the previous step, and enter a second command to open the gripper. Now your program window reads:

1 Go to Position 11 fast
2 Open Gripper

k) Add a command that will send the robot to position #1 at the slowest speed (1). Now your program reads:

1 Go to Position 11 fast
2 Open Gripper
3 Go to Position 1 speed 1

In cases when the object is moved from its original Pick position, and the gripper moves down, it may hit the object. Therefore a slow speed, which reduces impact and possible damage, is used for this movement.

l) Add the following commands:

4 Close Gripper
5 Go to Position #11 fast
6 Go to Position #12 fast
7 Go to Position #2 speed 1
8 Open Gripper
9 Go to Position # 12 fast.

You have finished writing the program!!

10. <u>Using Wait Commands</u>

The wait command halts program execution. The units of duration are tenths of a second. To insert a *Wait* command set for 3 seconds at the end of the program:

g) Select the empty line at the end of program.
h) In the command tree open the Program flow folder.
i) Double click the command Wait or type WT. A dialog box appears:

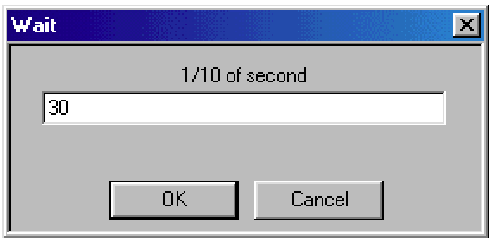


Figure 4-3: Wait command dialog box

j) Type 30 (30 tenths of a second) and choose OK.
The command Wait 30 causes the program (and robot) to pause for three seconds after executing the command in line #9 and before moving on to line #11 or, if Run Continuously is selected, to the beginning of the program.
k) Save the program as *USER4A*.
l) Run the program. Note the effect of the Wait command.

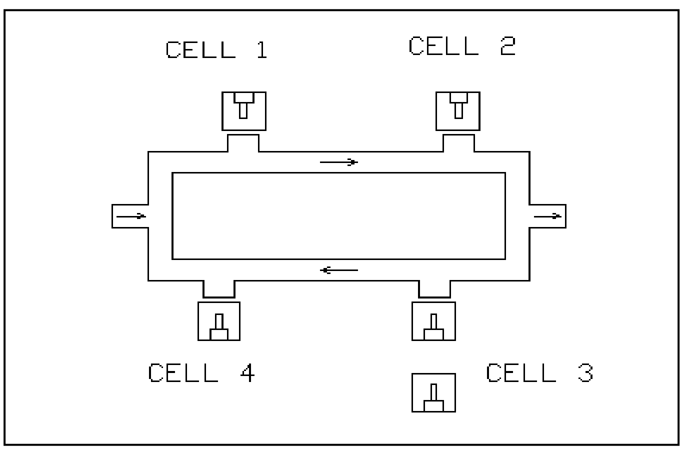
Robotics lab – Class- 5

Activity

(Make simulation model using Rockwell ARENA 11.0 to show the functions/predictions for a manufacturing work cell as per the details given below)

A layout for our small manufacturing system is shown in fig. The system to be modeled consists of part arrivals, four manufacturing cells, and part departures. Cells 1, 2, and 4 each have a single machine; Cell 3 has two machines. The two machines at Cell 3 are not identical; one of these machines is a newer model that can process parts in 80% of the time required by the older machine. The system produces three part types, each visiting a different sequence of stations. The part steps and process times (in minutes) are given in table. All process times are triangularly distributed; the process times given in table at Cell 3 are for the older (slower) machine.

The Inter-arrival times between successive part arrivals (all types combined) are exponentially distributed with a mean of 13 minutes; the first part arrives at time 0. The distribution by type is 26%, Part 1; 48%, Part 2; and 26%, Part 3. Parts enter from the left, exit at the right and move only in clockwise direction through the system. For now we'll assume that the time to move between any pair of cells is two minutes, regardless of the distance (we'll fix this up later). We want to collect statistics on resource utilization, time and number in queue, as well as cycle time (time in system, from entry to exit) by part time. Initially, we'll run our simulation for 32 hours.



Part Type	Cell/Time	Cell/Time	Cell/Time	Cell/Time	Cell/Time
1	***1***	***2***	***3***	***4***	
	6,8,10	5,8,10	15,20,25	8,12,16	
2	***1***	***2***	***4***	***2***	***3***
	11,13,15	4,6,8	15,18,21	6,9,12	27,33,39
3	***2***	***1***	***3***		
	7,9,11	7,10,13	18,23,28		

1. Using the above model, change the processing time for the second pass on Machine 1 to TRIA (6.7, 9.1, 13.6) using Sequences to control the flow of parts through the system and the assignment of process times at each machine. Run the simulation for 20,000 minutes. To the extent possible, indicate the batch-means-based confidence intervals on expected steady-state performance measures from this run.

2. Modify the model to account for acquiring a new customer, in addition to the one supplying the existing three part types. This new customer will supply two new types of parts – call them Type 4 and Type 5. The arrival process is in addition to and independent of that for the original three part types and has exponential Inter-arrival times with mean 15 minutes; the first arrival is at time 0. When the parts arrive, assign 40% of the new parts to be Type 4 and the rest to be Type 5. Here are the process plans and means processing times (in minutes) or the new part types.

Part Type	Cell/Time	Cell/Time	Cell/Time	Cell/Time	Cell/Time
4	***1***	***3***	***2***	***4***	
	6.1	5.2	1.3	2.4	
5	***2***	***3***	***4***	***1***	
	3.5	4.1	3.2	2.0	

While people feel comfortable with these mean processing times, there's not very good information on their distributions, so you're asked just to assume that the distributions are uniform with the indicated mean but plus or minus 0.2 minute in each case. Make all necessary changes to the model, including the modules, animation pictures (create new entity pictures for the new part types), and anything else required. Be sure that the animation works properly, including the clockwise-only movement of all entities and the new part types.

i. Clearly, adding this new customer is going to clog the system compared to what it was. Using the time-average total number of parts in all four queues combined, how bad does it get compared to the original system? Just make one run of each alternative (you'll do a better job on statistical analysis in the net part of this exercise)

ii. In an effort to alleviate the added congestion introduced by the new customer, you've been given a budget to buy one new machine to be placed either in Cell 1, 2, or 4 (not in Cell 3). Where would you recommend placing it in terms of the time-average total number of parts in all four queues combined? Assume that the new machine will work at the same rate as the machine it joins in whatever cell. Make 50 replications with the goal to select the best placement of the new machine; include for comparison a base-case scenario in which no machines are added anywhere.

Robotics lab – Class- 6

Activity

(Make simulation model using Rockwell ARENA 11.0 to simulate the drilling process in a manufacturing cell as per the details below)

a. Simulate a simple processing system which consists of a drilling machine and the processing time varies according to Triangular distribution of minimum 1, value 3 and maximum 6. The part enters the system with a random exponential value of 5 minutes. And then leaves the system. All time units are in minutes. Animate the resource and queue. Simulate the process for 20 minutes. Plot number waiting at drilling center queue and number busy at drill press. And report the following data

- The average total time in the system (part) and
- Utilization of drill press
- The last part number which entered the system
- Number of parts which leaves the system
- Average and maximum number of parts in process (wip)
- Make 5 replications of the above simulation. And observe the changes in output. Tabulate the readings.

b. Modify the above model with all of the following changes:
   - Add a second machine to which all parts go immediately after exiting the first machine for a separate kind of processing (rewash). Processing times at the second machine are the same as for the first machine. Gather all statistics as before, plus the time in queue, queue length and utilization at the second machine.
   - Immediately after the second machine, there's a pass fail inspection that takes a constant 5 minutes to carry out and has an 80% chance of passing result; queuing is possible at inspection, and the queue is first in and first out. All parts exit the system regardless of whether they pass the test. Count number that fail and number that pass, and gather statistics on the time in queue, queue length and utilization at inspection center.
   - Add plots to track the queue length and number busy at all three stations.
   - Run the simulation for 480 minutes.

c. In the above exercise suppose that parts that fail inspection after being washed are sent back and rewashed, instead of leaving; such re-washed parts must then undergo the same inspection, and have the same probability of failing. There's no limit on how many times a given part might have to loop back through the washer. Run this model under the same conditions and compare the results for the time in queue, queue length and utilization at the inspection center.

d. In the above problem suppose the inspection can result in one of the three outcomes; pass (probability 0.8) fail (probability 0.09) and rewash (probability 0.11). Failures leave immediately, and rewashes loop back to the washer. The above probabilities hold for each part undergoing inspection, regardless of its past history. Count the number that fail and number that pass and gather statistics.

e. Suppose that instead of having a single source of parts, there are three sources of arrival, one for each of the three different kinds of parts that arrive: blue, green and red. For each color arriving part, inter-arrival times are exponentially distributed with a mean of 15 minutes. Run the simulation for 480 minutes. Gather all the statistics as before. (Note: processing times for all of the three kinds of parts remain the same).

Robotics lab – Class- 7

Activity

(Make a model using Rockwell ARENA 11.0 to simulate the processing in small manufacturing cell as per the details below)

This is a model of a manufacturing cell. Parts arrive on a conveyor to the cell, which is picked up by an operator, who carries out the prescribed operation and as well transports it from machine to machine using a hand trolley. Finished parts are transported to an out bound conveyor after completion of all prescribed operations. This manufacturing cell has five resources out of which two are used for drilling, one each for cutting, finishing and shaping.

1. Part A and B enter the cell in the Ratio of 40:60 respectively

2. Entry Batch size is 100

3. The time between arrival of each batch into the system is NORM (75,3)

The process sequence of part A and B entering the cell is a s follows,

Part A: Drilling machine A Cutting,finishing,shipping

Part B: Drilling machine B Cutting,Finishing,Shipping

Process time of each resource is as follows

Drilling machine A is EXPO (80)

Drilling Machine B is NORM (70,3)

Cutting Machine EXPO (50)

Finishing TRIA (15,20,35)

Set up time for cutting and finishing is EXPO (10)

Shipping-Assume a process time

Shipping Lot size is 200

Trolley Velocity is 2 m second

Network links and guided transporters may be used to model the transportation process.

Task 1: we have to simulate the given manufacturing scenario for 100 hours, with a lunch break of 30 min and 2 tea breaks of 15 min each for every 8 hrs.

Task 2: Modify the model to include the use of conveyors in place of trolleys and by adding an additional manpower and relive the regular operator for moving next operation.

Task 3: Make necessary modification to increase the output and as well reduce WIP in critical areas which you interpret to be the bottlenecks of the system and justify.

Robotics lab – Class- 8

Activity

(Simulation modeling of four machine system using Rockwell ARENA 11.0 as per the details below)

Parts arrive at four-machine system according to an exponential Inter-arrival distribution with mean 10 minutes. The four machines are all different and there's just one of each. There are five part types with the arrival percentages and process plans given below. The entries for the process times are the parameters for a triangular distribution (in minutes).

Part Type	%	Cell/Time	Cell/Time	Cell/Time	Cell/Time
1	12	1	2	3	4
		10.5, 11.9, 13.2	7.1, 8.5, 9.8	6.7, 8.8, 10.1	6, 8.9, 10.3
2	14	1	3	2	
		8.9,13.5,18.1	9,15,21	15,18,21	
3	31	2	4	1	3
		8.7, 9.9, 12	8.6, 10.3, 12.8	10.3, 12.4, 14.8	8.4, 9.7, 11
4	24	3	4	3	2
		7.9, 9.4, 10.9	7.6, 8.9, 10.3	6.5, 8.3, 9.7	6.7, 7.8, 9.4
5	19	2	1	4	
		5.6, 7.1, 8.8	8.1, 9.4, 11.7	9.1, 10.7, 12.8	

The transfer time between arrival and the first machine, between all machines, and between the last machine and the system exit follows a triangular distribution with parameters 8, 10, 12 (minutes). Collect system cycle time (total time in system) and machine utilizations. Animate your model (including part transfers) and run the simulation for 10,000 minutes. If the run is long enough, give batch-means confidence intervals on the steady – state expected values of the results.

a. Modify your solution to include the travel times that are move specific. The travel times are given below as the parameters for a triangular distribution (in minutes). Compare your results. Is this a statistically reliable comparison?

From/To	Machine 1	Machine 2	Machine 3	Machine 4	Exit System
Enter System	7, 11, 19	7, 11, 16	8, 12, 19		
Machine 1		9, 13, 20	10, 13, 18	7, 9, 13	
Machine 2	8, 10, 15		7, 12, 18	7, 9, 12	8, 9, 14
Machine 3		9, 13, 20		9, 14, 21	6, 8, 11
Machine 4	11, 13, 17		10, 13, 21		6, 10, 12

Robotics lab – Class- 9

Activity

(To make a model using Rockwell ARENA 11.0 to simulate the process in three workstation system as per the details below)

A part arrives every ten minutes to a system having three workstations (A, B, and C), where each workstation has a single machine; the first part arrives at time 0. There are four part types, each with equal probability of arriving. The process plans for the four part types are given below. The entries for the process times are the parameters for a triangular distribution. (in minutes)

Part Type	Cell/Time	Cell/Time	Cell/Time
1	A	C	
	5.5,9.5,13.5	8.5,14.1, 19.7	
2	A	B	C
	8.9,13.5,18.1	9,15,21	15,18,21
3	A	B	
	8.4,12,15.6	5.3, 9.5, 13.7	
4	B	C	
	9.2, 12.6, 16	8.6, 11.4, 14.2	

Assume that the transfer time between arrival and the first station, between all stations, and between the last station and the system exit is three minutes. Use the Sequence feature to direct the parts through the system and to assign the processing times at each station. Use the Sets feature to collect cycle times (total time in system) or each of the part types separately. Animate your model (including part transfers) and run the simulation for 10,000 minutes.

a. Modify the above problem so that all parts follow the same path through the system: Workstation A – Workstation B – Workstation C. If a part does not require processing at a workstation, it must still wait in queue, but incurs a zero processing-time delay. Compare the results to those obtained.
b. Change your model to include fork trucks to transport the parts between stations. Assume that there are two fork trucks that teach travel at 85 feet per minute. Loading or unloading a part by the fork truck requires 0.25 minute. The distance between stations is given (in feet) in the following table; note that the distances are in general directional.

		To				
		Arrive	WS A	WS B	WS C	Exit
From	Arrive	0	100	100	200	300
	WS A	100	0	150	100	225
	WS B	100	150	0	100	200
	WS C	250	100	100	0	100
	Exit	350	250	225	100	0

Run the replication for 10,000 minutes (you may not want to turn off the animation via the Run>Run Control>Batch Run (No Animation) command after confirming that things are working properly. Assume that fork trucks remain at the station where they unloaded the last part if no request is pending. If both fork trucks are available assume that the closest one is selected.

c. Change your model to use non accumulating conveyors to transfer the parts between stations. Assume that there is a single conveyor that starts at the arrive area and continues to the exit area: Arrive – WS A – WS B – WS C – Exit. Assume that the distances between all adjacent stations on the conveyor are 100 feet. Further assume that the cells of the conveyor are 2 feet and that each part requires 4 feet of conveyor space. Load and unload times are each .2 minute. The conveyor speed is 20 feet per minute.