Chapter 6 Review Questions Answer Key

1. What are the five levels of robot programming?

   A. Level 1: No processor

   Level 2: Direct position control

   Level 3: Simple point-to-point

   Level 4: Advanced point-to-point

   Level 5: Point-to-point with AI

2. Describe programming a robot with a level 3 language.

   A: The programmer must still enter the motion type, data gathering, and other aspects of the program, but they do not manually enter the position. When programming these systems it is common practice to write the basic program offline with each position having a label, but no coordinate data. Once the programmer is happy with the program, they upload it into the robot and at that point physically move the robot to each point and record the positional data.

3. How do you write a program in a level 4 robotic language and what are some of the benefits?

   A: With level 4 languages, writing a program really is as simple as creating a new program, entering a string of points with the proper motion label to reach those points, and testing out the program. Instead of needing a heavy background in programming and logic, a week's worth of training at the manufacturer's site is enough for most to begin writing working programs. Some manufacturers will waive the training cost if an industry buys a certain number of their robots, helping to sweeten the sales deal. Another benefit

is that the simplicity of this programming language makes it easy to write a program directly from the teach pendant.

4. What is the difference between a level 4 and a level 5 programming system?

   A: Level 5 takes the convenience of level 4 languages and adds the ability of correction for error as well as advanced teaching methods. The standard way to teach a point is to use the teach pendant and manual mode to move the robot into position via various motion types and a combination of movements. Some of the level 5 systems allow the user to physically grab the robot and move it into whatever position they desire.

5. What are the seven questions you should answer during the task mapping phase of planning?

   A: What kind of tooling does the robot need?

   How do I want the robot to move between points?

   Are there any obstacles for the robot to avoid?

   What is the robot doing at each point?

   What is the robot doing between each point?

   Are there any conditions or other factors I need to consider in the process?

   Is the process logical?

6. What are the four main motion types and how does the robot move during each?

   A: There are four primary motion types in robotics: joint, linear, circular, and weave. Joint motion is point-to-point motion where all the axes involved move either as fast as they can or at the speed of the slowest axis with no correlation between the separate axes involved. Linear motion is where the controller moves all the axes involved at set speeds to insure straight-line motion. Circular motion is the formation of arcs and full circles as

described by no less than three points. Weave motion is straight line or circular motion that moves from side to side in an angular fashion while the whole unit moves from one point to another.

7. Which types of motion can joint movement generate between points? What is a potential danger from this motion?

   A: The tooling may curve, dip, and move in odd ways due to the amount of distance each axis moves. Joint movement could cause a crash with various objects in the work envelope due to this unexpected motion.

8. How do you create arcs and circles using circular motion commands?

   A: For arcs or part of a circle, you need to teach at least three points: the start point, a middle point, and an end point. For a full circle, you will need at least four points, but five or more is better with a start point, a point every quarter of the circle, and an end point that is the same or just past the start point along the circle.

9. What is the difference between a fine termination of a motion command and a continuous termination of a motion command?

   A: Fine termination stops the robot at the end of the move and adds approximately 500–750 ms to the cycle time every time the robot stops. Continuous terminations are advantageous for several reasons. Since the robot does not come to a complete stop, continuous terminations are less likely to significantly increase cycle time.

10. What is robotic kinematics?

    A: In the world of robotics, kinematics is the analytical study of motion of a robot manipulator and the basis for the complex math that goes into the movement of modern robotics with multiple axes or unique configurations, such as the delta style robots.

11. What are staging points and how are they typically used?

    A: Staging points are positions that get the robot close to the desired point, but are a safe distance away allowing for clearance and rapid movement. I highly recommend adding a staging point before and after any precision movements as coming in too fast or leaving too abruptly can cause problems.

12. What is singularity and how can we avoid this condition?

    A: Singularity is a condition in robotics where there is no clear-cut way for the robot to move between two points. This is the result of lining up two axes, such as four and five, in a straight line with one of the axes at zero degrees, usually axis 5, where the robot could go two different ways to reach the programmed point. To start with, careful planning of robot motion that avoids changes in wrist orientation goes a long way towards avoiding singularity. Mounting the tool so that it is offset or does not come straight out of the face plate helps to keep a bend in the arm and avoid the zero-point transition. If you have a singularity come up in your program, moving the problem axis around ten to fifteen degrees will usually correct the situation. Joint motion is the only motion type that is immune to singularity and thus your only option if none of the other fixes are possible.

13. What is the purpose behind using subroutines? Give some examples.

    A: The purpose of using subroutines is to reduce the lines of code in a program and to make it easier to write programs. Some examples of subroutines are opening or closing grippers, tooling change, alarm response actions, and other such actions that support the operation of the robot.

14. What is the difference between a global function and a local function?

    A: Global functions are variables, subroutines, and other code or data accessible by any program you create on the robot. Local function means only one program can access the data.

15. What is a frame and where is the origin point for the World frame of a robot?

    A: A frame is a set of three planes at right angles to each other and where the three planes intersect is the origin of the frame. Modern industrial robots come standard with a World frame that is a Cartesian system based off a point in the work envelope where the robot base attaches.

16. How does the right hand rule work?

    A: Orientate yourself in the same direction the robot is facing and using your right hand stick your thumb up, point your forefinger straight forward and point your middle finger towards the left side of your body at a ninety-degree angle to the forefinger. Your thumb is pointing in the positive direction for the Z-axis, your forefinger is pointing in the positive direction for the X-axis and your middle finger is pointing in the positive direction for the Y-axis. Bear in mind this works with the World frame and you must

orientate yourself in the same direction as the robot or looking from the robot towards the work envelope, not facing the robot from the other side of the work envelope.

17. What is the TCP? What is the default position for the TCP? Which of the frames use the TCP as the origin point?

    A: A critical element in setting a tool frame is the location of the Tool Center Point (TCP). The TCP is the reference point that your robot is moving through space and when you set a point, you are basically storing information that records the location of the TCP. Most robots use the center of the faceplate at the end of the wrist as a default TCP.

18. Which problems may potentially arise when you change the frame of a program after saving points?

    A: If you change the reference frame after creating the program and saving points, there is a high probability that your taught points will change and this could be a recipe for disaster.

19. What are the seven common categories of program control instructions and what is the basic function of each?

    A: Branching instructions cause the program to jump to another point in the program and fall into two broad types: conditional branching and unconditional branching instructions. Looping instructions cause the program to repeat a series of instructions either for a specified number of times or until a specific program condition is met. Register instructions allow the programmer to use arithmetic registers to store and manipulate data. Input/Output instructions are used to monitor conditions of and cause action in the robotic system as well as whatever else it is connected to. Arithmetic instructions are

used to perform arithmetic operations. A call (CALL) instruction is used to call other programs from a main program and can be considered as a type of unconditional branching instruction. A data structure is a collection of related items used to control a process, often arranged in a table format.

20. What is the difference between step mode testing and continuous mode testing?

    A: In step mode, the robot advances through the program one line at a time and requires you to press a button on the teach pendant keypad before it reads the next line of the program and responds. Continuous mode works as the name implies. Once you start the program in manual continuous mode it runs through the program until you stop it by releasing a button (the dead man switch), it reaches the end of the program, or something causes the system to alarm out.

21. What typically happens when you are testing in manual mode and the robot has to make a large axial movement? What is the danger associated with this?

    A: If you run into this, most systems will give you a warning alarm and once you accept or hit whatever is required to authorize the movement, the next movement the robot makes will be at or near full speed.

22. What should you do when running a program in automatic mode for the first time?

    A: When running a new or edited program for the first time in automatic, make sure you watch the robot go through the program a couple of times before you consider it ready to run with limited supervision. I would also recommend keeping your hand near the E-stop or some other stop button during that first run in automatic, in case something does go wrong.

23. What are some common tasks that fall under file maintenance?

A: You want to save the data you like, make sure you can find and use it when you need it, and delete the data you no longer want.