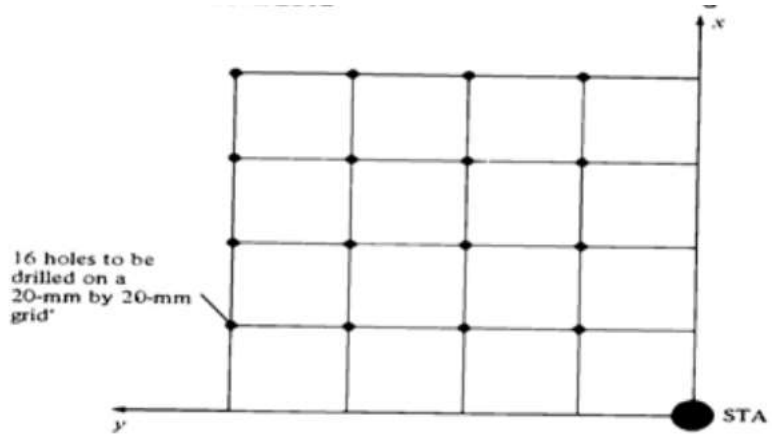
Unit-III: VAL-II

Q.1 Differentiate between the command structure of VAL-I and VAL-II language in Robot Programming.

Q.2 Explain how the sensory instructions for vision system using VAL-II.

Q.3 Develop a program using VAL II robot programming language for a PUMA 560 robot when setting output signal at 105th port and 5V voltage supply in controller it unloads a cylindrical part of 10mm diameter, from Machine1 positioned at point P1 with coordinates (200,250,0)mm and orientation(0,90,0)° and load the part on Machin2 positioned at P2 with coordinates(200,250,50)mm and orientation (0,90,0)°. The speed of robot motion is 40 in./s. However, because of safety precautions, the speed is reduced to 10 in./s while moving to a machine for an unloading or loading operation.

Q.4 Suppose we want to drill 16 holes according to the pattern shown in the figure below:



Write a program for drilling in a pattern of grid as shown in figure above using VAL-II Language. The speed during executing the motion will be 20% of normal speed.

Q.5 Explain the following instruction in VAL-II with example:

i. MOVET
ii. DRIVE
iii. APPRO
iv. DELAY
v. LISTP
vi. EXECUTE
vii. RETRY
viii. ENABLE
ix. GRASP
x. LISTL
xi. PCABORT
xii. RENAME
xiii. DISABLE

Q.6 Explain various program instructions used in VAL-II.

Q.7 Explain various motion instructions in VAL-II.

Q.8 Explain various monitor command instructions used in VAL-II.

Unit-IV: RAPID Language

Q.1 Define Data Type. Explain any *four* data type used in RAPID language with the help of examples of programs.

Q.2 Explain the Position Instructions and Input/ Output Signal Instructions in RAPID with the help of examples of programs.

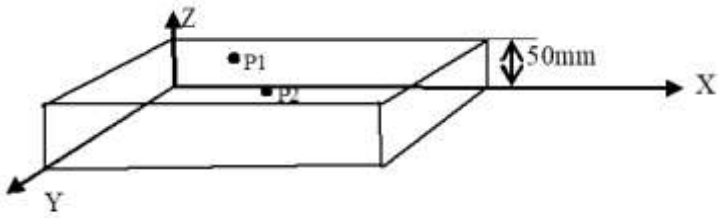
Q.3 Define Motion Command. Explain at least four Move Motion Commands used in RAPID language. Explain with examples for each Motion Command.

Q.4 Define function. Explain any *four* functions used in RAPID with the help of examples of programs.

Q.5 Explain the following instructions in RAPID with the help of examples of programs:

i. AccSet
ii. SetDO
iii. MoveAbsJ
iv. ISignalDO
v. WaitDO
vi. MoveL
vii. ClearPath
viii. GripLoad
ix. TriggL
x. StopMove
xi. CONNECT
xii. IDelete

Q.6 Develop a program using RAPID robot programming language using RAPID procedure for drilling operation from point P1(150,200,50) to P2(200,350,50) such that both the holes are of 10mm diameter and with depth of 50mm. While executing the program the orientation of end effector remains same as (0,90,0) °.



Q.7 Develop a program using RAPID robot programming language using RAPID procedure for pick and place operation from point P1(500,500,50) to P2(-500,500,50) such that it starts from *home* position and ends at *home* position. While executing the program the orientation of end effector remains same as (0,90,0) °.

Q.8 Differentiate between TRAP Routine and PROC Routine in RAPID Language.

Unit-V: AML (A Manufacturing Language)

Q.1 Define Motion Command. Explain at least three Move Motion Commands used in AML. Explain with examples for each Motion Command.

Q.2 Explain the following program used in AML for gripper:

```
GRASP: SUBR(GRIPPER_OPENING,<MIN_OFS,MAX_OFS>,F);
   TOGO:   NEW REAL;
   FMONS: NEW APPLY($MONITOR,PINCH_FORCE(F));
    CLEANUP($CLN);

    MOVE(GRIPPER,GRIPPER_OPENING+MIN_OFS,FMONS);

    IF QMONITOR(FMONS(1)) EQ O THEN
       BEGIN
       IF QMONITOR(FMONS(2)) EQ O THEN RETURN('TOO SMALL');
       TOGO = GRIPPER_OPENING+MIN_OFS-QPOSITION(GRIPPER);
       DMOVE(XYZ#<GRIPPER>,
              (TOGO/2*(HANDFRAME)(2,2))#<TOGO>,FMONS(1));
       END
    ELSE IF QMONITOR(FMONS(2)) EQ O THEN
       BEGIN
       TOGO = GRIPPER_OPENING+MIN_OFS-QPOSITION(GRIPPER);
       DMOVE(XYZ#<GRIPPER>,
              (-TOGO/2*HANDFRAME)(2,2))#<TOGO>,FMONS(2));
       END;

    RETURN( IF QPOSITION(GRIPPER)
                  LE GRIPPER_OPENING+MAX_OFFSET THEN'OK'
            ELSE 'TOO BIG');

  CLN: SUBR;
       ENDMONTIOR(FMONS);
       END;
    END;

PRESENCE: NEW <LED,1,ON,ON>;
NO_PRESENCE: NEW <LED,1,ON,ON>;


PINCH_FORCE: SUBR(F);
    RETURN(<<SLP,SRP>,
              1,O,F>>);
    END;

ANY_FORCE: SUBR(F);
    RETURN(PINCH_FORCE(F)
              #SIDE_FORCE(F)#TIP_FORCE(F));
    END;

NO_SENSING: NEW <>;

APPROACH_MOVE(OBJECT_PLACE,<0,0,0>,3.5,PRESENCE);
APPROACH_MOVE(HANDFRAME,<0,0,-1>,3.5,
                  TIP_FORCE(3.0*OZ),SLOWLY);
FINAL_MOVE(HANDFRAME,<0,0,.1>,3.5,NO_SENSING);
GRASP(1.5,<-.1,.1>,16.0*OZ);
```

Q.3 Explain the following program used in AML for palletization:

```
PALLET: SUBR(COUNTS,SPACING,PLACE);
   WHERE: NEW NILTRANS;
     IF ?PLACE THEN
        (IF AGGSIZE(PLACE) EQ 3 THEN &WHERE(1) ELSE &WHERE) =PLACE
     RETURN(<<1,1>,
                COUNTS,
                SPACING,
                WHERE,
                WHERE>);

     END;

-----------------------------------------------------------------------------

PALLET_GOAL: SUBR(!P);
     RETURN(P(5));
     END;

-----------------------------------------------------------------------------

INDEX_PALLET: SUBR(!P);
     IF P(1,1) LT P(2,1) THEN
        P(1,1) = P(1,1) + 1
     ELSE IF P(1,2) LT P(2,2) THEN
        P(1) = <1,P(1,2)+1>
     ELSE RETURN('EXHAUSTED');
     P(5,1) = DOT((P(1)- 1)*P(3)#<0>,P(4,2))+ P(4,1);
     RETURN('OK');
     END;
-----------------------------------------------------------------------------

RESET_PALLET: SUBR(!P,NEW_INDICES,NEW_LOC);
     IF ?NEW_LOC THEN
        IF AGGSIZE(NEW_XF) EQ 3 THEN
           P(4,1) = NEW_LOC
        ELSE P(4) = NEW_LOC;
     IF ?NEW_INDICES THEN

        IF NEW_INDICES GT P(2) OR NEW_INDICES LE O THEN
           RETURN('ILLEGAL_BOUNDS');
        ELSE P(1) = NEW_INDICES

     P(5,1) = DOT((P(1)-1)*P(3)#<O>,P(4,2))+ P(4,1)

     RETURN('OK');
     END;
```

Q.4 Explain the following instruction in AML with example:

i. ACCEL
ii. WAITMOVE
iii. SETTLE
iv. QGOAL
v. DEFIO
vi. ENDMONITOR
vii. AMOVE
viii. DMOVE
ix. EOD
x. QMONITOR
xi. ERASE
xii. PRINT

xiii. DECEL
xiv. STOPMOVE
xv. DELIO
xvi. REMONITOR

Q.5 Define Dynamic Variable. How dynamic variable is used in AML?
Q.6 Explain various sensor instructions used in AML?

Q.7 Explain the following code & output when executed in AML:

i.
```
MONITOR (LED, 2,0,0,1.5, 'passed');
MOVE (ARM, fgoal, LED);
```

ii.
```
ATTN: SUBR;
MOTPARMS: NEW STOPMOVE;
WAITMOVE;
BREAK (EOL, 'ATTENTION REQUESTED');
APPLY ('AMOVE', MOTPARMS);
END;
```

iii.
```
DMOVE (<4,5,6>,<30,-60,90>);
```

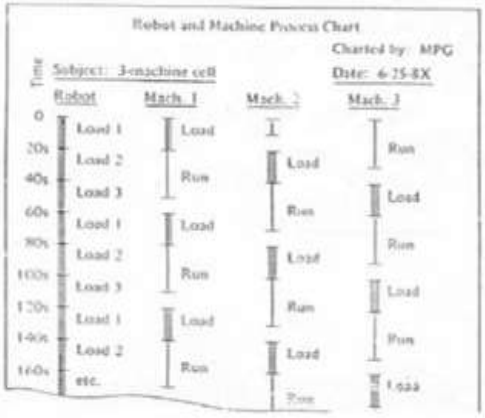
iv.
```
SPEED (0.8);
```

v.
```
IF SENSIO (LED, 0) THEN
BREAK ('Part gripped');
```

vi.
```
TRAP: SUBR(NUM, DATA);
DISPLAY ('ERROR NUMBER=', NUM, EOL);
IF? DATA THEN
DISPLAY ('ERROR DATA=', DATA, EOL);
BRANCH(RETRY);
END;
ERRTRAP('TRAP');
```

Unit-VI: Practical Study of Virtual Robot

Q.1 A three machine cell robot is used to load and unload machine. Each of the three machines are identical with identical cycles of 50s. This type of cycle time is divided between the run time (30s) and service time(load/unload) by the robot (20s). The organization of cycle time is shown in the robot and machine process chart given below:



It can be seen that each machine has idle time during its cycle of 10s while, the robot is fully occupied throughout its work cycle. Find:

a) Total idle time of robot
b) Total cycle time of robot
c) Machine Interference

Q.2 Write a short not on ‘Multiple Robot and Machine Interfacing’.
Q.3 Define Virtual Robotics. Explain the various user interfaces in Robot Studios for various applications.
Q.4 Define the concept of singularities. Explain the methods of detecting possible collision of robots and what are the features added to avoid it.
Q.5 Explain various types of singularities that can occur in robots.
Q.6 Write a short note on ‘Robot Economics’.
Q.7 Explain in detail about ‘Robot cycle time analysis’.
Q.8 Explain ‘the repeatability measurement of robot’.