Programarea si simularea offline a unei celule robotizate pentru aplicatii de tip pick and place utilizand mediul de lucru V-REP. Programarea pe baza de semnale.

OFFLINE PROGRAMMING AND SIMULATION OF A CELL FOR PICK AND PLACE APPLICATIONS USING THE V-REP WORKING ENVIRONMENT. SIGNAL-BASED PROGRAMMING.

Batman Gökhan

Facultatea:I.I.R, Specializarea:Robotica, Anul:II, e-mail:batmangokhan31@gmail.com

Conducător ştiinţific: Prof.dr.ing. Anania Dorel

Rezumat: This study comprises making a 3D cell in the V-rep working environment. In the presented cell, the pick and place application is made by a simple portal robot. He takes a box from the conveyor, puts it in the assembly machine where the box lid is placed. Will be presented the application programming and its operation.

1.Introducere

Obiective tinta urmarite in lucrare sunt acelea de a realiza programarea celulei robotizate si simularea acesteia in mediul de lucru V-rep.

2.Studiu actual

Pentru fundamenarea temei de cercetare a fost folosita o aplicatie robotizata.Aceasta se prezinta dupa cum urmeaza mai jos folosind o vedere de tip drafting si o imagine reala

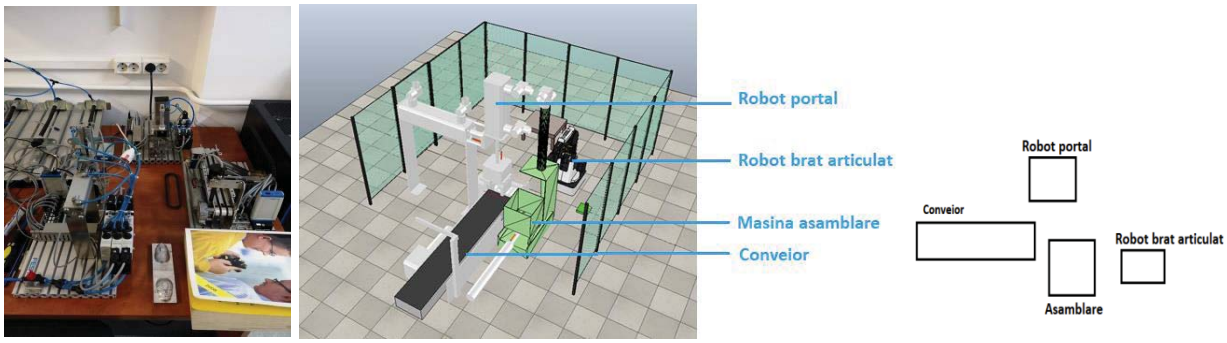

Fig.1.Prezentarea elementelor

Aceasta cuprinde un robot de tip portal simplu de la firma Festo. Aplicatia este una de pick and place. In aceasta aplicatie primul element intra pe conveior iar al doilea element este asamblat cu ajutorul dispozitivului de asamblare. Cel de-al doilea element este pus pe banda de urn robot. Piesa asamblata este o cutie la care se adauga un capac. In urmatoarea imagine este prezentata celula 3D asamblata in mediul de lucru V-rep.

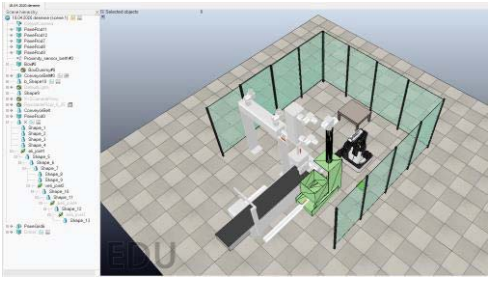

Fig.2.Aplicatia realizata

Definerea semnalelor

Pentru celula studiata avem ca si desfasurare a activitatilor urmatoarea:

Prima activitate este aceea de aprovizionare a celulei cu cutii. Aceasta aprovizionare este facuta cu ajutorul benzii transporotare. Acolo ea este sortata dupa culoare de catre un senzor vision acesta comunica cu alt senzor de la robot ce vizualizeaza capacul ce trebuie presat iar daca culoarea acestuia corespunde cu cea de pe banda cutia trece mai departe si este luata de al doilea robot. Daca culoarea capacului nu corespunde cu cea a cutiei atunci cutia este inlaturata de pe banda cu ajutorul unui mecanism.

Se va detalia sortarea cutiilor:

-daca pe banda ajunge cutia rosie iar la robot este capacul rosu atunci cutia trece

- daca pe banda ajunge cutia neagra iar la robot este capacul rosu atunci cutia nu trece

- daca pe banda ajunge cutia neagra iar la robot este capacul negru atunci cutia trece

- daca pe banda ajunge cutia neagra iar la robot este capacul rosu atunci cutia nu trece

Daca trece cutia de banda mai departe aceasta este luata de robotul portal si pusa mai departe in presa. Dupa ce este presat capacul robotul ia cutia si o pune in stiva

Semnalele celulei studiate sunt semnale de intrare si de iesire. Ca si semnale se vor folosi urmatoarele:

Tabel1.Definirea semnalelor

Semnale de intrare DI	Semnale de iesire DO	Senzorul pentru banda	Senzorul pentru presa
-senzor banda -senzor presa -robot brat -robot portal	-senzor banda -senzor presa -robot brat -robot portal	-senzor banda cutie culoare rosie -senzor banda cutie culoare neagra	-senzor presa capac culoare rosie -senzor presa capac culoare neagra

In continuare se vor detalia operatiile realizate in aplicatia studiata si programarea acestora astfel incat sa se poata realiza simularea offlie a acesteia.

Prezentarea banzii transportoare si senzor. Pe aceasta se afla un senzor de proximitate si unul vision pentru detectarea culorii obiectelor si a intrarii acestra pe bada.

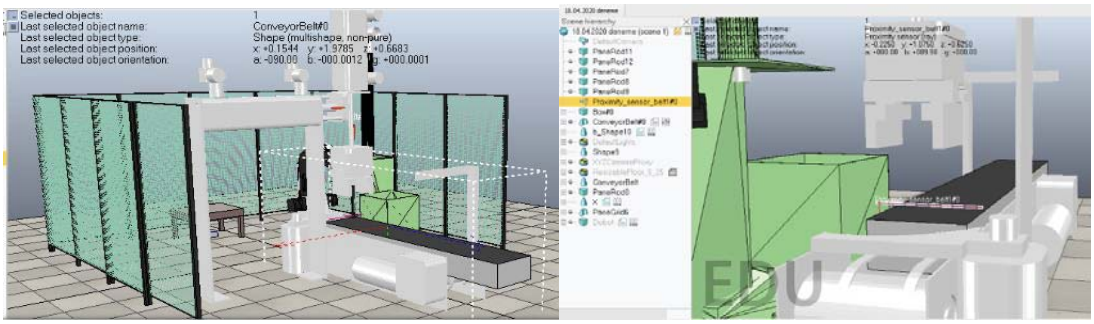

Fig.3.Prezentarea benzii transportoare si a senzorului aflat pe aceasta

Programarea. In continuare se va prezenta programarea pentru functionarea benzii transportoare.

In prima parte respectiv prima si a doua imagine de cod se va specifica viteza conveiorului,coordonatele,lungimea etc si inserarea obiectului pe banda.

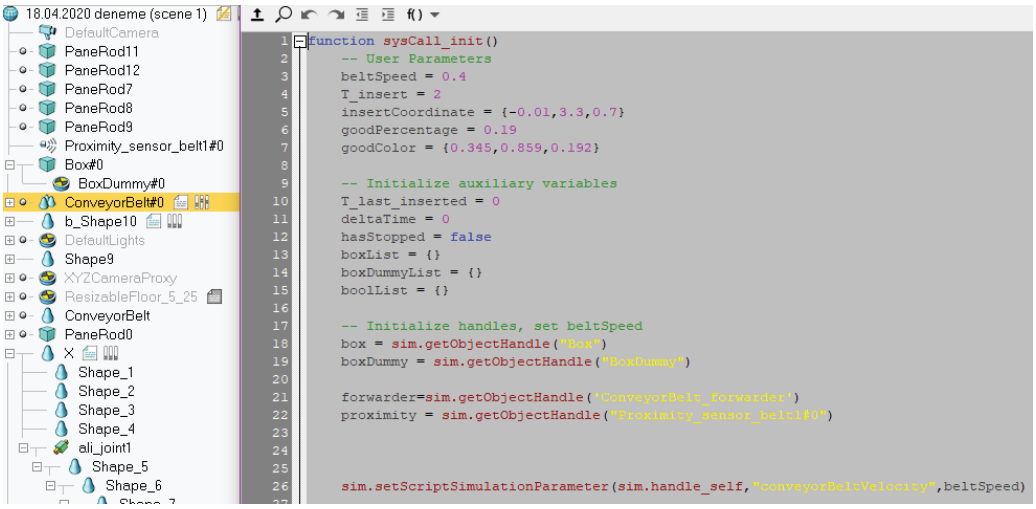

Fig. 4.Creare cod pentru programarea benzii

In imaginea urmatoare se va stabili functia de citire a obiectului daca semnalul este 0 atunci nu se citeste nici un obiect daca semnalul este 1 atunci senzorul citeste obiectul ce intra pe banda. Si se verifica daca se poate insera o noua cutie pe banda.

“”function sysCall_sensing()

-- Read vision sensor (0= nothing detected, 1 = object detected)

local res = sim.readProximitySensor(proximity)

-- Check if possible to insert an new box

if (sim.getSimulationTime()-T_last_inserted > T_insert) and not hasStopped then

insertBox()

end””

```
function sysCall_sensing()
    -- Read Proximity sensor (0= nothing detected, 1 = object detected)
    local res = sim.readProximitySensor(proximity)

    -- Check if possible to insert an new box
    if (sim.getSimulationTime()-T_last_inserted > T_insert) and not hasStopped then
        insertBox()
    end

    -- If proximity sensor detects an object, stop the belt, stop inserting objects
    if res == 1 and not hasStopped then
        if boolList[1] then
            sim.setScriptSimulationParameter(sim.handle_self,"conveyorBeltVelocity",0)
            deltaTime = sim.getSimulationTime()-T_last_inserted
            hasStopped = true
```

Fig. 5.Creare cod pentru programarea benzii

```
            sim.setIntegerSignal("objectAvailable",1)
        else
            local box = table.remove(boxList,1)
            local boxDummy = table.remove(boxDummyList,1)
            table.remove(boolList,1)

            sim.removeObject(box)
            sim.removeObject(boxDummy)
        end
    end

    -- If proximity sensor detects nothing and belt has stopped, start belt, continue inserting
    if res == 0 and hasStopped then
        sim.clearIntegerSignal("objectAvailable")
        sim.setScriptSimulationParameter(sim.handle_self,"conveyorBeltVelocity",beltSpeed)
        hasStopped = false
        T_last_inserted = sim.getSimulationTime()-deltaTime
    end
end
```

Fig. 6. Creare cod pentru programarea benzii

In cotinuare se va prezenta programarea robotului de tip portal simplu,pneumatic.

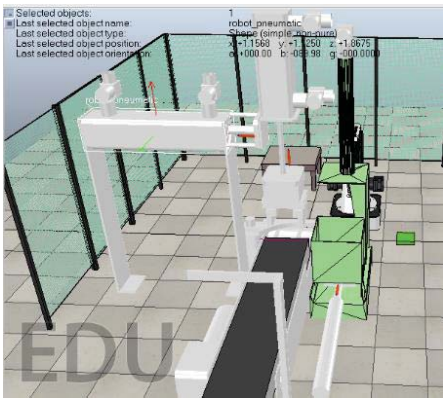

Fig.7. Prezentare robot de tip portal simplu

Robotul primeste semnalul si incepe sa se deplaseze pentru a prelua piesa. Daca senzorul detecteaza un obiect in timp ce exista deja un obiect in presa pentru asamblare robotul nu ia piesa.

Non-threaded child script (ConveyorBelt#0)

```
function sysCall_cleanup()

end

function sysCall_actuation()
    beltVelocity=sim.getScriptSimulationParameter(sim.handle_self,"conveyorBeltVelocity")


    relativeLinearVelocity={beltVelocity,0,0}

    sim.resetDynamicObject(forwarder)

    m=sim.getObjectMatrix(forwarder,-1)
    m[4]=0
    m[8]=0
    m[12]=0
    absoluteLinearVelocity=sim.multiplyVector(m,relativeLinearVelocity)

    sim.setObjectFloatParameter(forwarder,sim.shapefloatparam_init_velocity_x,absoluteLinearVelocity[1])
    sim.setObjectFloatParameter(forwarder,sim.shapefloatparam_init_velocity_y,absoluteLinearVelocity[2])
    sim.setObjectFloatParameter(forwarder,sim.shapefloatparam_init_velocity_z,absoluteLinearVelocity[3])
end


function sysCall_sensing()

    local res = sim.readProximitySensor(proximity)
```

Fig.8. Creare cod pentru programarea robotului portal simplu

Se va prezenta in imagine robotul de tip brat articulat.

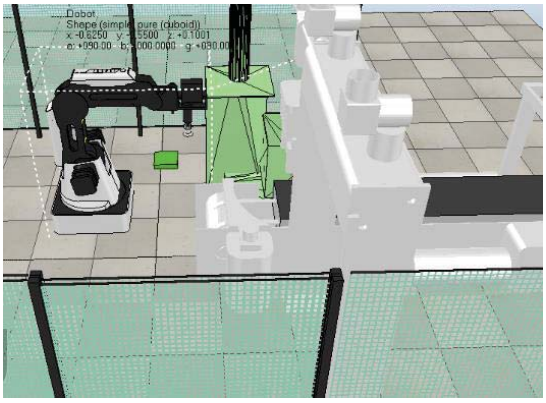

Fig.9. Prezentarea robotului tip brat articulat

Robotul primeste semnal ca a fost introdusa cutia si porneste sa ia capacul corespunzator culorii cutiei din presa.Pntru realizarea acestei operatii a fost facut urmatrul cod.

Main script

```
script=sim.getScriptHandle('ConveyorBelt#0')
function movetoPos(j1,j2,j3,j4,enable)
    modelBase=sim.getObjectAssociatedWithScript(sim.handle_self)
    modelName=sim.getObjectName(modelBase)

    motorHandles = {}
    j={j1*math.pi/180,j2*math.pi/180,j3*math.pi/180,j4*math.pi/180}

    for i=1,4,1 do
        motorHandles[i]=sim.getObjectHandle('Dobot_motor'..i)
    end
    for i=1,4,1 do
        sim.setJointTargetPosition(motorHandles[i],j[i])
        sim.wait(2)
    end

    if enable then
        sim.setIntegerSignal(modelName .."_suctionCup",1)
    else
        sim.setIntegerSignal(modelName .."_suctionCup",0)
    end
end

function sysCall_threadmain()
    --Motor1,Motor2,Motor3,Motor4,Suction Cup
    sim.waitForSignal('objectAvailable',1)
    sim.wait(25)
    movetoPos(50,50,65,0,true)
    movetoPos(50,0,0,0,true)
    movetoPos(0,90,0,-40,false)
   -- movetoPos(50,0,0,0,false)
    movetoPos(0,0,0,0,false)
end
```

Fig. 10. Creare cod pentru programarea robotului brat articulat

Pentru detectarea piesei pe banda a fost programat senzorul prin urmatoarul cod.Dupa efectuarea acestuia se va realiza asamblarea capacului.

```
Non-threaded child script (ConveyorBelt#0)
104 function insertBox()
105
106     local rand1 = math.random()
107     local rand2 = math.random()
108     local rand3 = math.random()
109
110
111     local dx = (2*rand1-1)*0.1
112     local dy = (2*rand2-1)*0.1
113     local dphi = (2*rand3-1)*0.5
114     local disturbedCoordinates = {0,0,0}
115     disturbedCoordinates[1] = insertCoordinate[1]
116     disturbedCoordinates[2] = insertCoordinate[2]
117     disturbedCoordinates[3] = insertCoordinate[3]
118
119
120     local insertedObjects = sim.copyPasteObjects({box,boxDummy},0)
121
122
123     T_last_inserted = sim.getSimulationTime()
124
125
126     sim.setObjectPosition(insertedObjects[1],-1,disturbedCoordinates)
127     sim.setObjectOrientation(insertedObjects[1],-1,{0,0,0})
128
129     table.insert(boxList,insertedObjects[1])
130     table.insert(boxDummyList,insertedObjects[2])
131
132
133     local decision = math.random()
134     if decision <= goodPercentage then
135
136         sim.setShapeColor(insertedObjects[1],nil,sim.colorcomponent_ambient_diffuse,goodColor)
137         table.insert(boolList,true)
```

Fig.11. Creare cod pentru programarea senzorului de detectare a piesie de pe banda

Realizarea asamblarii capacului este facuta de o masina de asamblare ce contine si un senzor de detectare a cutiei visual.Aceasta are un sistem de translatie care impinge capacul pentru al pune pe cutie.

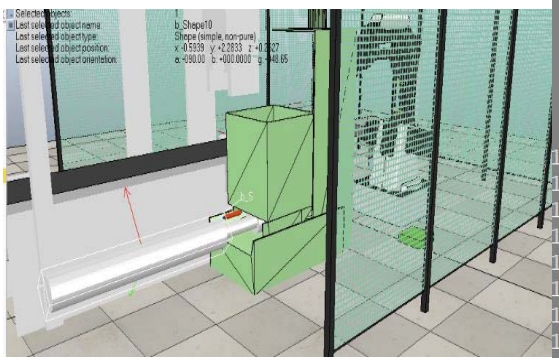

Fig.12.Prezentarea masinii de asamblare

```
1  script=sim.getScriptHandle('ConveyorBelt#0')
2  function sysCall_threadmain()
3      sim.waitForSignal('objectAvailable',1)
4
5       b=sim.getObjectHandle('b_joint1')
6
7
8      sim.setJointTargetPosition(b,0)
9
10     sim.wait(32)
11
12     sim.setJointTargetPosition(b,-3)
13     sim.wait(10)
14     sim.setJointTargetPosition(b,0)
15
16
17 end
```

Fig.13. Creare cod pentru programarea masinii de asamblare a capacului

Se va prezenta in continuare schema electica pentru aplicatia studiata si detalierea ei.

Cand este apasat butonul electric electricitatea ajunge la releu.Electricitatea care ajunge la releu inchide intrerupatoarele conectate la acesta. Are rolul de a-l deschide pe cel inchis.Mai departe electricitatea, supapele de cale sunt transmise electrovalvei deschizand sau inchizand supapa care ar trebui sa actioneze acolo .

Cand pistonul care incepe sa se miste atinge distanta prestabilita, cheia cu role o transmite pe intrerupator si activeaza cheia care trebuie inchisa sau deschisa.Logica generala a sistemului se bazeaza pe acest lucru.In figura de mai jos se regasete schema electrica si explicatia elementelor .

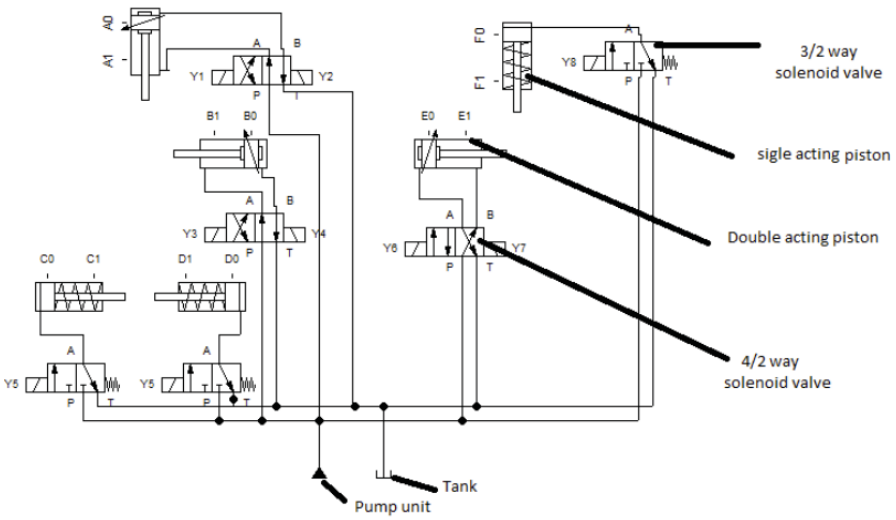

Fig.14.Schema electrica a aplicatiei studiate si explicatia ei.

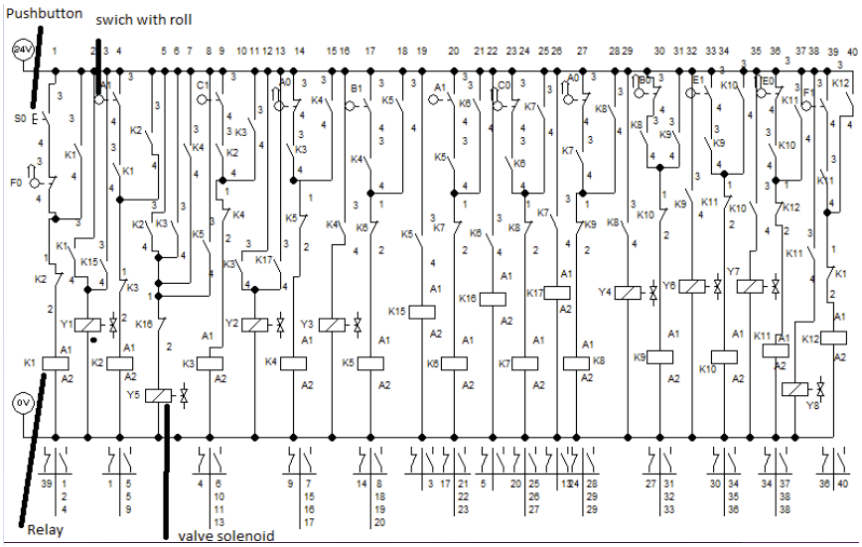

Fig15.Schema electrica a aplicatiei studiate si explicatia ei.

Capturi din simularea video realizata in V-rep

In urmatoarea imagine se prezinta intrarea cutiilor pe banda si preluarea acestora de catre robotul protal.

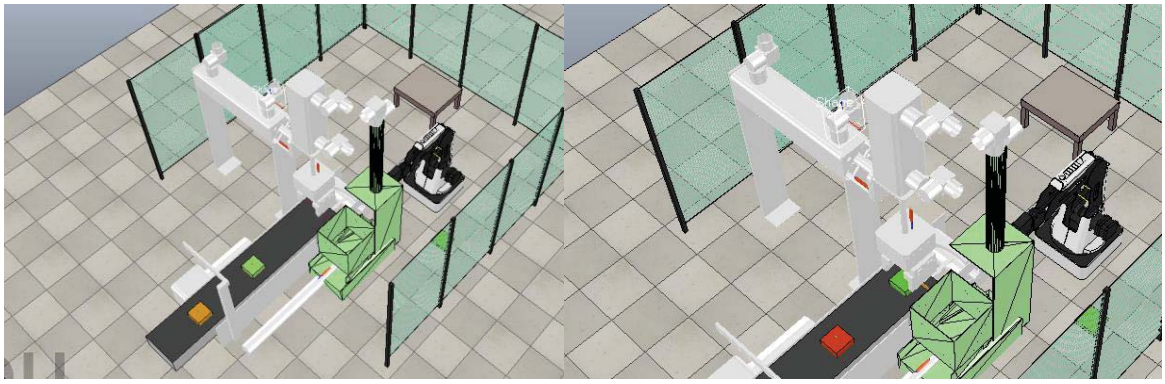

Fig.16.Intrarea si preluarea cutiilor de pe banda

In urmatoarele imagini este prezentat robotul protal cum ia cutia si o duce in masina de asamblarea a capacului.

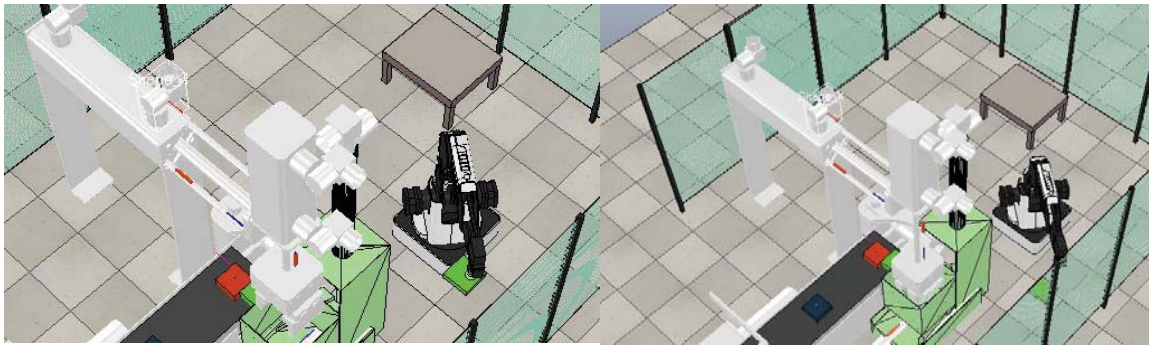

Fig.17.Punerea cutiei in masina de asamblare si preluarea capacului de catre robotul brat articulat

Capacul este luat de robotul tip brat si pus in masina de asamblare.Masina de asamblaere actioneaza o tija pentru a putea pune capacul pe cutie.

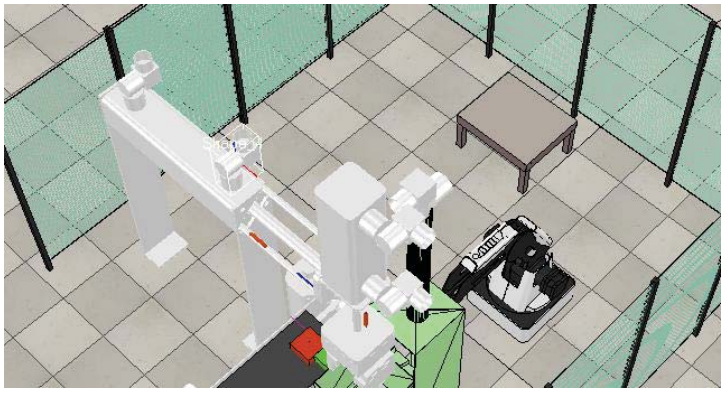

Fig.18.Punerea capacului in masina de asamblare de catre robotul brat articulat si asamblarea cutiei

Capitolul 4. Concluzii

Contributiile originale in aceasta lucrare sunt reprezentate de realizarea 3D a elementelor celulei in mediul de lucru Solidworks si importarea acestora in mediul de lucru V-rep,unde a fost creat ansamblul.Elementele realizate sunt:robotul portal simplu,conveiorul personalizat si presa ce asambleaza capacul cutiei de cutie.In continuarea cercetarii se va realiza simularea 3D a celulei robotizate.Pentru functionarea corecta a aplicatiei se vor folosii senzori ce vor fi programati astfel incat sa tansmita semnale digitale.Se va realiza programarea offline a aplicatiei si simularea acesteia pe baza de semnale si cod.

Bibliografie.

1. Bucuresteanu A. – Actionarea Pneumatica a Robotilor Industriali, note de curs UPB, 2017
2. Bucuresteanu A. – Elemente si sisteme pneumatice pentru actionarea robotilor industriali, Editura Printech, ISBN 978-606-23-0081-4, Bucuresti 2013.
3. Nicolescu, A., Roboti Industriali – Vol.1 Subsisteme si ansambluri componente. Structura axelor comandate numeric ale RI, ISBN 973 – 30 – 1244 – 0, Editura Didactica si Pedagogica RA, 2005, Bucuresti de curs si metodologii de proiectare, UPB, 2018