

ТЕХНИКАЛЫҚ ЖҮЙЕЛЕР: ОҚЫТУ-ҮЙРЕТУ ЖҮЙЕСІНІҢ СҮЛБАЛАРЫН ЖАНДАНДЫРУ

Тұрсын Шолпан

sholpan_tursyn@mail.ru

Л.Н. Гумилев атындағы ЕҰУ, Физика-техникалық факультетінің студенті,

Нұр-Сұлтан, Қазақстан

Ғылыми жетекшісі – Әубәкір Д.Ә.

Кіріспе. Робототехника – жобалармен, құрылыспен, роботтардың жұмыс жасауын қадағалаумен айналысатын, автоматтандыру мақсатында пайдаланылатын электрониканың, механиканың және информатиканың бір саласы. Осы секілді компьютерлік ғылымдар басқару үшін қолданылады. Осы жұмыста мен таңдауларды орындайтын және тапсырмаларды орналастыратын реттелген резеңке тұтқасы бар интеллектуалды роботтың дизайнын қарастыруды ұсынамын. Бұл робот түрлі пішіндерге ие кез-келген нысандарды басып алу үшін пайдаланылуы мүмкін.

1 Жабдықтау талаптары

Қозғалтқыш драйвері қозғалтқыштардың жұмысын басқару үшін қолданылады.

Тұрақты ток қозғалтқыштарының айналуы L298N қозғалтқышының басқару тізбегі арқылы бақыланады.

Роботты басқару үшін шамамен 4 тұрақты токтың қозғалтқыштары қолданылады. Резеңке тұтқасының пішінін реттеу үшін 2 тұрақты ток қозғалтқыштары қолданылады. Тұрақты ток арқылы берілген нұсқауларға сәйкес, екі қозғалтқыш айналады. Осы нысанның арқасында белдік нысанның объектісіне сәйкес автоматты түрде реттеледі.

Алюминий тақтайшасы қолдың пальмасы сияқты әрекет ететін роботты қолымен жасауға арналған. Бұл тақтайшалар губкамен жабылған. Бұл губка роботтың қолы мен заттың арасындағы үйкелісті арттырады. Қысым қосқышы осы алюминий тақтайшасына қосылады.

Алюминий тақтайшасында төрт қысымды ажыратқыш бар. Тұрақты ток арқылы берілген нұсқауларға сәйкес және нысанның қозғалтқыштарының пішініне сәйкес айналады және резеңке тұтқасының нысанын реттейді. Резеңке тұтқасы объектіні тығыз ұстағанда, ол қысымды ауыстырғышқа қысым көрсетеді. Қысым қосқышы орнатылған орнына тұрақты ток қозғалтқыштарының өзгеруі мен айналуы автоматты түрде тоқтатылады.

Әдетте резеңке қол полисилоксалы көміртекті материалдан тұрады. Бұл резеңке қол икемді, ол нысанның объектісіне сәйкес өзін-өзі реттей алады. Резеңке белдікті тұрақты ток қозғалтқыштарына орнатады. Пәрменді дербес компьютер арқылы тасымалдайтын болсақ, тұрақты ток қозғалтқыштары айнала бастайды және белдік нысанның объектісіне сәйкес пішінін реттей бастайды. Таспа толығымен реттелгенде, қысым датчигі оны сезеді және қозғалтқыш айналуын тоқтатады.

2 Түсіру үдерісі

Таңдау және орналастыру процесі болып табылатын түсіру процесі келесі үш кезеңде жүзеге асырылады:

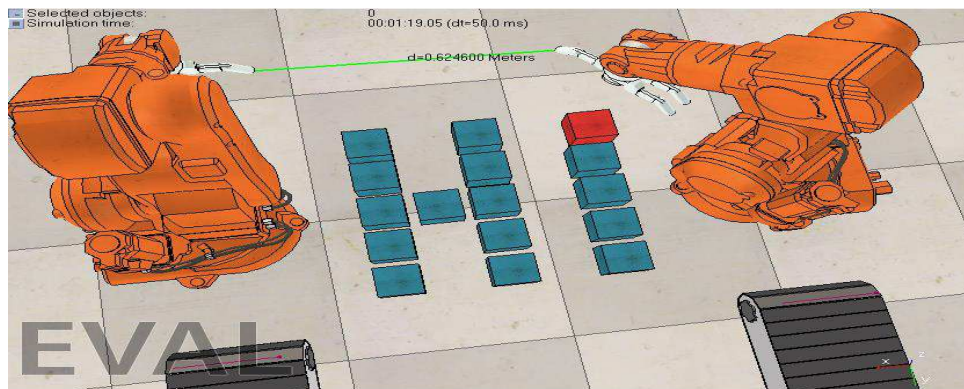
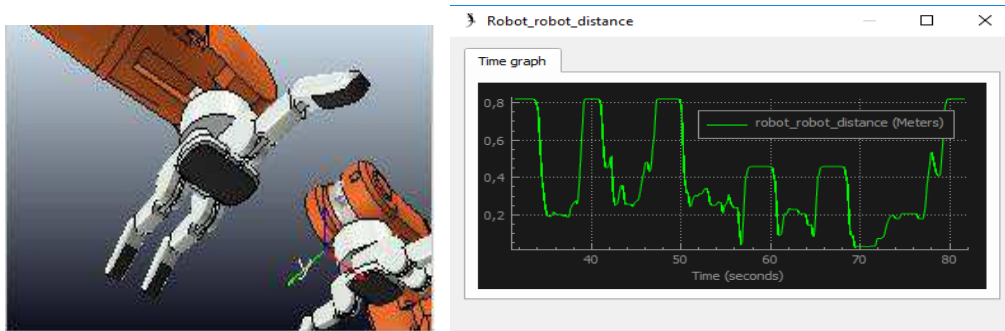
1. Жоспарлау қадамы
2. Түсіру қадамы
3. Босату қадамы

Барлық қалған есептер осы кезеңде аяқталады. Осы нақты параметр аяқталғаннан кейін процесс аяқталды, содан кейін нысан белгілі бір жерге орналастырылады.

3 V-REP-PRO программасы

Осы бағдарлама арқылы роботтардың сезіну процессін жасау.

Barret Hand Pick and Place (робот түрі)



V-REP-PRO, скетч:

```

function sysCall_init() // инициализация жасау
sim.handleSimulationStart() //резеңке қол орнату
sim.openModule(sim.handle_all) //модуль құрау
sim.handleGraph(sim.handle_all_except_explicit, 0) //резеңке қодың графигін //жасау
end
function sysCall_actuation() // роботты іске қосу
sim.resumeThreads(sim.scriptthreadresume_default)
sim.resumeThreads(sim.scriptthreadresume_actuation_first)
sim.launchThreadedChildScripts()
sim.handleChildScripts(sim.syscb_actuation)
sim.resumeThreads(sim.scriptthreadresume_actuation_last)
sim.handleCustomizationScripts(sim.syscb_actuation)
sim.handleModule(sim.handle_all,false)
simHandleJoint(sim.handle_all_except_explicit,sim.getSimulationTimeStep()) --
DEPRECATED
simHandlePath(sim.handle_all_except_explicit,sim.getSimulationTimeStep()) --
DEPRECATED
sim.handleMechanism(sim.handle_all_except_explicit)
sim.handleIkGroup(sim.handle_all_except_explicit)
sim.handleDynamics(sim.getSimulationTimeStep())
sim.handleMill(sim.handle_all_except_explicit)
end
function sysCall_sensing() // роботты зондау
-- put your sensing code here
sim.handleSensingStart()
sim.handleCollision(sim.handle_all_except_explicit)
sim.handleDistance(sim.handle_all_except_explicit)
sim.handleProximitySensor(sim.handle_all_except_explicit)

```

```

sim.handleVisionSensor(sim.handle_all_except_explicit)
sim.resumeThreads(sim.scriptthreadresume_sensing_first)
sim.handleChildScripts(sim.syscb_sensing)
sim.resumeThreads(sim.scriptthreadresume_sensing_last)
sim.handleCustomizationScripts(sim.syscb_sensing)
sim.handleModule(sim.handle_all,true)
sim.resumeThreads(sim.scriptthreadresume_allnotyetresumed)
sim.handleGraph(sim.handle_all_except_explicit,sim.getSimulationTime()+sim.getSimulationTimeStep())
end

```

```

function sysCall_cleanup() // оптимизация жасау
sim.resetMilling(sim.handle_all)
sim.resetMill(sim.handle_all_except_explicit)
sim.resetCollision(sim.handle_all_except_explicit)
sim.resetDistance(sim.handle_all_except_explicit)
sim.resetProximitySensor(sim.handle_all_except_explicit)
sim.resetVisionSensor(sim.handle_all_except_explicit)
sim.closeModule(sim.handle_all)
end

```

```

function sysCall_suspend() //жұмысты уақытша тоқтату
sim.handleChildScripts(sim.syscb_suspend)
sim.handleCustomizationScripts(sim.syscb_suspend)
end

```

```

function sysCall_resume() //қорытынды
sim.handleChildScripts(sim.syscb_resume)
sim.handleCustomizationScripts(sim.syscb_resume)
end

```

-- Мұнда қосымша жүйелік қоңырауларды анықтай аласыз:

```

function sysCall_beforeCopy(inData)//дейінгі копия
for key,value in pairs(inData.objectHandles) do
print("Object with handle "..key.." will be copied")
end
end
function sysCall_afterCopy(inData) //кейінгі копия
for key,value in pairs(inData.objectHandles) do
print("Object with handle "..key.." was copied")
end
end
function sysCall_beforeDelete(inData)
for key,value in pairs(inData.objectHandles) do
print("Object with handle "..key.." will be deleted")
end
-- inData.allObjects indicates if all objects in the scene will be deleted
end
function sysCall_afterDelete(inData)
for key,value in pairs(inData.objectHandles) do
print("Object with handle "..key.." was deleted")
end
end

```

-- *inData.allObjects* indicates if all objects in the scene were deleted
end

Қорытынды. Жоғарыда аталған барлық талқылау, біз озық роботталған қару коммерцияландыруға кедергі келтіретін мәселелер қаралды, және осы қиындықтарды жеңуге көптеген зерттеушілер қабылдаған әр түрлі тәсілдерді қарап, практикалық негізінде шешімін таптық, сондай-ақ, мысалы, қазіргі заманғы технологиялар, ол таңдау проблемасын шешу үшін қарапайым роботты ұсынды. Бұл робот бейімделу мен икемділіктің тиімді ерекшеліктеріне байланысты 95% табыстары бар нақты өмірдің түрлі нысандарын тұрақты түрде басып шығарады.

Қолданылған дереккөздер тізімі:

1. V-REP_PRO_V3_5_0_Setup_15926
2. Программирование с помощью STEP 7 v 5.3. Siemens AG, – 2004. –602с
3. Әубәкір Д.Ә. Жүйелер теориясының негіздері. Основания теории систем. Bases of system theory. Оқулық/ Учебник/ Textbook. Астана: ЕҰУ баспасы, 2011, 500 б.