Универзитет „Св. Кирил и Методиј“ – Скопје, Р. Македонија  
Природно – математички факултет  
Институт за информатика

Насока: Програмско инженерство

**  
Дипломска работа**

Компјутерска визија во роботика

|  |  |
| --- | --- |
| ***Ментор:***  *Доц. Д-р* | ***Студент:***  *број на индекс:* |

Скопје, 2012

|  |  |
| --- | --- |
| **Ментор:** | **Доц. Д-р** Природно – математички факултет, Институт за информатика |
| **Членови на комисијата:** | **Доц. Д-р**  Природно – математички факултет, Институт за информатика  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Природно – математички факултет, Институт за информатика  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Природно – математички факултет, Институт за информатика |
| **Датум на одбрана:**  **Научна област:** | \_\_\_\_.\_\_\_\_.\_\_\_\_\_\_\_\_ год.  Информатика – |

Содржина

**Апстракт4**

1. **Вовед5**
2. **Пребарување информации (IR)7**
   1. Почетоци на областа за пребарување информации7
   2. Дефинирање на систем за пребарување информации8
   3. Типови на модели за пребарување информации8
      1. Модел на множества9
      2. Модел на векторски простори10
      3. Модел на веројатности11
      4. Модел на јазик12
3. **Опис на системот14**
   1. Технички опис на системот14
   2. Дизајн на системот14
   3. Компоненти на системот16
   4. Опис на функционалноста на системот, компонентите и обработени модели и техники17
   5. Перформанси на системот23
4. **Експерименти и резултати од практичната примена на системот24**
   1. Колекцијата од документи и прашања за тестирање24
   2. Резултати и статистики од спроведените анализи25
      1. Резултати за моделот на векторски простори25
      2. Резултати за Okapi BM2529
      3. Резултати за јазичниот модел31
5. **Заклучок и идна работа35**

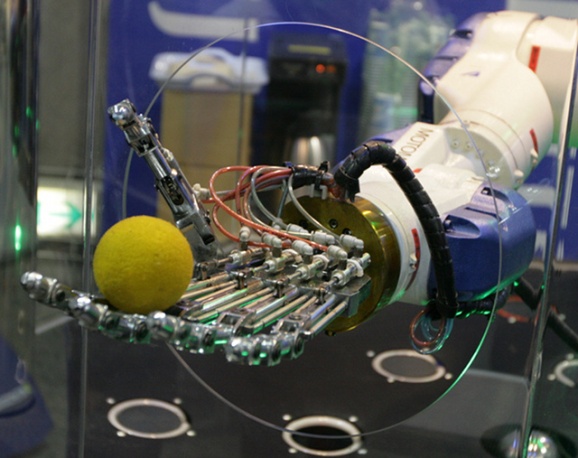
**Библиографија36**

1. Апстракт

Визијата е екстремно важно сетило за луѓето и роботите, обезбедувајќи детални информации за околината. Робустен систем за визија треба да биде во можност да детектира објекти со голема сигурност и да обезбеди точна репрезентација на светот за процесите на повисоко ниво. Системот за визија мора да биде ефикасен овозможувајќи агент со ограничени ресурси да одговори на промените во околината. Секоја слика добиена од дигитална камера мора да биде обработена во кратко време. Поради ова алгоритамската сложеност ограничена, вклучувајќи компромис помеѓу времето на обработка и квалитетот на добиените информации. Примена во роботика , системите за визија се главни уреди за перцепција и автономните роботи мора да бидат во можност да ги користат системите за визија за да може самите да се лоцираат во околината и да лоцираат објекти кои што треба да ги манипулираат.

1. Вовед

Повеќето копнени животни имаат способност да перцепираат објекти со различни димензии и форми а некои можат да ги препознаваат нивните карактеристики и да ги запомнат. Животните како и луѓето се во состојба да препознаваат објекти користејќи го нивното сетило за вид во околината во која се наоѓаат и успешно да манипулираат со нив користејќи ги нивните екстремитети. Инспирирани од природата предизвикот за роботските системи е насочен кон копирање на вештините на биолошките организми. Предизвикот се состои во препознавање на објекти во реално време од вештачки систем за визија и манипулирање со нив. Ваквиот систем треба да биде робустен и во ограничено време со ограничени ресурси да препознае објекти и да одучи кои акции ќе ги изврши за манипулирање со објектите. Роботските системи вклучуваат сетило за вид-дигитална камера и манипулатори -роботски раце. Некои од овие роботски системи можат да бидат и мобилни, со тенденција за постојано усовршување со цел да се постигне комплетна автономност на системот.



1. Опис на проблемот

Во работната околина на роботот може да се наоѓаат објекти со различна големина, боја и форма. Се поставува прашањето како да се препознаат објектите во работната околина на роботот, да се определи нивната релативна положба во однос на основата на роботот, да се најде нивната боја форма и ориентација и овие информации да се пренесат на процес на повисоко ниво. Овој процес во зависност од задачата треба да одлучи како ќе манипулира со дадените објекти. Пример да се сортираат објектите според дадена боја или форма. За да се реши овој проблем потребно е да се разгледаат неколку аспекти.

Дали дадената точка е во работната околина на роботот. Доколку дадената точка може да се достапи како да се најдат вредности за зглобовите на роботот да ја достигне целната точка. Доколку се најде решение за вредностите на зглобовите дали тоа решение е во границите на роботот. Ако решението е во границите на роботот системот треба да провери дали даденото решение нема да доведе до судир со некој објект на сцената. Овие aаспекти се разгледани во областа на кинематика каде се имплементирани различни техники и методи за да најдат решенија со дадените ограничувања.

Сега кога имаме кинематички модел кој што има можност за придвижување на роботот кон дадената цел со дадени ограничувања се поставува проблемот како да се детектираат објектите кои се наоѓаат на сцената и информациите за нивната локација и ориентација да се пренесат до системот за кинематика кој е задолжен за движење на роботот.

Доколку системот за детекција детектира објекти и испраќа информации за детектирааните објекти, а системот за кинематика соодветно наоѓа решенија за вредностите на зглобовите на роботот и успешно го придвижува, се поставува прашањето како да се симулира робота на системот за да се потврди исправното функционирање на системот во целина и воедно да се избегнат можни оштетувања на роботот при погрешно пресметани вредности.

1. Кинематика
   1. Frames

Секоја точка во 3д просторот е определена со координати (x ,y, z). Ориентацијата на дадена точка во просторот може да се претстави со три единечни вектори: Rx(1, 0, 0), Ry(0, 1, 0) и Rz(0, 0, 1). Овие вектори запишани во матрица со хомогена репрезентација тогаш секоја точка во просторот зададена со локација и ориентација може да се запише:

* 1. Трансформации
     1. Координатен систем

Координатниот систем е Декартов. Притоа се користи правило на десна рака, односно Z+ има ориентација нагоре, X+ кон нас, Y+ на лево. Позитивна ротација е во спротивна насока на стрелките на часовникот, додека негативната ротација е во насока на стрелките на часовникот.

* + 1. Операции

**Транслација**Транслацијата може да се изврши во било која насока, за произволна вредност. Пример транслација на фрејм fr1 за дадени вредности (10,12,15) соодветно по X, Y и Z оски

X =

**Ротација**

Ротацијата може да може да се изврши околу било која оска. Пример ротација околу X оска за 90 степени.

X =

* 1. DH parameters

Данавит-Хатенберг параметри (DH параметри) се четири параметри поврзани со конвенција за поставување референтни фрејмови на врските во кинематички синџир или роботски манипулатор. Референтните фрејмови се поставени на следниов начин :

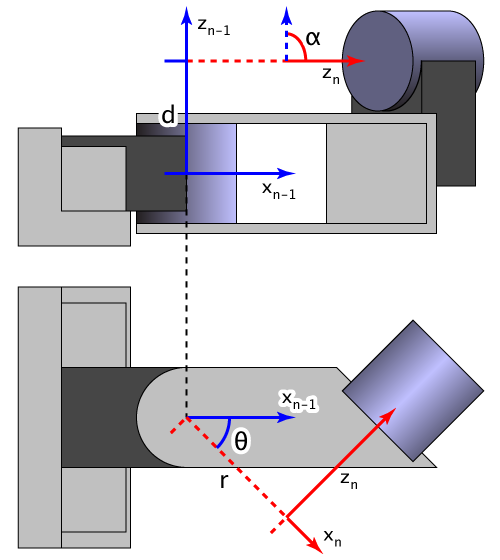
* + 1. Z - оската е во насока на оската на зглобот
    2. X - оската е паралелна со заедничката нормала . Доколку не постои заедничка нормала d е слободен параметар.
    3. Y – оската се користи да го комплетира фрејмот, користејќи го правилото на десната рака

Трансформацијата е опишана преку четирите параметри :

**d** –поместување по претходната Z-оска до заедничката нормала

**θ**  – агол околу претходна Z оска од старата/претходна X оска до новата

***r*** – должина на заедничка нормала. Доколку станува збор за ротирачки зглоб овој параметар претставува радиус на ротација

**α** –агол околу заедничка нормала од претходната Z оска до новата Z оска

* 1. Forward kinematics

Проблемот на кинематика се однесува на употреба на кинематички равенки на роботот за да се пресмета положбата на крајната точка на роботоскиот манипулатор од вредности на параметрите на зглобовите. Кинематичките равенки се користат во роботика, компјутерски игри и анимација. Обратниот процес кој за дадена положба на крајната точка на манипулаторот ги пресметува параметрите на зглобовите се нарекува инверзна кинематика. Кинематичките равенки за робот се добиени со користење на крути трансформации[Z]( ротација), да го опише релативното движење дозволено на секој зглоб и посебна трансформација [X] да ги дефинира димензиите на секој линк на роботот. Резултатот е низа од низа од крути трансформации (само ротација и транслација ). Оваа низа се состои од наизменични трансформации на зглобовите и линковите од основата на манипулаторот од крајната точка:

Каде Т е трнсформација која ја дефинира крајната точка на манипулаторот. Овие равенки се наречени кинематички равенки на сериски кинемтички синџир.

* 1. Inverse kinematics

Тривијално е да се пресмета крајната точка на кинематички синџир за дадено множествово на агли на зглобовите. Обратниот процес не е тривијален бидејќи многу често има повеќе решенија а во некои случаи нема решенија. Треба да го решиме проблемот на инверзна кинематика во анимација и роботика, каде роботот се состои од неколку кинематички синџири и треба да ја достигне целната точка во правоаголен координатен систем, каде што само координатите се познати.

Моментално многу алгоритами кои го решаваат проблемот на инверзна кинематика користат нумерички и итеративни процедури. Друг вообичаен метод е конструкција на матрица на Јакобиан и потоа за оваа матрица се бара инверзна матрица или се транспонира. Воедно методот на инверзна матрица на Јакобиан е искористен во овој проект за да се најадат решенија за вредностите на зглобовите.

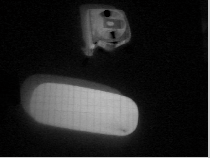
1. Препознавање на објекти
   1. Background subtraction

**Background subtraction** е широко користена класа техники за сегментирње на објекти од интерес на сцената. Популарноста на алгоритамот главно доаѓа од ефикасноста на пресметките, што овозможува примена во области како што се интеракција човек - компјутер, видео надзор и набљудување на сообраќајот. Објектите на сцената се детектираат преку разликата помеѓу моменталната слика и сликата со статичка позадина:

Пример:

Позадина Објекти на сцена

Разлика Бит мапа

* 1. Connected components

Connected component алгоритамот се користи во компјутерска визија за детектирање на поврзани региони во бинарни дигитални слики, како и слики во боја и податоци со повисоки димензии можат да бидат обработени. Кога овој алгоритам е интегриран во препознавање на објекти или интеракција човек-коппјутер може да се примени на широк опсег на информации. Генерално се користи за во конструкција на региони од добиената бинарна слика. Во понатамошните чекори регионите можат да бидат филтрирани, броени, и следени.

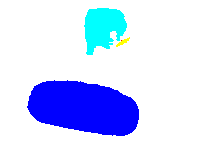


* 1. Labeling

Кога ќе се определи поврзаноста на регионите потребно е секој регион да се означи со цел да се идентификуваат различни региони за понатамошна обработка. Алгоритамот означувањето на различни региони може да го изврши според параметрите за поврзаност. Пример поврзаност со 8 соседни точки или поврзаност со 4 точки.



Пример, секој регион е означен со различна вредност и таа вредност е претворена во RGB за да се конструира слика која е видлива и разбирлива за човек.



Означени региони на бит мапа

* 1. Бит мапа

Бинарна мапа или бит мапа претставува матрица со исти димензии како и сликата што се процесира. Вредностите на бит мапата како што и самото име имплицира се 0 или 1. Битмапата се користи најчесто во процесот за сегментирање на региони на сликата кои што се од интерес. Исто така може да се употребат различни техники на филтрирање на регионите, според површина, форма итн. Пример бит мапа добиена со примена на алгоритамот background subtraction.



Бит мапа

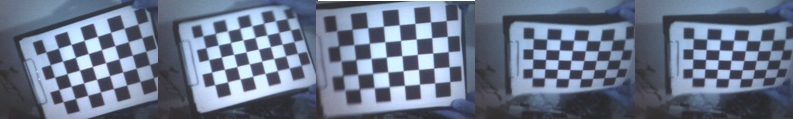
* 1. Калибрирање на видео камера

Најчесто како цел за фат на објектите е центарот на самиот објект . Со техниките споменати погоре имаме механизам за детектирање на објекти во работната околина на роботот. Но координатите што се добиени преку детекција на објектите се во координатен систем на сликата односно (x, y). Потребно е да се најде трансформација од координатниот систем на сликата во реалниот свет така што за секоја точка на сликата I=(x, y), би добиле точка во реалниот свет W=(x, y ,z), во координатен систем на камерата , притоа мерните единици се во милиметри. За да ја најдеме трансформацијата потребно е да извршиме постапка наречена калибрирање на камерата. Оваа постапка овозможува да ги добиеме внатрешните параметри на камерата преку кои се извршува проекцијата.

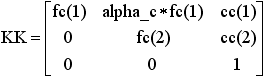
Внатрешни параметри на камерата :

* **Фолкално растојание :** Фокалната должина во пиксели е зачувана во вектор 2х1 означен со **fc.**
* **Внатрешна точка (или центар на сликата):** Точка на пресек помеѓу оптичките оски и рамнината на сликата. Координатите на внатрешната точка се зачувани во 2х1 вектор означен со **сс**
* **Коефициент на накосеност:** Коефициентот на накосеност дефинира агол помеѓу x и y оските на пикселите, и е зачуван во скалар **alpha\_c.**
* **Дисторзии**: Коефициенти на дисторзија на сликата(радијални и тангентни) се зачувани во вектор 5х1 означен со **kc**

Процесот на калибрирање на камерата започнува со зачувување на индивидуални слики на печатена шема слична на шаховска табла која се наоѓа пред камерата. Притоа шаховската табла на секоја слика треба да има различна ориентација од претходната. Осум до десет слики се доволни за да се определат внатрешните параметри на камерата. Софтверот користен за калибрирање на камерата е “[**Camera Calibration Toolbox for Matlab**](http://www.vision.caltech.edu/bouguetj/calib_doc/)”



Од добиените параметри може да се конструира матрица на проекција, со која секоја точка од сликата I=(x,y) може да се трансформира во 3Д координатен систем на камерата:



1. Дизајн на системот
   1. Избор на платформа

Платформата во која е имплементиран софтверскиот систем е MATLAB. При изборот на платформа за реализација на софтверскиот систем земени се повеќе критериуми:

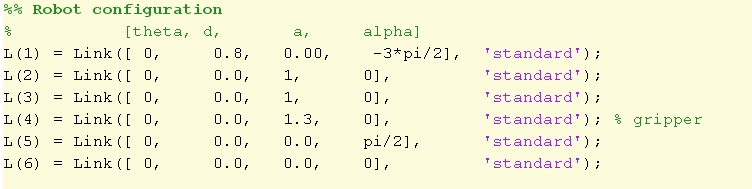
* Лесен начин на изразување на математички пресметки.
* Достапност на различни алатки со цел максимално искористување на постоечки технологии.
* Еден од најважните критериуми при изборот е и можноста за дизајнирање на симулатор кој ќе овозможи симулирање на роботскиот систем во реално време со голема точност.
* Алатки кои овозможуваат обработка на слики.
* Алатки кои овозможуваат калибрирање на дигитални камери
  1. Модули

Системот се состои од посебни модули кои комуницираат помеѓу себе преку глобални променливи кои се сместени во меморијата на Матлаб. Системот се состои од следниве модули :

* Модул за кинематика. (Кинематички солвер кој овозможува пресметување на инверзна кинематика со неколку алгоритми)
* Модул за детектирње на објекти
* Модул за трансформација на координати од сликата во координати на роботот.
* Модул за виртуелна реалност
* Модул за комуникација со роботот/ контролер на роботот
* Модул за графички интерфејс со корисникот
  1. Модул за кинематика

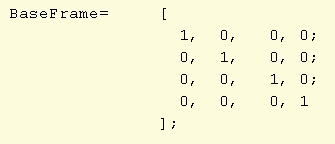
Прв чекор во кинематичките пресметки е опис на конфигурацијата на роботот. Најчест метод за опишување на конфигурацијата се параметрите на Данавит – Хатенберг кои на едноставен начин овозможуваат дефинирање на кинематичките линкови на роботот.

DH –параметри за роботот



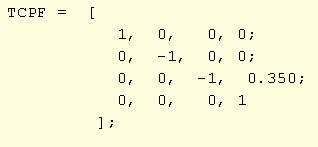
Вредностите за должините на линковите на роботот се скалирани со што се олеснува репрезентацијата на роботот. Роботот има 4 зглобови, што значи четири степени на слобода, но како што може да се забележи од сликата погоре во конфигурацијата на роботот има 6 степени на слобода од кои последните 2 се редундантни и се наоѓаат на крајната точка на роботот. Причината за додавање на редундантни степени на слобода е да се овозможи достапност на секоја локација која се наоѓа во работната површина на роботот.

Покрај конфигурацијата на роботот потребно е да го дефинираме фрејмот на основата(Robot base frame) . Овој фрејм ја дефинира локацијата на роботот во 3Д просторот, притоа сите локации кои се дефинирани во работната околина на роботот се релативни во однос на фрејмот на основата. Во нашиот случај роботот е поставен во центарот на координатниот систем.



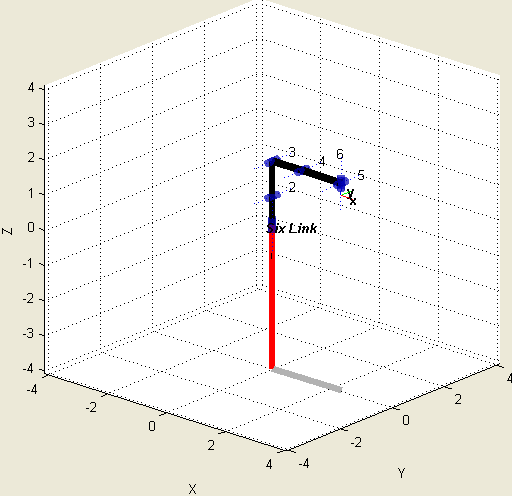
Слика фрејм на основата на роботот

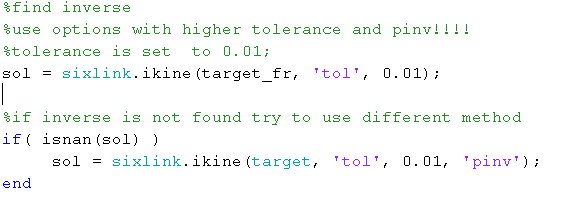
Сега кога ја имаме дефинирано конфигурацијата на роботот следен чекор е дефинирање на TCPF(Tool Point Center Frame). Обично TCPF е поместување од крајната точка на роботот. Со ова поместување е претставена крајната точка на алатката што е поставена на роботот. Во нашиот случај алатката е „клешта“ монтирана на роботската рака.



Конфигурација на TCPF

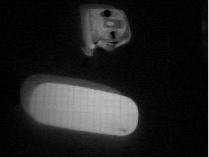
Многу често за дадена локација во просторот постојат повеќе од едно решение за вредностите на зглобовите на роботот. Обично локацијата е достигната со лактот нагоре, или со лактот надолу. Во нашиов случај единствено решение е конфигурација на роботот со лактот нагоре, бидејќи доколку го избереме решението со лактот надолу, вредностите ќе ги надминат границите на зглобовите на роботот или ќе доведат до судирање на роботот со подот.



Модулот за кинематика наоѓа решенија за вредностите на зглобовите за дадена локација. Притоа ја користи методата на транспонирана матрица на Јакобиан. Алгоритамот користи итеративен метод за наоѓање на решенијата. Доколку овој алгоритам не може да најде решенија за дадената локација тогаш се применува псевдо-инверзен метод наместо транспонирана матрица на Јакобиан.

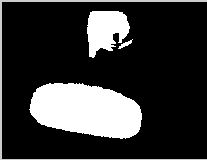
* 1. Модул за детектирање на објекти

Модулот за детекитрање на објекти овозможува детекција на објекти кои се наоѓаат во работната околина на роботот. Со примена на алгоритамот за Background Subtraction, модулот е во можност да конструира бинарна мапа. По употребата на овој алгоритам во модулот за детектирање на објекти се употребени неколку додатни техники кои овозможуваат попрецизно и посигурно детектирање на објектите. Постапката се состои од неколку чекори:

* Сликите во боја добиени од дигиталната камера се претвораат во gray-scale (црно бели слики). Ова постапка се применува со цел да се намалат пресметките и да се зголеми брзината на обработка.
* Кога референтната слика и сликата со објекти се претворени во gray-scale, се применува алгоритамот Background Subtraction, од што се добива слика во која се истакнати разликите во вредности на секој пиксел од сликата
* Следен чекор е отстранување на “шумот” од добиената слика. Со примена на **wiener2** филтерот кој е достапен во алатките на Матлаб за обработка на слики со интензитет се отстранува шумот кој е добиен од камерата. Овој метод користи статистички податоци од соседните пиксели за да го отстрани шумот. Во нашиот случај е земена околина од 5х5 пиксели.
* Сега може да се изгради бинарна мапа со примена на гранична вредност. Односно сите пиксели од сликата кои имаат вредност поголема од граничната се обележани со 1 додека пикселите чии интензитет е под граничната вредност се обележани со 0. Резултатот од оваа постапка е прикажан на следата слика:



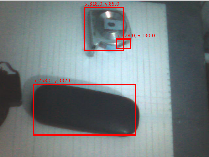
* Како што може да се забележи на силиката погоре во горниот објект се наоѓаат празнини кои се добиени поради различно осветлување. За да се овозможи детектирање на објектите со поголема сигурност се користи метода за исполнување на вакви празнини. По примената на ова метода се добива бинарна мапа во која се отстранети празнините во регионот.



* Следен чекор е да се обележат регионите со примена на алгоритамот Connected Components:

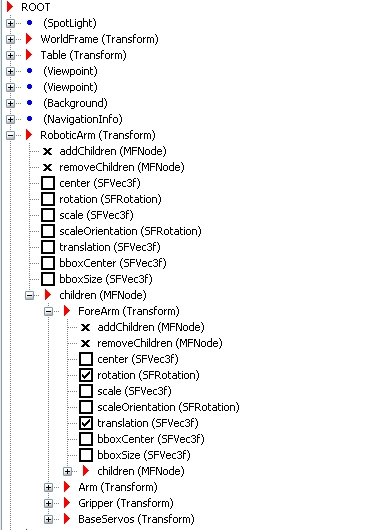


* Како што може да се забележи на сликата мал сегмент од објектот е отцепен. Ова е исто така резултат од различното осветлување на самиот објект како и својствата на материјалот. Идеално, за да се избегне фрагментација на објектот пости техника која врз основа на оддалеченоста на регионите и нивната големина одлучува дали два блиски региони ќе ги поврзе.
* Сега кога имаме мапа со означени региони може да примениме уште еден алгоритам кој ќе ги отстрани сите региони кои се помали од дадена вредност со цел да се добие почиста слика. Во нашиот случај сите региони со површина помала од 200 пиксели се отстранети.
* Бидејќи од интерес се објекти кои се наоѓаат во центарот на сликата може да примениме уште едена техника која отстранува региони кои се наоѓаат по рабовите на сликата.
* Последен чекор е да го идентификуваме центарот на секој регион, неговите димензии како и површина.
* Откако овие информации ни се познати околу регионите може да исцртаме правоаголници кои ќе ги обележат регионите.



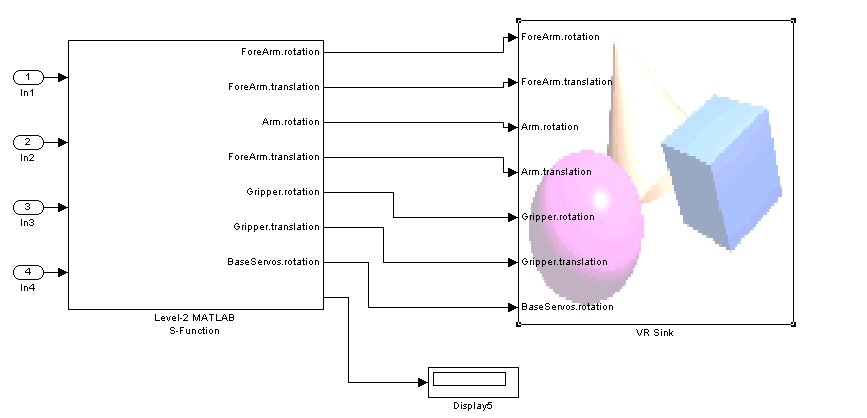
* Сега информациите за центарот на објектите може да се пренесат до наредниот модул за понатамошна обработка
  1. Модул за виртуелна околина

Модулот за виртуелна околина овозможува симулирање на работата на роботот во реално време. Симулацијата овозможува да ја провериме работата на секој модул индивидуално како и резултатите од нивната интеграција. Модулот се состои од под систем кој ги извршува пресметките за анимација на роботските компоненти. Влез на системот се вредности за зглобовите на роботот. Под системот соодветно ја пресметува транслацијата и ротацијата за секоја индивидуална компонента.

Технологијата која се користи за создавање на виртуелна околина е **VRML (Virtual Reality Nodeling Languge)** . VRML е стандарден формат за претставување на 3 димензионална интерактивна вектор графика. Овој стандард е дизајниран според World Wide Web односно Интернет. Бидејќи станува збор за јазик, рабовите и линиите на 3 димензионален пологон можат да бидат специфицирани заедно со бојата на површината, текстури, како и осветленост на објектите итн. Оваа технологија Матлаб ја интегрира во Simulink овозможувајќи да се претстават некои системи во виртуелна реалност и извршувањето на симулацијата да биде прикажана во виртуелен свет. Во VRML постои хиерархија на компоненти со што се овозможува групирање. Соодветните компоненти се сместени во трансформација, притоа компонентите се деца на трансформацијата.

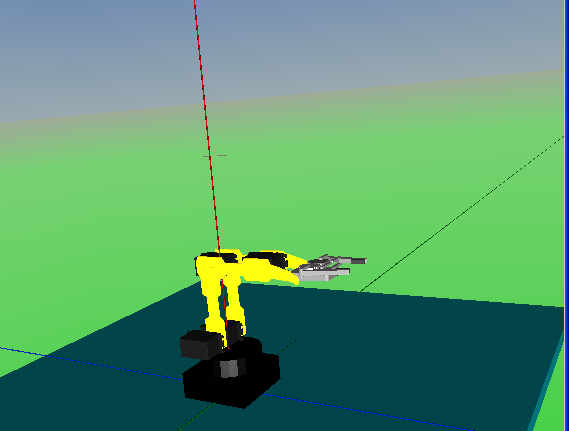
Оваа трансформација ни овозможува потоа да ги анимираме компонентите кои се наоѓаат во нејзината хиерархија преку транслација, ротација, скалирање итн. Во нашиов случај од интерес се „крути“ трансформации односно ротација и трансформација.

Simulink овозможува да бидат селектирани соодветните операции. Селектираните операции се појавуваат во блокот како влезни порти.

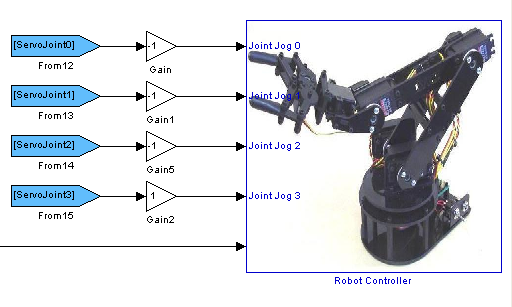
Роборската рака се состои од следниве компоненти чии операции се прикажани на сликата погоре.

* Серво мотори на основата
* Подлактица
* Рака??
* Клешта

Соодветните операции (ротација и транслација) се пресметани во под системот кој за дадени вредности на зглобовите(влезови) пресметува каде ќе биде позицијата во виртулениот свет на компонентите на роботската рака



* 1. Модул за контрола на роботска рака



1. Графички интерфејс

Референвцесс

K. C. Gupta and K. Kazerounian, “Improved

numerical solutions of inverse kinematics of

robots,” in Proceedings of the 1985 IEEE International

Conference on Robotics and Automation,

vol. 2, March 1985, pp. 743–748.

[2] A. Goldenberg, B. Benhabib, and R. Fenton,

“A complete generalized solution to the inverse

kinematics of robots,” IEEE Journal of

Robotics and Automation, vol. 1, pp. 14–20,

March 1985.

[3] V. J. Lumelsky, “Iterative coordinate transformation

procedure for one class of robots,”

IEEE Transactions on Systems Man and Cybernetics,

vol. 14, pp. 500–505, Jun. 1984.

[4] V. D. Tourassis and J. Ang, M.H., “A modular

architecture for inverse robot kinematics,”

IEEE Transactions on Robotics and Automation,

vol. 5, pp. 555–568, October 1989.

[5] M. Fedor, “Application of inverse kinematics

for skeleton manipulation in real-time,” in

SCCG ’03: Proceedings of the 19th spring

conference on Computer graphics. New York,

NY, USA: ACM Press, 2003, pp. 203–212.