

Vodenje robota z Android OS

Uroš Sadek , Bojan Kseneman
Mentorja: izr. prof. dr. Aleš Hace, doc. dr. Boris Curk
Univerza v Mariboru, FERI, Smetanova ul. 17, Maribor
skull199@gmail.com , ksenchy@ksenchy.org

Robot control with Android OS

The following article briefly describes the concept and implementation of a robotic arm with 5 degrees of freedom. The control itself is implemented with a smart phone or potentially any other device that is based on Android OS with a present Bluetooth module. For the purpose of control an application was developed, which works in conjunction with a microcontroller unit to control servomotors. Communication between microcontroller unit and a smart phone is based on Bluetooth protocol. The developed application allows the control in manual mode, control by individual axes of the robot (inner coordinate system) as well as in the outer coordinate system. It also allows moving a specific joint of the arm with desired speeds or synchronous with the given maximum speed (PTP). In manual mode we have the option to save the coordinates of the current position (point) of the arm. They can later be edited, deleted from a list of all saved points. In the automatic mode we can run a “program” of all saved points.

Kratek pregled prispevka

V danem prispevku je na kratko opisana zamisel ter hkrati predstavljena izvedba vodenja robotske roke s 5 prostostnimi stopnjami. Vodenje poteka s pametnim mobilnim telefonom ali kakršnokoli drugo napravo, ki ima nameščen operacijski sistem Android in ima dostop do Bluetooth komunikacijskega vmesnika. V ta namen je bila razvita aplikacija za operacijski sistem Android, katera deluje skupaj z mikrokrmilniško enoto za krmiljenje servomotorjev. Komunikacija med mikrokrmilniško enoto in pametno napravo poteka preko bluetooth vmesnika. Razvit je bil uporabniški vmesnik pametnega telefona, ki omogoča v sklopu ročnega režima vodenje po posameznih oseh robota(notranje koordinate), kot tudi v zunanjem koordinatnem sistemu. Pri tem lahko pomikamo posamezni sklep manipulatorja z različnimi hitrostmi, ali pa sinhrono s podano maksimalno hitrostjo(PTP). V ročnem režimu vodenja imamo možnost shranjevanja koordinat trenutnega položaja. Slednje lahko urejamo in brišemo iz seznama točk, ki predstavljajo koordinate različnih položajev. V sklopu avtomatskega režima poženemo robotski program, ki je sestavljen iz vseh shranjenih točk.

1 Uvod

Glavni cilj nastale projektne naloge je bil izdelava robotskega sistema za 5 osni robotski manipulator, voden s pametno mobilno napravo. Izdelava takšnega sistema zajema uporabniški vmesnik in krmilni del sistema.

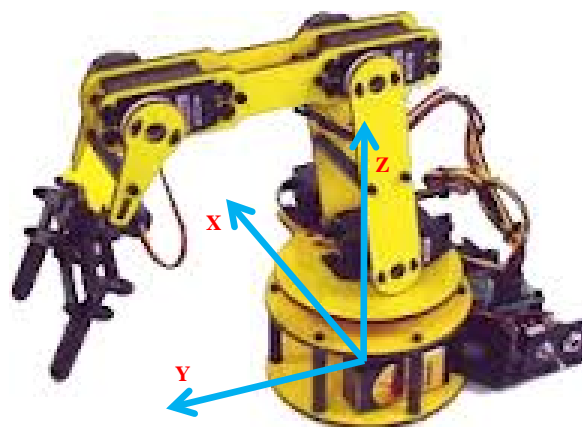
Krmilni del robotskega sistema smo izvedli z mikrokrmilnikom. Na njem se izvajajo protokoli, ki so povezani z gibanjem manipulatorja kot so položaj, hitrost, način gibanja in izvajanje shranjenega robotskega programa. Krmilni del skupaj z uporabniškim vmesnikom tvori celoto, za kar je obvezna komunikacija med njima. Slednja poteka preko Bluetooth protokola.

Uporabniški vmesnik služi kot posrednik med uporabnikom ter krmilnim delom sistema. Preko njega uporabnik dostopa do robotskih funkcij, ki jih sam sistem ponuja. Ker večinoma uporabnik robotskega sistema strokovnjak iz nekega drugega področja, za kar potrebuje robota (varjenje, skladiščenje...), mora biti uporabniški vmesnik preprost, enostaven in učinkovit za uporabo. V sklopu naše projektne naloge, smo izdelali uporabniški vmesnik za Android OS.

Prednost izdelave takšnega uporabniškega vmesnika je vsekakor brezžična komunikacija, pri kateri ne potrebujemo dodatnih vodnikov, ki bi nas lahko omejevali pri delu z robotskim sistemom. Prav tako pa veliko prednost prinaša tudi prenosljivost takšnega uporabniškega vmesnika (upravljalne konzole), kajti z lahkoto ga prenesemo na druge naprave, ki imajo nameščen Android OS 4.0 ali novejšega.

2 Robotska roka (Lynx6)

Lynx6 je 5-osna robotska roka namenjena za laboratorijske preizkuse (*Slika 1*). Sestavljena je iz petih gibljivih osi in sicer baze, ramena, komolca, ter dveh osi v zapestju. Rotacijski aktuatorji v sklepah so servomotorji, z omejenim kotom rotacije med -90° in 90° . Glede na njeno strukturo lahko doseže poljubno točko v dosežnem prostoru vendar ne pri poljubni orientaciji.

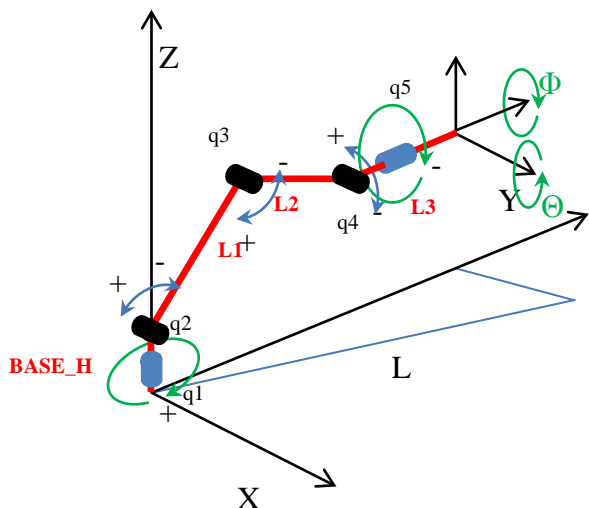


Slika 1: Robotska roka Lynx6

Načinov vodenja robotske roke je več. In sicer najenostavnejši način, pri katerem direktno pomikamo posamezne osi robota, se imenuje vodenje po notranjih koordinatah. Druga možnost je pomikanje vrha robota glede na zunanje koordinate, ki so v našem primeru postavljene tako kot prikazuje *slika 1*. Seveda so možnosti neskončne, saj lahko postavimo poljubne koordinatne sisteme glede na okolje v katerem robot deluje. V našem projektu smo realizirali vodenje na dva osnovna načina, in sicer v notranjih in zunanjih koordinatah. Za vodenje po zunanjih koordinatah potrebujemo inverzni kinematični model, s pomočjo katerega izračunamo vrednosti posameznih notranjih koordinat pri poljubnih zunanjih koordinatah. Prav tako pa smo implementirali tudi direktni kinematični model za izračun zunanjih koordinat, pri poljubnih notranjih koordinatah. Slednjega uporabljamo pri spremembi načina vodenja iz notranjih v zunanje koordinate, za izračun trenutnega položaja v zunanjih koordinatah.

2.1 Inverzna kinematika

Kot smo že omenili, je inverzna kinematika sklop enačb, ki omogoča izračun notranjih koordinat iz poljubnih zunanjih koordinat vrha robota. Te enačbe so povezane z geometrijo samega manipulatorja (dolžine segmentov...), za to ni enotnih modelov in se razlikujejo med različnimi roboti. Geometrijo robotske roke za katero smo izpeljali kinematični model, prikazuje *slika 2*.



Slika 2: Geometrija robotske roke

Pri izpeljavi inverzne kinematike smo si pomagali z ravnino YZ, kajti do koordinate X je odvisen samo kot q_1 . Glej *slika 2*.

$$SW_y = \sqrt{z^2 - y^2} - L_3 \cos(\Theta) \quad (1)$$

$$SW_z = z - L_3 \sin(\Theta) - \text{BASE}_H \quad (2)$$

$$SW = \sqrt{SW_y^2 + SW_z^2} \quad (3)$$

$$A_1 = \arctan \frac{SW_z}{SW_y} \quad (4)$$

$$A_2 = \frac{\arccos(L_1^2 - L_2^2 + SW^2)}{2 \cdot L_1 \cdot SW} \quad (5)$$

$$q_1 = \arctan \frac{x}{y} \quad (6)$$

$$q_2 = A_1 + A_2 \quad (7)$$

$$q_3 = \frac{\arccos(L_1^2 + L_2^2 - SW^2)}{2 \cdot L_1 \cdot L_2} \quad (8)$$

$$q_4 = \Theta - q_3 - q_2 \quad (9)$$

$$q_5 = \Phi \quad (10)$$

SW – razdalja od izhodišča k.s. do osi (q4)

SW_z – z-koordinata zapestja (q4)

SW_y – y-koordinata zapestja (q4)

A₁ – kot premice SW glede na ravnino XY

A₂ – kot premice SW glede na L1

V prvem koraku izpeljave inverzne kinematike, z enačbama (1) in (2), izračunamo položaj zapestja glede na orientacijo Θ (q4). Z dano bazno višino in položajem zapestja smo izračunali razdaljo med zapestjem in ramo (3), ter določili kot te daljice glede na tla (4) ter glede na L1 (5). V tretji fazi lahko poračunamo vse notranje koordinate. V tem delu pri računanju kota q_1 , v enačbi (6), upoštevamo tudi koordinato X. Tako smo pridobili inverzni kinematični model robotske roke, ki upošteva položaj vrha robota in orientacijo(Θ), ki določa kot med orodjem in ravnino XY [1].

2.2 Direktna kinematika

Direktna kinematika je za izpeljavo enostavnejša, kajti iz podanih vseh notranjih kotov in z znanimi dolžinami posameznih segmentov, enostavno poračunamo koordinate vrha ter orientacijo Θ [1].

$$z = \text{BASE}_H + L_1 \cos(-q_2) + L_2 \sin(q_2 - q_3) \quad (11)$$

$$+ L_3 \sin(q_2 - q_3 + q_4)$$

$$L = L_1 \sin(-q_2) + L_2 \cos(q_2 - q_3) \quad (12)$$

$$+ L_3 \cdot \cos(q_2 - q_3 + q_4)$$

$$x = L \cdot \sin(q_1) \quad (13)$$

$$y = L \cdot \cos(q_1) \quad (14)$$

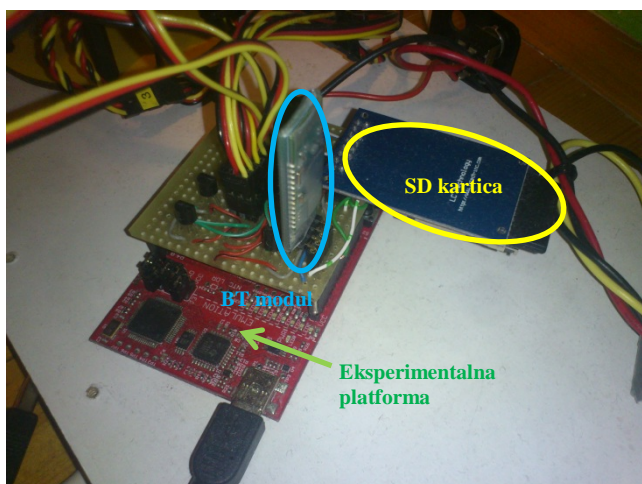
$$\Theta = q_2 - q_3 + q_4 \quad (15)$$

Koordinato Z izračunamo kot vsoto vseh prispevkov posameznih členov v smeri Z osi(11), podobno tudi pridobimo dolžino L(12) vidno na *sliki 2*. X in Y koordinati izračunamo glede na kot q_1 (13)(14). Orientacijo Θ pa pridobimo z vsoto druge, tretje in četrte notranje koordinate. Preračunavanje enačb kinematičnih modelov poteka na Android OS, na katerem teče tudi uporabniški vmesnik.

3 Krmilni del robotskega sistema

Na Android OS se v ozadju uporabniškega vmesnika pripravijo vse koordinate v notranjem koordinatnem sistemu, ki jih preko bluetooth komunikacije nato posreduje do krmilnega dela sistema. Krmilni del robotskega sistema glede na vrsto funkcije uredi ustrezen pomik manipulatorja z določeno hitrostjo in določenim načinom giba. Hitrostni profil, ki smo ga implementirali v našem primeru, je pravokotne oblike.

Za izvedbo krmilnega sistema robotske roke, smo izbrali eksperimentalno platformo TI MSP-EXP430FR5739, prikazano na *sliki 3*. Na platformo smo dodatno priključili bluetooth modul HC-06 s katerim smo vzpostavili komunikacijo z uporabniškim vmesnikom na Android OS [2].

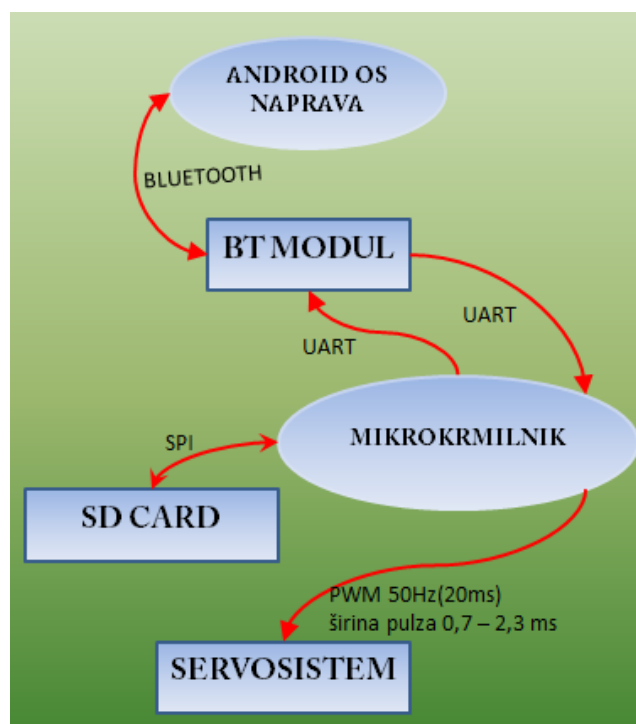


Slika 3: Eksperimentalna platforma z dodatki

Želja po brezžični komunikaciji med uporabniškim vmesnikom in krmiljem robotske roke, nas je privedla do bluetooth komunikacije, ki je na voljo na skoraj vseh napravah z Android OS. Kot vemo, potekajo vse priprave in transformacije koordinat na notranje kote v ozadju uporabniškega vmesnika. Za to smo hitro našli na težave pri izvajanju robotskega programa v realnem času, saj vnaša bluetooth komunikacija kar precej nepredvidljivih zakasnitev. To nas je privedlo do predhodnega pošiljanja vseh točk iz baze podatkov robotskega programa na mikrokontrolnik. Zaradi

precej omejenega delovnega pomnilnika (RAM) na mikrokontrolniku, smo morali zraven vključiti tudi vmesnik za SD pomnilniško kartico [3]. Tako pred izvajanjem samega robotskega programa v avtomatskem režimu, preko bluetooth vmesnika, prenesemo vse podatke o robotskem programu na SD pomnilniško kartico. Med samim izvajanjem robotskega programa, mikrokontrolnik sproti prenaša po blok podatkov v delovni pomnilnik. Dostopanje do podatkov na SD kartici je mnogo hitrejše od bluetooth komunikacije, prav tako ne vnaša nepredvidljivih in zelo nezaželenih zakasnitev.

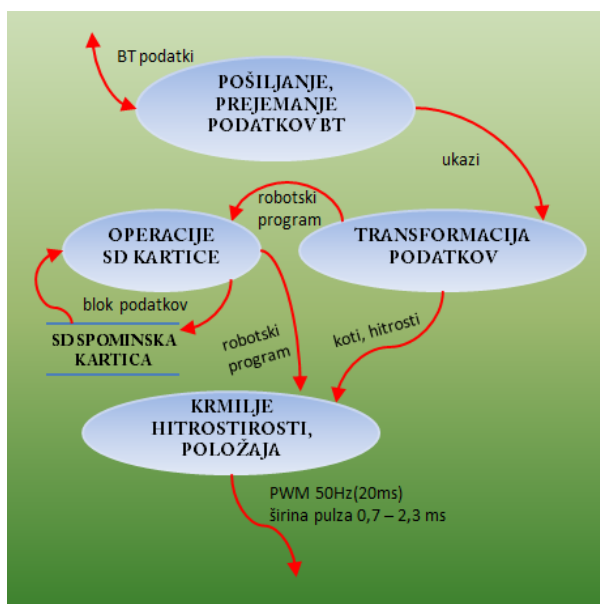
Vse komponente robotskega sistema povezane v celoto prikazuje *slika 4*.



Slika 4: Procesorsko okolje celotnega sistema

3.1 Struktura kode krmilnega dela

Koda krmilniškega sistema je razdeljena na štiri večje module. Prvi modul je namenjen bluetooth komunikaciji, drugi modul za transformacijo podatkov glede na vrsto podatkov. Tretji modul upravlja z robotsko roko, krmili položaj in hitrost gibanja, medtem ko četrti modul izvaja operacije z SD kartico. Medsebojno povezanost in tok podatkov prikazuje *slika 5*.



Slika 5: Prikaz modulov krmilja

Tok podatkov označen kot »robotski program« na sliki 5, predstavljajo shranjene koordinate v različnih točkah za uporabo v avtomatskem režimu delovanja.

4 Uporabniški vmesnik na Android OS

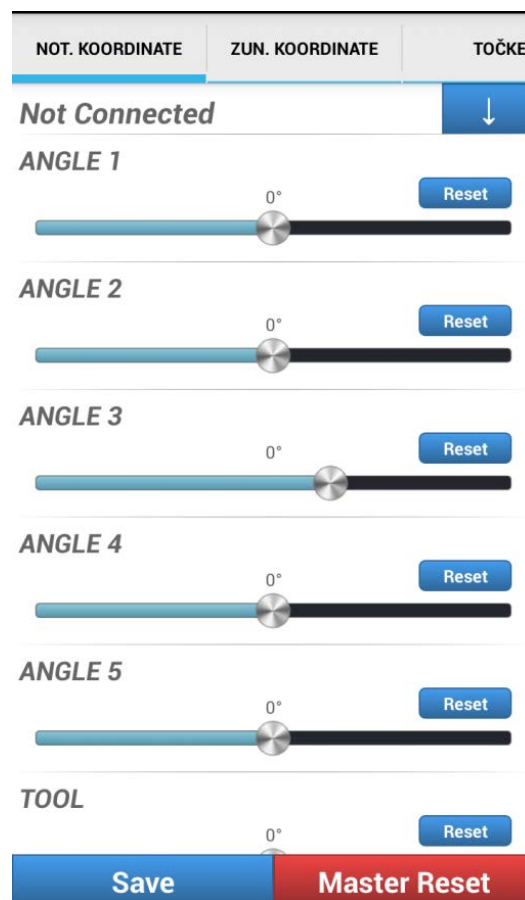
V predhodnih poglavjih smo že večkrat omenili uporabniški vmesnik, katerega si bomo podrobneje ogledali v nadaljevanju. Namreč gre za grafični uporabniški vmesnik(GUI), kjer uporabnik s pomočjo grafičnih objektov, upravlja z robotskim mehanizmom.

Grafični vmesnik je razdeljen na 3 dele in sicer:

- Vodenje po notranjih koordinatah,
- Vodenje po zunanjih koordinatah
- Seznam shranjenih točk(robotski program)

Vodenje robota je realizirano s pomočjo drsnikov, katerih vsak predstavlja položaj sklepa na robotu. Premik roke po določenemu sklepu, je možno s premikanjem drsnika, pri tem se posodablja prikaz za trenutno vrednost položaja posameznega sklepa, kar v notranjih

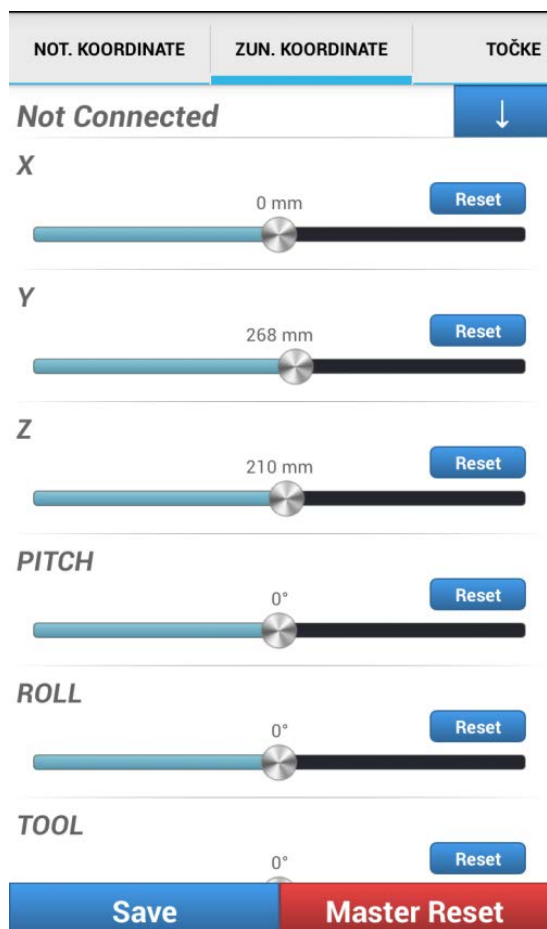
koordinatah predstavlja kot osi(Slika 6), pri zunanjih pa koordinate vrha robota v prostoru(Slika 7).



Slika 6: Vodenje v notranjih koordinatah

Takšen način vodenja imenujemo ročni položajni način, saj določamo končni položaj manipulatorja. V našem projektu smo realizirali samo tega. Bolj razširjen pa je tako imenovani hitrostni način ročnega vodenja, kjer s pritiskom na smerni gumb gibanja osi, sklep rotira z konstantno hitrostjo dokler gumba ne spustimo. Z gumbom naprave za dostop do menija, dostopamo do možnosti spreminjanja načina gibanja(neodvisno,PTP), nastavitve posameznih hitrosti ali skupne sinhrono hitrosti.

Vsak drsник ima omejeno področje premika npr. kot od -90° pa do $+90^\circ$. Poleg vsakega drsnika imamo gumb »Reset«, katerega funkcija je, da določeni sklep oziroma koordinato povrne v privzeto vrednost, ki je v naprej sprogramirana.



Slika 7: Vodenje v zunanjih koordinatah

Seveda je v vmesniku dodan gumb »master reset«, katerega funkcija je povratak privzetih vrednosti za vse sklepe oziroma koordinate. Dodatno imamo še gumb »save«, s pomočjo katerega shranimo trenutne koordinate v SQL bazo podatkov, do katere dostopamo v zavihku seznama shranjenih točk, kjer je mogoče te koordinate naknadno urejati, brisati, ter jim spremeniti vrstni red v tako imenovanem robotskem programu.

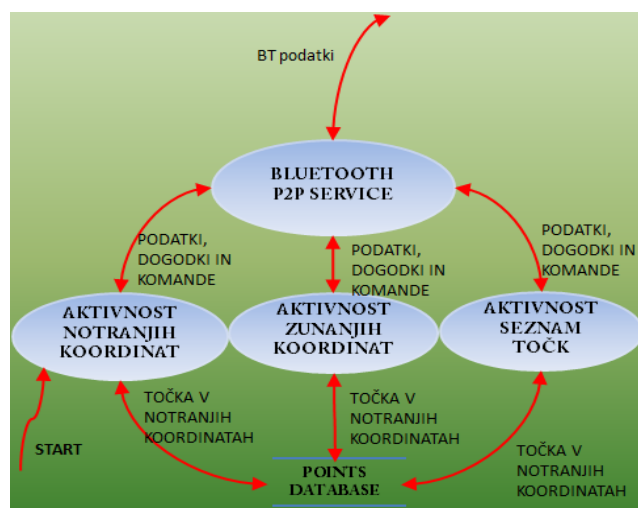
Možno je tudi ročno vodenje robota od točke do točke v seznamu točk, v okviru avtomatskega vodenja pa se izvrši robotski program, ki je sestavljen iz vseh shranjenih točk v seznamu. Robotski program poženemo z gumbom start in se zaključi s pomikom manipulatorja na zadnjo koordinato ali s pritiskom na gumb stop.

Skupno vsem trem zavihkom vmesnika je prikaz trenutnega stanja bluetooth povezave, ter

modri gumb, ki ob kliku pokaže dodatne opcije, kot je npr. vnosno polje, s pomočjo katerega lahko ročno pošiljamo ukaze na krmilnik.

4.1 Struktura kode uporabniškega grafičnega vmesnika

Programska oprema grafičnega uporabniškega vmesnika je razdeljena na štiri glavne module, kot prikazuje slika 8. Modul, ki upravlja z bluetooth vmesnikom, tudi komunicira z vsemi tremi ostalimi moduli, kateri opravljajo vsak svojo funkcijo v določenem načinu vodenja robota. Prav tako vsi trije moduli za vodenje dostopajo do podatkovne baze shranjenih točk, kamor lahko kot smo že omenili, shranjujemo trenutne položaje, in jih urejamo v seznamu.



Slika 8: Prikaz modulov grafičnega vmesnika

5 Zaključek

Izdelani GUI je izpolnil glavne cilje, saj omogoča brezžično komunikacijo, kar uporabnika ne omejuje z dolžino vodnika in se s tem lahko neovirano giblje v radiju 10m od Bluetooth modula. Prenosljivost uporabniškega vmesnika je zelo učinkovita, saj lahko vsaki napravi, ki ima nameščen Android OS 4.0 ali novejšega, dodamo funkcijo upravljalvske konzole z enostavno namestitvijo aplikacije.

Na robotskem sistemu je možno izvesti vrsto nadgradenj kot so boljši hitrostni profili v okviru krmiljenja položaja, hitrostno vodenje v ročnem režimu, linearna ter krožna interpolacija točk, upravljanje robota z vgrajenimi orientacijskimi senzorji v mobilni napravi in mnoge druge nadgradnje za boljšo praktičnost učinkovitost in uporabnost sistema.

6 Literatura

- [1] <http://www.lynxmotion.com/images/html/proj057.htm>, (februar 2013)
- [2] http://thingylab.com/wiki/hc06_linvor_1.5_at_command_set, (februar 2013)
- [3] <http://www.dharmanitech.com/2009/01/sd-card-interfacing-with-atmega8-fat32.html>, (februar 2013)