

Primjena LinuxCNC sustava kod hibridnih alatnih strojeva

Novak, Dino

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:790611>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-14**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Dino Novak

Zagreb, 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:
Prof. dr. sc. Tomislav Staroveški, dipl. ing.

Student:
Dino Novak

Zagreb, 2022.



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za završne i diplomske ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo
materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 – 04 / 22 – 6 / 1	
Ur.broj: 15 - 1703 - 22 -	

ZAVRŠNI ZADATAK

Student: **Dino Novak**

JMBAG: **0035217873**

Naslov rada na hrvatskom jeziku: **Primjena LinuxCNC sustava kod hibridnih alatnih strojeva**

Naslov rada na engleskom jeziku: **Application of LinuxCNC system for hybrid machine tools**

Opis zadatka:

U Laboratoriju za alatne strojeve Fakulteta strojarstva i brodogradnje razvijen je CNC ispitni postav za glodanje zasnovan na javno dostupnom upravljačkom sustavu otvorene arhitekture LinuxCNC. Postav je trenutno opremljen motorvretenom kao glavnim prigonom. Integracijom prigona za lasersko graviranje na predmetni postav ostvarila bi se mogućnost glodanja i graviranja. Razvoj takvog sustava omogućio bi istraživanje na području hibridnih strojeva, što je uočljivi trend u razvoju suvremenih numerički upravljanih strojeva.

U radu je potrebno:

1. Istražiti načine na koje se prigoni za lasersko graviranje mogu integrirati sa LinuxCNC sustavom.
2. Koristeći dostupnu opremu, povezati postojeći ispitni postav sa prigonom za lasersko graviranje i parametrirati ga.
3. Testirati sustav.
4. Dati zaključke rada.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

9. 5. 2022.

Zadatak zadao:

Izv. prof. dr. sc. Tomislav Staroveški

Datum predaje rada:

2. rok (izvanredni): 6. 7. 2022.
3. rok: 22. 9. 2022.

Predviđeni datumi obrane:

2. rok (izvanredni): 8. 7. 2022.
3. rok: 26. 9. – 30. 9. 2022.

Predsjednik Povjerenstva:

Prof. dr. sc. Branko Bauer

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru, prof. dr. sc. Tomislavu Staroveškom, na prenesenom znanju, pruženoj podršci, savjetima i odvojenom vremenu.

Zahvaljujem se i djelatnicima Laboratorija za alatne strojeve za pomoć tijekom izrade ovog rada.

Također se zahvaljujem obitelji i prijateljima na pruženoj podršci tijekom studiranja.

Dino Novak

Sadržaj

Sadržaj	I
Popis slika	II
Popis tablica	III
Sažetak	IV
Summary	V
1 Uvod	1
1.1 Alatni strojevi	1
1.2 Glodanje	1
1.3 Laseri	2
2 Sustav LinuxCNC	3
2.1 O LinuxCNC - u	3
2.2 Komponente LinuxCNC sustava	4
2.2.1 HAL	4
2.2.2 EMC	7
2.2.3 GUI	9
2.3 Integracija prigona za lasersko graviranje s LinuxCNC sustavom	11
2.3.1 Prigoni za lasersko graviranje	11
2.3.2 XY2-100	12
2.3.3 LinuxCNC xy2mod	13
3 Postavljanje i parametrisiranje hibridnog stroja	14
3.1 Pisanje upravljačkog programa za galvo glavu	14
3.1.1 HAL program	14
3.1.2 INI datoteka	17
3.2 Testiranje prigona za lasersko graviranje	19
3.3 Integracija galvo glave s glodalicom	20
4 Zaključak	21
Literatura	22
Prilog	23

Popis slika

1.1	Rast tržišta CNC strojeva, 2018. - 2029. [1]	1
1.2	Petoosna glodalica OPTImill FU 5-600 HSC 24 [3]	2
1.3	Stroj za lasersko graviranje twotrees TT-50 [6]	2
1.4	Shema Nd:YAG lasera čvrstog stanja [7]	2
2.1	Blok dijagram upravljanja motorom [9]	3
2.2	Spajanje HAL komponenata [8]	4
2.3	Arhitektura EMC sustava [11]	8
2.4	Axis GUI	9
2.5	Touchy GUI [8]	10
2.6	Gscreen GUI [8]	10
2.7	GMOCCAPY GUI [8]	10
2.8	NGCGUI [8]	10
2.9	Galvo glava SINO-GALVO SG7110 [12]	11
2.10	Shema XY2-100 [13]	12
2.11	DB25 konektor [12]	12
3.1	Shema regulacijske petlje galvo glave	16
3.2	Testni postav galvo glave	19
3.3	Axis GUI hibridnog stroja	20

Popis tablica

2.1	HAL komponente korištene u radu	5
2.2	INI sekcije [8]	7
2.3	Pinovi xy2mod komponente [15]	13

Sažetak

Hibridni alatni strojevi suvremeni su pravac razvoja CNC tehnologije, a definiraju se kao strojevi koji kombiniraju dva ili više obradna procesa. Jedna od tih kombinacija je lasersko graviranje integrirano s glodanjem. Lasersko graviranje je postupak obrade materijala odvajanjem čestica pomoću topline, a samo je jedna od mnogih primjena lasera u industriji. U ovom radu proveden je i opisan postupak postavljanja i parametriranja hibridnog alatnog stroja koji se sastoji od eksperimentalne troosne glodalice ADRISS i prigona za lasersko graviranje SINO-GALVO SG7110. Upravljački program hibridnog stroja napravljen je koristeći LinuxCNC, sustav za upravljanje i razvoj CNC strojeva temeljen na otvorenoj arhitekturi. U sklopu rada napravljena je konfiguracija za upravljanje prigonom za lasersko graviranje te je ista integrirana s upravljačkim sustavom glodalice. Sustav je testiran, a programski kod detaljno je opisan tijekom rada.

Ključne riječi: alatni stroj, hibridni stroj, CNC, glodanje, lasersko graviranje, LinuxCNC

Summary

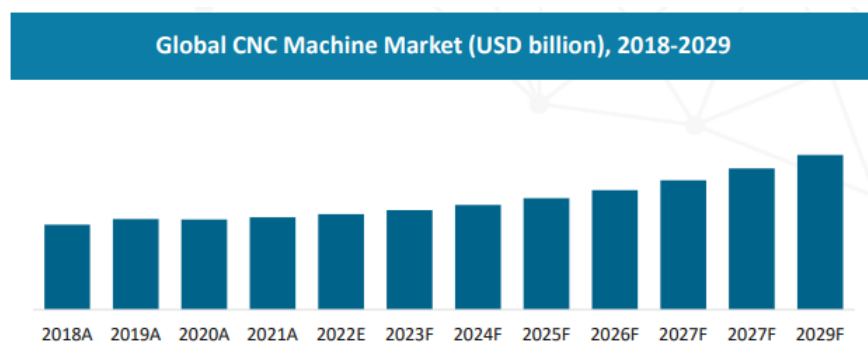
Hybrid machine tools represent a modern direction in the development of CNC technology, and can be defined as machines that combine two or more machining processes. One of the mentioned combinations is laser engraving integrated with milling. Laser engraving, as one of many applications of lasers in industry, is a machining process that works by the principle of removing material using heat. In this bachelor's thesis, a process of setting up and parameterization of a hybrid machine consisting of an experimental 3-axis CNC milling machine named ADRISS and a galvo head SINO-GALVO SG7110 was carried out and described. Firmware for the hybrid machine was developed using LinuxCNC, an open source system for control and development of CNC machines. As part of the thesis, configurations for control of the galvo head were made and integrated with configuration of the CNC milling machine. The resulting system was tested and the firmware code is explained in detail in the following text.

Keywords: machine tool, hybrid machine, CNC, milling, laser engraving, LinuxCNC

1 Uvod

1.1 Alatni strojevi

Alatni strojevi pogonjeni CNC tehnologijom srce su moderne proizvodne industrije. Kao dio proizvodnog pogona ili kao zaseban sustav, svaki alatni stroj obavlja na sirovini ili poluproizvodu svoju zadanu funkciju s ciljem dovođenja obratka jedan korak bliže gotovom proizvodu. Najčešće funkcije koje obavljaju CNC strojevi su obrada odvajanjem čestica (glodanje, tokarenje, lasersko rezanje i graviranje, elektroerozija žicom), aditivna proizvodnja (3D tiskanje, stereolitografija) te zavarivanje. Bilo da se radi o tokarilici, 3D pisaču ili nekoj drugoj vrsti stroja, svima im je zajednička jedna karakteristika: središnji numerički upravljački sustav (CNC). Tehnologija je to koja je znatno proširila proizvodne mogućnosti konvencionalnog, ručno upravljano alata. Istraživanje provedeno 2022. godine od strane istraživačke grupe *Fortune Business Insights* pokazuje da tržišna vrijednost CNC industrije u svijetu raste za 7,1 % godišnje [1].



Slika 1.1: Rast tržišta CNC strojeva, 2018. - 2029. [1]

Graf na slici 1.1 prikazuje predviđeni godišnji rast tržišta CNC strojeva u svijetu do 2029. godine. Analizom trenutno dostupnih podataka može se zaključiti da je pred spomenutom industrijom svijetla budućnost i da u njezino daljnje unaprijeđivanje i te kako ima smisla ulagati. Jedan od pravaca razvoja CNC tehnologije su hibridni alatni strojevi. Hibridni alatni stroj je sustav koji kombinira dva različita procesa obrade s ciljem istovremenog korištenja najvećih prednosti obiju procesa te izbjegavanja ograničenja svakog individualnog procesa [2]. Takav stroj može obaviti zadatak za koji bi bez njega bilo potrebno koristiti dva konvencionalna stroja. Hibridni procesi obrade najčešće spajaju konvencionalne načine obrade (glodanje, tokarenje, brušenje i bušenje) s modernim postupcima poput elektroerozije žicom, elektrokemijske obrade, laserskog odvajanja čestica, ultrazvučne obrade i obrade vodenim mlazom [2]. Ovaj rad bavi se postavljanjem i parametranjem hibridnog alatnog stroja koji može obavljati funkcije glodanja i laserskog graviranja. U sklopu rada objašnjen je proces laserskog graviranja i vrste prigona za graviranje, objašnjen je softver korišten za upravljanje strojem te je pokazan postupak postavljanja i parametranja hibridnog stroja.

1.2 Glodanje

Glodanje je postupak obrade materijala odvajanjem čestica koji se ostvaruje rotacijskim gibanjem alata - glodala i posmičnim gibanjem obratka ili alata u smjeru osi. Prema vrsti alata, glodanje može biti čeono i obodno. Kod obodnog glodanja, rezne oštrice smještene su po obodu alata što omogućuje rezanje kružne geometrije dok čeono glodalo rezne oštrice ima na ravnoj površini - čelu. Glodalica može imati dvije ili više osi gibanja. Svaka dodatna os znači dodatni stupanj slobode što omogućuje obradu kompleksnije geometrije. Moderni obradni sustavi, zahvaljujući mogućnostima koje otvara CNC tehnologija, kombiniraju glodalice i tokarilice u hibridne alatne strojeve s do dvanaest osi. Takvi strojevi nerijetko su opremljeni podsustavima za automatsku

izmenu alata i obradaka te skladištenje i dobavu sredstava za hlađenje, ispiranje i podmazivanje. Primjer takvog obradnog sustava prikazan je na slici 1.2.



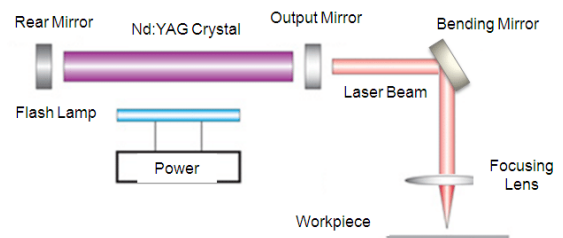
Slika 1.2: Petoosna glodalica OPTImill FU 5-600 HSC 24 [3]

1.3 Laseri

Laser, skraćena pojava "pojačanje svjetlosti s pomoću stimulirane emisije zračenja" (*engl. Light Amplification by Stimulated Emission of Radiation*), uređaj je za stvaranje i pojačavanje koherentnoga elektromagnetskog usko usmjerenog zračenja velike jakosti [4]. Laser pronalazi primjenu u širokom spektru područja. Koristi se u mjeriteljstvu, medicini, vojnoj industriji, komunikaciji, proizvodnji i mnogim drugim industrijama. Ovaj rad bavi se primjenom lasera u obradi metala. Metali, ali i mnogi drugi materijali poput drva, plastike i kože, laserom se mogu rezati i gravirati. Razlika u ta dva obradna procesa svodi se na odabir leće, vrste lasera i parametara poput snage i posmične brzine. Slika 1.3 prikazuje stolni CNC stroj za lasersko graviranje pogodan za obratke manjih dimenzija. Laseri se dijele prema tipu aktivnog medija koji imaju u sebi. Pet glavnih vrsta lasera su: plinski, poluvodički, vlaknasti, laser s čvrstom jezgrom i laser s bojiлом [5]. Slika 1.4 prikazuje pojednostavljenu shemu Nd:YAG lasera koji spada u lasere s čvrstom jezgrom.



Slika 1.3: Stroj za lasersko graviranje twotrees TT-50 [6]

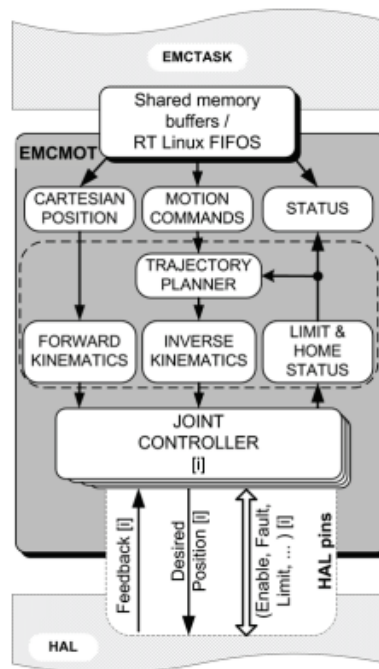


Slika 1.4: Shema Nd:YAG lasera čvrstog stanja [7]

2 Sustav LinuxCNC

2.1 O LinuxCNC - u

LinuxCNC je sustav koji se sastoji od skupa aplikacija čija je zajednička svrha upravljanje CNC strojevima [8]. Svaku od aplikacija karakterizira visoka mogućnost prilagodbe potrebama korisnika. Prema riječima jednog od razvojnih programera LinuxCNC - a, Dougu Gwynu, *LinuxCNC nije napravljen da spriječi korisnika da napravi pogrešku jer bi ga tada spriječavao i da dođe do kreativnih rješenja*. Upravo se iz ovih riječi može vidjeti da je jedna od najvažnijih karakteristika LinuxCNC sustava da se korisniku pruži potpuna sloboda i kontrola nad svakom komponentom koja se pojavljuje u izradi softvera za upravljanje gibanjem stroja. LinuxCNC se može koristiti za kontrolu mnogo različitih vrsta strojeva, od glodalica i tokarilica, preko galvoglava, transportnih traka i 3D pisača, do robotskih manipulatora. Iako su to sve uređaji s potpuno različitim namjenama, njihovo programiranje svodi se u principu na isti zadatak, upravljanje pogonskim motorima posmičnih i glavnih prigona, za što je LinuxCNC idealna platforma. Blok dijagram na slici 2.1 shematski prikazuje način na koji LinuxCNC upravlja motorom. Program interpretacijom G koda u kojem su sadržane naredbe o gibanjima prigona dobiva instrukcije o zahtijevanim koordinatama u kartezijskom koordinatnom sustavu. Prema tim koordinatama, komponenta za planiranje gibanja interpolacijom određuje trajektoriju koju prigon mora opisati. Nakon toga izračunava se kinematika planiranog gibanja i takve se informacije šalju kontroleru motora koji onda izvršava zadane naredbe.



Slika 2.1: Blok dijagram upravljanja motorom [9]

Sustav LinuxCNC sastoji se od tri glavne komponente koje predstavljaju osnovnu strukturu svakog upravljačkog programa [8]:

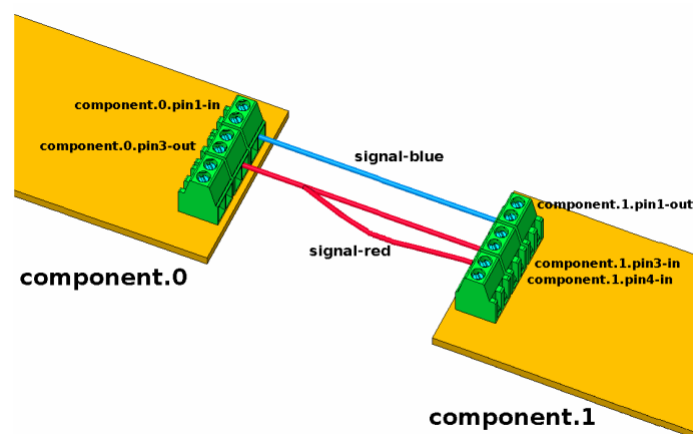
- GUI (*engl. Grapical User Interface*), grafičko korisničko sučelje
- HAL (*engl. Hardware Abstraction Layer*), sloj za apstrakciju hardvera
- EMC (*engl. Enhanced Machine Control*) komponente za kontrolu gibanja: EMCOT, EMCIO i EMCTASK

Sustav sadrži mnoštvo dodatnih komponenta. Čarobnjaci za postavljanje koračnih motora (*Stepconf wizard*) i kontrolnih pločica (*PNCconf wizard*) služe za jednostavljanje postavljanje i parametrisiranje hardvera. Sustav ima integrirani virtualni osciloskop, *HALscope*, koji omogućuje očitavanje i grafički prikaz vrijednosti bilo kojeg signala u stvarnom vremenu. Uz postojeće grafičko korisničko sučelje moguće je pomoću komponente *PyVCP* (*Python Virtual Control Panels*) izradivati vlastite grafičke komponente koje se mogu namijeniti za praćenje i postavljanje vrijednosti signala. LinuxCNC podržava i programiranje PLC (engl. *Programmable Logic Controller*) sustava te sadrži aplikaciju za izradu PLC ljestvičastih dijagrama. PLC kontroleri se kod projektiranja CNC sustava koriste za upravljanje sekundarnim sustavima poput podmazivanja i izmjene alata.

2.2 Komponente LinuxCNC sustava

2.2.1 HAL

U općenitom smislu, na razini operacijskog sustava, HAL (engl. *Hardware Abstraction Layer*) je apstraktni međusloj između hardvera i softvera koji operacijskom sustavu računala omogućuje komunikaciju s hardverom bez obzira na kojoj je arhitekturi hardver građen. HAL osigurava univerzalno i standardizirano sučelje za povezivanje različitih perifernih komponenta s jezgrom operacijskog sustava [10]. U kontekstu LinuxCNC sustava, HAL se definira kao sloj koji omogućuje učitavanje i povezivanje različitih građevnih jedinica upravljačkog programa u kompleksan i cjelovit sustav [8]. Navedene građevne jedinice nazivaju se HAL komponentama. HAL komponenta može se shvatiti kao blok (*black-box*) koji jednu ili više ulaznih vrijednosti (*inputa*) nekom funkcijom transformira u izlazne vrijednosti (*outpute*). Prema definiciji, HAL komponenta je softverski blok s jednoznačno definiranim ulazima, izlazima i funkcijom koji se može pokrenuti i spojiti s drugim komponentama prema potrebi [8]. Označavaju se prema pravilu *komponenta.N* gdje *komponenta* označava naziv korištene HAL komponente, a *N* je redni broj instance komponente koja se koristi. Zadana vrijednost parametra *N* je 0, a povećava se za 1 za svaku dodatnu istovrsnu komponentu. Ulazi i izlazi HAL komponenta realizirani su pomoću izvodnica (*pinova*) koji se mogu shvatiti kao sučelja za spajanje signala na komponente. Pinovi također imaju svoje nazive koji se dodjeljuju prema pravilu *komponenta.N.pin* gdje *pin* označava naziv dotičnog pina. Komponente se preko pinova povezuju signalima. Signal je informacija o vrijednosti neke veličine. Unutar HAL okruženja svakom signalu se dodjeljuje proizvoljni naziv. Signal s jedne strane može biti spojen na točno jedan izlazni pin, a s druge strane na proizvoljni broj ulaznih pinova. Kao što prikazuje slika 2.2, spajanje HAL komponenta signalima analogno je spajanju elektroničkih uređaja vodovima. U ovoj usporedbi, elektronički uređaj predstavlja HAL komponentu, sučelje za spajanje vodova je HAL pin, a vodovi su signali.



Slika 2.2: Spajanje HAL komponenta [8]

Svaki pin i signal ima svoj tip koji se određuje prema vrsti podatka koji prenosi. Postoje četiri tipa:

- bit - binarna veličina, može poprimiti vrijednosti 1 ili 0,
- float - realni broj veličine do 64 bita,
- s32 - cijeli broj s predznakom, može poprimiti vrijednosti od -2 147 483 647 do +2 147 483 647 i
- u32 - cijeli broj bez predznaka, može poprimiti vrijednosti od 0 do 4 294 967 295.

Postoji velik broj ugrađenih HAL komponenata, a moguće je izrađivati i vlastite komponente. U tablici 2.1 je dan prikaz s nazivima i opisima komponenata korištenih u izradi upravljačkog programa za galvo glavu u sklopu ovog rada.

Tablica 2.1: HAL komponente korištene u radu

HAL komponente	Opis
hostmot2	Driver za HostMot2 firmware Mesa kontrolera
hm2_eth	Driver za ethernet konekciju Mesa kontrolera s računalom
pid	PID regulator
not	Inverter vrijednosti tipa bit
trivkins	Definira kinematiku u kartezijskom koordinatnom sustavu stroja
motmod	Kontrola gibanja motora, generira putanju od trenutne do zadane pozicije
scale	Skalira vrijednost ulazne vrijednosti prema formuli $izlaz = koeficijent \cdot ulaz + pomak$

HAL komponente se u LinuxCNC-u učitavaju i međusobno povezuju u .hal datotekama. U nastavku je dan pregled sintakse HAL programskog kôda.

Svaka HAL konfiguracijska datoteka počinje učitavanjem HAL komponenti koje će biti korištene. Komponenta se ne može koristiti ako prethodno nije učitana. Postoje dvije vrste HAL komponenti: *real-time* i *userspace* komponente. *User-space* komponente izvode se u stvarnom vremenu u točno određenim, preciznim, vremenskim periodima i njihovo izvođenje ima prednost nad *user-space* komponentama koje se izvode kada su za njihovo izvođenje dostupni potrebni resursi. *User-space* komponente pogodne su za pokretanje aplikacija poput *PyVCP* upravljačkih ploča pošto su kod njihovog korištenja kašnjenja reda veličine jedne milisekunde prihvatljiva. Prilikom učitavanja komponente, ovisno o kojoj se vrsti komponente radi, mogu se prema potrebi navesti različiti parametri koji definiraju njezino ponašanje.

```
loadrt <komponenta> <parametri>
loadusr <komponenta> <parametri>
```

Svaka *real-time* komponenta treba se povezati s funkcijom osvježavanja. Zbog same prirode rada računala, koje radi s diskretnim podacima, potrebno je zadati takt po kojem se obavljaju operacije kako bi se dobila iluzija kontinuiranog rada HAL komponente. *Userspace* komponentama nije potrebno dodavati funkcije osvježavanja.

```
addf <komponenta> <thread>
```

Nakon što se učitaju sve komponente te im se pridruže petlje funkcija osvježavanja, može se početi s povezivanjem signala i komponenta čime se formira logička struktura upravljačkog sustava. Signal se deklarira i povezuje s komponentama pomoću naredbe `net`:

```
net <ime_signala> <komponenta1.N.pin> <komponenta2.N.pin>
```

Ako se signal povezuje na K različitih ulaznih pinova, sve konekcije mogu se osvariti jednom `net` naredbom na način da se ostali ulazni pinovi dodaju nakon prvog. Ako postoji potreba da se vrijednost na nekoj izvodnici postavi ručno, ta se vrijednost može postaviti naredbom `setp` bez da je se poveže na signal.

```
setp <komponenta.N.pin> <vrijednost>
```

Isto se tako vrijednost može zadati i signalu naredbom `sets`:

```
sets <ime_signala> <vrijednost>
```

HAL konfiguracije se često, zbog lakše čitljivosti, dijele na više `.hal` datoteka. Svaka `.hal` datoteka sadrži jednu logičku cjelinu programa. Tako se, na primjer, u jednoj datoteci učitavaju sve komponente, u drugoj se pridružuju funkcije osvježavanja, u trećoj se spaja regulacijska petlja za x os, itd. U tom slučaju sve se `.hal` datoteke naredbom `source` pozivaju u glavnoj `.hal` datoteci.

```
source <lokacija_na_disku>
```

Uz HAL datoteku, svaka LinuxCNC konfiguracija mora sadržavati i INI datoteku. INI datoteka je popis varijabli s njihovim vrijednostima. HAL i INI datoteke međusobno su povezane na način da se u INI datoteci navodi lista HAL konfiguracijskih datoteka koje će se učitati, nakon čega je unutar HAL datoteke moguće referencirati parametre iz INI datoteke. Na taj se način izbjegava upisivanje brojevnih vrijednosti izravno u HAL datoteci što povećava čitljivost programskog kôda. Također, ako se neki parametar koristi više puta, dođe li do promjene njegove vrijednosti dovoljno ga je promijeniti samo na jednom mjestu unutar INI datoteke. Ispod je prikazana općenita struktura INI datoteke.

```
[SEKCIJA_1]
PARAMETAR_1 = vrijednost_1
PARAMETAR_2 = vrijednost_2
[SEKCIJA_2]
PARAMETAR_3 = vrijednost_3
.
.
.
```

Ako se u HAL datotekama koristi parametar definiran u INI datoteci, on se referencira prema pravilu: `[SEKCIJA]PARAMETAR`, npr.:

```
setp komponenta.N.pin [SEKCIJA_1]PARAMETAR_2
```

Unutar INI datoteke može se definirati proizvoljan broj sekcija i varijabli s proizvoljnim imenima i vrijednostima. Svaka INI datoteka sadrži predefinirane sekcije čiji je pregled dan u tablici 2.2.

Tablica 2.2: INI sekcije [8]

INI sekcija	Opis
[EMC]	Opće informacije (verzija, ime stroja)
[DISPLAY]	Postavke GUI-a, koji GUI se koristi
[FILTER]	Popis datoteka s ekstenzijama i interpreterima koji se za svaku ekstenziju mogu pozvati
[RS274NGC]	Parametri interpretera G-koda
[EMCMOT]	Parametri komponente za kontrolu gibanja koja se izvršava u stvarnom vremenu
[TASK]	Parametri EMCTASK komponente
[HAL]	Definicija HAL konfiguracijskih datoteka koje će se učitati tijekom podizanja sustava
[HALUI]	Dodatni parametri koji mogu proširiti mogućnosti upravljanja CNC strojem
[APPLICATIONS]	Dodatne aplikacije koje se pokreću prije pokretanja GUI-a, npr. <i>halscope</i>
[TRAJ]	Dodatni parametri kontrolera gibanja (planiranje putanje alata)
[AXIS_n]	Parametri osi, svaka os ima zasebnu sekciju (brzina, akceleracija, granice)
[JOINT_n]	Parametri pogonskog motora posmičnog prigona, svaki motor ima zasebnu sekciju
[KINS]	Definicija kinematike stroja (broj osi, vrsta koordinatnog sustava)
[EMCIO]	Parametri I/O kontrolera

Sekcije [AXIS] i [JOINT] često imaju jednake vrijednosti svojih parametara pošto će, na primjer, brzina osi odgovarati brzini motora koji ju pokreće. Sekcija [AXIS] odnosi se na gibanje s primijenjenim kinematskim modelom, odnosno na gibanje same osi stroja koje može biti linearno. Čak i ako je kinematika osi linearna, gibanje pogonskog motora je i dalje rotacijsko zbog čega [JOINT] postoji kao zasebna sekcija i koristi se za parametrisiranje samog stupnja slobode. Isto tako, parametri motora i osi odvojeni su u dvije sekcije i iz razloga što jednu os može pokretati više od jednog motora.

2.2.2 EMC

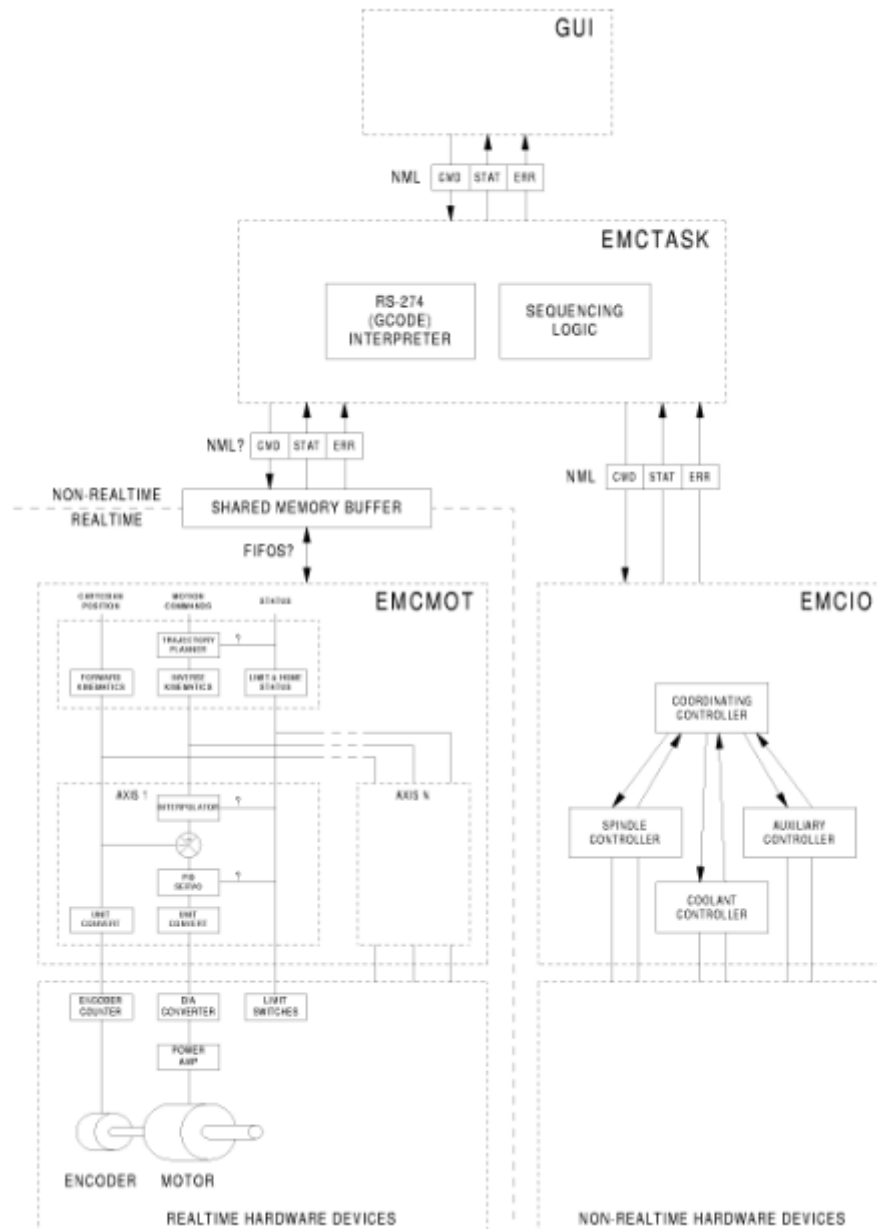
EMC (*Enhanced Machine Control*) je modul za računalnu kontrolu alatnim strojevima. 2003. godine razvijena je nova verzija, EMC2, u kojoj je uveden ranije opisani HAL. EMC2 se sastoji od četiri modula: EMCMOT, EMCIO, EMCTASK i GUI.

EMCMOT je kontroler gibanja. EMCMOT prema danj naredbi računa koordinate sljedeće pozicije te interpolira putanju između te dvije točke. Podaci o brzini i potrebnom zakretu se nakon toga šalju podređenim regulatorima pogonskih motora posmičnih osi čime se zatvara petlja regulacije

gibanja tog motora.

EMCIO je diskretni I/O kontroler, modul koji upravlja sekundarnim funkcijama stroja kao što su paljenje i gašenje stroja, aktivnost glavnog vretena, protok rashladnog sredstva i izmjena alata. Učitava se u [EMCIO] sekciji INI datoteke gdje se može odabrati koristi li se *io* ili *iov2* verzija I/O kontrolera. *iov2* sadrži proširenu podršku i dodatne pinove za automatsku izmjenu alata.

EMCTASK je izvršni modul EMC-a. Kao što je prikazano na slici 2.3, u hijerarhiji arhitekture EMC-a nalazi se ispod GUI-a te iznad modula EMCMOT i EMCIO.

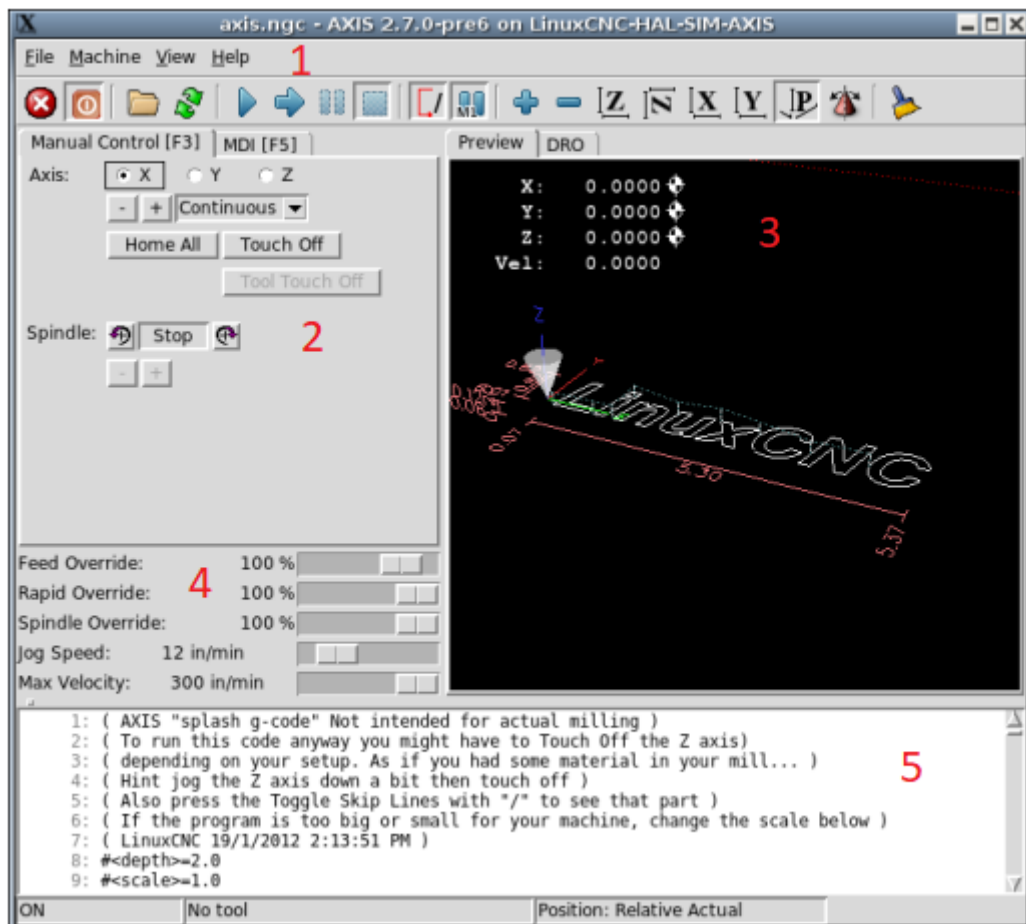


Slika 2.3: Arhitektura EMC sustava [11]

EMCTASK interpretira i izvršava naredbe G koda te nadgleda i koordinira rad modula EMCMOT i EMCIO [9]. G kod može se zadavati unutar GUI-a, naredbu po naredbu, koristeći MDI (*Machine Device Interface*) ili u posebnoj datoteci *.ngc* formata. Datoteka s G kodom se učitava u [DISPLAY] sekciji INI datoteke.

2.2.3 GUI

GUI (*Graphical User Interface*) je grafičko korisničko sučelje koje korisniku omogućuje jednostavno i intuitivno upravljanje strojem bez potrebe za ulaženjem u kod upravljačkog programa. Kao što je prikazano na slici 2.3, GUI je na vrhu hijerarhije EMC sustava. Komunicira direktno s EMCTASK modulom kojem šalje naredbe koje korisnik unese. LinuxCNC dolazi s pet ugrađenih GUI-a: Axis, Touchy, Gscreen, GMOCCAPY i NGCGUI. Odabrani GUI se navodi u [DISPLAY] sekciji INI datoteke pod parametrom DISPLAY. U ovom radu korišten je Axis GUI koji je i postavljen kao zadani GUI ako se ne odabere neki drugi.



Slika 2.4: Axis GUI

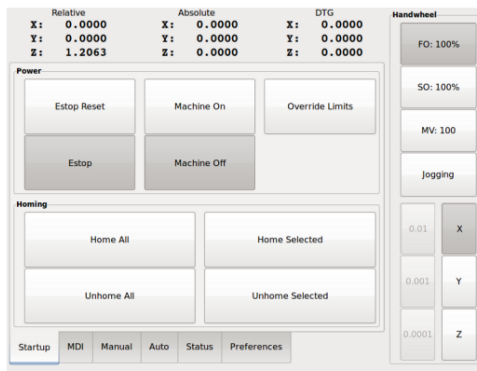
Na slici 2.4 prikazan je Axis GUI. Brojčanim oznakama od 1 do 5 označeni su njegovi glavni dijelovi:

1. Alatna traka
2. Alati za ručno upravljanje strojem
3. Prikaz pozicije vrha alata u 3D prostoru
4. Alati za kontrolu brzine
5. Prikaz aktivnog G koda

Kartica za ručnu kontrolu prikazuje osi stroja. Svaku os moguće je pokretati u oba smjera na dva načina: kontinuirano i inkrementalno. Osi se ovdje također mogu dovesti u nultu poziciju (*Home*) i pomaknuti za određeni interval (*Touch Off*). Osim osi, ovdje se može kontrolirati i gibanje glavnog

vretena. Prostorni prikaz u stvarnom vremenu prati poziciju vrha alata i ispisuje njegove koordinate i brzinu svake osi. Ovdje je također prikazan i CAD model izratka. Sučelje za kontrolu brzine omogućuje promjenu pet parametara. *Feed override* i *Rapid override* upravljaju dozvoljenim prekoračenjem maksimalne posmične brzine kod obrade odnosno brzine pozicioniranja alata. *Spindle override* omogućava ručno podešavanje programski zadane frekvencije vrtnje glavnog vretena. *Jog speed* se odnosi na brzinu gibanja kod ručnog upravljanja strojem, a *Max velocity* je sveukupna maksimalna brzina.

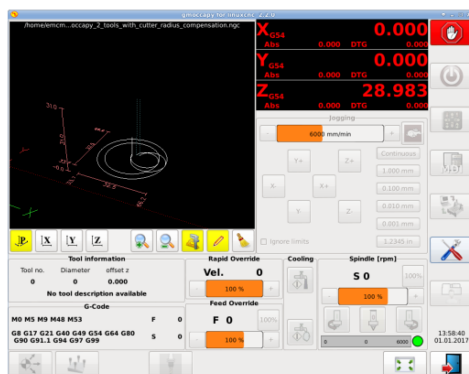
Axis je GUI najbolje prilagođen upravljanju mišem i tipkovnicom. Touchy, Gscreen i GMOCCAPY, prikazani na slikama , zamišljeni su kao grafička sučelja za zaslone osjetljive na dodir. NGCGUI, prikazan na slici , poseban je po činjenici da ima ugrađen čarobnjak za pisanje G koda.



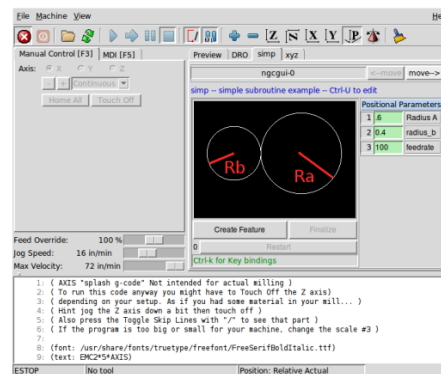
Slika 2.5: Touchy GUI [8]



Slika 2.6: Gscreen GUI [8]



Slika 2.7: GMOCCAPY GUI [8]



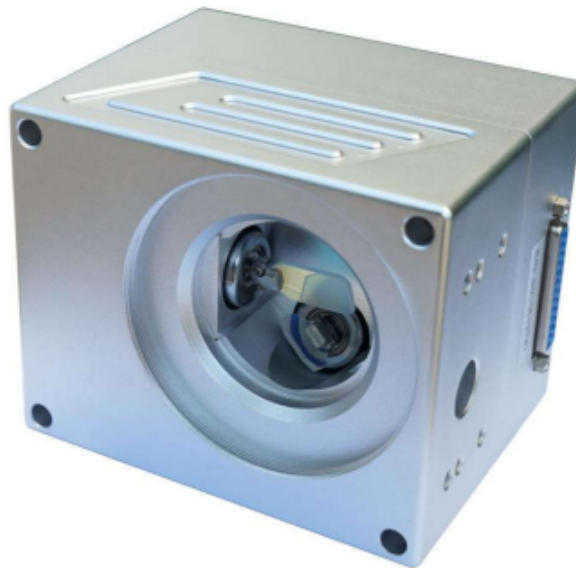
Slika 2.8: NGCGUI [8]

2.3 Integracija prigona za lasersko graviranje s LinuxCNC sustavom

2.3.1 Prigoni za lasersko graviranje

Proces laserskog graviranja realizira se ugradnjom prigona za lasersko graviranje na numerički upravljani stroj (CNC) koji usmjerava njegovu svjetlost prema području materijala koje treba biti obrađeno. Stroj za lasersko graviranje može se izvesti na tri načina: korištenjem galvo glave, ugradnjom lasera na z os stroja portalne izvedbe ili ugradnjom lasera na mjesto izvan radnog područja stola uz korištenje zrcala.

Galvo glava je uređaj koji, koristeći fizikalnu pojavu refleksije, usmjerava lasersku zraku po ravnini. Sastoji se od dva galvanometra sa zrcalima pričvršćenim na njihova vratila, leće i sučelja za montiranje lasera. Laserska zraka upada na zrcalo, prvo na zrcalo koje služi za usmjeravanje po y osi, a od njega se odbija na drugo zrcalo koje je usmjerava po x osi. Zraka nakon toga prolazi kroz leću koja ju fokusira na određenoj udaljenosti. Na toj udaljenosti treba biti postavljen materijal. Postoje i galvo glave s lećama promijenjive žarišne duljine što dodaje i treću os. U ovom radu korištena je dvoosna galvo glava SINO-GALVO SG7110 prikazana na slici 2.9. Montirana je na z os eksperimentalne glodalice ADRISS.



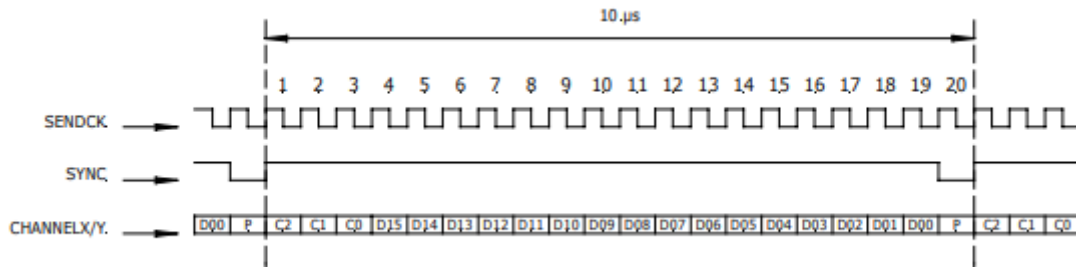
Slika 2.9: Galvo glava SINO-GALVO SG7110 [12]

Ako se ne koristi galvo glava, laser za graviranje može se montirati na z os alatnog stroja. Laserska zraka tada se ne usmjerava kutnim pomacima motora sa zrcalima već gibanjem x i y osi stroja. Razlika između te dvije izvedbe vidi se u brzini procesa graviranja. Galvo glava omogućuje puno brže pomake laserske zrake.

Treći način izvedbe prigona za lasersko graviranje koristi se ako je odabran laser velikih dimenzija što je čest slučaj kod CO₂ lasera. Takvi laseri preveliki su da bi ih se montiralo na z os stroja pa se oni montiraju tako da budu nepomični izvan radnog područja stroja. Laserski snop tada se do materijala dovodi pomoću zrcala koje se ugrađuju na svaku od osi stroja. Na z os stroja ugrađuje se leća kroz koju snop prolazi između zrcala koje ga usmjerava prema radnom komadu i samog radnog komada. Takva izvedba zahtijeva vrlo preciznu i čestu kalibraciju svih zrcala jer bi svaka eventualna nepreciznost mogla uzrokovati da snop ne pogodi središte leće.

2.3.2 XY2-100

Galvo glava s računalom komunicira pomoću XY2-100 protokola. XY2-100 je protokol koji šalje x i y koordinate između kontrolera i galvo glave [13]. Na slici 2.10 prikazana je shema signala za komunikaciju putem XY2-100 protokola.

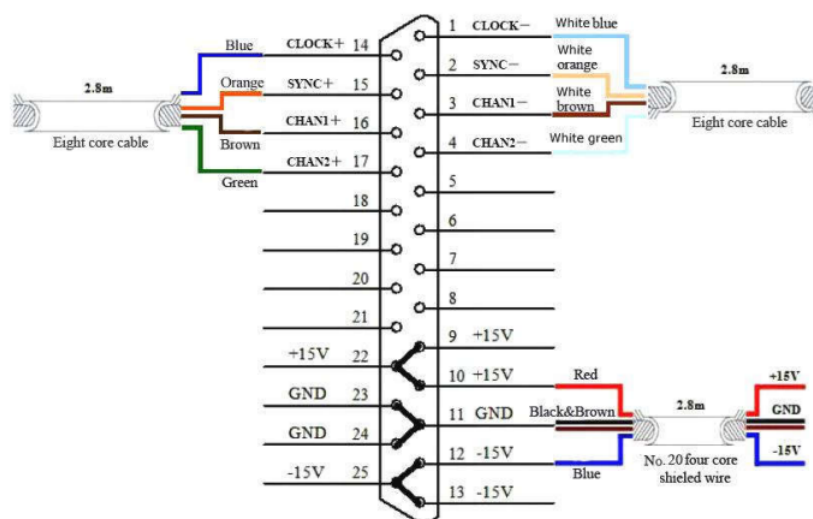


Slika 2.10: Shema XY2-100 [13]

XY2-100 protokol sastoji se od pet signala [13]:

- *SENDCK* - taktni signal frekvencije 2 MHz, na višem naponu kontroler šalje podatke, na nižem galvo glava obrađuje podatke
- *SYNC* - sinkronizacija prijenosa podataka
- *CHANNELX* - prijenos podataka za x os, 20 bitova
- *CHANNELY* - prijenos podataka za y os, 20 bitova
- *STATUS* - 1 bit o statusu galvo glave, ne ovisi o *SENDCK* signalu

Ovaj protokol je opće prihvaćeni standard u industriji laserskog graviranja. Galvo glava SINO-GALVO SG7110 također radi na njemu. Protokol je fizički realiziran preko DB25 konektora kojime se galvo glava spaja s kontrolerom. Može se vidjeti na slici 2.9, a shema je prikazana na slici 2.11.



Slika 2.11: DB25 konektor [12]

XY2-100 može prenositi 16 bitova podataka za svaku os i podržava do 3 zasebne osi. Zbog ograničenja XY2-100 protokola, razvijena je nova verzija. XY3-100 je novi, poboljšani protokol s

većom preciznošću. Podržava do pet osi, može prenositi do 26 bitova podataka i kompaktilan je s XY2-100 protokolom i DB25 konektorom. Osim navedenih protokola, postoje dva alternativna: SL2-100 i RL3-100, no njih koristi samo nekoliko proizvođača [14].

2.3.3 LinuxCNC xy2mod

LinuxCNC ima ugrađenu podršku za XY2-100 protokol. U sklopu *hostmot2* drivera za Mesa kontrolere nalazi se HAL komponenta *xy2mod* koja omogućuje integraciju galvo glava u HAL programe i LinuxCNC konfiguracije. Iako relativno nova i slabo dokumentirana, sadrži sve funkcije potrebne za potpunu kontrolu galvo glave. U tablici dan je pregled pinova koje *xy2mod* komponenta daje na korištenje.

Tablica 2.3: Pinovi *xy2mod* komponente [15]

Pin	Tip	Opis
posx_cmd / posy_cmd	float in	Tražena pozicija x/y
posx_fb / posy_fb	float out	Povratna veza po poziciji x/y
velx_cmd / vely_cmd	float in	Tražena brzina x/y
velx_fb / vely_fb	float out	Povratna veza po brzini x/y
accx_cmd / accy_cmd	float in	Tražena akceleracija x/y
posx_scale / posy_scale	float in	Skaliranje ulaznih ili izlaznih vrijednosti
enable	bit in	Omogućavanje rada prigona
controlx / controly	u32 in	Kontrolira 3 kontrolna bitova CHANNEL signala
commandx / commandy	u32 in	16 podatkovnih bitova CHANNEL signala
commandmodex / commandmodey	bit in	Dozvola za slanje naredbi za gibanje
18bitmodex / 18bitmodey	bit in	Uključuje način rada s 18 bitnim podacima
posx-overflow / posy-overflow	bit out	Indikacija prekoračenja maksimalne ili minimalne pozicije
velx-overflow / vely-overflow	bit out	Indikacija prekoračenja maksimalne ili minimalne brzine
status	u32 out	16 bitni statusni registar

3 Postavljanje i parametriranje hibridnog stroja

3.1 Pisanje upravljačkog programa za galvo glavu

3.1.1 HAL program

Prvi korak postavljanja hibridnog stroja je izrada HAL konfiguracijske datoteke za galvo glavu kako bi se ona mogla pokrenuti i testirati. U nastavku je prikazan i opisan svaki dio prve verzije programa. Tijekom rada dolazilo je do promjena programa što je kasnije opisano.

Na početku HAL programa učitavaju se *real-time* komponente.

```
loadrt hostmot2
loadrt hm2_eth board_ip="10.10.10.10" config="num_encoders=1
        num_stepgens=1 num_pwmgens=1
        sserial_port_0=0000xxxx enable_raw"
loadrt pid num_chan=2
loadrt trivkins coordinates=uv
loadrt motmod servo_period_nsec=400000 num_joints=2
loadrt scale count=4
```

U prvoj i drugoj liniji učitavaju se *hostmot2* i *hm2_eth*, driveri za Mesa kontroler. Dalje se učitavaju dvije instance, po jedna za svako zrcalo, komponente *pid* koja derivira poziciju i aplicira parametre Pgain, FF1 i FF2, a *num_chan=2* naređuje da se moraju učitati dvije instance komponente. Komponenta *trivkins* učitana u četvrtoj liniji definira kinematiku i koordinatni sustav stroja. U petoj liniji učitava se *motmod*, planer gibanja, kojem se ovdje definira takt update funkcije (400000 ns) i broj osi. U posljednjoj liniji učitavaju se četiri instance komponente *scale* koja skalira vrijednost pozicije.

U sljedećem dijelu komponentama se pridružuju funkcije osvježavanja. Svakoj komponenti dodaje se petlja koja se naziva *servo-thread*. To je petlja koja pokreće funkciju osvježavanja i njezin period definira se u INI datoteci parametrom *servo-thread*.

```
addf hm2_7i96.0.read          servo-thread
addf hm2_7i96.0.write        servo-thread
addf pid.3.do-pid-calcs      servo-thread
addf pid.4.do-pid-calcs      servo-thread
addf motion-command-handler  servo-thread
addf motion-controller      servo-thread
addf scale.0                 servo-thread
addf scale.1                 servo-thread
addf scale.2                 servo-thread
addf scale.3                 servo-thread
```

Nakon što su sve komponente učitane i spojene, dodaju se vrijednosti određenim ulaznim pinovima koji definiraju postavke Mesa 7i96 kontrolera:

```
setp hm2_7i96.0.watchdog.timeout_ns 25000000
setp hm2_7i96.0.dpll.01.timer-us -50
setp hm2_7i96.0.stepgen.timer-number 1
setp hm2_7i96.0.xy2mod.read-timer-number 1
setp hm2_7i96.0.xy2mod.write-timer-number 1
setp hm2_7i96.0.packet-read-timeout 70
setp hm2_7i96.0.packet-error-limit 50
```


Počevši od prve linije gornjeg bloka koda: *watchdog.timeout_ns* definira, prema LinuxCNC dokumentaciji, koliko vremena treba proći prije nego što će *watchdog* "ugristi" [15]. *Watchdog* predstavlja sigurnosni sustav koji radi na način da od upravljačkog programa zahtijeva da ga on periodički osvježava. Ako iz bilo kojeg razloga *Watchdog* nije osvježen unutar zadanog roka, zaustavlja se slanje svih upravljačkih signala. *dpl.01.timer-us* definira vremenski odmak u kojem će se provesti neka akcija prije nego *hostmot2* očita podatke. *stepgen.timer-number* postavlja timer koji služi za zaustavljanje *stepgen* brojač. *xy2mod.read-timer-number* i *xy2mod.write-timer-number* postavljaju timere za uzorkovanje registara pozicije i brzine prije očitavanja podataka i za update podataka nakon slanja.

U sljedećem bloku definiraju se vrijednosti parametara PID regulatora i *scale* komponente:

```
setp pid.3.Pgain      [AXIS_U]PGAIN
setp pid.3.FF1       [AXIS_U]FF1
setp pid.3.FF2       [AXIS_U]FF2
setp pid.3.maxoutput [AXIS_U]MAXOUTPUT

setp pid.4.Pgain      [AXIS_V]PGAIN
setp pid.4.FF1       [AXIS_V]FF1
setp pid.4.FF2       [AXIS_V]FF2
setp pid.4.maxoutput [AXIS_V]MAXOUTPUT

setp scale.0.gain     [SCALE]MIN_SCALE
setp scale.1.gain     [SCALE]MAX_SCALE
setp scale.2.gain     [SCALE]MIN_SCALE
setp scale.3.gain     [SCALE]MAX_SCALE
```

Osi x i y galvo glave u ovoj konfiguraciji nazvane su osima u i v jer su x, y i z ostali rezervirani za osi glodalice. Prve četiri linije gornjeg bloka postavljaju vrijednosti parametara Pgain, FF1, FF2 i maxoutput za os u, a druge četiri za os v. Pgain odnosi se na proporcionalnu konstantu K_p kod PID regulatora. FF1 i FF2 su parametri unaprijednog pojačanja (*feed-forward*) regulatora prvog i drugog reda (odnosno po brzini i akceleraciji) koji se množe s prvom, odnosno drugom derivacijom referentne vrijednosti pozicije i kompenziraju za greške slijeđenja. *maxoutput* postavlja maksimalnu vrijednost izlaznog signala. Sljedeće četiri linije postavljaju vrijednosti koeficijenata kojima se množe signali kako bi se oni skalirali.

Nakon što su postavljene vrijednosti svih potrebnih ulaznih pinova, učitane HAL komponente povezuju se u upravljačku strukturu. Prvo signali koji nose traženu vrijednost pozicije izlaze iz planera gibanja, skaliraju se te njihove skalirane vrijednosti ulaze u PID kontroler:

```
net u-pos-cmd  <= joint.3.motor-pos-cmd
net u-pos-cmd  => scale.0.in
net scaled-u-pos-cmd  scale.0.out => pid.3.command

net v-pos-cmd  <= joint.4.motor-pos-cmd
net v-pos-cmd  => scale.2.in
net scaled-v-pos-cmd  scale.2.out => pid.4.command
```

Iako su učitane dvije instance planera gibanja i PID kontrolera, one su ovdje označene s *joint.3* i *joint.4* te *pid.3* i *pid.4* jer se vrijednosti od 0 do 2 koriste za tri osi glodalice.

Dalje se izlazne vrijednosti PID kontrolera, tražene brzine, spajaju u driver Mesa 7i96 kontrolera i xy2mod komponentu:

```

net u-vel-cmd <= pid.3.output
net u-vel-cmd => hm2_7i96.0.xy2mod.00.velx-cmd

net v-vel-cmd <= pid.4.output
net v-vel-cmd => hm2_7i96.0.xy2mod.00.vely-cmd

```

Izlazna vrijednost ove komponente je integrirana tražena brzina, a predstavlja povratnu vezu pozicije. Ta se povratna veza spaja u *feedback* ulazni pin PID kontrolera te se ponovo skalira i spaja u planer gibanja:

```

net u-vel-cmd <= pid.3.output
net u-vel-cmd => hm2_7i96.0.xy2mod.00.velx-cmd

net v-vel-cmd <= pid.4.output
net v-vel-cmd => hm2_7i96.0.xy2mod.00.vely-cmd

```

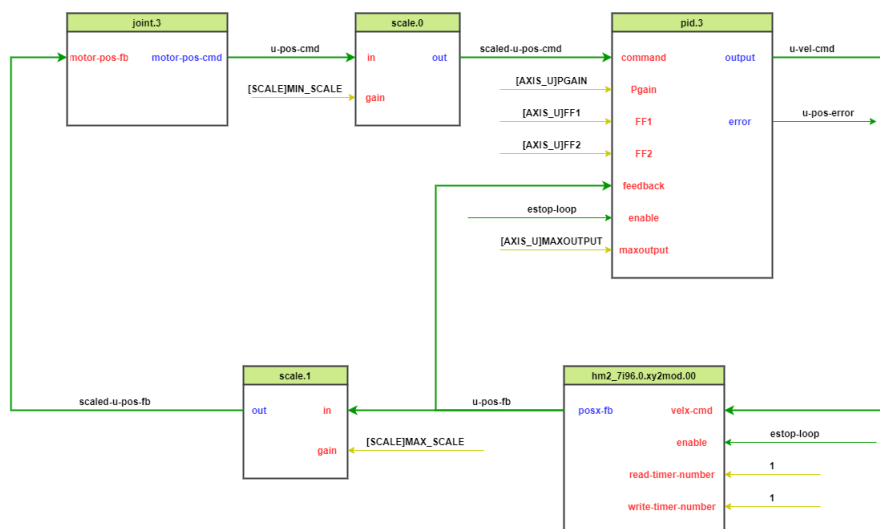
Ovime je gotova kontrola gibanja osi galvo glave. Potrebno je još spojiti signal za uključivanje i isključivanje u petlju:

```

net estop-loop iocontrol.0.user-enable-out
                    iocontrol.0.emc-enable-in
net estop-loop => hm2_7i96.0.xy2mod.00.enable
net estop-loop => pid.3.enable
net estop-loop => pid.4.enable

```

Ovim se kodom komponentama *pid* i *hm2_7i96.0.xy2mod* daje naredba trebaju li se pokrenuti ili ne. Prva linija omogućuje paljenje i gašenje konfiguracije te resetiranje tipke za hitno zaustavljanje. Time je HAL program završen i sljedeći je korak izrada INI datoteke. Petlja za kontrolu gibanja napravljena u ovom HAL programu shematski je prikazana na slici 3.1. Crvenom bojom označeni su ulazni pinovi, a plavom izlazni. Deblje zelene linije označavaju signale koji sudjeluju u regulacijskoj petlji, tanke zelene linije su signali dozvole, a žute linije su signali čije vrijednosti su postavljene ručno u gornjem kodu.



Slika 3.1: Shema regulacijske petlje galvo glave

3.1.2 INI datoteka

Sljedeći korak je izrada pripadajuće INI datoteke u kojoj su sadržani parametri konfiguracije. Slijedi prikaz i opis sekcija INI datoteke koje pripadaju galvo glavi i glodalici. U sekciji [EMC] nalaze se osnovni podaci EMC komponente kao što su naziv stoja i verzija.

```
[EMC]
VERSION = 1.1
MACHINE = adriss_galvo
DEBUG = 0
```

DEBUG = 0 znači da se tijekom rada konfiguracije neće ispisivati debug informacije. Sljedeća sekcija je [DISPLAY] u kojoj se definira i parametrira GUI.

```
[DISPLAY]
DISPLAY = axis
MAX_LINEAR_VELOCITY = 780
DEFAULT_LINEAR_VELOCITY = 25
POSITION_OFFSET = RELATIVE
POSITION_FEEDBACK = ACTUAL
PROGRAM_PREFIX = /home/dbagaric/linuxcnc/configs
                  /galvo_v3/gcod
OPEN_FILE = /home/dbagaric/linuxcnc/configs/galvo_v3
            /gcod/test1.ngc
```

Četiri sekcije koje slijede definiraju parametre EMCMOT, EMCIO i EMCTASK komponenti te interpretera G koda:

```
[TASK]
TASK = milltask
CYCLE_TIME = 0.010

[RS274NGC]
PARAMETER_FILE = adriss_galvo.var

[EMCMOT]
EMCMOT = motmod
SERVO_PERIOD = 400 000
COMM_TIMEOUT = 1.0
COMM_WAIT = 0.01

[EMCIO]
EMCIO = io
CYCLE_TIME = 0.100
TOOL_TABLE = alati.tbl
```

U [EMCIO] sekciji odabran je I/O kontroler *io* jer nema potrebe za proširenim mogućnostima *iov2* kontrolera. Ovdje je navedena i datoteka u kojoj se nalazi popis alata, *alati.tbl*, u kojoj se navode svi alati ako G program radi promjene alata. U [EMCMOT] sekciji učitana je planer gibanja te je definiran takt update funkcije.

U sekciji [HAL] referencira se HAL program kojem pripada ova INI datoteka:

```
[HAL]
HALFILE = adriss_main_emc.hal
HALUI = halui
```

U sljedeće dvije sekcije definirana je kinematika i koordinatni sustav stroja.

```
[TRAJ]
AXES = 5
COORDINATES = xyzuv
LINEAR_UNITS = mm
ANGULAR_UNITS = degree

[KINS]
JOINTS = 5
KINEMATICS = trivkins coordinates=xyzuv
```

Navedeno je da stroj ima pet osi i samim time pet stupnjeva slobode gibanja. Također su definirane mjerne jedinice za pomak i kut.

Ostalo je još parametrirati svaku od osi i stupnjeva slobode gibanja stroja u sekcijama [AXIS] i [JOINT]. Pošto se svaka od osi parametrira na isti način, ovdje je prikazana samo jedna os i to u os galvo glave. Sekcije za ostale osi razlikuju se po vrijednostima parametara, a potpuna INI datoteka može se vidjeti u Prilogu.

```
[AXIS_U]
MIN_LIMIT = -110.0
MAX_LIMIT = 110.0
MAX_VELOCITY = 780
MAX_ACCELERATION = 3100
PGAIN = 500
FF1 = 1.2
FF2 = 0.001
MAXOUTPUT = 780
```

Granične vrijednosti pomaka osi, *MIN_LIMIT* i *MAX_LIMIT*, određene su iz tehničkih podataka galvo glave. Prema tim podacima, maksimalni kutni pomak zrake je $\pm 15^\circ$ u odnosu na horizontalu [12]. Iz kutnog pomaka od 15° i žarišne duljine leće od 420 mm (navedeno na leći) dobiva se pomak prema izrazu:

$$pomak = 420 \cdot \tan(15)$$

$$pomak = 112,54mm \approx 110mm$$

Maksimalna brzina (*MAX_VELOCITY*) određena je tijekom testiranja postupnim povećavanjem vrijednosti. Maksimalna akceleracija (*MAX_ACCELERATION*) određena je prema iskustvenom pravilu da njezina vrijednost bude otprilike četiri puta veća od vrijednosti maksimalne brzine.

```
[JOINT_3]
TYPE = LINEAR
MAX_VELOCITY = 780
MAX_ACCELERATION = 10000
MIN_LIMIT = -110.0
MAX_LIMIT = 110.0
HOME_SEQUENCE = 0
HOME_SEARCH_VEL = 0.1
HOME_LATCH_VEL = -0.1
HOME = 0
HOME_OFFSET = 0
HOME_IGNORE_LIMITS = YES
HOME_ABSOLUTE_ENCODER = 2
```

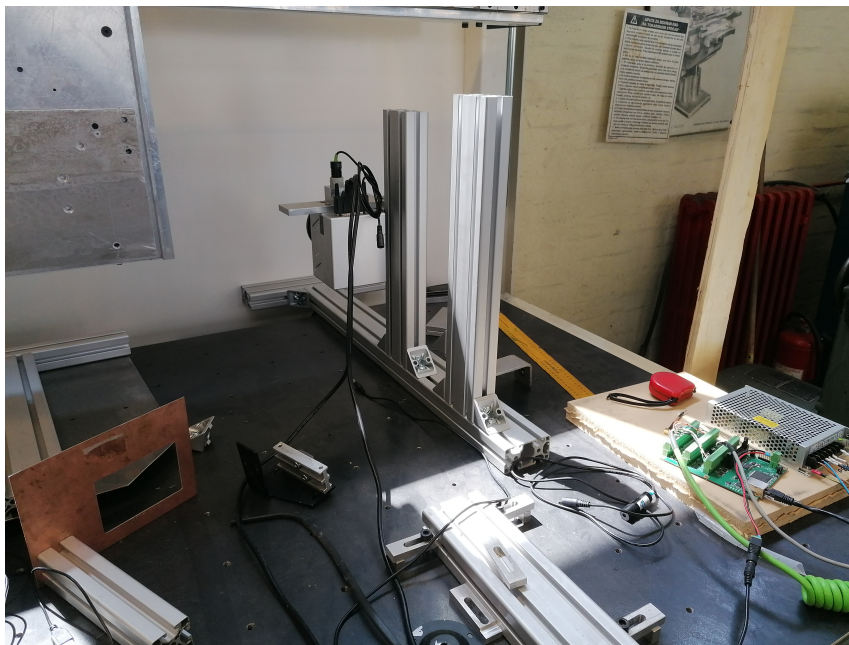
Parametar *TYPE* definira hoće li se granične vrijednosti pozicije, brzine i akceleracije motora zadavati u linearnim ili kutnim jedinicama. Granične vrijednosti pozicije, brzine i akceleracije motora jednake su vrijednostima osi jer samo taj motor upravlja tom osi.

Dodana je i sekcija [SCALE] u kojoj su definirane vrijednosti koeficijenata skaliranja. Razlog dodavanja komponente *scale* opisan je u sljedećem poglavlju rada.

```
[SCALE]
MIN_SCALE = 0.0064102564
MAX_SCALE = 156
```

3.2 Testiranje prigona za lasersko graviranje

Galvo glava je za potrebe testiranja motirana na nosač od aluminijskih profila postavljen na radni stol glodalice. Na stražnju stranu glodalice postavljena je ravna ploča koja služi kao ravnina za mjerenje pozicije laserske zrake. Galvo glava je postavljena tako da joj je leća od ploče udaljena 420 mm što je žarišna duljina leće. Testni postav prikazan je na slici 3.2. Za potrebe testiranja korišten je laserski pokazivač kao zamjena za industrijski laser za graviranje.



Slika 3.2: Testni postav galvo glave

Nakon pokretanja galvo glave, ustanovljeno je da postignuti pomak nije jednak željenom. Sljedeći korak je određivanje koeficijenta skaliranja kojim će se ispraviti greška. Na ploču je stavljen milimetarski papir u radnom području galvo glave. Određivanje koeficijenta provedeno je na sljedeći način:

1. Na papiru je označen položaj zrake u nultom položaju
2. Izvršen je pomak od 10 mm u smjeru osi u
3. Označena je pozicija zrake lasera na papiru
4. Koraci 2 i 3 ponavljani do granice osi
5. Izmjerena je udaljenost između točaka na papiru
6. Koraci od 1 do 5 ponovljeni su za os v

Koeficijent skaliranja definiran je kao omjer traženog i dobivenog pomaka. Dijeljenjem tražene i izmjerene vrijednosti dobiven je koeficijent skaliranja koji iznosi 0,0064102564. Komponenta *scale* dodana je u HAL program kako bi se tim koeficijentom pomnožila vrijednost signala zadane pozicije. Na taj način kontroleru se zadaje da pomak motora bude skaliran za koeficijent skaliranja čime se postiže točan pomak u stvarnosti. Prije vraćanja povratne veze u planer gibanja, vrijednost pozicije potrebno je ponovo skalirati za recipročnu vrijednost koeficijenta kako bi se motoru vratio točan podatak o trenutnoj poziciji. Test koji je bio proveden za određivanje koeficijenta skaliranja proveden je opet kako bi se utvrdilo odgovara li postignuti pomak traženom nakon skaliranja. Iz- mjerena vrijednost dobivena testiranjem odgovara traženoj iz čega se može zaključiti da je galvo glava ispravno parametirana.

3.3 Integracija galvo glave s glodalicom

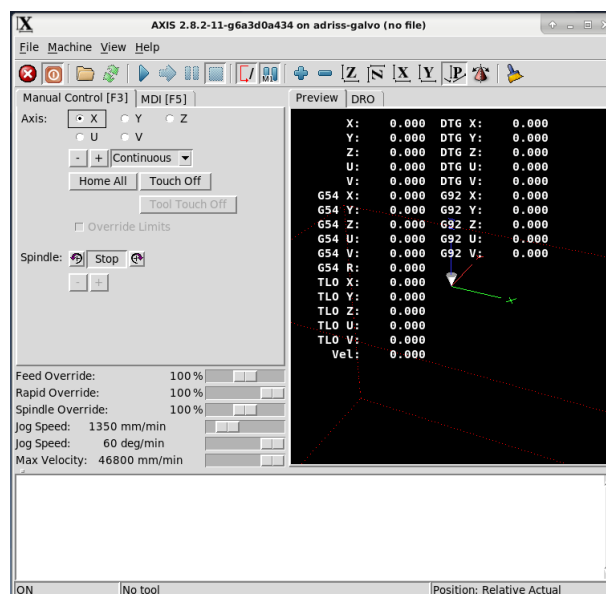
Nakon što je dovršen upravljački program galvo glave te je ona parametirana, njezina konfiguracijska HAL datoteka integrirana je s konfiguracijom za glodalicu. Glodalica je u trenutku testiranja već imala napravljenu HAL konfiguracijsku datoteku pa ona neće biti prikazana u sklopu ovog rada. HAL kod koji u upravljački program galvo glave učitava komponente koje se koriste i u programu glodalice je maknut jer iste komponente ne smiju biti učitane više puta. U HAL program glodalice dodane su dodatne instance komponenata rezervirane za galvo glavu. U glavnu HAL datoteku hibridnog stroja u kojoj se učitavaju svi ostali HAL programi dodana je linija:

```
source ./galvo_v3.hal
```

Ova naredba dodaje HAL program galvo glave u konfiguraciju hibridnog stroja. Posljednji korak je spajanje postojeće INI datoteke ADRISS glodalice s novom INI datotekom galvo glave u zajedničku datoteku koja parametrira cijeli hibridni stroj. Njezini dijelovi objašnjeni su u poglavlju 3.1.2 te je u cijelosti prikazana u Prilogu. Dovršena LinuxCNC konfiguracija pokrenuta je upisivanjem sljedeće naredbe u Linux terminal pokrenut u direktoriju s INI datotekom:

```
linuxcnc adriss-galvo.ini
```

Pokretanjem konfiguracije otvara se Axis GUI prikazan na slici 3.3.



Slika 3.3: Axis GUI hibridnog stroja

4 Zaključak

Glavni cilj ovog rada je integracija prigona za lasersko graviranje SINO-GALVO SG7110 s ispitnim postavom troosne glodalice ADRISS u hibridni alatni stroj te parametriranje istog. Postupak integracije, parametriranja i testiranja proveden je koristeći sustav za upravljanje i razvoj CNC strojeva otvorene arhitekture, LinuxCNC. Praktični dio rada sastojao se od pokretanja početne ispitne LinuxCNC upravljačke konfiguracije na prigonu za lasersko graviranje, izrade nove upravljačke konfiguracije koja omogućuje direktno upravljanje prigonom, kalibracije galvo glave i dodavanja kompenzacije greške u upravljačku konfiguraciju, integracije te upravljačke konfiguracije u postojeću konfiguraciju glodalice te testiranja rezultirajućeg hibridnog alatnog stroja.

Postavljanje i parametriranje hibridnog stroja započelo je testiranjem galvo glave koja je izvodila testnu upravljačku konfiguraciju. Ta konfiguracija nije sadržavala sučelje koje omogućuje direktnu kontrolu gibanja nego je driver kontrolera galvo glave bio spojen na *siggen*, HAL komponentu koja generira pravilne sinusne (ili slične) signale koji simuliraju korisnika. Naj taj način parametriran je PID regulator galvo glave. Nakon toga napravljena je upravljačka konfiguracija koja omogućuje zadavanje naredbi. Tijekom testiranja upravljačke konfiguracije dva nedostatka izašla su na vidjelo: postignuti pomak nije bio jednak zahtijevanom i konfiguracija bi se srušila prilikom dolaska zrake do granične pozicije u smjeru osi. Problem usklađivanja pomaka riješen je skaliranjem vrijednosti signala unutar upravljačke petlje pomoću HAL komponente *scale*. Faktor skaliranja vrijednosti dobiven je provođenjem mjerenja stvarnog pomaka. To mjerenje izvršeno je na način da je laserska zraka usmjerena na milimetarski papir na kojem su se označavale postignute pozicije. Udaljenosti pozicija su izmjerene te su vrijednosti traženih pomaka podijeljene izmjerenim vrijednostima čime je dobiven faktor skaliranja. Taj postupak nije najprecizniji no dovoljno je dobar za svrhu ove testne konfiguracije. Prema tehničkim karakteristikama galvo glave izračunate su minimalne i maksimalne granice pozicije te su one uvedene u [JOINT] i [AXIS] sekcije INI datoteke čime je riješen problem rušenja konfiguracije. Nakon što je prigon za lasersko graviranje parametriran, njegova upravljačka konfiguracija integrirana je s postojećom konfiguracijom glodalice. Za potrebe integracije napravljene su potrebne izmjene koda upravljačke konfiguracije. Cijeli sustav na kraju je ponovo testiran pokretanjem svake od pet osi, u svim smjerovima i različitim brzinama.

Postignuti sustav ima mjesta za poboljšanja. Fizičke komponente prigona laserskog graviranja potrebno je precizno montirati na glodalicu, a nakon toga može se eksperimentalnom metodom odrediti u kojem položaju treba biti z os glodalice kako bi žarišna točka laserske zrake točno po- gađala površinu obratka. Isto tako, laserski pokazivač korišten tijekom rada mora se zamijeniti laserom veće snage pogodnim za graviranje. Test korišten za kalibraciju galvo glave potrebno je ponovo provesti preciznijom metodom. Daljnjim parametriranjem prigona lasera moguće je proširiti funkcionalnost hibridnog stoja i uz graviranje omogućiti i lasersko rezanje.

Literatura

- [1] *CNC Machine Global Market Analysis, Insights and Forecast, 2022 - 2029*. Teh. izv. Fortune Business Insights, 2022.
- [2] Kapil Gupta, Neelesh K. Jain i R. F. Laubscher. *Hybrid Machining Processes: Perspectives on Machining and Finishing*. Springer, 2016.
- [3] shorturl.at/fmyIQ. Posjećeno 01.09.2022.
- [4] <https://enciklopedija.hr/Natuknica.aspx?ID=35494>. Posjećeno 01.09.2022.
- [5] <https://www.laserax.com/blog/types-lasers>. Posjećeno 01.09.2022.
- [6] <https://twotrees3d.com/laser-cutter-laser-engraver/>. Posjećeno 01.09.2022.
- [7] https://www.abmsale.com/?category_id=2950555. Posjećeno 01.09.2022.
- [8] *LinuxCNC V2.8.3-3-g92cfd719, 2022-08-20*. dokumentacija. 2022.
- [9] Staroveški T., Brezak D. i Udiljak T. "LinuxCNC – Napredni sustav CNC upravljanja: primjena i kritički osvrt". *Tehnički vjesnik* 20.6 (2013.). Preuzeto s <https://hrcak.srce.hr/112334>, str. 1103–1110.
- [10] <https://www.techopedia.com/definition/4288/hardware-abstraction-layer-hal>. Posjećeno 05.09.2022.
- [11] http://wiki.linuxcnc.org/cgi-bin/wiki.pl?EMC_Components. Posjećeno 15.09.2022.
- [12] *SG7110 Technical Specification*. Preuzeto s https://www.photonics.com/Buyers_Guide/Prod_Spec/Galvanometers/SG7110_2-Axis_Galvanometer_Scanner/psp7416.
- [13] <http://www.newson.be/doc.php?id=XY2-100>. Posjećeno 17.09.2022.
- [14] <https://www.halaser.de/compare.phpXY3>. Posjećeno 17.09.2022.
- [15] *LinuxCNC Manual Pages*. dokumentacija. 2022.

Prilog

I programski kod HAL upravljačke konfiguracije prigona za lasersko graviranje

II Kod INI datoteke

III CD-R disk

Programski kod HAL upravljačke konfiguracije prigona za lasersko graviranje: galvo_v3.hal

```
loadrt hostmot2
loadrt hm2_eth board_ip="10.10.10.10" config="num_encoders=1
        num_stepgens=1 num_pwmgens=1
        sserial_port_0=0000xxxx enable_raw"
loadrt scale count=4

addf hm2_7i96.0.read          servo-thread
addf pid.3.do-pid-calcs      servo-thread
addf pid.4.do-pid-calcs      servo-thread
addf hm2_7i96.0.write        servo-thread
addf scale.0                 servo-thread
addf scale.1                 servo-thread
addf scale.2                 servo-thread
addf scale.3                 servo-thread

setp hm2_7i96.0.watchdog.timeout_ns 25000000
setp hm2_7i96.0.dpll.01.timer-us -50
setp hm2_7i96.0.stepgen.timer-number 1
setp hm2_7i96.0.xy2mod.read-timer-number 1
setp hm2_7i96.0.xy2mod.write-timer-number 1
setp hm2_7i96.0.packet-read-timeout 70
setp hm2_7i96.0.packet-error-limit 50

setp pid.3.Pgain [AXIS_U]PGAIN
setp pid.3.FF1 [AXIS_U]FF1
setp pid.3.FF2 [AXIS_U]FF2
setp pid.3.maxoutput [AXIS_U]MAXOUTPUT

setp pid.4.Pgain [AXIS_V]PGAIN
setp pid.4.FF1 [AXIS_V]FF1
setp pid.4.FF2 [AXIS_V]FF2
setp pid.4.maxoutput [AXIS_V]MAXOUTPUT

setp scale.0.gain [SCALE]MIN_SCALE
setp scale.1.gain [SCALE]MAX_SCALE
setp scale.2.gain [SCALE]MIN_SCALE
setp scale.3.gain [SCALE]MAX_SCALE

net u-pos-cmd <= joint.3.motor-pos-cmd
net u-pos-cmd => scale.0.in
net scaled-u-pos-cmd scale.0.out => pid.3.command

net v-pos-cmd <= joint.4.motor-pos-cmd
```

```
net v-pos-cmd => scale.2.in
net scaled-v-pos-cmd scale.2.out => pid.4.command

net u-pos-fb <= hm2_7i96.0.xy2mod.00.posx-fb
net u-pos-fb => pid.3.feedback
net u-pos-fb => scale.1.in
net scaled-u-pos-fb scale.1.out => joint.3.motor-pos-fb

net v-pos-fb <= hm2_7i96.0.xy2mod.00.posy-fb
net v-pos-fb => pid.4.feedback
net v-pos-fb => scale.3.in
net scaled-v-pos-fb scale.3.out => joint.4.motor-pos-fb

net u-pos-error <= pid.3.error
net v-pos-error <= pid.4.error

net u-vel-cmd <= pid.3.output
net u-vel-cmd => hm2_7i96.0.xy2mod.00.velx-cmd

net v-vel-cmd <= pid.4.output
net v-vel-cmd => hm2_7i96.0.xy2mod.00.vely-cmd

net estop-loop iocontrol.0.user-enable-out
                iocontrol.0.emc-enable-in
net estop-loop => hm2_7i96.0.xy2mod.00.enable
net estop-loop => pid.3.enable
net estop-loop => pid.4.enable
```

Kod INI datoteke: adriss-galvo.ini

```
[EMC]
VERSION = 1.1
MACHINE = adriss_galvo
DEBUG = 0

[DISPLAY]
DISPLAY = axis
MAX_LINEAR_VELOCITY = 780
DEFAULT_LINEAR_VELOCITY = 25
POSITION_OFFSET = RELATIVE
POSITION_FEEDBACK = ACTUAL
PROGRAM_PREFIX = /home/dbagaric/linuxcnc/configs/galvo_v3/gcod
OPEN_FILE = /home/dbagaric/linuxcnc/configs/galvo_v3/
                gcod/test1.ngc

[FILTER]
PROGRAM_EXTENSION = .png,.gif,.jpg Grayscale Depth Image
PROGRAM_EXTENSION = .py Python Script
png = image-to-gcode
gif = image-to-gcode
jpg = image-to-gcode
```

```
py = python

[TASK]
TASK = milltask
CYCLE_TIME = 0.010

[RS274NGC]
PARAMETER_FILE = addriss_galvo.var

[EMCMOT]
EMCMOT = motmod
SERVO_PERIOD = 400 000
COMM_TIMEOUT = 1.0
COMM_WAIT = 0.01

[EMCIO]
EMCIO = io
CYCLE_TIME = 0.100
TOOL_TABLE = alati.tbl

[HAL]
HALFILE = addriss_main_emc.hal
HALUI = halui

[TRAJ]
AXES = 5
COORDINATES = xyzuv
LINEAR_UNITS = mm
ANGULAR_UNITS = degree

[KINS]
JOINTS = 5
KINEMATICS = trivkins coordinates=xyzuv

[AXIS_X]
MIN_LIMIT = -455.0
MAX_LIMIT = 480.0
MAX_VELOCITY = 166.666666667
MAX_ACCELERATION = 650.0

[JOINT_0]
TYPE = LINEAR
MAX_VELOCITY = 166.666666667
MAX_ACCELERATION = 650.0
BACKLASH = 0.000
FERROR = 5.0
MIN_FERROR = 1.0
INPUT_SCALE = 1.0
OUTPUT_SCALE = -1.000
OUTPUT_OFFSET = 0.0
```

```
MAX_OUTPUT = 166.666666667
MIN_LIMIT = -455.0
MAX_LIMIT = 480.0
HOME = 0.000
HOME_ABSOLUTE_ENCODER = 2
DEADBAND = 0.000015
PGAIN = 30.0
IGAIN = 0.000
DGAIN = 0.000
FF0 = 0.000
FF1 = 1.000
FF2 = 0.0018
BIAS = 0.000
```

```
[AXIS_Y]
```

```
MIN_LIMIT = -315.0
MAX_LIMIT = 295.0
MAX_VELOCITY = 166.666666667
MAX_ACCELERATION = 650.0
```

```
[JOINT_1]
```

```
TYPE = LINEAR
MAX_VELOCITY = 166.666666667
MAX_ACCELERATION = 650.0
BACKLASH = 0.000
FERROR = 5.0
MIN_FERROR = 1.0
INPUT_SCALE = 1.0
OUTPUT_SCALE = -1.000
OUTPUT_OFFSET = 0.0
MAX_OUTPUT = 166.666666667
MIN_LIMIT = -315.0
MAX_LIMIT = 295.0
HOME = 0.000
HOME_SEARCH_VEL = -0.10
HOME_ABSOLUTE_ENCODER = 2
DEADBAND = 0.000015
PGAIN = 30.0
IGAIN = 0.000
DGAIN = 0.000
FF0 = 0.000
FF1 = 1.000
FF2 = 0.0018
BIAS = 0.000
```

```
[AXIS_Z]
```

```
MIN_LIMIT = -190.0
MAX_LIMIT = 200.0
MAX_VELOCITY = 166.666666667
MAX_ACCELERATION = 650.0
```

```
[JOINT_2]
TYPE = LINEAR
MAX_VELOCITY = 166.666666667
MAX_ACCELERATION = 650.0
BACKLASH = 0.000
FERROR = 5.0
MIN_FERROR = 1.0
INPUT_SCALE = 1.0
OUTPUT_SCALE = -1.000
OUTPUT_OFFSET = 0.0
MAX_OUTPUT = 166.666666667
MIN_LIMIT = -190.0
MAX_LIMIT = 200.0
HOME = 0.000
HOME_SEARCH_VEL = -0.10
HOME_ABSOLUTE_ENCODER = 2
DEADBAND = 0.000015
PGAIN = 30.0
IGAIN = 0.000
DGAIN = 0.000
FF0 = 0.000
FF1 = 1.000
FF2 = 0.0018
BIAS = 0.000
```

```
[SCALE]
MIN_SCALE = 0.0064102564
MAX_SCALE = 156
```

```
[AXIS_U]
MIN_LIMIT = -110.0
MAX_LIMIT = 110.0
MAX_VELOCITY = 780
MAX_ACCELERATION = 10000
PGAIN = 500
FF1 = 1.2
FF2 = 0.001
MAXOUTPUT = 780
```

```
[JOINT_3]
TYPE = LINEAR
MAX_VELOCITY = 780
MAX_ACCELERATION = 10000
MIN_LIMIT = -110.0
MAX_LIMIT = 110.0
HOME_SEQUENCE = 0
HOME_SEARCH_VEL = 0.1
HOME_LATCH_VEL = -0.1
```

```
HOME = 0
HOME_OFFSET = 0
HOME_IGNORE_LIMITS = YES
HOME_ABSOLUTE_ENCODER = 2
```

```
[AXIS_V]
MIN_LIMIT = -110.0
MAX_LIMIT = 110.0
MAX_VELOCITY = 780
MAX_ACCELERATION = 10000
PGAIN = 500
FF1 = 1.2
FF2 = 0.001
MAXOUTPUT = 780
```

```
[JOINT_4]
TYPE = LINEAR
MAX_VELOCITY = 780
MAX_ACCELERATION = 10000
MIN_LIMIT = -110.0
MAX_LIMIT = 110.0
HOME_SEQUENCE = 1
HOME_SEARCH_VEL = 0.1
HOME_LATCH_VEL = -0.1
HOME = 0
HOME_OFFSET = 0
HOME_IGNORE_LIMITS = YES
HOME_ABSOLUTE_ENCODER = 2
```

```
[HOSTMOT2]
DRIVER=hm2_pci
BOARD=5i22
CONFIG="firmware=hm2/5i22-1.5/SVST8_8.BIT num_encoders=3
      num_pwmgens=3 num_stepgens=0"
```

```
[ECAT]
IDX_NCTMK15 = 0.0
IDX_ESTUNJ0 = 0.1
IDX_ESTUNJ1 = 0.2
IDX_ESTUNJ2 = 0.3
```