



Универзитет „Св. Кирил и Методиј” - Скопје
Факултет за електротехника и информациски
технологии



РАЗВОЈ НА НАПРЕДНИ АЛГОРИТМИ ЗА УПРАВУВАЊЕ НА
ХИБРИДНИ СНС МАШИНИ И НИВНА ИМПЛЕМЕНТАЦИЈА
НА ВГРАДЛИВ СИСТЕМ СО ОГРАНИЧЕНИ РЕСУРСИ

-докторска дисертација-

Ментор

проф. д-р Јосиф Косев

Кандидат

м-р Никола Јовановски

Скопје, 2026

Кандидат: **м-р. Никола Борис Јовановски**
број на индекс: 3/2018

Ментор: **проф. д-р Јосиф Ќосев**
Универзитет „Св. Кирил и Методиј“ – Скопје
Факултет за електротехника и информациски технологии

Членови на комисија:

Претседател: **проф. д-р Катерина Ралева**
Универзитет „Св. Кирил и Методиј“ – Скопје
Факултет за електротехника и информациски технологии

Ментор: **проф. д-р Јосиф Ќосев**
Универзитет „Св. Кирил и Методиј“ – Скопје
Факултет за електротехника и информациски технологии

Член: **проф. д-р Зоран Ивановски**
Универзитет „Св. Кирил и Методиј“ – Скопје
Факултет за електротехника и информациски технологии

Член: **проф. д-р Глигорче Вртаноски**
Универзитет „Св. Кирил и Методиј“ – Скопје
Машински факултет

Член: **проф. д-р Бранислав Гераров**
Универзитет „Св. Кирил и Методиј“ – Скопје
Факултет за електротехника и информациски технологии

Лектор: **Јулијана Јосифовска**

Датум на одбрана: _____

Научна област: Електротехника и информациски технологии

Содржина

Листа на слики	5
Листа на табели	7
Листа на кратенки	8
Апстракт	9
Abstract	10
1 ВОВЕД.....	11
1.1 Преглед на достигнувањата во научната дисциплина поврзани со предметот на истражување	11
1.2 Предмет на истражувањето.....	17
1.3 Цели на истражувањето.....	17
1.4 Образложение на работните хипотези.....	18
2 Хардверска имплементација на предложената хибридна CNC-машина	19
2.1 Машински дизајн	19
2.2 Хардверска архитектура на предложената CNC-машина (JLABsmartCNC)....	21
2.3 Напојување, електрична шема и контролни прекинувачи на JLABsmartCNC	23
2.4 Погонски мотори на предложената хибридна CNC-машина	25
2.5 Глава за обработка (спиндла)	31
2.6 Екструдер и маса за печатење.....	33
2.7 Интерфејси човек-машина	36
2.8 Електронска управувачка единица (EVE)	37
3 Софтверска имплементација на предложената хибридна CNC-машина	44
3.1 Имплементација на фирмвер	44
3.1.1 GRBL слободен и отворен софтвер за прецизно управување на CNC-машини.....	45
3.1.2 Клипер напреден фирмвер наменет за контрола на 3D-печатари.....	66
3.1.3 Архитектура на фирмверот на предложениот управувачки систем	87
3.2 Имплементација на компјутерски софтвер за комуникација со предложената машина	102
4 Експериментална евалуација на предложениот управувачки фирмвер.....	106
4.1 Искористеност на РАМ-меморијата кај предложениот управувачки фирмвер.....	106

4.2	Анализа на алгоритмот за пресметка на временските интервали за чекорите во реално време	107
4.3	Експериментална валидација на временската точност на генерираните чекори преку споредба со пресметаните вредности и анализа на временските отстапувања	113
4.4	Практична валидација на предложениот систем преку експериментални обработки со различни алати изведени на самата машина	116
4.5	Перформанси на предложениот систем во конфигурација на 3D-печатење во хибридна и одвоена варијанта	118
5	Заклучок.....	119
5.1	Придонеси во областа на истражувањето	120
5.2	Насоки за идни истражувања.....	121
	Список на користена литература.....	123

Листа на слики

Слика 2.1. 3D-модел на JLABsmartCNC-хибридната машина.	19
Слика 2.2. Хардверска архитектура на предложената хибридна CNC-машина (JLABsmartCNC).	22
Слика 2.3. Контролна табла на JLABsmartCNC.	23
Слика 2.4. Електрична шема на поврзување на главните контролни прекинувачи на напојувањето со сите компоненти на машината.	24
Слика 2.5. Поврзување на податочните дигитални излези од EYE со влезовите на серводрајверот.	28
Слика 2.6. Поврзување на податочните дигитални излези од серводрајверот со влезовите на EYE.	28
Слика 2.7. Поврзување на влезни пинови за импулси од серводрајверот со EYE.	30
Слика 2.8. Пневматска кочница за Z-оската.	31
Слика 2.9. Фреквентен регулатор и мотор за спиндла.	32
Слика 2.10. MICRON3DP-екструдер со два различни краеви за топење на материјалот.	34
Слика 2.11. Драјвер за управување на чекорниот мотор на екструдерот.	35
Слика 2.12. 3D-модел на електронската управувачка единица (EYE).	38
Слика 2.13. Изглед на електронската плочка за напојување.	38
Слика 2.14. Поврзување на индуктивните сензори со EYE.	39
Слика 2.15. Коло за кондиционирање на температурните сензори.	40
Слика 2.16. Изглед на електронската плочка од второто ниво на EYE.	41
Слика 2.17. Шема на поврзување на EYE со драјверот на чекорниот мотор на екструдерот.	42
Слика 2.18. Изглед на електронската плочка на третото ниво од EYE.	42
Слика 2.19. Микроконтролерска плочка на четвртото ниво од EYE.	43
Слика 3.1. Кружна интерполација.	45
Слика 3.2. Разлика помеѓу функциите atan2 и arctan.	47
Слика 3.3. Зависност на бројот на сегменти од радиусот на лакот.	48
Слика 3.4. Одредување на бројот на сегменти.	49
Слика 3.5. Мемориска структура на баферот кој го користи GRBL-планерот.	52
Слика 3.6. Линеарна интерполација од точка P1 до точка P2.	53
Слика 3.7. Одредување на аголот помеѓу два сегменти на движење.	54
Слика 3.8. Одредување на максимална брзина при премин помеѓу два сегменти.	55
Слика 3.9. Алгоритам за пресметка на оптималните брзини на блоковите во баферот.	57
Слика 3.10. Алгоритам за одредување на профилот на брзина на секој од блоковите.	59
Слика 3.11. Примери за триаголен и трапезоиден профил на брзина.	60
Слика 3.12. Пример за сегментација на блок од планерот.	62
Слика 3.13. Пример за користење на Брезенхам алгоритмот за генерирање на линија од точка (0,0) до точка (50,25).	63
Слика 3.14. Блок-дијаграм на хардверот на Клипер системот.	67

Слика 3.15. Споредба помеѓу брзината на движење на главата на печатарот и посакуваната и реалната брзина на екструдирање на материјал.	68
Слика 3.16. Различни модели на брзини на екструдирање и компензација на притисокот на два сегменти со брзина на екструдирање поголема од нула во точката на спој.	69
Слика 3.17. Елементи на класата Move.	72
Слика 3.18. Измазнување на кратките патеки со триаголни профили на брзини со користење на виртуелно забрзување.....	74
Слика 3.19. Агол помеѓу два сегменти и дефинирање на центрипетална брзина.....	76
Слика 3.20. Алгоритам за пресметување на оптималните брзини по сегменти кај Клипер.....	78
Слика 3.21. Споредба на виртуелните брзини кои ги користи алгоритмот за пресметка на оптимални брзини на блоковите.	81
Слика 3.22. Примери за блокови само со забрзување проследени од триаголен блок (лев график) и мали кратки сегменти само со забавување (десен график).	82
Слика 3.23. Двојно поврзана листа во која дополнително се делат блоковите по сегменти со забрзување, забавување и константни брзини.	83
Слика 3.24. Алгоритам кој применува нумерички методи за пресметка на времетраењето на секој импулс/чекор на моторите.	85
Слика 3.25. Структура во која се чуваат пакуваните чекори за праќање кон микроконтролерот.....	86
Слика 3.26. Поедноставена верзија на управувачкиот алгоритам на предложениот фирмвер.....	88
Слика 3.27. Структура на планер баферот која се користи кај предложениот фирмвер.	90
Слика 3.28. Хиерархиска структура на мемориските бафери кај предложениот фирмвер.....	91
Слика 3.29. Структура на Intermediate-баферот кај предложениот фирмвер, со приказ на зачуваните кинематички параметри, податоците за бројот на чекори и променливите за компензација на нумеричките грешки.	92
Слика 3.30. Структура на StepISR баферот.....	95
Слика 3.31. Алгоритам кој се извршува во прекинската рутина на тајмерите за една оска (идентичен за сите оски).	96
Слика 3.32. Структура на Intermediate axis-баферите.	96
Слика 3.33. Структура и димензии на Step-баферите.	98
Слика 3.34. Хардверска мастер-слејв топологија на синхронизација помеѓу тајмерите кои ги управуваат моторите на оските на машината и екструдерот и блок-дијаграм на преносот на податоците од StepDMA-баферот кон баферираните регистри на тајмерите (shadow регистри).	101
Слика 3.35. Алгоритам кој се изведува во прекинската рутина при завршување на тековниот DMA-трансфер и иницирање на нов. За секоја оска се користи посебна прекинска рутина со истиот алгоритам имплементиран за истата.	102
Слика 3.36. Изглед на основното мени на компјутерскиот софтвер каде се избира дали машината ќе се користи за обработка или 3D-печатење.	103

Слика 3.37. Мени од компјутерскиот софтвер наменето за 3D-печатење.	103
Слика 3.38. Мени од компјутерскиот софтвер наменето за CNC-обработка.	104
Слика 3.39. Протокол на комуникацијата помеѓу компјутерскиот софтвер и машината.	105
Слика 4.1. Промена на брзината и поместувањето по сите оски во зависност од времето за првите 16 точки од G-кодот.	111
Слика 4.2. Промена на вкупната брзина (брзината со која се движи главата на машината) во зависност од времето.	111
Слика 4.3. Споредба помеѓу генерираниот пат од страна на предложениот алгоритам и идеалниот пат добиен со интерполација во 3D-просторот.	112
Слика 4.4. Позициона грешка на оските во зависност од времето.	112
Слика 4.5. Вкупна позициона грешка на главата на машината во зависност од времето.	113
Слика 4.6. Временско отстапување на периодот (jitter) во функција од времето за X-оската.	114
Слика 4.7. Временско отстапување на периодот (jitter) во функција од времето за Y-оската.	115
Слика 4.8. Хистограм на измерените временски отстапувања на периодот за X оската. Сите измерени вредности се наоѓаат во границите од $\pm 0.05 \mu s$, што одговара на интервалот на семплирање на осцилоскопот.	115
Слика 4.9. Изглед на изработената CNC-машина користена во сите експерименти.	116
Слика 4.10. Машинска CNC-обработка на основни геометриски форми на меко дрво со користење на двоперно глодало со дијаметар од 12 mm.	117
Слика 4.11. Машинска CNC-обработка во висококвалитетно суво дрво од орев со користење на троперно глодало со дијаметар од 4 mm.	117
Слика 4.12. Машинска CNC-обработка на прототип на мала електронска плочка со димензии $13 \times 16 \text{ mm}$ со користење на прецизен алат за гравирање.	118

Листа на табели

Табела 3.1. Координати на сегментите за дадениот пример од слика 3.1.	51
Табела 3.2. Пример за користење на Брезенхам алгоритмот за генерирање на дискретна линија од точка (0,0) до точка (7,4).	65
Табела 4.1. Големина на секој од баферите во RAM-меморијата.	106
Табела 4.2. Првите 16 координати добиени од G-кодот кои се користат за евалуација на алгоритмот за генерирање на временските интервали на чекорите.	110

Листа на кратенки

Кратенка	Значење
ABS	Acrylonitrile Butadiene Styrene
ADC	Analog-to-Digital Converter
ARM	Advanced RISC Machine
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CAN	Controller Area Network
CNC	Computer Numerical Control
CPU	Central Processing Unit
DAC	Digital-to-Analog Converter
DMA	Direct Memory Access
FDM	Fused Deposition Modeling
FFF	Fused Filament Fabrication
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
G-code	Geometric Code
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HMI	Human–Machine Interface
IC	Inter-Integrated Circuit
ISR	Interrupt Service Routine
MCU	Microcontroller Unit
NURBS	Non-Uniform Rational B-Splines
PCB	Printed Circuit Board
PID	Proportional–Integral–Derivative
PLA	Polylactic Acid
PWM	Pulse Width Modulation
RAM	Random Access Memory
RPM	Revolutions Per Minute
RTOS	Real-Time Operating System
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VFD	Variable Frequency Drive

Апстракт

Оваа докторска дисертација опфаќа развој, имплементација и експериментална верификација на напредна архитектура за управување на хибридни компјутерски нумерички управувани (CNC) системи, кои ги интегрираат адитивните и супстрактивните технологии на производство во една машина. Предложениот управувачки систем е целосно реализиран на хардверска платформа со ограничени ресурси и обезбедува детерминистичко и прецизно однесување во реално време, без употреба на дополнителни процесорски платформи или оперативни системи. Во рамките на оваа дисертација е дизајнирана и изработена комплетна хибридна CNC-машина која може да се користи и за CNC-обработка и за 3D-печатење. Покрај механичката конструкција, развиена и имплементирана е електронска управувачка единица базирана на еден микроконтролер со соодветни интерфејси за комуникација и управување со сите периферии на предложената машина.

Со цел формулирање на напреден и оптимизиран математички модел за управување со машината, направена е детална анализа и споредба на најчесто користените отворени и слободни фирмвери, наменети како за платформи базирани на микроконтролери, така и за архитектури базирани на сложени процесорски системи. Врз основа на идентификуваните предности и недостатоци, развиен е нов математички модел соодветен за имплементација на предложената хардверска платформа со ограничени ресурси. Предложениот модел се темели на хиерархиска организација на мемориски бафери која започнува со резултатите од кинематичките пресметки и завршува со временските интервали на поединечните чекори. Пресметаните временски параметри се пренесуваат директно во регистрите на тајмерите преку контролерот за директен мемориски пристап (DMA), со што се овозможува прецизно и синхронизирано генерирање на управувачките сигнали кон погонските мотори.

Експерименталната валидација на предложениот алгоритам вклучува симулациски анализи на реално генерираните чекори и практични тестови со CNC-обработка и 3D-печатење на реални модели. Добиените резултати потврдуваат висока геометричка точност од ранг на микрометри, стабилна работа при различни брзини и забрзувања, прецизна синхронизација помеѓу оските и глатка работа при високи динамички оптоварувања. Анализите покажуваат дека предложениот алгоритам успешно функционира на хардверски платформи со ограничени ресурси, обезбедувајќи детерминистичка работа во реално време со значително помала потребна процесорска моќ и големина на меморија во споредба со современите решенија. Предложената архитектура претставува достапна, флексибилна и скалабилна алтернатива на конвенционалните индустриски CNC-управувачки системи и поставува цврста основа за понатамошни истражувања во насока на адитивно производство со високи брзини и управување со сложени повеќеосни производствени системи.

Клучни зборови: Хибридна CNC-машина, адитивно и супстрактивно производство, управување во реално време, архитектура базирана на микроконтролер, кинематичко управување, DMA-контрола, детерминистички системи, 3D-печатење, CNC-обработка.

Abstract

This doctoral dissertation addresses the development, implementation, and experimental verification of an advanced control architecture for hybrid computer numerical control (CNC) systems that integrate additive and subtractive manufacturing technologies within a single machine. The proposed control system is fully realized on a hardware platform with limited computational resources and ensures deterministic and precise real-time behavior without the use of additional processor-based platforms or operating systems. Within the scope of this dissertation, a complete hybrid CNC machine capable of both CNC machining and 3D printing was designed and manufactured. Besides the mechanical design, an electronic control unit based on a single microcontroller was developed and implemented, incorporating appropriate communication interfaces and control mechanisms for all machine components.

Based on detailed analysis and comparison of widely used open-source firmware solutions, in both microcontroller-based platforms and complex processor systems architectures, a novel mathematical model suitable for implementation on the proposed resource-constrained hardware platform was developed. The proposed model is based on a hierarchical organization of memory buffers starting with the results of the kinematic computations, and ending with the time intervals of individual motion steps. A precise and synchronized generation of control signals for the drive motors is achieved via direct memory access (DMA) transfer of the computed timing parameters into the timer registers.

Experimental validation of the proposed algorithm is performed by simulation-based analysis of the actually generated motion steps as well as practical tests involving CNC machining and 3D printing of real models. The obtained results confirm high geometric accuracy at the micrometer level, stable operation across a wide range of speeds and accelerations, precise axis synchronization, and smooth performance under high dynamic loads. The analyses demonstrate that the proposed algorithm operates successfully on hardware platforms with limited memory and processing resources, providing deterministic real-time performance with an order of magnitude lower processing power than current solutions. The proposed architecture represents an affordable, flexible, and scalable alternative to conventional industrial CNC control systems and establishes a solid foundation for further research in high-speed additive manufacturing and the control of complex multi-axis manufacturing systems.

Keywords: Hybrid CNC machine, additive and subtractive manufacturing, real-time control, microcontroller-based architecture, kinematic control, DMA-based control, deterministic systems, 3D printing, CNC machining.

1 ВОВЕД

1.1 Преглед на достигнувањата во научната дисциплина поврзани со предметот на истражување

Денес живееме во светот на четвртата индустриска револуција (Industry 4.0). Таа се одликува со преклопување на границите меѓу физичкиот и дигиталниот свет со сè поголема автоматизација на машините, комуникација меѓу нив и нивно самоследење и дијагностика без интервенција на човек [1-4].

Двете основни технологии за производство на индустриски делови за секаква намена се:

- Супстрактивната технологија (со одземање на материјал од парче суровина)
- Адитивната технологија (со додавање на материјал во три димензии)

Во првата категорија спаѓаат CNC (компјутерски нумерички управувани) машини за обработка, а во втората категорија 3D-печатарите чие користење во последната деценија енорно се интензивира. Индустриските CNC-фрези се со висока прецизност, релативно мали брзини на обработка, а специфичниот дизајн и комплексните управувачки системи ја прават нивната цена многу висока, додека пак 3D-печатарите се со ниска цена, но со мала прецизност, полош квалитет и ниска брзина на изработка што во многу случаи не ги задоволува денешните потреби во индустријата. Иако постојат и други технологии (лиење, извлекување, пресување, дувана пластика и др.), најголем дел од индустриското производство се темели на овие две технологии, па истражувањата во оваа област за подобрување и забрзување на процесите на производство се интензивираат, особено кај 3D-печатарите со цел да се добие побрз процес на производство и повисок квалитет на изработените делови [5-6].

Кај секоја од различните технологии на адитивно производство кои се користат денес идентификувани се слабостите и предизвиците поради што сè повеќе се оди во насока на хибридни технологии со кои несовершеностите на адитивната технологија можат да се надминат преку комбинирање со супстрактивната технологија [7].

Постојат истражувања во областа на хибридно производство каде што во една машина се комбинираат двете основни технологии – супстрактивната и адитивната, но комерцијалната употреба на вакви машини сè уште е ограничена. Пример за комбинација на овие две технологии е производството на комплексни делови како што се 3D-печатените сензорски системи. За разлика од нив, конвенционалните хибридни системи се состојат од две одвоени CNC-машини – една за супстрактивно производство (CNC-глодалка) и една за адитивно производство (најчесто 3D-печатар) што има големи недостатоци. Интегрирањето на двете технологии во една CNC-машина дава многу повисока точност, го редуцира времето на производство и ја намалува цената на чинење [8,9]. Инкорпорирањето на адитивното производство на веќе постоечка CNC-машина претставува многу комплексен процес кој бара многу промени и во хардверот и во софтверот на самата машина.

Експерименти се правени во секвенцијалното хибридно производство каде во процесот на 3D-печатење на одредени секвенци на парчето се прави супстрактивна машинска дообработка со цел добивање на поквалитетно изработени делови. Секвенците на обработка зависат од големината на алатот и обликот на парчето. Друг тип на хибридно производство е со супстрактивна обработка на секој печатен слој од

делот кој што се изработува. Експериментите за овој тип на производство покажуваат десет пати подобри перформанси во однос на мазноста на сидовите на парчето. За овој хибриден начин на производство се користат посебни софтвери, како за раслојување (слајсирање) на самото парче, така и за генерирање на G-кодот за обработка и 3D-печатење [10].

Развојот на ваквите хибридни машини и нивното користење ги намалува трошоците и времето за производство, задржувајќи го квалитетот на изработените делови. Комбинација на метално печатење со ласер, како адитивна технологија, со традиционалната CNC-обработка, како супстрактивна технологија, овозможува висококвалитетни изработени делови. Оваа хибридна технологија е особено добра за производство на мали количини на делови од материјали кои што тешко се сечат, како што се легурите за воздухопловната индустрија и материјали со висока цврстина, како и компонентите за медицински помагала со висока прецизност. Истражувањата покажуваат дека хибридните машини ќе го револуционизираат производството на следната генерација [11].

Адитивното производство со користење на стандардни 3D-печатари со филамент (fused filament fabrication) нуди лесна и брза изработка на прототипови, флексибилност во дизајнот и економичност што ја прави оваа технологија поедноставна во споредба со традиционалното супстрактивно производство. Сепак, квалитетот и прецизноста на печатењето кај денешните комерцијално достапни 3D-печатари се ограничени, што ја наметнува потребата од истражување на факторите кои влијаат врз квалитетот на процесот на печатење. Експериментални тестови, како што се тестовите за утврдување на девијациите при кружни движења, оперативна анализа на вибрациите и точноста на повторливо позиционирање откриваат дека сервомоторите ги надминуваат чекорните мотори во однос на точноста на позиционирањето и редуцирањето на вибрациите. Дополнително на тоа резултатите покажуваат дека изборот на моторите има ограничено влијание врз квалитетот на печатењето, додека вибрациите предизвикани од движењето значително влијаат врз квалитетот на испечатените делови [12].

Целиот процес на 3D-печатење и CNC-обработка се сведува на интерполации помеѓу зададени точки преку G-код добиен од CAD/CAM-софтверите за дизајн на 3D-модел. Линеарните интерполации се лесни за имплементација и го забрзуваат времето на печатење кога се работи за движења по прави линии, но можат да предизвикаат грешки при кружни движења и десинхронизација помеѓу брзината на движење на оските и брзината на екструдирање на материјал. Кај фирмверите кои што се слободни и отворени и се користат за оваа намена, најчесто се имплементирани линеарни, кружни и Безиерови (Bezier) интерполации. Аналитички и експериментални методи се користени за да се споредат овие методи на интерполација базирани на нивните профили на брзини [13]. Резултатите од овие анализи и експерименти имаат за цел да ги оптимизираат стратегиите на пресметување на патеките на екструдерите и да се подобри ефикасноста на адитивното производство [14].

Технологијата на 3D-печатење се користи и за производство на прототипови на електронски печатени кола. Дефектите во 3D-печатењето кај овие кола може да резултираат со неисправна работа на истите. Методи базирани на невронски мрежи се користат за надминување на овие проблеми. Камери со висока резолуција снимаат слики за секој вод за време на печатењето, а конволуциските невронски мрежи ги сегментираат водовите со цел детекција на грешки. Тренираните невронски мрежи ги идентификуваат прекините во водовите и генерираат патеки за корекција на истите, кои печатарот ги

извршува. Експериментите покажуваат успешна автоматска детекција на сите намерно предизвикани дефекти, што ветува зголемување на надежноста кај 3D-печатарите наменети за печатени електронски кола. Сепак, за посложени електронски кола потребно е проширување на ова истражување [15].

Проценката на резонантната фреквенција на декартовите 3D-печатари е од клучно значење за оптимален дизајн и перформанси. Во [16] се воведува метод базиран на Ојлер-Лагранж формализмот за проценка на резонантните фреквенции земајќи ги во предвид различните можни дизајни во рамките на декартовата кинематика. Методот помага во изборот на оптимални позиции на оските и линеарните погонски механизми за време на раната фаза на дизајн. Споредбата со методот на конечни елементи и експерименталната валидација на предложениот метод резултираат со висока точност и надежност на предложениот пристап. Методот е особено корисен за развој на прелиминарни дизајни и поддршка на новите технологии за 3D-печатење.

Во поново време 3D-печатењето се користи и во металната индустрија. Печатењето на метал е многу посложено во споредба со печатењето на стандардни пластични материјали (PLA, ABS, PET...). За разлика од стандардните печатари кои најчесто користат готови едноставни дизајни и управувачки единици, печатарите за метал бараат специфичен машински дизајн и управувачка единица со посебни прилагодувања особено за екструдирањето на метал кое е многу посложено и бара многу повисоки температури и прилагодување на брзините [17-21]. Начините на интерполација и управувањето со движењата на главата на машината е сепак слично како кај стандардните 3D-печатари.

Развојот и напредокот на CNC-интерполаторите одигра главна улога во последните 6-7 децении за успешна имплементација на компјутерските нумерички управувачи. Иницијално, интерполацијата се темелеше на едноставна поделба на кривите на мали апроксимативни линеарни сегменти. Сепак, модерните интерполатори еволуираа од комплексни хардверски решенија во моќни и флексибилни софтверски решенија, овозможувајќи лесна обработка на сложени форми и контури. Напредните параметарски криви како што се Cubic splines, B-splines, Bezier кривите и неуниформните рационални B-splines (NURBS) ги заменија конвенционалните техники на интерполација, обезбедувајќи зголемена точност на контурите, прецизност и глаткост на движењето [22]. Во некои истражувања за пресметка на интерполацијата на патеката на алатот се користат и брзите Фуриеви трансформации (FFT) [23].

При високи брзини на обработка динамичките грешки кај CNC-машините често се поголеми во однос на квази-статичките грешки и значително влијаат врз квалитетот на обработената површина. Динамичките грешки се дефинираат како отстапување на вистинското поместување на главата на CNC-машината во однос на референтното поместување за време на движењето. Овој тип на грешки се категоризирани во две групи: оние кои што се јавуваат во самата повратна врска на сервомоторите вклучително со нивните драјвери и грешки кои се јавуваат од управувачката единица односно системот за генерирање сигнали за управување на сервомоторите. За редуцирање на динамичките грешки се користат различни стратегии за секоја оска поединечно како и за комбинираниот систем заради генерирање на координираната траекторија. Фокусот на истражувањата во ова поле е ставен на управувачката единица која ги генерира сигналите за сервомоторите и ја изведува комплетната интерполација по сите оски адаптирајќи ја согласно динамиката на системот за да постигне редуцирање на грешките при високи брзини на обработка [24].

Во [25-26] се предложени алгоритми за измазнување на аглите кај линеарни траектории базирани на B-spline криви, со цел обезбедување континуирани транзиции и контролирана геометриска девијација. Во [25] се користат B-splines од петти ред за спојување на соседни линеарни сегменти со оптимална закривеност, при што се контролираат грешките кои се јавуваат во зоните на аглите, а на корисникот му се овозможува дефинирање на дозволената толеранција. Дополнително, во [26] е предложена шема за B-spline апроксимација со ограничена грешка, базирана на доминантни точки, наменета за CNC-интерполација на микролинеарни патеки на алатот, со што се постигнува подобра рамнотежа помеѓу точноста и пресметковната ефикасност. Генерирањето на сигналите за синхронно управување со моторите при поминување низ зоните на аглите се реализира со примена на S-профил на забрзување, оптимизиран за минимален временски циклус, како и интерполација во реално време за обезбедување глатко и прецизно движење на алатот. Имплементација на B-spline метод за интерполација на вградлив систем базиран на микроконтролер со ограничени ресурси е прикажана во [27]. Понатамошен пристап за измазнување на аглите, кој ги комбинира позицијата, тангентата, закривеноста и остријата на соседните сегменти, е реализиран со користење на 3D-clothoid splines (базирани на Ојлерова спирала), при што се постигнува G^3 -континуитет [28].

Врз основа на овие пристапи засновани на B-spline криви, понатамошниот развој на интерполациските алгоритми кај CNC-системите води кон примена на NURBS-криви (Non-Uniform Rational B-Splines), кои овозможуваат поголема флексибилност, подобра контрола на закривеноста и зголемена прецизност при сложени траектории. NURBS-интерполациските криви овозможуваат континуирана закривеност и висока геометриска точност, со што ги надминуваат ограничувањата на класичните линеарни и кружни интерполациски функции. Дополнително, ефикасноста на интерполацијата може значително да се зголеми со извршување на дел од пресметките пред самиот интерполациски процес, со што се редуцира пресметковното оптоварување во реално време [29,30]. Примената на NURBS-интерполација е исто така проширена и кај функциите за компензација на радиусот на алатот, каде што се постигнува зголемена прецизност и стабилност на обработката [31], како и кај системи за генерирање на референтни импулси со константна брзина по должина на патеката на алатот [32].

Сепак, закривеноста на NURBS-кривите варира по должината на траекторијата, што ги прави традиционалните методи за планирање на брзината, засновани на класични модели на забрзување и забавување, тешки за директна примена кај CNC-обработката. Во [33] е предложен метод за планирање на брзините на движење на главата на CNC-машината базиран на концептот на критична ограничувачка крива (Critical Constraint Curve – CCC). Овој пристап ги интегрира ограничувањата на брзината, кинематичките параметри и динамичките услови за време на процесите на забрзување и забавување, обезбедувајќи концизен и математички конзистентен модел. Експерименталните резултати покажуваат дека предложениот метод го редуцира времето на забрзување и забавување за приближно 40%, при што се задржуваат реално-временските перформанси, стабилноста и квалитетот на обработката, што го прави овој пристап ефикасна стратегија за планирање на брзините кај NURBS-базираната CNC-обработка.

Кај CNC-машините со пет оски, наменети за обработка на сложени форми и површини, минимизирањето на геометриските грешки и задржувањето на континуираноста од висок ред се круцијални за постигнување висок квалитет на обработената површина. Во [34] е предложен метод за генерирање на траекторијата на алатот базиран на компактна репрезентација на кривите и оптимизација на контролните

точки со цел подобрување на квалитетот на обработката. Притоа, се воведува прецизна шема за пребарување на оптималните контролни точки за секоја крива, при што се применува механизам за поделба на кривите кој резултира со прогресивно зголемување на бројот на контролни точки за време на процесот на оптимизација, овозможувајќи зголемена пресметковна ефикасност и подобра мазност на обработената површина.

Покрај тоа, кај CNC-машините со пет оски се применуваат и методи засновани на двојни Безиереви криви, кои не се ограничени исклучиво на вакви машини, туку можат да се користат и кај системи со помал број на оски. Овие алгоритми обезбедуваат континуираност на сегментите на патеката, како и на нивните тангенти, со што значително се подобрува глаткоста на траекторијата, конзистентноста на брзината и точноста на движењето [35]. Слично на тоа, во [36] е предложен метод за измазнување на траекторијата со G^3 -континуитет кај пет-осни CNC-машини, со што дополнително се редуцираат динамичките нарушувања и грешките при обработка. Контролата на контурните грешки кај повеќеосните CNC-системи дополнително се унапредува преку итеративни алгоритми кои ја користат закривеноста на траекторијата како водечки параметар за корекција, како што е прикажано во [37], каде што се постигнува подобра точност на следење на траекторијата и стабилност на процесот на обработка.

Вибрациите предизвикани од наглите забрзувања при обработка со големи брзини, исто така влијаат врз квалитетот на обработените делови. Се користат различни шеми за минимизирање на овие вибрации. Авторите на [38] воведуваат филтер со конечен импулсен одзив (FIR Finite impulse response) дизајниран да генерира референтни профили на забрзувањето кои ги ублажуваат несаканите вибрации во широк спектар на фреквенции. Ваквиот филтер овозможува реално-временска интерполација за линеарните сегменти и континуираните траектории, додека распоредувањето на брзините овозможува грешките на контурите да останат во границите на зададената толеранција. Експерименталните тестови ја покажуваат ефикасноста на овој метод во редуцирањето на инерцијалните вибрации и постигнувањето висок квалитет на изработените површини без да се зголеми времето на изработка.

Друго решение за намалување на вибрациите е дадено во [39] каде што за разлика од традиционалните методи со симетрично измазнување на аглите на споевите, се предлага метод кој генерира асиметрична траекторија за измазнување на аглите која овозможува полесно забрзување и забавување на споевите на сегментите. Овој метод обезбедува аналитичко решение во еден чекор и конзистентност на патеките на алатот во двете насоки. Експерименталните резултати ја потврдуваат ефективноста на методот за подобрување на точноста на контурите и намалување на времето на обработка.

Кај модерните CNC-системи се прават и истражувања во сферата на точното предвидување на брзините на движење по оските во тек на обработката, како и на временските циклуси, со моделирање на генерирањето на траекторијата. За разлика од традиционалните системи кои симулираат машинско движење базирано само на зададената брзина и геометријата на патеката, овој пристап го вклучува и системот за генерирање сигнали за управување на моторите заедно со интерполациските стратегии инкорпорирани во него [40].

Постојат и методологии за редуцирање на времето на производство со оптимизирање на G-код командните секвенци. Овие методологии се наменети за компјутерските програми кои го генерираат G-кодот пред да биде испратен до управувачкиот систем на машината [41]. Со развојот на индустрија 4.0 машините и

уредите сè повеќе се поврзуваат со сервер преку различни комуникациски стандарди и протоколи. Управувачките системи на CNC-машините обезбедуваат многу податоци кои се однесуваат на процесот на работа на истите, па се прават експерименти и истражувања во насока на стандардизација и поврзување и на старите управувачки системи со централни сервери [42].

Интеграцијата на роботиката сè повеќе напредува, така што во последно време роботски раце се користат како платформи за обработка. Предноста кај роботската обработка е што овозможува сложени форми на обработка, голема флексибилност и економичност. Слично како кај CNC-машините и роботските платформи се управуваат со нумерички управувачи (NC) со помош на G-код, дефиниран со меѓународниот стандард ISO 6983 [43]. Предмет на истражување кај роботските платформи се методи со кои ќе се избегнат сложените пресметки на сферичните интерполации во реално време, потребни за правилно позиционирање на алатот. Модифицираниот метод на Haversine ги елиминира нелинеарните интерполациски грешки кои се јавуваат при линеарна интерполација на позицијата кај ротационите оски [44,45].

Покрај развојот на алгоритмите за интерполација и планирање на траекторијата, во последните години значително внимание се посветува и на архитектурата на управувачките системи кои ги реализираат овие алгоритми во реално време. Современите CNC-машини и системите за адитивно производство поставуваат строги барања во однос на временската точност на генерираните управувачки сигнали, синхронизацијата на движењата по повеќе оски и детерминистичкото извршување на алгоритмите за управување. Кај системите базирани на чекорни или сервомотори, дури и мали временски отстапувања при генерирањето на управувачките импулси можат да доведат до акумулација на позициони грешки, намалување на квалитетот на обработката и појава на несакани вибрации. Поради тоа, проблемот на детерминистичко извршување на алгоритмите за управување претставува едно од клучните истражувачки прашања кај современите вградливи системи за управување во реално време [46-48].

Во литературата предложени се бројни архитектури насочени кон подобрување на временската точност и намалување на процесорското оптоварување. Една група истражувања се базира на употреба на FPGA-платформи, при што временски критичните задачи се реализираат хардверски со цел да се обезбеди висока паралелност и детерминистичко однесување на системот [49-56]. Иако ваквите решенија овозможуваат високи перформанси и висока временска предвидливост, нивната примена обично е поврзана со поголема комплексност на хардверската и софтверската архитектура, како и со повисоки барања во однос на развојот, интеграцијата и одржувањето на системот.

Алтернативен пристап претставуваат архитектурите базирани на микроконтролерски системи кои користат хардверски периферии и оптимизирани софтверски механизми за намалување на оптоварувањето на јадрото на микроконтролерот. Во оваа насока се предложени решенија за планирање на траекторијата имплементирани на вградливи системи, употреба на директен мемориски пристап (DMA), хардверско генерирање на управувачки импулси и реализација на управување со повеќе оски во реално време [57-59]. Овие истражувања укажуваат дека со соодветна организација на мемориските структури и искористување на хардверските ресурси може значително да се подобри временската точност и да се намали влијанието на доцнењата поврзани со извршувањето на софтверот.

Покрај тоа, бројни истражувања се насочени кон развој на вградливи CNC-архитектури базирани на оперативни системи во реално време и процесорски платформи со повеќе јадра [60–63]. Дополнително, современите истражувања ја разгледуваат примената на методи за оптимизација и алгоритми за управување со цел подобрување на точноста на движењето, намалување на грешките при интерполацијата и поефикасна реализација на сложени траектории [64,65].

Меѓу најраспространетите отворени системи за управување кои денес се применуваат кај CNC-машините и системите за адитивно производство се: GRBL, grblHAL, Klipper и LinuxCNC [66–69]. Иако овие системи овозможуваат високи перформанси и наоѓаат широка примена во пракса, најголемиот дел од нив се базираат на сегментација на траекторијата, прекински механизми со висока фреквенција или употреба на дополнителни процесорски платформи за реализација на управувањето. Оттука, се наметнува потребата од развој на архитектури кои ќе овозможат реализација на напредни алгоритми за синхронно управување на движењата по повеќе оски и директна пресметка на временските интервали помеѓу последователните чекори, без употреба на дополнителни процесорски системи или FPGA базирани платформи.

1.2 Предмет на истражувањето

Предмет на истражување на оваа докторска дисертација е развој на напредни управувачки алгоритми наменети за CNC-машини и 3D-печатари и нивна имплементација на вградлив систем со ограничени ресурси и ниска цена на чинење. Експериментите и евалуацијата на развиените алгоритми ќе се изведуваат на хибридна машина која ги комбинира двете основни технологии на производство во една. Целта на овој систем е да се забрза процесот на изработка на 3D-делови (модели) во индустријата и да се подобри квалитетот на истите.

За таа цел, првично ќе биде дизајнирана комплетна машина која ќе ги комбинира двете основни технологии на производство: адитивното и супстрактивното. Детални анализи ќе бидат направени за потребните сензори, пневматски компоненти и актуатори заедно со драјвери за истите. Следно ќе се имплементира интеграција на сите електрични компоненти во машинскиот дизајн. За соодветно дизајнираната машина ќе се развие вградена електронска управувачка единица со соодветни периферии наменети за управување со сите компоненти на машината. Понатаму, ќе се развие фирмвер за сите периферии на управувачката единица со цел тестирање на динамиката на целиот систем.

Поставувањето на математички модел на алгоритмите кои ќе се користат ќе биде клучно за ефикасна интерполација која ќе овозможи брзо и прецизно движење на главата на хибридната CNC-машина. Утврдувањето на точноста на предложениот модел ќе се изведе со софтверска симулација и анализи на добиените резултати. Следно ќе биде изведена имплементација на алгоритмите во управувачката единица на машината.

Дополнително ќе бидат изведени експериментални тестирања со изработка на пробни 3D-делови (модели) со различни брзини. За изработените модели ќе се извршат анализи на квалитетот. На крај ќе се направи споредба со комерцијалните системи во однос на вибрациите, брзината, прецизноста и квалитетот на изработката.

1.3 Цели на истражувањето

Основната цел на докторската дисертација е да придонесе кон Индустрија 4.0 во развојот и унапредувањето на процесите на производство со CNC-машини и 3D-печатари како и во напорите за нивна интеграција во една машина. Акцентот посебно ќе

биде ставен онаму каде што тековните истражувања најмалку се насочуваат, а тоа е напредните алгоритми за генерирање на импулсите кон моторите од чија прецизност најмногу зависи квалитетот на изработените делови. Имплементацијата на развиените алгоритми на платформи со ограничени ресурси и ниска цена на чинење ќе придонесе за поголема достапност на овие технологии. Во насока на овие цели дефинирани се различни задачи што треба да се извршат:

- Систематски преглед на релевантната литература од областа на CNC-машините, 3D-печатарите и хибридните машини;
- Компаративна анализа на постоечки машини, идентификација на предности и недостатоци;
- Компаративна анализа на математички модели и алгоритми кои се користат за управување на овие машини;
- Дизајн и имплементација на хардверско решение на машина која ќе биде соодветна за хибридно производство;
- Поставување на математички модел за алгоритмите наменети за хибридната машина;
- Поставување на модели за брзините и забрзувањата при процесот на интерполација;
- Симулација на предложените модели;
- Имплементација на предложените алгоритми во фирмверот;
- Анализа и приказ на резултатите.

1.4 Образложение на работните хипотези

Во оваа докторска дисертација главната хипотеза е дека со предложените алгоритми за управување, имплементирани исклучиво на микроконтролерска платформа со ограничени пресметковни ресурси, е возможно да се постигне реално-временско управување, точност и квалитет на CNC-обработка и 3D-печатење споредливи со комерцијалните CNC и 3D-системи базирани на процесорски платформи. Дополнително, предложениот пристап ќе овозможи зголемување на брзината на обработка и печатење без компромис на квалитетот на изработените делови. Оваа хипотеза се базира на неколку посебни хипотези дадени во продолжение:

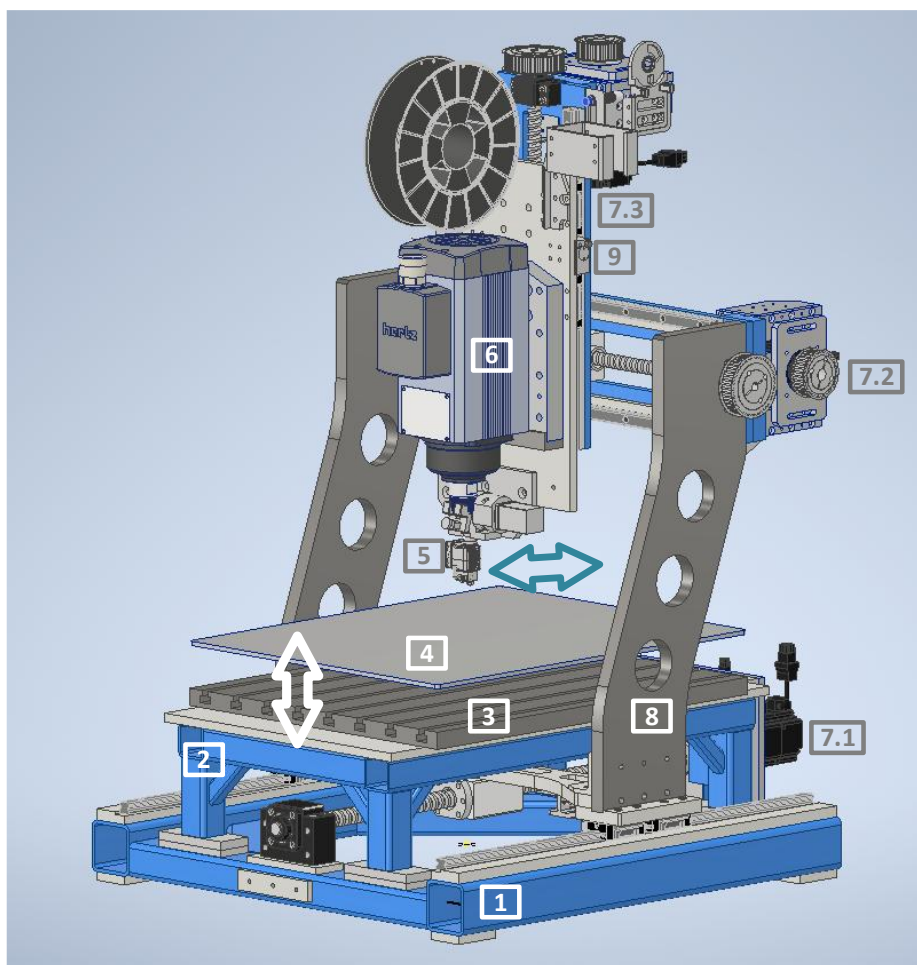
- **Хипотеза 1:** Сите неопходни пресметки за планирање на траекторија, интерполација и управување со движењата можат да се извршуваат во реално време исклучиво на микроконтролер, без потреба од надворешна процесорска платформа.
- **Хипотеза 2:** Предложениот микроконтролерски систем ќе обезбеди динамички и позициони перформанси споредливи со оние кај комерцијалните CNC-системи и 3D-печатари.
- **Хипотеза 3:** Предложените алгоритми ќе овозможат зголемување на брзината на CNC-обработка и 3D-печатење без нарушување на стабилноста, точноста и квалитетот на изработените делови.
- **Хипотеза 4:** Примената на предложените алгоритми ќе резултира со зголемен геометриски и површински квалитет на CNC-обработените и 3D-печатените делови без намалување на продуктивноста.
- **Хипотеза 5:** Предложените алгоритми се хардверски независни и можат да се применуваат кај CNC-машини и 3D-печатари со различни механички конструкции и типови на погонски системи (серво и чекорни мотори) со минимални прилагодувања.

2 Хардверска имплементација на предложената хибридна CNC-машина

2.1 Машински дизајн

Механичката конструкција на хибридната CNC-машина има фиксна работна маса и подвижна алатна глава која се движи во тридимензионалниот простор. 3D-моделот на машината е прикажан на слика 2.1. Материјалот од кој е изработена машината е конструктивен челик (0361) погоден и најчесто користен материјал за ваков тип на конструкции. Основата на конструкцијата се состои од неподвижна носечка рамка (1) и подлогата за масата (2), кои се клучни компоненти што обезбедуваат стабилност и цврстина на целата машина. Масата за обработка со Т-жлебови (3) е фиксирана на подлогата со завртки и овозможува сигурно држење на суровиот материјал за време на машинската обработка.

Подвижниот дел на конструкцијата (8) е поврзан со статичката основа преку лизгачки елементи (водилки), кои овозможуваат прецизно линеарно движење. Преносот на вртежниот момент од моторот за X-оската (7.1) до навојното вретено е реализиран со помош на еластична спојка (spider). Кај оските Y (7.2) и Z (7.3) преносот на движењето се изведува со синхрон ремен и соодветни ременици со преносен однос 1:1, што овозможува директен и стабилен пренос на вртежниот момент.



Слика 2.1. 3D-модел на LABsmartCNC-хибридната машина.

Преносот со синхрон ремен овозможува флексибилност во дизајнот, преку лесна промена на димензиите на ремениците и можност за експериментирање со различни преносни односи. Поради овие предности, во следната верзија на конструкцијата се планира и преносот за X-оската да се изведе со ременски систем, со цел да се изедначи и поедностави преносниот механизам за сите оски. Таквата конфигурација ќе овозможи и спроведување на експерименти со различни преносни односи, со цел оптимизација на динамичките карактеристики на системот. Навојните вретена служат за трансформација на кружното движење од моторите во линиско по оските на CNC-машината. Составени се од осовина и топчиња во внатрешниот дел (слично како кај лежиштата со топчиња) со што се овозможува прецизен пренос на движењето од моторот без никаков зјај.

Во тековната верзија на машината, хибридно производство се реализира на секвенцијален начин – најпрво се врши 3D-печатење на делот, по што следи машинска обработка со цел постигнување на бараните геометриски форми и точности. Во идните верзии се предвидува интеграција на адитивните и супстрактивните процеси во единствен производствен циклус, со можност за нивно паралелно извршување. Со тоа што машината и понатаму ќе остане флексибилна, овозможувајќи да се користи само како 3D-печатар или само како CNC-фреза, во зависност од производствените потреби.

Двојната функција на машината бара едноставна и брза промена на главата и работната маса, во зависност од потребниот производствен процес – CNC-обработка или 3D-печатење. Главната спиндла (6), наменета за глодање како суптрактивен процес, е трајно фиксирана на машината поради нејзината масивност и тежина. Наспроти тоа, екструдерот (5), кој се користи за 3D-печатење, претставува модуларен елемент што се прицврстува со две завртки и е поврзан преку единствен конектор што ги обединува сигналните линии и напојувањето. Поради различната природа на двете технологии, машината користи и различни типови на работни маси. За машинската обработка се применува масивната работна маса со T-жлебови (3), цврсто фиксирана на подлогата на машината, која овозможува стабилно прицврстување на суровиот материјал. За 3D-печатење, се користи лесна алуминиумска маса со интегриран грејач (4), која се поставува врз масата со T-жлебови со помош на магнети и се поврзува преку посебен конектор.

При премин во режим на 3D-печатење, потребно е да се постави масата за 3D-печатење, која се прицврстува исклучиво со магнети и се поврзува со првиот конектор, како и да се монтира екструдерот, кој се прицврстува со две завртки и се поврзува преку втор конектор. На тој начин, со само две механички точки на прицврстување и два електрични конектори, се овозможува брза и практична транзиција кон 3D-печатење, без потреба од дополнителен алат или сложени процедури. За машинска обработка соодветно само се вадат масата и екструдер.

Номиналните механички димензии на движење по должините на трите оски изнесуваат: X – 400 mm, Y – 325 mm и Z – 300 mm. Сепак, овие вредности не претставуваат целосно ефективни димензии, бидејќи реалниот опсег на обработка е намален поради присуството на крајни сензори (лимит прекинувачи), како и поради физичките димензии и позиционирањето на работната маса, масата за печатење, спиндлата и екструдерот. Овие елементи ја ограничуваат ефективната работна површина и висина, што мора да се земе во предвид при планирање на производниот процес. За поголема сигурност за секоја оска се користат дупли крајни прекинувачи за кои се изработени посебни држачи прицврстени на самата конструкција.

Поради големата сопствена маса на спиндлата, во случај кога моторот на оската Z е исклучен, може да дојде до неконтролирано спуштање на истата, што може да предизвика механичко оштетување на навојното вретено на долниот краен дел. Овој проблем често се јавува и кај комерцијалните CNC-машини, па за да се надмине кај JLABsmartCNC е имплементирана пневматска кочница (означена со број 9 на слика 2.1). Кочницата работи комплементарно со моторот на оската Z, односно автоматски се активира кога моторот е исклучен со што се спречува неконтролирано паѓање на истата и автоматски се деактивира при вклучување на моторот со што се овозможува непречено и безбедно движење на оската без механички отпор.

2.2 Хардверска архитектура на предложената CNC-машина (JLABsmartCNC)

Главната компонента која управува со целата машина е електронската управувачка единица (ЕУЕ) сместена во центарот на хардверската архитектура на машината дадена на слика 2.2. Како погонски мотори за оските на машината се користат 3 идентични сервомотори со моќност од по 400 W со соодветни хардверски драјвери. Развиени се посебни интерфејси за комуникација помеѓу електронската управувачка единица и драјверите на сервомоторите. За управување на пневматската кочница за Z-оската се користи соодветен електромагнетен вентил за пневматски цилиндри кој е управуван преку соодветен релеен излез од ЕУЕ. На краевите од секоја оска се поставени дупли крајни прекинувачи со цел да се обезбеди сигурна заштита од механички удар и оштетување на машината при достигнување на крајните позиции. Први се позиционирани индуктивни сензори, кои се поврзани со електронската управувачка единица (ЕУЕ) и софтверски се читаат. Тие служат и за автоматска калибрација и одредување на почетната референтна позиција на секоја оска. Како дополнителна заштита, веднаш по нив се поставени механички микропрекинувачи кои, при активација, хардверски го прекинуваат напојувањето на моторите, спречувајќи удар во крајните позиции на навојните вретена. Овој двостепен систем на заштита придонесува за повисоко ниво на оперативна безбедност и сигурност, како за време на реалната работа на машината, така и во фазата на развој и тестирање на фирмверот.

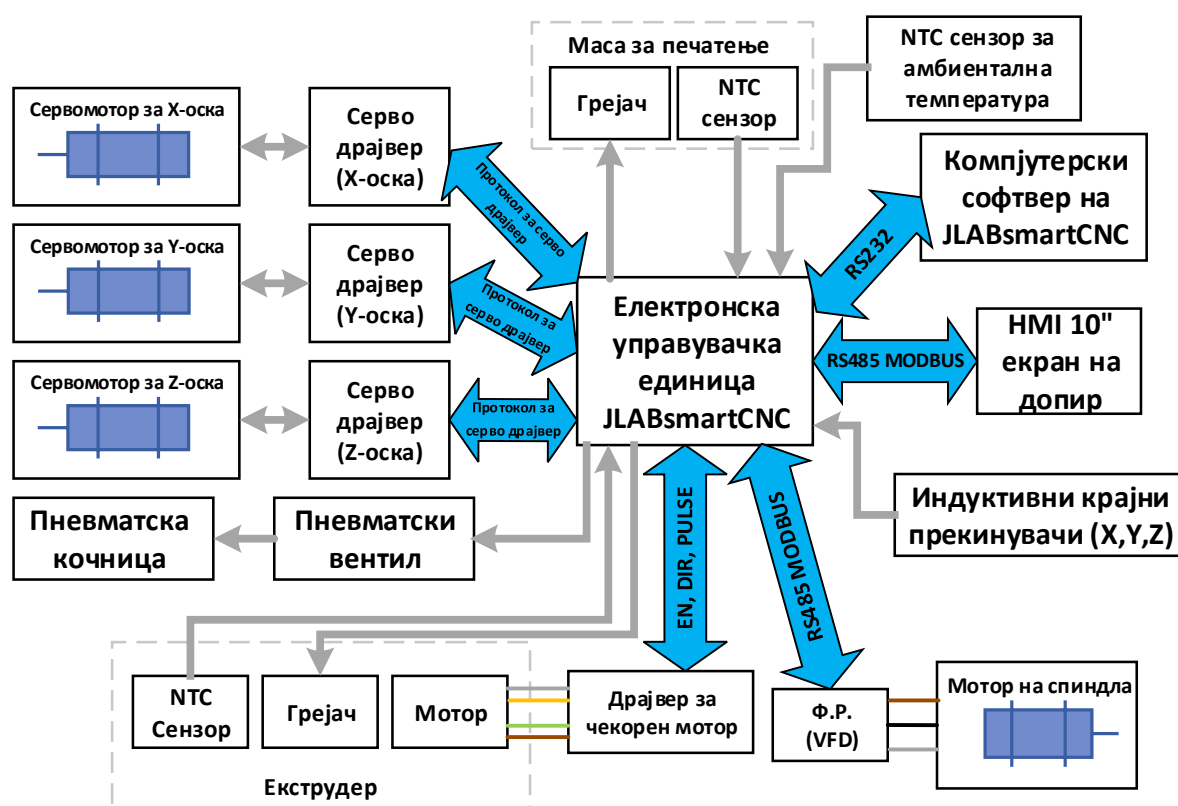
Екструдерот е составен од три основни компоненти: мотор за повлекување на материјалот, грејач за негово топење и температурен сензор што овозможува повратна информација за прецизна регулација на температурата. За придвижување на материјалот се користи чекорен мотор со вграден механички редуктор, со што се обезбедува висока резолуција и стабилност при процесот на екструдирање. Управувањето со моторот се реализира преку соодветен хардверски драјвер. Температурата се мери со аналоген NTC-сензор, чиј сигнал се обработува преку посебно кондиционирано коло интегрирано во електронската управувачка единица (ЕУЕ). Прецизното управување на грејачот е од клучно значење за одржување на константна температура, што директно влијае врз квалитетот на 3D-печатените делови. Поради тоа, за управување со греењето се користи посебен PWM (импулсно-ширински модулиран) излез со интегриран хардверски драјвер во ЕУЕ, кој овозможува фино регулирање на интензитетот на греење во согласност со повратната информација од температурниот сензор.

Во процесот на 3D-печатење, неопходно е и масата за печатење да се загрее на соодветна температура, која зависи од типот на материјалот што се користи. Работната површина на масата изнесува $500 \times 400 \text{ mm}$, па затоа е инсталиран грејач со моќност од 800 W за да се обезбеди рамномерно и ефикасно загревање. Управувањето со грејачот се реализира преку специјален контролен излез од електронската управувачка единица (ЕУЕ), додека за повратна информација за температурата се користи NTC температурен

сензор, како и кај екструдерот. Прецизното управување со температурата на масата за печатење е клучно за квалитетна адхезија на првиот слој при печатењето.

Спиндлата, односно моторот за машинска обработка, претставува трифазен електромотор со моќност од 4 kW, кој се управува преку соодветен фреквентен регулатор (VFD – Variable Frequency Drive). Комуникацијата помеѓу електронската управувачка единица (EUE) и регулаторот е реализирана преку RS485 сервиска линија, користејќи го индустрискиот MODBUS-протокол [70], што овозможува сигурно и флексибилно управување со брзината и статусот на моторот.

Користењето и управувањето со машината се реализира преку 10-инчен екран на допир (HMI), кој располага со интуитивен и едноставен кориснички интерфејс. Покрај основните функции, преку HMI се овозможува и напредно мануелно управување со машината, што е особено корисно при рачни интервенции или фини дообработки на одредени делови. Комуникацијата помеѓу HMI и електронската управувачка единица (EUE) е воспоставена преку посебна RS485 сервиска линија, независна од онаа што се користи за фреквентниот регулатор, при што е имплементиран индустрискиот MODBUS-протокол. Дополнително, развиен е наменски компјутерски софтвер преку кој машината може да се стартува, контролира и надгледува директно од компјутер. Комуникацијата помеѓу компјутерскиот софтвер и EUE е реализирана преку RS232 сервиска врска, која овозможува стабилен пренос на податоци и сигурна двонасочна комуникација. За оваа комуникација е развиен и наменски протокол што обезбедува сигурна и ефикасна размена на податоци помеѓу машината и компјутерскиот софтвер.



Слика 2.2. Хардверска архитектура на предложената хибридна CNC-машина (JLABsmartCNC).

2.3 Напојување, електрична шема и контролни прекинувачи на JLABsmartCNC

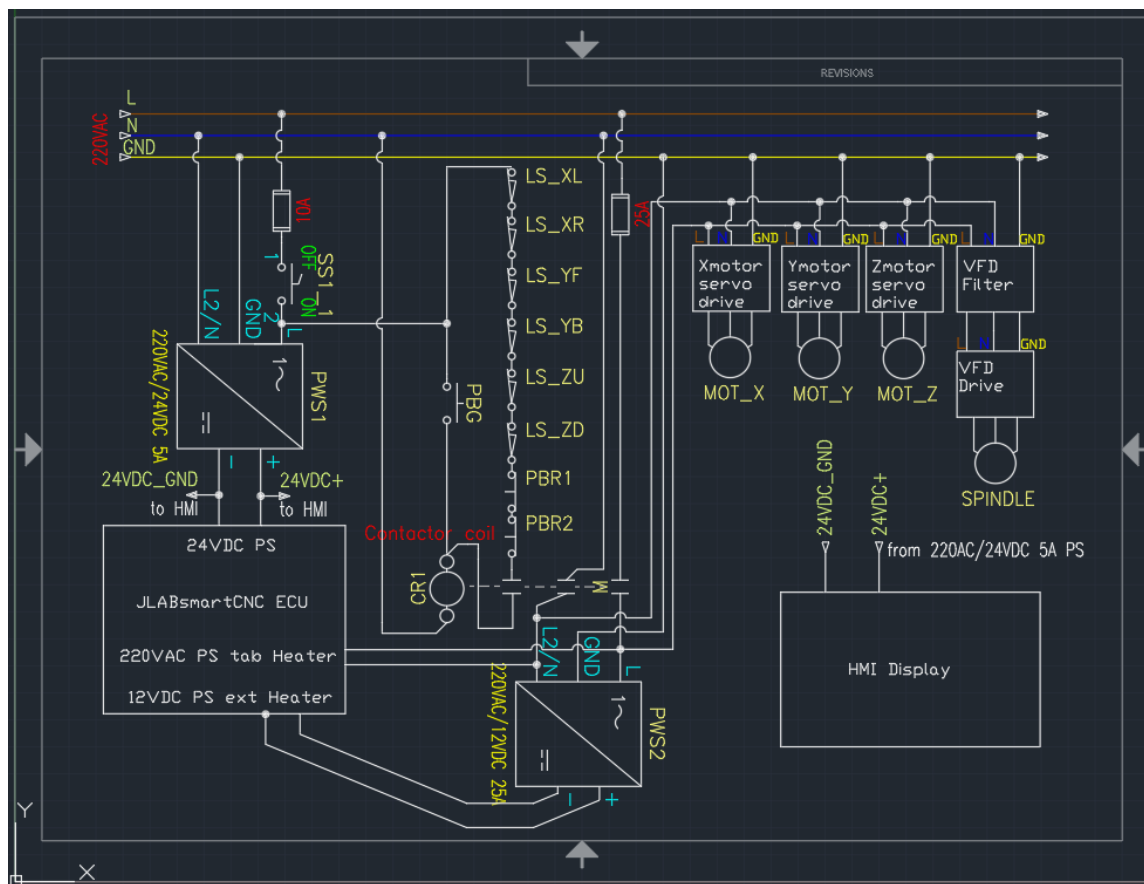
Хибридната CNC-машина има поголем број компоненти во споредба со конвенционалните CNC-системи, што резултира со покомплексна електрична конфигурација и архитектура на напојувањето. Основната идеја на системот е управувачката електроника и сензорите да се вклучуваат преку посебен прекинувач, додека електромоторите се активираат преку централен контактор. На слика 2.3 е прикажана контролната табла на машината, која претставува основен интерфејс за интеракција помеѓу операторот и машината.

Вклучувањето на управувачкиот систем се реализира преку прекинувач со три позиции поставен во горниот десен дел. Во почетната (нулта) положба, целиот контролен систем е исклучен. Во првата позиција, се активира само резервниот батериски систем на електронската управувачка единица, чија функција е да овозможи зачувување на работната состојба при ненадеен прекин на напојувањето од електромрежата. Втората позиција го активира комплетниот управувачки систем, овозможувајќи нормална работа на машината.

Активирањето на електромоторите се врши со притискање на зелениот тастер, при што се активира главниот контактор. Откако ќе се активира, контакторот останува во вклучена состојба сè додека не се притисне црвениот тастер за исклучување или сигурносниот вртлив прекинувач (emergency stop). Во ситуации на ризично однесување, грешка или потенцијално небезбедна работа на системот, операторот може итно да го активира сигурносниот прекинувач, со што се прекинува напојувањето кон сите мотори и се овозможува нивно моментално запирање. Овој механизам претставува клучен дел од безбедносниот систем на машината и обезбедува брза реакција во критични услови.



Слика 2.3. Контролна табла на JLABsmartCNC.



Слика 2.4. Електрична шема на поврзување на главните контролни прекинувачи на напојувањето со сите компоненти на машината.

На слика 2.4 е прикажана електричната шема за поврзување на напојувањето на сите компоненти на хибридната CNC-машина, вклучувајќи ги и главните контролни прекинувачи. Електронската управувачка единица (ЕУЕ) се напојува преку еднонасочен извор со напон од 24 VDC и максимална струја од 5 A, обележан како PWS1 на шемата. Активирањето на ова напојување се изведува преку прекинувачот SS1_1, кој одговара на втората положба на прекинувачот со три позиции прикажан на слика 2.3. Истото напојување се користи и за напојување на HMI-дисплејот, кој претставува примарен кориснички интерфејс за комуникација помеѓу операторот и машината.

Вклучувањето на главниот контактор се иницира со притискање на нормално отворениот тастер PBG (зелениот тастер од слика 2.3), додека неговото исклучување се реализира преку активирање на некој од нормално затворените контакти PBR1 (црвениот тастер) или PBR2 (сигурносниот прекинувач). Дополнително, во серија со овие прекинувачи се поврзани и сите крајни микропрекинувачи (вкупно шест, по два на секоја оска), кои при активирање го исклучуваат главниот контактор, а со тоа и напојувањето до сите мотори и грејачи, со што се овозможува хардверска заштита од потенцијални судири или прегревање.

Напојувањето означено како PWS2 обезбедува 12 VDC со струја до 25 A и се користи за напојување на грејачот на екструдерот. Излезот од ова напојување прво се доведува до ЕУЕ, преку која се изведува управувањето со грејачот. Истовремено, преку главниот контактор се обезбедува 220 VAC напојување до контролната единица која го користи за напојување и управување на грејачот на масата за печатење. На истиот излез

од контакторот се поврзани и драјверите за погонските мотори и главниот мотор за обработка (спиндлата), при што пред фреквентниот регулатор се вметнува соодветен филтер за електромагнетна интерференција (EMI) со цел да се спречи враќање на електромагнетни пречки во електричната мрежа [71,72].

2.4 Погонски мотори на предложената хибридна CNC-машина

Во процесот на дизајн и развој на CNC-машини, изборот на погонски мотори игра клучна улога во определувањето на прецизноста, динамичката стабилност и вкупната функционалност на системот. Во двете претходни верзии на машината, кои беа наменети исклучиво за 3D-печатење, беа користени чекорни мотори поради нивната едноставност и ниска цена. Сепак, со зголемувањето на технолошките барања и интеграцијата на адитивни и супстрактивни процеси во единствен хибриден систем, се наметна потребата од примена на индустриски АС сервомотори. Преминот кон сервомотори овозможи значително подобрување на перформансите, особено во однос на контролата на позицијата, брзината и вртежниот момент. Во продолжение е дадена подетална споредба помеѓу чекорните и сервомоторите од повеќе аспекти на работа:

1. Начин на управување

- **Чекорни мотори:** Користат систем со отворена јамка. Позиционирањето се заснова на бројот на пратени импулси, без потврда дали моторот ја достигнал целната позиција.
- **Сервомотори:** Користат систем со затворена јамка каде повратната информација најчесто се добива од енкодер, мерни ленти или нивна комбинација. Контролата е базирана на реално-временски мерења, што овозможува автоматска корекција при грешки и постигнување на висока точност.

2. Точност и резолуција

- **Чекорни мотори:** Стандардно имаат резолуција од 200 чекори/револуција ($1,8^\circ$), со можност за користење на техниката микростепинг што ги дели чекорите со фактор до максимум 32. Меѓутоа, точноста зависи и од механичкиот товар.
- **Сервомотори:** Енкодерите можат да обезбедат резолуција од 10.000 до 20.000 импулси по револуција или повеќе. Реалната точност е стабилна без разлика на товарот.

3. Вртежен момент и динамика

- **Чекорни мотори:** Обезбедуваат висок вртежен момент при ниски брзини, но тој драматично опаѓа со зголемување на брзината.
- **Сервомотори:** Одржуваат стабилен вртежен момент низ широк опсег на брзини. Идеални се за апликации со брзи забрзувања и нагли промени на движењата.

4. Однесување при оптоварување

- **Чекорни мотори:** Склони се кон губење на чекори при нагли промени на оптоварувањето. Контролата со отворена јамка не овозможува механизам за детекција на грешки.
- **Сервомотори:** Автоматски го зголемуваат вртежниот момент доколку отпорот се зголеми. Преку енкодерот се детектираат грешки или преголем товар.

5. Енергетска ефикасност

- **Чекорни мотори:** Континуирано трошат струја дури и кога мируваат, што доведува до загревање и неефикасност.
- **Сервомотори:** Струјата се модулира според реалната потреба. Нема непотребно загревање при мирување.

6. Загревање и генерирање на бучава

- **Чекорни мотори:** Склони се кон прегревање, особено кога долго мируваат под товар. Генерираат повеќе вибрации и бучава.
- **Сервомотори:** Работат тивко и остануваат ладни при нормално оптоварување.

7. Цена и комплексност

- **Чекорни мотори:** Поевтини, едноставни за инсталација и управување.
- **Сервомотори:** Поскапи и со посложено управување (покомплексен драјвер за управување со повратна врска, енкодер, I/O или сервиска комуникација), но со значително подобри перформанси.

Во согласност со техничките барања на системот, како и со цел да се постигне повисок степен на динамичност, сигурност и прецизност при движењето на оските, во оваа верзија на хибридната CNC-машина е имплементиран **Delta ASDA-B2** сервосистемот (мотори со соодветни хардверски драјвери) [73]. Овие индустриски сервомотори со номинална моќност од 400 W претставуваат оптимален компромис помеѓу цената и перформансите и се широко користени во различните CNC-машини и автоматизирани системи.

Клучни карактеристики на Delta ASDA-B2 – 400 W:

- Номинален вртежен момент: 1.27 Nm.
- Максимален вртежен момент: 3.82 Nm.
- Максимална брзина: 3000 RPM.
- Седумнаесет битен инкрементален енкодер.
- RS-485/Modbus поддршка за конфигурација, параметризација и мониторинг.
- Функции за автоматско подесување (auto-tuning) на PID-параметрите.

Кај овој сервосистем постојат три основни режими на работа:

- Режим на управување со позицијата (Position Control Mode) во кој серводрајверот прима команди за позицијата и со висока прецизност го позиционира роторот на моторот на посакуваната цел. Командите доаѓаат преку влезно-излезниот терминал во форма на импулси.
- Режим на управување со брзината (Speed Control Mode) во кој серводрајверот прима команди за брзината и го придвижува моторот со посакуваната брзина. Командите можат да се зададат со запишување во регистрите на драјверот (преку сервиска комуникација со MODBUS-протокол) или преку аналогните влезови со аналоген сигнален напон (± 10 V).
- Режим на управување со вртежениот момент (Torque Control Mode) во кој серводрајверот прима команда за вртежниот момент и го движи моторот со тој вртежен момент. Командите се задаваат со запишување во регистрите на драјверот (преку сервиска комуникација со MODBUS протокол) или преку аналогните влезови на драјверот со аналоген сигнален напон (± 10 V).

Овој сервосистем овозможува и дуален режим на работа во кој може да се комбинираат два од основните режими на работа. Дуалниот режим овозможува менување на режимот на работа со контролен влезен сигнал што значи дека надворешниот контролен систем одлучува кога и кој режим на работа ќе го користи.

Во контекст на дизајнот на оваа хибридна CNC-машина, како најсоодветен е избран режимот за позиционирање (Position mode), кој овозможува прецизна контрола на положбата на оските според бројот на примени импулси од електронската управувачка единица (EUE). Во овој режим, клучен параметар е линеарната резолуција, односно линеарното поместување на оската што се предизвикува од еден импулс испратен од EUE кон серводрајверот.

За сите оски на системот се користи идентична механика и резолуција. Навојните вретена на машината JLABsmartCNC имаат преносен однос од 5 mm/rev, односно при една ротација на моторот се добива линеарно поместување од 5 милиметри на оската. Енкодерите вградени во сервомоторите имаат резолуција од 160 000 импулси по револуција, што доведува до теоретска линеарна резолуција од 0.03125 μm по импулс. Ова е премногу висока резолуција која многукратно ја надминува и хардверската точност на машината, а за управување бара импулси со многу висока фреквенција, што може да предизвика компликации при генерирање и поголема чувствителност на грешки. Резолуцијата што најчесто се користи кај индустриските CNC-системи е 1 μm по импулс што е прифатено и кај JLABsmartCNC. За постигнување на оваа резолуција хардверските драјвери овозможуваат поставување на множители и делител на преносниот однос. Линеарната резолуција се пресметува со следната формула:

$$\text{Линеарна резолуција} = \frac{L \times N_x}{R \times D} \quad (2.1)$$

каде што:

- L – линеарно поместување при една револуција на навојното вретено (преносен однос на навојното вретено)
- N_x – електронски множител кој се задава во самиот драјвер со кој се мултиплицира бројот на импулси. Постојат 4 вакви множители кои можат да се дефинираат од страна на корисникот и EUE да ги менува со помош на влезните линии на драјверот во зависност од потребите на апликацијата. Во конкретната верзија на машината се работи само со еден фиксен множител и фиксна резолуција од 1 $\mu\text{m}/\text{pulse}$.
- R – претставува резолуција на енкодерот изразена во импулси во една револуција
- D – вредност на електронскиот делител со кој се редуцира бројот на импулси.

Во тековната верзија множителот е поставен на 160, а делителот на 5. Со овие вредности се добива линеарна резолуција од 1 μm според изразот (2.1):

$$\text{Линеарна резолуција} = \frac{5000 \mu\text{m}/\text{rev} \times 160}{160\,000 \text{ pulse}/\text{rev} \times 5} = 1 \mu\text{m}/\text{pulse} \quad (2.2)$$

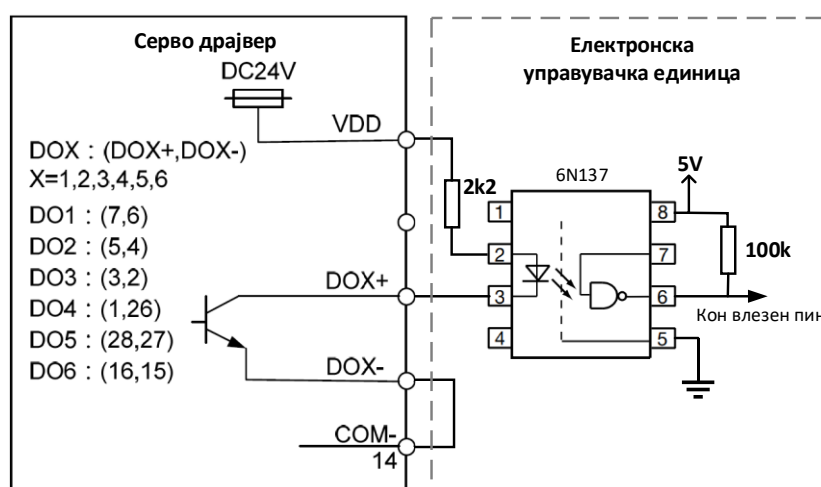
Комуникацијата помеѓу секој серводрајвер и електронската управувачка единица е имплементирано преку 7 дигитални излези и 4 дигитални влезови. Пет од дигиталните излези се помошни управувачки излези и нивната конфигурација, односно шема на поврзување е дадена на слика 2.5. За секој од петте излези се користи NPN биполарен брз транзисторски излез кој го активира оптички изолираниот влез на драјверот.



Слика 2.5. Поврзување на податочните дигитални излези од ЕУЕ со влезовите на серводрајверот.

Детален опис на функциите на овие пет контролни сигнали е даден во продолжение:

- **Servo ON** - при активирање на овој сигнал моторот се вклучува и е во закочен режим.
- **Pulse clear** - со активирање на овој сигнал се бришат баферите со примени импулси во самиот драјвер. Оваа опција се користи кога се вклучени филтрите кај серводрајверот и во оваа верзија на машината овој сигнал не се користи.
- **Alarm reset** – овој сигнал се користи за бришење на грешките настанати во драјверот.
- **GNUM0** и **GNUM1** – овие два сигнали се користат за селекција на електронскиот множител N_x . Селектирањето се прави од четирите предефинирани множители. Во оваа верзија се користи фиксен множител, но хардверски оваа опција е овозможена за да можат да се прават експерименти на алгоритмите за интерполација со променлив множител, односно променлива резолуција.



Слика 2.6. Поврзување на податочните дигитални излези од серводрајверот со влезовите на ЕУЕ.

На слика 2.6 е прикажана шемата на поврзување помеѓу дигиталните излезни сигнали од серводрајверите и соодветните влезни канали на електронската управувачка единица (ЕУЕ). Серводрајверите користат NPN-транзисторски дигитални излези, додека на страната на ЕУЕ се имплементирани оптички изолирани влезови, базирани на брзи оптокаплери од типот 6N137, кои овозможуваат сигурна детекција на сигнали со висока фреквенција.

Секоја излезна сигнална линија од серводрајверот носи одредена статусна или дијагностичка информација, која е неопходна за правилно функционирање и надгледување на сервосистемот. Во продолжение е даден опис на функцијата на секој од излезните сигнали:

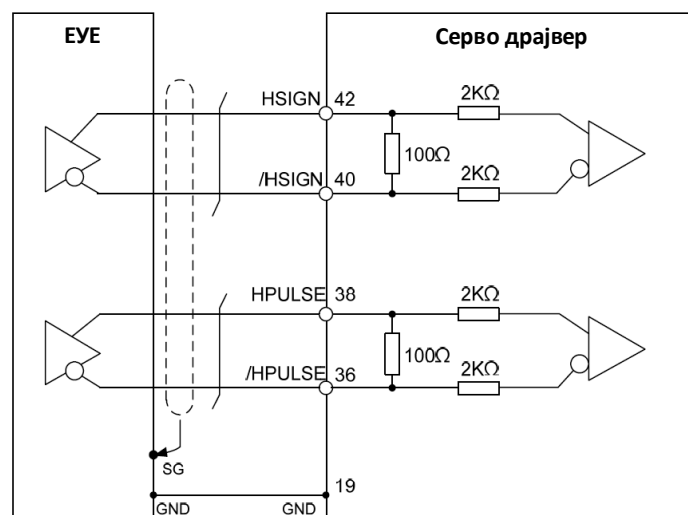
- **Servo Ready** – Сигнал кој се активира кога серводрајверот е подготвен за работа. Тој укажува дека моторот е иницијализиран, успешно вклучен и е подготвен да прифаќа команди од ЕУЕ.
- **Zero Speed Reached** – Контролен излез кој е активен кога моторот е во состојба на мирување, односно кога тековната брзина е нула.
- **Target Position Reached** – Сигнал што го активира серводрајверот кога моторот ја достигнал зададената позиција. Оваа повратна информација е клучна за потврда на успешно извршеното позиционирање и за синхронизација со следната фаза од процесот.
- **Servo Alarm** – Алармен сигнал кој индицира присуство на грешка во сервосистемот, како што се прегревање, пречекорување на максималната струја за дадените мотори, заглавување или загуба на повратната врска од енкодерот и др. Овој сигнал се чита од страна на ЕУЕ, која ја прикажува грешката на екранот и овозможува соодветно дијагностицирање и реакција од операторот.

Прецизната интерпретација и обработката на дигиталните влезно-излезни сигнали од серводрајверите е од суштинско значење за стабилна и безбедна работа на CNC-системот. Сепак, најкритичната комуникација помеѓу ЕУЕ и серводрајверите се линиите преку кои се испраќаат импулсите за чекорите и сигналите за насоката на ротација на секој од моторите.

Серводрајверите од серијата ASDA-B2 поддржуваат повеќе типови на влезни интерфејси, вклучувајќи:

- Оптички изолирани влезови, кои нудат галванска изолација и се посоодветни за заштита во индустриска средина, но имаат ограничена максимална брзина до 500 kpps (500 000 импулси во секунда), дополнително редуцирање на оваа брзина се прави со филтрите присутни на овие влезови од драјверот.
- Диференцијални влезови, кои овозможуваат далеку поголема пропусна моќ и сигурност при преносот, со поддршка за комуникација до 4 Mrps (4 милиони импулси во секунда), што ги прави значително попогодни за системи кои бараат висока прецизност и брзина на движење.

Во тековната верзија на JLABsmartCNC, се користи диференцијален начин на поврзување, при што во самата ЕУЕ се имплементирани соодветни линиски драјвери кои генерираат и испраќаат командните сигнали во диференцијален формат. Овој пристап овозможува значително намалување на електромагнетни пречки (EMI), зголемена отпорност на шум, како и стабилна и сигурна комуникација при високи фреквенции на импулсите.



Слика 2.7. Поврзување на влезни пинови за импулси од серводрајверот со EYE.

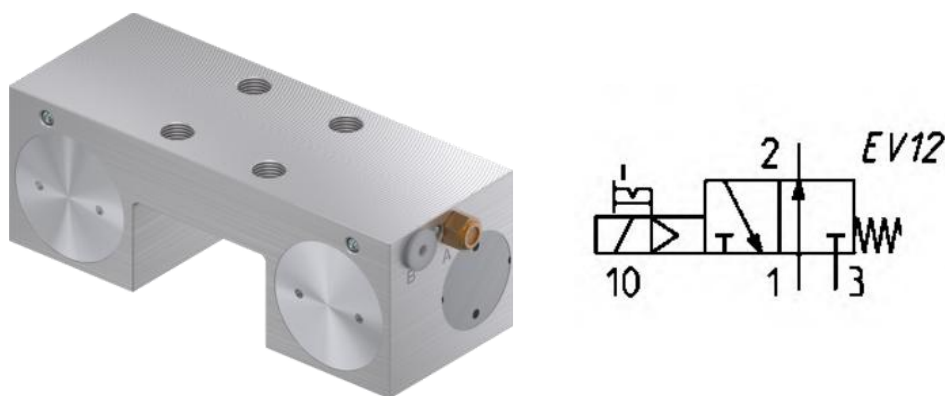
Важно е да се напомене дека при употреба на диференцијален интерфејс, референтните маси (GND) на EYE и серводрајверите мора да бидат кусо поврзани, со што оптичката изолација на останатите влезови од слика 2.6 ја губи својата функција, но ова е стандардна конфигурација наменета за индустриски услови.

Шемата за поврзување на сигналните линии за импулсите (HPULSE) и насоката (HSIGN) со диференцијалните излези од EYE кон серводрајверите е прикажана на слика 2.7. Оваа конфигурација овозможува доверливо управување со сервомоторите при високи работни брзини и овозможува целосна искористеност на прецизноста и динамиката на системот.

Дополнително, за сите сигнални линии, вклучувајќи ги и оние за импулси и насока, се користат екранирани (оклопени) кабли, со цел да се обезбеди заштита од надворешни електромагнетни пречки и да се гарантира стабилен и сигурен пренос на сигналите во индустриско опкружување. Екраните на каблите се соодветно поврзани со сигнална маса на едната страна, во согласност со препораките за заземјување кај сензитивни комуникациски линии.

На слика 2.8 е прикажана пневматската кочница што се користи за механичко задржување на Z-оската во случај кога моторот е деактивиран. Оваа кочница е особено важна за спречување на неконтролирано паѓање на Z-оската поради тежината на спиндлата, што може да доведе до оштетување на навојното вретено или алатот. Кочницата се активира кога на нејзиниот приклучок е поврзан довод на компресиран воздух. Нејзиното управување е имплементирано со помош на нормално затворен (NC) електро-пневматски вентил, чија пневматска шема е прикажана десно на истата слика. На приклучокот 1 од пневматскиот вентил се носи доводот со компримиран воздух, а на приклучокот 2 е поврзана пневматската кочница. Излезот 3 служи за испуштање на воздухот од кочницата кога вентилот е во активна состојба (2 и 3 се споени).

При активирање на моторот за Z-оската, EYE преку релеен излез го активира електро-пневматскиот вентил, се ослободува кочницата и се овозможува слободно движење на оската. Овој механизам обезбедува дополнителна сигурност и стабилност во работата на Z-оската, особено при исклучување или при губење на напојувањето за моторите.



Слика 2.8. Пневматска кочница за Z-оската.

2.5 Глава за обработка (спиндла)

За реализација на суптрактивниот процес кај хибридната CNC-машина, се користи индустриски асинхрон индуccionен мотор (спиндла) со висока брзина од серијата на Hertz од типот HM90-ER25S-CK-CR-2R, со номинална моќност од 4 kW и максимална брзина на ротација од 18 000 вртежи во минута (rpm). Моторот работи на еднофазно напојување од 220 V, што е во согласност со дизајнерскиот пристап целата машина да биде компатибилна со стандардна електрична мрежа, без потреба од трофазно напојување.

Во самата обработка најбитно за спиндлата е да се овозможи прецизно управување со брзината на ротација и да се избере мотор со доволна моќност за обработка на посакуваните материјали. Избраниот мотор е повеќе од доволен за тековната машина, а за негово управување се користи фреквентен регулатор од серија Yaskawa GA500, кој е исто така наменет за еднофазен 220 V влез и е оптимизиран за управување со асинхронни мотори со висок број на вртежи [74]. Фреквентниот регулатор нуди повеќе комуникациски и контролни опции, меѓу кои:

- Индустриски аналогни влезови (0–10 V или 4–20 mA).
- Дигитални влезови за контрола на основни функции на фреквентниот регулатор.
- Импулсни влезови за контрола на брзината.
- RS485 сервиска линија со поддршка за MODBUS-RTU-протокол.
- Опционални модули за индустриски комуникациски протоколи (Ethernet/IP, PROFINET, EtherCAT и др.).

Во рамките на JLABsmartCNC, за комуникација меѓу фреквентниот регулатор и електронската управувачка единица (EYE) е избрана RS485 сервиската комуникација со MODBUS-RTU-протокол, поради нејзината сигурност, стабилност и можност за двонасочен пренос на податоци со само две жици. Преку оваа врска, EYE испраќа команди за вклучување/исклучување, задавање на брзината (фреквенцијата), како и прима информации за моменталната состојба на фреквентниот регулаторот, вклучувајќи алармни сигнали и излезни параметри.

Изборот на MODBUS протоколот, имплементиран преку RS485 сервиската линија, се покажа како најпрактично и најсоодветно решение за оваа машина, бидејќи овозможува лесна интеграција во контролниот систем, сигурен пренос на податоци дури и во индустриско опкружување, како и можност за проширување со повеќе уреди на истата комуникациска линија, иако во тековната верзија оваа комуникациска порта на

EYE се користи исклучиво за управување со фреквентниот регулатор. Во рамките на MODBUS-имплементацијата, EYE има улога на мастер, додека фреквентниот регулатор функционира како слејв уред.

За да се обезбеди стабилна и оптимизирана работа на индуkциониот мотор во рамките на хибридна CNC-машина, пред пуштањето во работа е извршена постапка на автоматска калибрација (auto tuning) користејќи го вградениот алгоритам во фреквентниот регулатор Yaskawa GA500. Оваа постапка овозможува прецизно пресметување на електричните и механичките параметри на моторот, неопходни за прецизна регулација на брзината и моментот, како и за адаптација кон реалните услови на оптоварување.

Во конкретниот случај е избран режимот на ротационо автоматско калибрирање (Rotational Auto-Tuning), при што моторот краткотрајно се пушта да ротира без товар под надзор на фреквентниот регулатор, со цел да се изврши мерење и анализа на неговите параметри. Пред активирањето на функцијата за автоматска калибрација, преку корисничкиот интерфејс на регулаторот, се задаваат следните фабрички параметри на моторот:

- T1-01: Режим на автоматска калибрација – поставен на 0 (Rotational Auto-Tuning);
- T1-02: Номинална моќност на моторот – 4 kW;
- T1-03: Номинален напон на моторот – 220 V;
- T1-04: Номинална струја на моторот – 16,4 A;
- T1-05: Максимална фреквенција на моторот – 300 Hz;
- T1-06: Број на полови на моторот – 2;
- T1-07: Максимална брзина на моторот – 18 000 rpm;
- T1-13: Напон без оптоварување – се внесува истиот напон од параметарот T1-03 (220 V).

По внесувањето на параметрите, се активира функцијата за автоматска калибрација преку менито на фреквентниот регулатор. При овој процес, драјверот автоматски го вклучува моторот да ротира со различни фреквенции со цел да се измерат вредностите за неговите електромагнетни и динамички карактеристики. Процедурата завршува по неколку минути, по што сите добиени параметри автоматски се запишуваат во трајната меморија на фреквентниот регулатор.



Слика 2.9. Фреквентен регулатор и мотор за спиндла.

Оваа функција значително го подобрува однесувањето на системот, овозможувајќи стабилно стартување, брза реакција на промените во оптоварувањето и минимизирање на грешките при регулација на брзината. Постапката за автоматска калибрација се покажа како многу ефикасен и едноставен начин за соодветно подесување на параметрите на дадениот мотор без комплицирани процедури кои треба да ги изведува корисникот на овој систем. На слика 2.9 е прикажан изгледот на драјверот (фреквентниот регулатор) и моторот кој се користи како спиндла на JLABsmartCNC-хибридната машина.

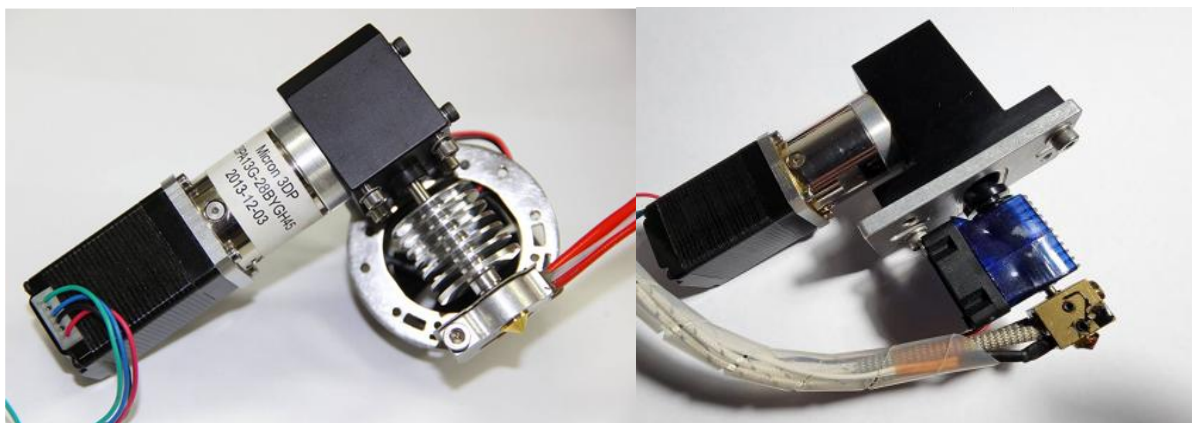
2.6 Екструдер и маса за печатење

Главата на печатарот има клучна улога во адитивното производство, бидејќи директно влијае врз прецизността, стабилноста и квалитетот на изработените 3D-модели. Меѓу различните методи за нанесување на материјал, технологијата на екстудирање се издвојува како најфлексибилно и широко применливо решение, овозможувајќи работа со различни типови термопластични материјали. Основната функција на екструдерот е повлекување на материјал во форма на филамент, негово топење преку контролиран термички процес и прецизно нанесување на подлогата за печатење. Поради ова, екструдерот се смета за највиталната компонента на системот за 3D-печатење, од која во најголема мера зависи успешноста и квалитетот на производствениот процес.

Во претходните верзии на машината беше користен MICRON3DP-екструдерот, кој припаѓа на втората генерација на целосно метални и интегрирани екструдери. Овој тип на екструдер обединува два главни потсклопа: механизам за повлекување на материјалот и загревачки дел за негово топење, познат како хот-енд. Ваквата конструкција овозможува лесно демонтирање на единицата за топење заедно со млазницата, што значително го олеснува одржувањето и го минимизира времето на прекин при затнување на системот – проблем што е често присутен кај екструдерите од постарите генерации. Иако MICRON3DP-екструдерот донесе значителни подобрувања во однос на стабилноста, конструктивната цврстина и во прецизността на повлекувањето на материјалот, во пракса проблемот со заглавување на материјалот во хот-ендот сè уште беше релативно чест.

Со цел да се зголеми оперативната стабилност и да се надмине овој проблем, во оваа верзија на машината, оригиналниот хот-енд е заменет со широко користена алтернативна верзија, која покрај тоа што беше компатибилна со постојниот дизајн, имаше и значително пониска цена, а притоа го немаше проблемот со заглавување на материјалот. На слика 2.10 на левата страна е прикажан MICRON3DP-екструдерот со оригиналниот хот-енд, а на десната страна со новата алтернативна верзија на хот-ендот кој се користи кај JLABsmartCNC.

Она што го издвојува MICRON3DP од останатите екструдери со пониска цена е неговиот робустен и сигурен дизајн, кој овозможува стабилно и прецизно повлекување на филаментот. Ова е постигнато преку употреба на чекорен мотор со висок вртежен момент и погонски механизам изработен од нерѓосувачки челик со специјално профилирани запци. Дополнително, интегриран е и целосно метален редуктор со цел зголемување на прецизността и резолуцијата при процесот на екстудирање. Промената на материјалот е поедноставена благодарение на конструктивното решение што овозможува брза и лесна замена. Патот на движење на филаментот е внимателно дизајниран со мазна внатрешна површина и континуирана геометрија, што значително го намалува ризикот од заглавување на материјалот.



Слика 2.10. MICRON3DP-екструдер со два различни краеви за топење на материјалот.

Екструдерот е составен од следните клучни компоненти:

- **Мотор за повлекување на суровината** – се користи чекорен мотор со номинален напон од 4,56 V и номинална струја од 0,67 A. Отпорноста по фаза изнесува $6,8\Omega$, а индуктивноста 4,9 mH. Моторот се поместува за агол од $1,8^\circ$ при еден чекор (200 чекори во една револуција), а преку вграден механички редуктор со преносен однос 1:13,76, овие 200 чекори се зголемуваат на 2752 во една револуција, што значително ја зголемува прецизностa на повлекувањето на филаментот.
- **Hot-end (глава за топење и нанесување на материјалот)** – се состои од млазница преку која се нанесува стопениот материјал. Млазницата е дел од грејниот склоп кој овозможува топење на филаментот до температури од 350°C .
- **Електричен грејач** – со номинален напон од 12 V и излезна моќност од приближно 60 W, поставен во близина на млазницата, со задача да го топи материјалот пред неговото нанесување.
- **Температурен сензор** – NTC термистор со отпорност од $100\text{ k}\Omega$ при 25°C , се користи за мерење на температурата на екструдерот и како повратна информација во PID-алгоритмот за регулација на температурата.
- **Систем за ладење** – кој се состои од вентилатор и алуминиумски ладилник поставени на горниот дел од екструдерот, кои ја стабилизираат температурата во областа каде филаментот сè уште е во цврста состојба. На овој начин се спречува прерано топење и заглавување на материјалот во цевката кон млазницата.
- **Механичка конструкција** – која обезбедува механичка стабилност и правилно позиционирање на сите компоненти на екструдерот, овозможувајќи непречено работење и лесно одржување.

Прецизностa на повлекувањето на материјалот претставува еден од најкритичните аспекти на системот за адитивно производство, со директно влијание врз квалитетот и прецизностa на печатењето. За да се обезбеди стабилно и точно повлекување на филаментот, неопходна е употреба на квалитетен и прецизен хардверски драјвер за управување на чекорниот мотор на екструдерот. Иако механичката конструкција на екструдерот игра значајна улога, без соодветно управување преку прецизен драјвер не е можно да се постигнат барањата за висока резолуција и стабилност во екструдирањето.

Во рамките на овој систем, за управување на моторот на екструдерот се користи индустриски драјвер базиран на чипот TV6600, прикажан на слика 2.11. Драјверот поддржува микростепинг техника на делење на чекорите, со што се овозможува многу

поголема резолуција во движењето на моторот. Максималниот фактор на делење на чекорот кај овој драјвер е $1/32$, што значи дека секој основен чекор на моторот може да се подели на 32 микрочекори, овозможувајќи фино и прецизно дозирање на материјалот [75].

Управувачката електроника на драјверот користи прекинувачка (chopper) топологија, при што се применува повисок напон од номиналниот за напојување на моторот (24 V), додека струјата низ намотките се контролира со прекинувачки режим, што овозможува стабилна работа дури и при големи оптоварувања [76]. Во конкретната конфигурација, границата на струјата е поставена на околу 0,5 A, во согласност со номиналната струја на чекорниот мотор, додека факторот за делење на чекорите е конфигуриран на максималната вредност ($1/32$) за постигнување на највисока можна прецизност.

За правилно користење на екструдерот главна улога има познавањето на неговата резолуција. Моторот на екструдерот има 200 чекори во една револуција односно аголот на ротација на еден чекор изнесува $1,8^\circ$. Односот на редукторот е $1:13,76$ што значи дека бројот на чекори во една револуција го зголемува од 200 на:

$$13,76 \times 200 = 2752 \text{ чекори.} \quad (2.3)$$

Дијаметарот на запчаникот што го повлекува материјалот е 12 mm, па неговиот периметар е:

$$3,14 \times 12 = 37,68 \text{ mm.} \quad (2.4)$$

Ако моторот работи во full-step режим, бројот на чекори во еден милиметар ќе биде:

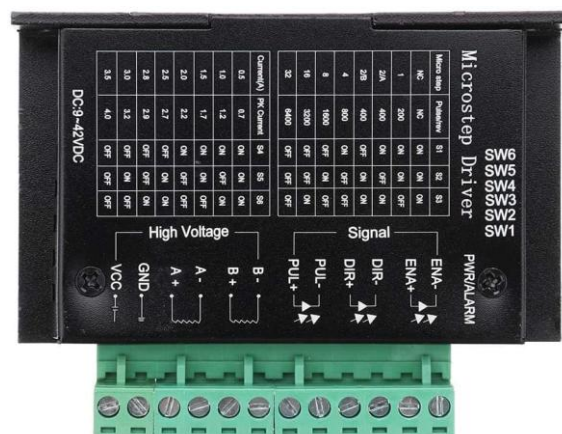
$$2752/37,68 = 73,03 \text{ step/mm.} \quad (2.5)$$

Бидејќи конкретниот драјвер го дели чекорот со фактор $1/32$ бројот на чекори во еден милиметар се зголемува на:

$$32 \times 73,03 = 2336,96 \text{ step/mm.} \quad (2.6)$$

Па според тоа резолуцијата на еден чекор ќе биде:

$$1/2336,96 = 0,000427906 \text{ mm} \quad (2.7)$$



Слика 2.11. Драјвер за управување на чекорниот мотор на екструдерот.

Подлогата за печатење (масата за печатење) има суштинска улога во процесот на 3D-печатење, особено во формирањето на првите слоеви, каде што е критично да се обезбеди соодветна адхезија помеѓу материјалот и подлогата. За таа цел, потребно е прецизно загревање на масата за печатење на точно дефинирана температура, зависно од видот на материјалот што се користи (на пр. PLA, ABS, PETG итн.).

Во претходните верзии на машината се користеше алуминиумска подлога за печатење со димензии од 200×300 mm, на која беше прицврстен грејач со моќност од 120 W и работен напон од 12 V. Иако функционален, овој систем се покажа како неефикасен во однос на брзината на загревање, а дополнително поради нискиот работен напон, грејачот повлекуваше релативно висока струја, што резултира со дополнителен термички стрес на управувачките компоненти.

Во тековната верзија на JLABsmartCNC е интегрирана значително поголема алуминиумска маса за печатење со димензии од 400×500 mm, па загревањето на оваа маса со ист тип на грејач само ќе ги мултиплицира проблемите од претходната верзија. Затоа во оваа верзија се користи високоефикасен силиконски грејач залепен на самата маса со моќност од 800 W, кој работи на стандардно AC 220 V напојување. Ова овозможува значително побрзо постигнување на работната температура и подобра термална стабилност во текот на целиот процес на печатење.

Контролата на температурата е реализирана преку PID-управување, идентично на регулацијата кај екструдерот, при што како сензор за повратна врска се користи NTC-термистор. Параметрите на регулацијата се соодветно подесени за да овозможат одржување на константна температура со висока прецизност, што е од клучно значење за квалитетот на печатените делови и избегнување на деформации на 3D-моделите.

2.7 Интерфејси човек-машина

Во рамките на хибридна CNC-машина JLABsmartCNC, имплементирани се два независни интерфејси за комуникација помеѓу корисникот и системот: 10-инчен индустриски екран на допир (HMI) и наменски развиен компјутерски софтвер. Овие интерфејси овозможуваат контрола, конфигурација и мониторинг на состојбата на машината во реално време, при што комуникацијата со електронската управувачка единица (EUE) е реализирана преку две посебни сериски линии, со што се обезбедува стабилност, независност и сигурност во преносот на податоци.

Индустрискиот MODBUS-RTU-протокол се користи како комуникациски протокол помеѓу HMI-дисплејот и EUE имплементиран преку RS485 сериска линија. Во оваа комуникација дисплејот е конфигуриран како мастер, а EUE како слејв. Корисничкиот интерфејс на HMI е развиен во специјализиран развоен софтвер за конкретниот модел на екран, при што посебно внимание е посветено на ергономијата, интуитивноста и функционалноста на дизајнот, со цел олеснување на ракувањето дури и за оператори без напредни технички познавања.

Дополнително, за потребите на напредното управување на работата на машината, развиен е наменски компјутерски софтвер кој овозможува целосно управување на машината преку персонален компјутер. Со овој софтвер програмите за машината можат да се пуштаат директно од компјутер, без потреба од префрлање на програмата преку USB-мемориски уред во електронската управувачка единица. Иако првично комуникацијата со EUE беше имплементирана преку RS485 сериска линија, поради ограничувањата на half-duplex карактерот на овој интерфејс (не е можно двонасочна

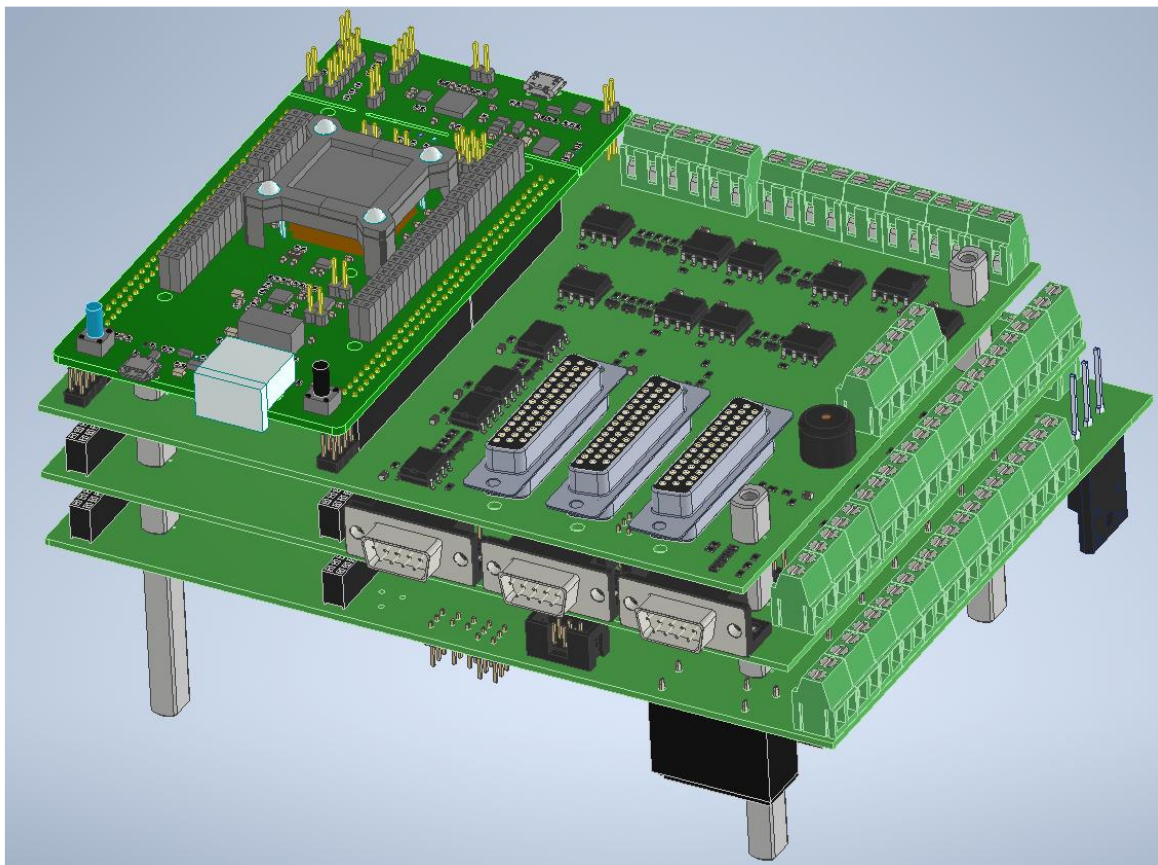
комуникација во ист момент), истата беше заменета со RS232 сервиска комуникација. RS232 овозможува full-duplex пренос на податоци (двонасочна комуникација во секој момент), што е од суштинско значење за правилно функционирање на комплексниот комуникациски протокол развиен за оваа намена. Со цел оптимизација на системската архитектура и намалување на бројот на физички врски, целокупната човек-машина комуникација е реализирана преку само две независни сервиски линии, што дополнително ја зголемува прегледноста, стабилноста и ја намалува комплексноста на системот.

2.8 Електронска управувачка единица (EUE)

Електронската управувачка единица (EUE) претставува централниот дел, односно мозокот на системот, задолжен за контрола на сите компоненти и периферни уреди на хибридната CNC-машина. Со цел оптимална организација на функциите, како и заради намалување на електромагнетните пречки и зголемување на стабилноста и електричната безбедност, системот е реализиран преку четири физички одвоени, но логички поврзани електронски плочки (PCB). Распоредот и поврзаноста на плочките се прикажани на слика 2.12.

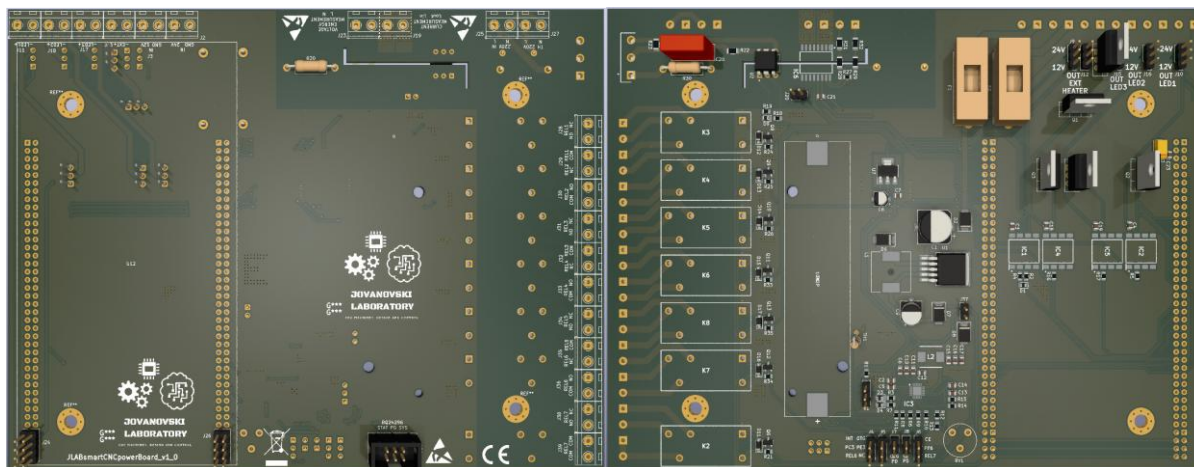
Основната, најдолна плочка во системот е напојувачката плочка, која врши конверзија на влезниот напон од 24VDC (стандарден напон за целата електронска управувачка единица) во соодветни секундарни напонски нивоа потребни за работата на различните делови од EUE. На оваа плочка се генерираат еднонасочен напон од 5V преку прекинувачки (switching) стабилизатор со излезен капацитет од 3A, како и 3,3V преку линеарен регулатор од петте волти. Дополнителни регулатори за 3,3V се интегрирани и на останатите плочки во системот, со цел да се минимизираат евентуални влијанија од падови на напон, шумови и електромагнетни пречки, како и да се избегне преоптоварување на единечен регулатор. Напонот од 5V се дистрибуира до сите плочки од централниот модул.

Од задната страна на напојувачката плочка е интегриран резервен батериски систем, изведен со литиум-јонска батерија од типот 18650 со номинален напон од 3.7V и капацитет од 1500mAh. За управување со батеријата се користи интелигентен контролер BQ24296, кој поддржува брзо полнење до 3A. Системот е дизајниран автоматски да се префрла од надворешен извор на батериско напојување во случај кога надворешното напојување е прекинато. Комуникацијата помеѓу микроконтролерот и контролерот на батерискиот систем е реализирана преку I²C комуникациски протокол. Во нормални услови, контролерот ја одржува батеријата наполнета, додека во случај на прекин на надворешното напојување, батерискиот систем веднаш го презема напојувањето на главниот микроконтролер. Во таква ситуација, сите потрошувачи со голема моќност (моторите, грејачите и сл.) се исклучуваат, а микроконтролерот ја зачувува моменталната состојба на машината. По повторно воспоставување на надворешното напојување, контролерот сигнализира за грешка преку корисничкиот интерфејс и чека потврда од операторот. По потврдата, системот автоматски врши рекалибрација на координатите и го продолжува извршувањето на програмата од точката на прекин. Дополнително, на оваа плочка е имплементиран и систем за мерење на потрошувачката на електрична енергија, со што се овозможува енергетска анализа и проценка на трошоците за изработка на индивидуален дел. За таа цел, на фазата на главниот доведен кабел е поставен струен трансформатор, чии сигнали се обработуваат преку специјализираниот интегриран сензор ACS37800, комуникацијата со микроконтролерот повторно се одвива преку I²C интерфејс.

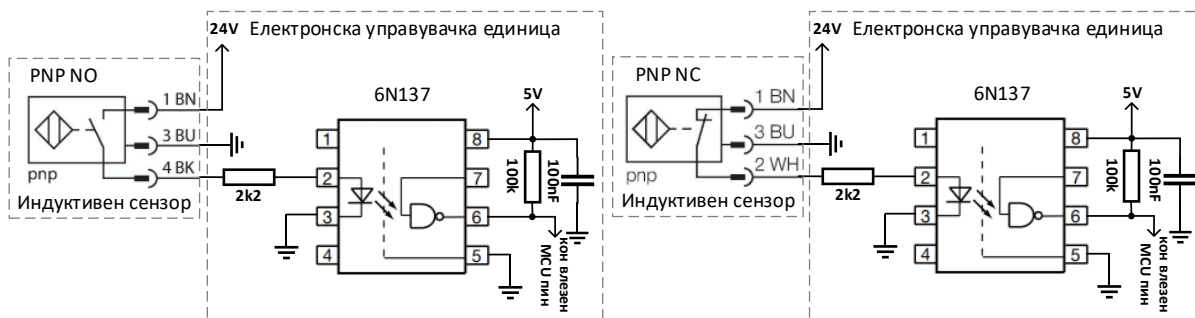


Слика 2.12. 3D-модел на електронската управувачка единица (EUE).

Плочката исто така содржи 7 изолирани релејни излези со максимална прекинувачка способност на контактите од 220VAC/10A или 30VDC/10A, со што е овозможено сигурно вклучување и исклучување на различните компоненти во системот. Дополнително, вградени се 4 MOSFET излези, од кои секој е конфигурабилен за работа со 12V или 24V грејачи, при што во оваа верзија на системот се користи само еден излез за управување со грејачот на екструдерот. За управување со грејачот на масата за печатење, користен е оптоизолиран излез со триак, кој овозможува безбедна и прецизна контрола на АС-напојување. На слика 2.13 е даден изгледот на оваа електронска плочка и тоа горна страна на левиот дел од сликата и долна страна на десниот дел од сликата.



Слика 2.13. Изглед на електронската плочка за напојување.



Слика 2.14. Поврзување на индуктивните сензори со ЕУЕ.

Електронската плочка на второто ниво од електронската управувачка единица (ЕУЕ) е наменета за кондиционирање и обработка на сигналите од сензорите, како и за реализација на комуникациските интерфејси.

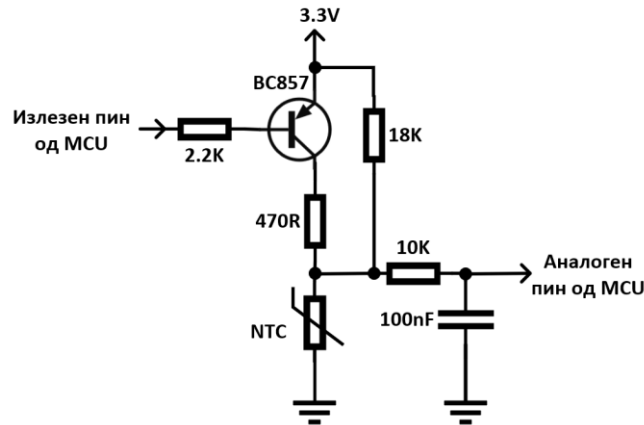
Во рамките на оваа плочка, имплементирани се шесте индуктивни сензори за детекција на крајните позиции на сите оски, чие читање се изведува софтверски од страна на ЕУЕ. Поради нивната критична улога во ограничувањето на движењата и спречување на механички оштетувања од удар во крајните позиции на машината, избрани се сензори со соодветни електрични карактеристики, заедно со внимателно дизајнирани кола за кондиционирање на сигналите. За да се обезбеди сигурна дијагностика и откривање на потенцијални дефекти (како што е прекината жица), употребени се PNP NC (нормално затворени) индуктивни сензори кај пет крајни позиции од трите оски (шемата на поврзување е дадена на слика 2.14 десно). Кај овие позиции механичкото поставување на сензорите е такво што само во крајните положби на оските во видното поле на сензорот има присуство на метален дел што го активира сензорот. Во нормална состојба, кога не е активирана крајна позиција, овие сензори пропуштаат струја и го активираат влезот на оптоизолаторот (тип 6N137), со што се овозможува ЕУЕ да го следи нивниот статус. Во случај на оштетување на сензорот или прекин на жица, системот детектира состојба идентична со крајна позиција, но поради неможноста за активирање при движење во спротивната насока, се активира аларм и машината прекинува со работа.

Кај Z-оската, поради специфичниот механички распоред, за долната крајна позиција е применет PNP NO (нормално отворен) сензор (шемата на поврзување е дадена на слика 2.14 лево). Сензорот детектира метал во сите позиции освен во крајната со што во рамките на безбедните позиции е активиран и контактот е затворен. Принципот на детекција на грешка и алармирање е идентичен како кај NC-сензорите.

Дополнително, за поголема безбедност, зад индуктивните сензори се додадени механички крајни прекинувачи (микропрекинувачи) како хардверска заштита, кои директно го исклучуваат главниот контактор, како што беше објаснето во поглавјето 2.2.

Во однос на комуникациските интерфејси, на оваа плочка се интегрирани:

- три RS485 комуникациски линии, од кои две имаат можност за изведба на стандардни DB9 конектори или на конектори со завртки (terminal blocks), додека третата линија е достапна исклучиво преку конектор со завртки;
- една RS232 комуникациска линија со излез на стандарден DB9 конектор, наменета за поврзување со персонален компјутер или друг терминал.



Слика 2.15. Коло за кондиционирање на температурните сензори.

Како дополнителна функционалност, на оваа плочка е интегриран слот за мемориска SD-картичка, кој овозможува директно вчитување на програми во ЕУЕ. И покрај тоа што оваа опција е поддржана, во пракса ретко се користи поради постоењето на USB-интерфејс, кој нуди поголема практичност и флексибилност при префрлање на податоци.

Следно, во рамките на оваа плочка е имплементиран и системот за читање на температурите на машината, со вкупно три аналогни влезови поврзани со NTC-термистори кои ја мерат температурата на:

- главата на екструдерот,
- масата за печатење,
- амбиенталната температура.

Секој сензор е поврзан со посебно коло за кондиционирање прикажано на слика 2.15. За зголемена точност и проширување на опсегот на мерење, особено за високите температури до 300 °C, во колото се користат две паралелни гранки составени од напонски делители и заеднички излезен нископропусен RC-филтер со временска константа $\tau = 1 \mu s$. Отпорностите на фиксните отпорници од напонските делители се:

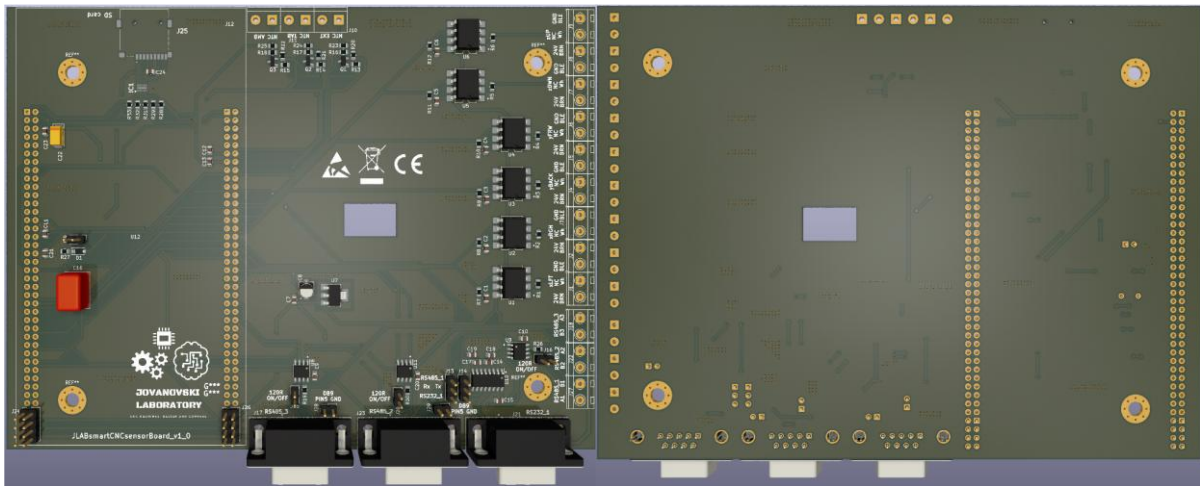
- 470 Ω кој се користи за високите температури,
- 18 k Ω кој се користи за ниските температури.

Отпорноста на NTC-сензорот кога PNP-транзисторот е исклучен се пресметува според следната формула:

$$R_{NTC}[\Omega] = \frac{V_{sen}[mV] \times 18\,000[\Omega]}{3300[mV] - V_{sen}[mV]} \quad (2.8)$$

каде V_{sen} е напонот кој се мери на влезниот аналоген пин од микроконтролерот. Во случајот кога PNP-транзисторот е вклучен отпорноста на NTC-сензорот се пресметува според друга формула дадена во продолжение:

$$R_{NTC}[\Omega] = \frac{V_{sen}[mV]}{\frac{3300 - V_{EC,SAT}[mV] - V_{sen}[mV]}{470[\Omega]} + \frac{3300 - V_{sen}[mV]}{18000[\Omega]}} \quad (2.9)$$



Слика 2.16. Изглед на електронската плочка од второто ниво на ЕУЕ.

каде $V_{EC,SAT}$ е напонот помеѓу емитерот и колекторот на транзисторот во подрачје на заситување. Микроконтролерот добиената отпорност на NTC-сензорот со помош на пребарувачка табела (lookup table) за соодветниот сензор ја конвертира во температура. Мерењето секогаш се прави и со исклучен и со вклучен PNP-транзистор, па во зависност од добиените вредности и прагот за соодветниот сензор се избира која температура ќе се земе во предвид. На слика 2.16 е даден изгледот на електронската плочка од второто ниво на ЕУЕ и тоа горната страна на левиот дел, а долната страна на десниот дел од сликата.

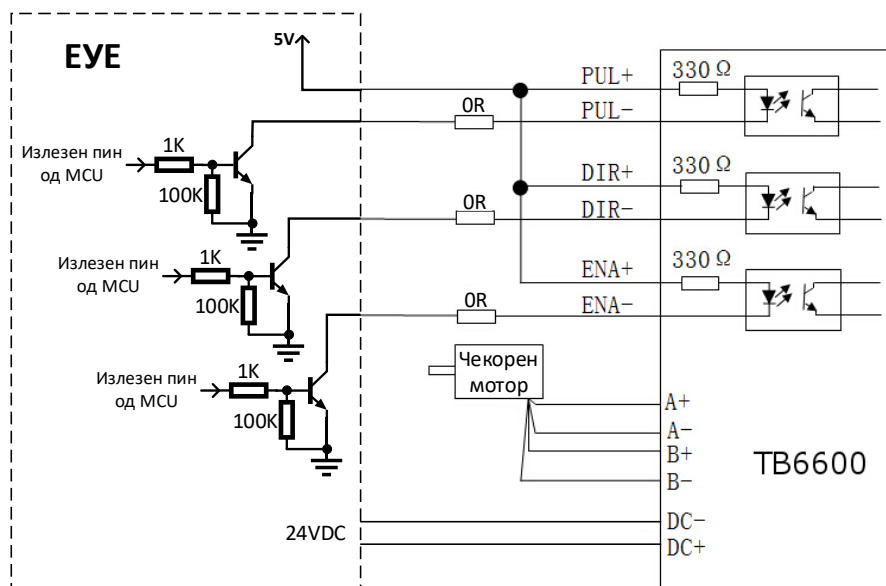
Електронската плочка на третото ниво од електронската управувачка единица (ЕУЕ) е наменета за управување на погонските сервомотори на машината и чекорниот мотор на екструдерот задолжен за повлекување на материјалот. Управувањето со моторот на спиндлата е реализирано преку сервиска комуникација како што беше опишано во поглавјето 2.5 и не бара директна обработка од оваа плочка.

Шемите на поврзување помеѓу ЕУЕ и серводрајверите беа дадени во делот 2.4. Со цел да се поедностави поврзувањето на ЕУЕ со серводрајверите, сите влезно-излезни контролни линии се извадени на стандардни DB44 конектори на горниот дел од оваа плочка, идентични со оние што се користат на драјверите. Со ваквиот пристап се овозможува директно поврзување со стандарден DB44 кабел без потреба од индивидуално поврзување на сигналните линии, што значително го олеснува инсталирањето, сервисирањето и ја намалува можноста за грешки при поврзувањето.

За управување на чекорниот мотор на екструдерот се користат три сигнални линии:

- EN (Enable) – се користи за вклучување на драјверот и активирање на моторот во закочен режим.
- DIR (Direction) – го дефинира правецот на ротација на моторот, односно дали ќе се врши екструзија на материјал или повлекување на истиот.
- PUL (Pulse) – прима низа од импулси, при што секој импулс претставува еден микрочекор на моторот.

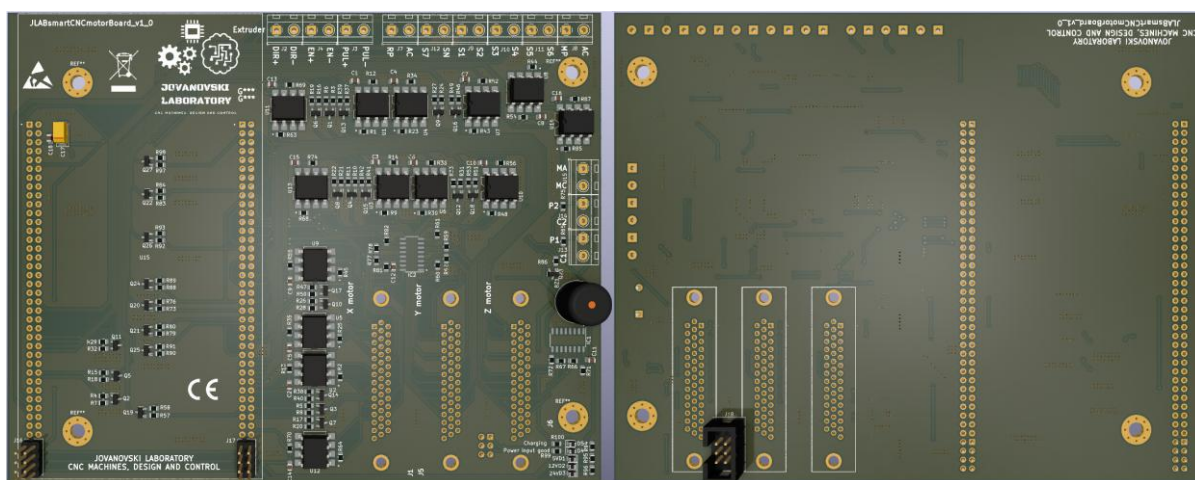
Шемата на поврзување помеѓу ЕУЕ и драјверот на чекорниот мотор на екструдерот е дадена на слика 2.17. Драјверот има оптички изолирани влезови кои ЕУЕ ги активира преку транзисторски излези. Изгледот на оваа плочка е даден на слика 2.18 и тоа горната страна на левата половина, а долната страна на десната половина од сликата.



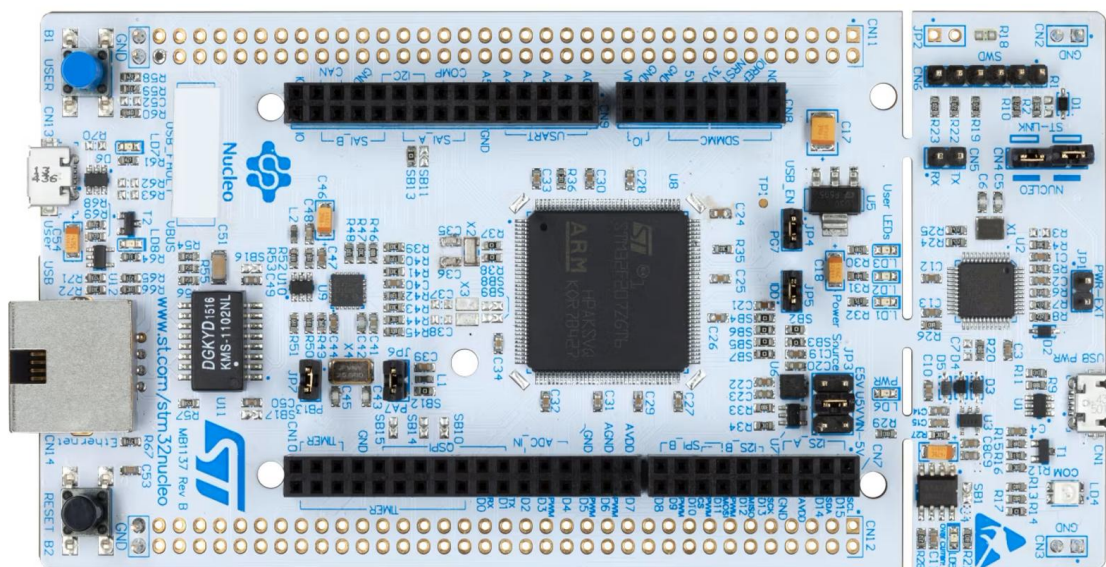
Слика 2.17. Шема на поврзување на ЕУЕ со драјверот на чекорниот мотор на екструдерот.

На четвртото, најгорно ниво од електронската управувачка единица (ЕУЕ) е сместена микроконтролерската плочка, која претставува централна пресметковна и управувачка компонента на целиот систем. За оваа намена, избрана е готовата развојна плочка NUCLEO-F767ZI од STMicroelectronics [77].

Во срцето на оваа плочка се наоѓа STM32F767ZIT6 микроконтролерот, кој припаѓа на F7 серијата на 32-битни микроконтролери базиран на ARM Cortex-M7 процесорско јадро, со работна фреквенција до 216 MHz, што овозможува високи перформанси, мала латенција и напредна обработка на сигнали и податоци. Контролерот располага со 2MB-флеш меморија, 512 KB-рам, копроцесор за работа со броеви со подвижна записка, посебни инструкции за дигитално процесирање на сигнали и голем број на влезно-излезни пинови (I/O), што го прави исклучително погоден за комплексни индустриски апликации како што е хибридната CNC-машина. Пресметковните перформанси на овој микроконтролер се од големо значење при реализацијата на напредните управувачки алгоритми [78].



Слика 2.18. Изглед на електронската плочка на третото ниво од ЕУЕ.



Слика 2.19. Микроконтролерска плочка на четвртото ниво од ЕУЕ.

Дополнително, STM32F767 поддржува широк спектар на комуникациски протоколи како USART, SPI, I2C, CAN, USB OTG, Ethernet MAC, што се од клучно значење за поврзувањето на машината со сите нејзини периферии. Големiot број на тајмери (15 тајмери) со кои располага овој микроконтролер беше една од клучните точки за избор на овој микроконтролер, бидејќи тајмерите се потребни за прецизно управување на моторите во системот.

Изборот на NUCLEO-F767ZI како готова развојна платформа беше и стратешки направен поради нејзината стабилност, достапност, сигурност и одлична интеграција со STM32Cube-екосистемот кој овозможува брз развој и одржување на firmware-от. Дополнително плочката веќе интегрира корисничка USB-порта и Ethernet-порта за LAN-поврзување, кои директно се користат во апликацијата за пренос на програмите преку USB-мемориски стик и поврзување на машината на интернет за оддалечен надзор и контрола на истата.

Со оваа архитектура, STM32F767ZIT6 микроконтролерот функционира како главен управувачки модул на сите потсистеми на хибридната CNC-машина и централен процесирачки дел во алгоритмите за напредна контрола на машината. Тој ја извршува централната логика и е одговорен за синхронизација и управување на критичните функции, вклучувајќи го прецизното управување со моторите на сите оски, контролата на екструдерот, нивната синхронизација, обработката на сензорските податоци, управувањето со корисничките интерфејси, како и комуникацијата со надворешни системи преку стандардизирани индустриски протоколи.

Благодарение на високата пресметковна моќ, интегрираниот копроцесор за работа со броеви со подвижна запирка и поддршката за реално-временски обработки, микроконтролерот овозможува паралелно и сигурно извршување на повеќе процеси, со што се гарантира стабилна, скалабилна и флексибилна архитектура на управувачкиот систем. Ова го прави системот не само прилагодлив на моменталните потреби, туку и лесен за надградување, т.е. за идни проширувања и додавање на нови функционалности. На слика 2.19 е даден изгледот на микроконтролерската електронска плочка како дел од електронската управувачка единица на четвртото ниво.

3 Софтверска имплементација на предложената хибридна CNC-машина

Софтверската имплементација претставува клучна компонента во функционалната реализација на хибридната CNC-машина, овозможувајќи прецизно управување, синхронизација и надзор на сите потсистеми. Во имплементацијата на машината реализирани се три софтверски делови: вградениот фирмвер на електронската управувачка единица (EUE), софтверот на HMI-дисплејот и компјутерскиот софтвер за управување на машината и пуштање на програми од персонален компјутер.

Во продолжение на ова поглавје детално се разработуваат сите три софтверски компоненти, нивната архитектура, алгоритми и интеракција со хардверските модули, со цел целосно разбирање на логиката и интелигенцијата што стои зад управувањето на хибридната CNC-машина.

3.1 Имплементација на фирмвер

Фирмверот, кој го управува главниот микроконтролер на EUE, претставува јадрото на целокупниот управувачки систем. Во него се интегрирани напредни алгоритми за интерполација на движењата, управување со моторите, регулација на температурите, читање и обработка на сигналите од крајните сензори, како и комуникација со сите периферни уреди. Покрај тоа, фирмверот овозможува и поддршка за различни индустриски комуникациски протоколи. Со цел да се одговори на барањата за висока прецизност, стабилност и брзина, фирмверот е развиен како реално-временски систем, овозможувајќи сигурно и предвидливо однесување при паралелна обработка на повеќе задачи во реално време.

Акцентот во оваа докторска теза е ставен на развојот на напредни контролни алгоритми за управување на хибридни CNC-машини, особено во делот на имплементацијата на движењата, интерполациите помеѓу точките добиени од G-кодот и управувањето со различните профили на брзини. Секоја CNC-обработка или 3D-печатење се сведува на интерполации помеѓу зададени точки од G-кодот добиен како излез од софтверите за раслојување на 3D-моделите кои се обработуваат или печатат. Имплементацијата на интерполациите може да се изведе едноставно ако се работи со константни брзини, но инерцијата на системот не го дозволува тоа и потребно е сложено управување и со профилите на самите брзини. Така целата проблематика станува покомплексна, особено што треба да се изведе прецизно и синхронизирано движење на сите мотори во системот и при тоа прецизно да се управува со брзините.

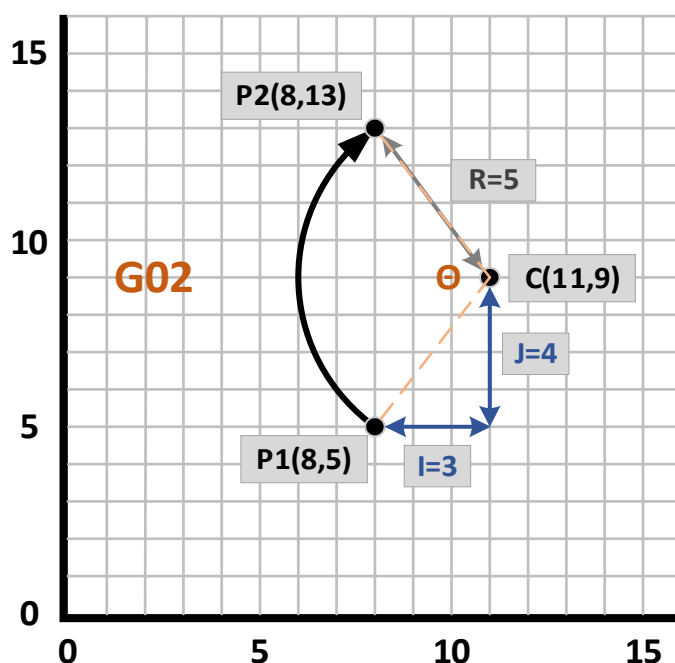
Резолуцијата на машината го дефинира просторот за обработка како дискретно множество од точки. Линеарните и кружни интерполации се изведуваат во даденото множество од точки во просторот на машината. Фирмверот треба реално временски да ги пресметува брзините и патеките на движење на главата на машината во дискретниот простор за обработка и печатење. Пред да се пристапи на имплементација на фирмверот за JLABsmartCNC-хибридната машина беше направена детална анализа на достапните отворени и слободни фирмвери кои се користат за оваа намена. Прво направена е детална анализа на GRBL-фирмверот, првично наменет за управување на CNC-машини и имплементиран на хардвер со ограничени ресурси, но подоцна користен како основа за сите денешни 3D-печатари и CNC-машини. Следно е направена детална анализа на Клипер фирмверот кој го користат најновите генерации на 3D-печатари и кој направи револуција во оваа технологија правејќи значителна дистинкција со претходните генерации на 3D-печатари како во квалитетот, така и во големата брзина на печатење.

3.1.1 GRBL слободен и отворен софтвер за прецизно управување на CNC-машини

GRBL претставува слободен и отворен софтвер/фирмвер за прецизно управување на CNC-машини кој за прв пат се појавува во 2009 година. Првично развиен за работа на Atmega328p осум битните микроконтролери набрзо станува број еден избор на хобистите, производителите и истражувачите во оваа област кои сакаат достапен и флексибилен начин за контрола на нивните CNC-машини. Во основа GRBL ги трансформира дигиталните инструкции добиени од компјутерот во вид на G-код команди во прецизни движења на моторите на машината овозможувајќи прецизна обработка, гравирање и сечење на различни типови на материјали. Денес, развојот на GRBL продолжува преку неговиот наследник GRBL HAL, кој е наменет за помодерни и помоќни микроконтролерски платформи, овозможувајќи повисоки перформанси, повеќе оски, подобра комуникација и проширени функционалности.

Вистинската моќ на овој фирмвер е во тоа што е отворен и слободен софтвер лиценциран под GPLv3 генералната јавна лиценца што му овозможува на пошироката заедница на корисници и инженери да придонесат за неговото подобрување и развој [79]. Токму ова го прави GRBL основа за развој на многуте отворени и слободни фирмвери за управување на 3D-печатари. Најголемиот дел од комерцијалните 3D-печатари денес користат фирмвер/софтвер базиран на GRBL.

Кружната интерполација претставува движење на главата на машината по кружен лак од тековната позиција P1 до дестинациската координата P2 дадена во G-код командата. Движењето исто како кај линеарната интерполација се изведува со предефинирана брзина. G-код командите за кружна интерполација се: G02 во насока иста со насоката на стрелките на часовникот и G03 во насока обратна од насоката на стрелките на часовникот. Кружна интерполација G02 од точка P1 со координати (8,5) до точка P2 со координати (8,13) е дадена на слика 3.1. Овие команди за кружна интерполација може да имаат две синтаксички верзии:



Слика 3.1. Кружна интерполација.

- Верзија 1 – со користење на радиусот на кружниот лак како параметар:

G02/G03 X8.0 Y13.0 R5.0 F100.0

- **X, Y** – координати на дестинациската точка на движењето.
- **R** – радиус на кружниот лак.
- **F** – брзина со која главата на машината треба да се движи од тековната до дестинациската точка.

- Верзија 2 – со користење на параметрите: I, J, K:

G02/G03 X8.0 Y13.0 I3.0 J4.0 F100.0

- **X, Y** – координати на дестинациската точка на движењето.
- **I, J** – координати на центарот на кружниот лак релативни во однос на почетната точка P1.
- **F** – брзина со која главата на машината треба да се движи од тековната до дестинациската точка.
- Радиусот на кружниот лак може да се пресмета со помош на I и J.

Кружните интерполации кај GRBL се имплементирани преку сегментација на кружниот лак во мали линеарни сегменти. Кај овој фирмвер се користи I, J репрезентацијата на G-код командите за кружна интерполација. Дополнително GRBL имплементира и спирална интерполација која е често користена за правење на навој кај CNC-машините. За поедноставување ќе биде разгледана имплементацијата на кружната интерполација на XY-рамнината, но истиот алгоритам работи и за XZ и YZ-рамнините. Нека **P1(X₁,Y₁)** е тековната позиција на главата на машината, а **P2(X₂,Y₂)** е дестинациската точка добиена од G-код командата и нека **I, J** се координатите на центарот на кружниот лак релативни во однос на почетната точка. Алгоритмот започнува со пресметување на координатите на центарот:

$$X_c = X_1 + I, \quad Y_c = Y_1 + J \quad (3.1)$$

Следно се пресметуваат координатите на почетната и на дестинациската точка релативно во однос на центарот:

$$r_x = -I, \quad r_y = -J \quad (3.2)$$

$$r_{tx} = X_2 - X_c, \quad r_{ty} = Y_2 - Y_c \quad (3.3)$$

Радиусот на кружниот лак може да се пресмета од релативните координати I, J со користење на следниот израз:

$$R = \sqrt{I^2 + J^2} \quad (3.4)$$

За одредување на бројот на сегменти на дадениот кружен лак потребно е да се најде аголот θ кој го формираат почетната точка P1 центарот на лакот C и крајната точка P2. Овој агол се одредува со помош на математичката функција **atan2** која е инверзна тангенс функција со два аргументи. По дефиниција $\theta = \text{atan2}(y, x)$ е аголот помеѓу позитивниот дел на X-оската и правата од центарот на координатниот систем до точката со координати (x, y). Стандардната инверзна тангенс операција со еден аргумент враќа агол од $-\frac{1}{2}\pi < \theta \leq \frac{1}{2}\pi$, додека пак **atan2** враќа агол во опсег од $-\pi < \theta \leq \pi$. Разликата помеѓу овие две функции е дадена на слика 3.2, а дефиницијата на **atan2** преку стандардната инверзна тангенс операција е дадена во продолжение:

$$\text{atan2}(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right), & \text{за } x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi, & \text{за } x < 0 \text{ и } y \geq 0 \\ \arctan\left(\frac{y}{x}\right) - \pi, & \text{за } x < 0 \text{ и } y < 0 \\ +\frac{\pi}{2}, & \text{за } x = 0 \text{ и } y > 0 \\ -\frac{\pi}{2}, & \text{за } x = 0 \text{ и } y < 0 \\ \text{недефинирано}, & \text{за } x = 0 \text{ и } y = 0 \end{cases} \quad (3.5)$$

За пресметување на бараниот агол Θ од слика 3.1 се користат релативните координати на крајната точка P2 и почетната точка P1 во однос на центарот C. Од аголот помеѓу X-оската и правата CP2 се одземе аголот помеѓу X-оската и правата CP1 (X-оската е поместена во центарот C).

$$\Theta = \text{atan2}(r_{ty}, r_{tx}) - \text{atan2}(r_y, r_x) \quad (3.6)$$

Со примена на тригонометрискиот идентитет:

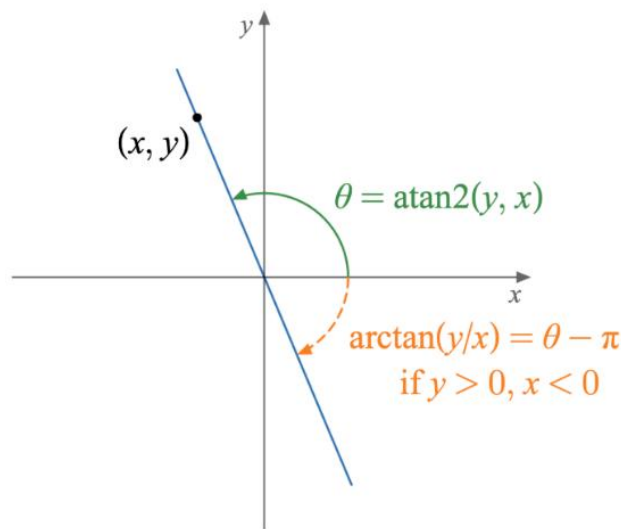
$$\text{atan2}(y_A, x_A) \pm \text{atan2}(y_B, x_B) = \text{atan2}(y_A x_B \pm x_A y_B, x_A x_B \mp y_A y_B) \quad (3.7)$$

во изразот (3.6) се добива:

$$\Theta = \text{atan2}(r_{ty}, r_{tx}) - \text{atan2}(r_y, r_x) = \text{atan2}(r_{ty} r_{tx} - r_{tx} r_y, r_x r_{tx} + r_y r_{ty}) \quad (3.8)$$

и конечно со користење на изведениот израз од (3.8) алгоритмот го пресметува аголот Θ (со изразот 3.8 се добива аголот во опсег $-\pi$ до π).

За примерот од слика 1 каде се дадени конкретни координати за точките P1(8,5), P2(8,13) и релативните координати I = 3, J = 4 на центарот во однос на почетната точка P1 пресметките на алгоритмот за наоѓање на аголот Θ се дадени во продолжение:



Слика 3.2. Разлика помеѓу функциите atan2 и arctan.

$$X_C = X_1 + I = 8 + 3 = 11, \quad Y_C = Y_1 + J = 5 + 4 = 9 \quad (3.9)$$

па така координатите на центарот се C(11,9). Следно алгоритмот ги пресметува релативните координати на почетната и крајната точка:

$$r_x = -I = -3, \quad r_y = -J = -4 \quad (3.10)$$

$$r_{tx} = X_2 - X_C = 8 - 11 = -3, \quad r_{ty} = Y_2 - Y_C = 13 - 9 = 4 \quad (3.11)$$

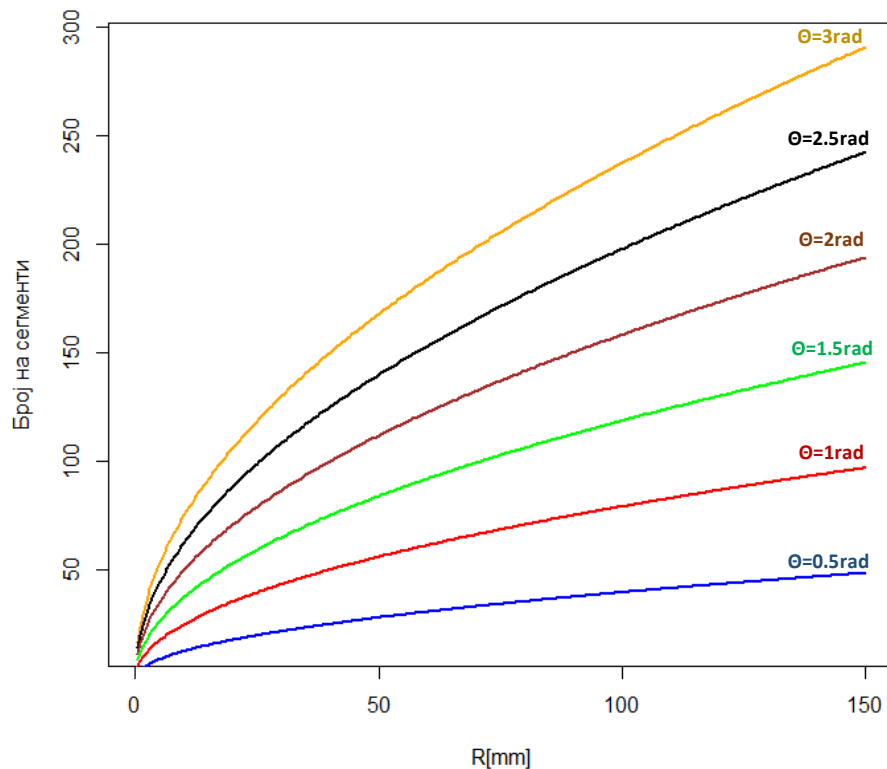
и од релативните координати се пресметува аголот Θ :

$$\Theta = \text{atan2}(r_{ty}r_x - r_{tx}r_y, r_xr_{tx} + r_yr_{ty}) = \text{atan2}(-24, -7) = -1,85\text{rad} = -106^\circ \quad (3.12)$$

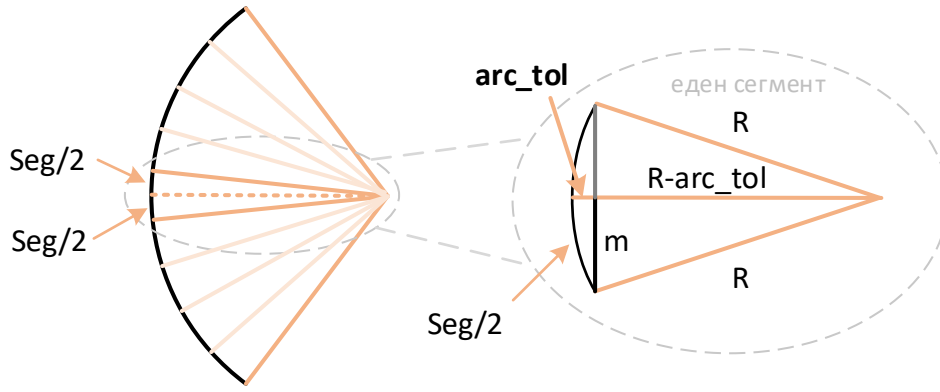
Следно GRBL го пресметува бројот на сегменти на кои треба да се подели кружниот лак според следниот израз:

$$\text{segments} = \frac{\left\lceil \frac{|\Theta R|}{2} \right\rceil}{\sqrt{\text{arctolerance}(2R - \text{arctolerance})}} \quad (3.13)$$

и притоа секогаш го заокружува добиениот резултат на помалиот цел број. Точките од сегментите се наоѓаат на самиот лак, а со линеарната интерполација меѓу нив се прави грешка т.е. намалување на дијаметарот до $2 \times \text{arctolerance}$ која претставува константа дефинирана во кодот. Стандардната вредност на оваа константа е 0.002mm, но корисникот може да ја менува во зависност од потребите. Зависноста на бројот на сегменти од големината на радиусот R за различни агли Θ е дадена на слика 3.3.



Слика 3.3. Зависност на бројот на сегменти од радиусот на лакот.



Слика 3.4. Одредување на бројот на сегменти.

Броителот од изразот (3.13) претставува половина од должината на кружниот лак кој треба да го помине главата на CNC-машината, а се добива со следното изведување:

$$\frac{2\pi R \times \frac{\theta}{2\pi}}{2} = \frac{\theta R}{2} \quad (3.14)$$

следствено на тоа именителот од изразот (3.13) треба да претставува половина од должината на сегментот на кој се дели кружниот лак, но тука се зема должината m од слика 3.4 што секако е помала во однос на половината од должината на сегментот, па со тоа бројот на сегменти ќе биде незначително поголем и грешката на лакот ќе се задржи во дефинираните граници со $arc_{tolerance}$. Во продолжение е дадено изведувањето на изразот за пресметување на m :

$$m = \sqrt{R^2 - (R - arc_{tolerance})^2} = \sqrt{arc_{tolerance}(2R - arc_{tolerance})} \quad (3.15)$$

Со најдениот број на сегменти GRBL го пресметува аголот на секој сегмент кој воедно претставува и чекорот на сегментацијата:

$$\theta_{seg} = \frac{\theta}{segments} \quad (3.16)$$

и дополнително се пресметува сегментот за линеарното поместување на третата оска (во конкретниот пример оската Z) ако станува збор за спирална интерполација:

$$Z_{seg} = \frac{\Delta Z}{segments} \quad (3.17)$$

Спиралната интерполација може да се изведува и со кружна интерполација во рамнината XZ со линеарна интерполација по Y-оската или кружна интерполација на рамнината YZ со линеарна интерполација по X-оската.

Генерирањето на сегментите се прави со векторска ротација со трансформациона матрица, каде центарот на лакот претставува оската на ротација, а радиус-векторот е дефиниран од центарот на лакот до почетната точка на лакот $\overrightarrow{CP1}$. Секој линеарен сегмент се добива со сукцесивна векторска ротација за агол θ_{seg} . Трансформационата матрица за ротација е дадена во продолжение:

$$RM = \begin{bmatrix} \cos(\theta_{seg}) & -\sin(\theta_{seg}) \\ \sin(\theta_{seg}) & \cos(\theta_{seg}) \end{bmatrix} \quad (3.18)$$

Ротацијата се изведува со фиксен агол, односно чекор, па пресметувањето на синус и косинус операциите се прави еднаш на почетокот пред генерирањето на сегментите со цел да се избегне нивното пресметување повеќе пати. Овие операции особено за осум битните микроконтролери за кои беше првично наменет овој фирмвер се сложени операции и земаат големо процесорско време, па затоа целта е да се минимизира нивното користење. Дополнително на тоа бидејќи θ_{seg} претставува мал агол се користи апроксимација за пресметка на тригонометриските функции на мали агли, па така пресметувањето на овие операции е изведено според изразите во продолжение:

$$\cos(\theta_{seg}) = 1 - \frac{\theta_{seg}^2}{2} \quad (3.19)$$

$$\sin(\theta_{seg}) = \theta_{seg} - \frac{\theta_{seg}^3}{6} \quad (3.20)$$

Со ова дополнително се оптимизира и скратува времето на пресметка и се избегнуваат сложени операции за осум битните микроконтролери. Со множење на тековната координата запишана во матрица со трансформационата матрица од (3.18) се добива следната координата односно следниот линеарен сегмент:

$$\begin{bmatrix} r_{x(i+1)} \\ r_{y(i+1)} \end{bmatrix} = \begin{bmatrix} r_{xi} \\ r_{yi} \end{bmatrix} \times \begin{bmatrix} \cos(\theta_{seg}) & -\sin(\theta_{seg}) \\ \sin(\theta_{seg}) & \cos(\theta_{seg}) \end{bmatrix} \quad (3.21)$$

GRBL имплементира и корекција на грешката што би се акумулирала од тригонометриската апроксимација, така што на секој 12 сегменти (константа која може да се менува од страна на корисникот) пресметувањето на следниот сегмент се прави со примена на трансформационата матрица на иницијалниот радиус-вектор со егзактно пресметување на тригонометриските функции за аголот $i \times \theta_{seg}$. Ова се прави со цел да се заштеди на пресметување на тригонометриските функции при секоја итерација, што кај микроконтролерите со ограничени ресурси (особено кај осум битните) овие функции бараат значително процесорско време. Генерираните сегменти се проследуваат за понатамошна обработка. Во случај кога има спирална интерполација она што се проследува за понатамошна обработка за третата линеарна оска е:

$$Z_{new} = Z_{old} + Z_{seg} \quad (3.22)$$

За примерот од слика 3.1 со дадените конкретни координати и со користење на иницијалната вредност за $arc_{tolerance}$ за бројот на сегменти се добива:

$$segments = \frac{\left| \frac{-1,85 \times 5}{2} \right|}{\sqrt{0.002(2 \times 5 - 0.002)}} = 32 \quad (3.23)$$

За аголот на секој сегмент се добива:

$$\theta_{seg} = \frac{-1,85}{32} = -0,0578125 = -3,3^\circ \quad (3.24)$$

Табела 3.1. Координати на сегментите за дадениот пример од слика 3.1.

S1-S8		S9-S16		S17-S24		S25-S32	
X	Y	X	Y	X	Y	X	Y
7,773893	5,180024	6,408422	7,020793	6,007731	9,277418	6,656006	11,47574
7,558566	5,372812	6,301736	7,289401	6,032102	9,565408	6,806313	11,7226
7,354737	5,577719	6,210749	7,563726	6,073073	9,851508	6,970633	11,96036
7,163089	5,794061	6,135764	7,842849	6,130506	10,13476	7,148416	12,18823
6,984261	6,021114	6,077031	8,125838	6,204209	10,41423	7,339068	12,40545
6,818852	6,258122	6,034749	8,411748	6,293937	10,68896	7,541952	12,61129
6,667413	6,50429	6,009056	8,699623	6,399389	10,95806	7,75639	12,80506
6,53045	6,758797	6,00004	8,988501	6,520214	11,22061	7,981667	12,98612

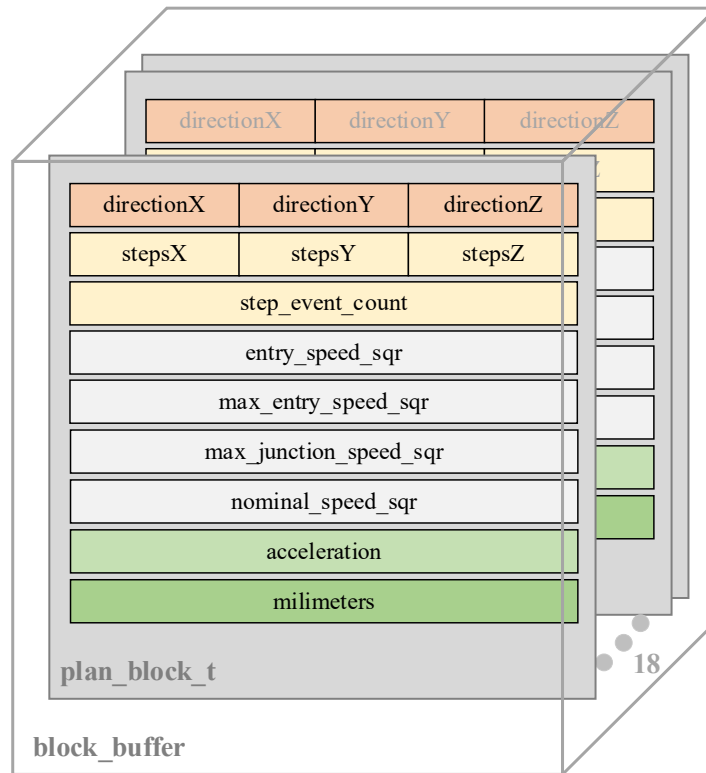
Апроксимативната пресметка за тригонометриските функции за аголот θ_{seg} е дадена во продолжение:

$$\cos(\theta_{seg}) = 1 - \frac{(-0,0578125)^2}{2} = 0,9983289 \quad (3.25)$$

$$\sin(\theta_{seg}) = (-0,0578125) - \frac{(-0,0578125)^3}{6} = -0,0577803 \quad (3.26)$$

Генерираните точки од алгоритмот за кружна интерполација за примерот од слика 3.1 се дадени во табела 3.1. Генерираните сегменти од кружната интерполација и линеарните сегменти добиени од G-код командите се проследуваат на GRBL-планерот за понатамошна обработка. GRBL-планерот користи бафер од 18 елементи во кој ги запишува обработените линеарни сегменти (блокови). Секој елемент од овој бафер претставува структура во која се чуваат повеќе кинематички параметри за тековниот сегмент опишани во продолжение :

- **Direction X, Y, Z** – запишани се насоките на секоја од оските за тековниот сегмент/блок.
- **Steps X, Y, Z** – се чуваат бројот на чекори кои треба да ги помине секоја од оските за да се реализира движењето на тековниот сегмент.
- **Step_event_count** – претставува максималниот број на чекори од оските, ако StepsX има најголема вредност таа вредност ќе се зачува и во оваа променлива. Максималниот број на чекори се користи од Бресенхам (Bresenham) алгоритмот.
- **Entry_speed_sqr** – ја чува тековно пресметаната почетна брзина за тековниот сегмент/блок.
- **Max_entry_speed_sqr** – претставува максималната можна почетна брзина на тековниот сегмент добиена од номиналните брзини на соседните сегменти и максималната можна брзина во точката на спојување на истите.
- **Max_junction_speed_sqr** – максималната брзина добиена врз основа на насоките на векторите на брзините на соседните сегменти.
- **Nominal_speed_sqr** – номиналната брзина за тековниот сегмент која се добива од G-код командата.
- **Acceleration** – пресметаното линиско забрзување.
- **Milimeters** – останатите милиметри од блокот кој треба да се поминат за да се комплетира движењето по тековниот блок.



Слика 3.5. Мемориска структура на баферот кој го користи GRBL-планерот.

Во структурата се чуваат квадратите на брзините и тоа во единица мерка $(mm/min)^2$ со цел оптимално користење на брзините во кинематичките изрази и намалување на бројот на пресметки. Забрзувањето се чува во единица мерка mm/min^2 , а растојанието во mm . Мемориската структура на баферот кој го користи GRBL-планерот е дадена на слика 3.5.

Со сегментацијата на кружните движења во линеарни, целото движење на главата на CNC-машината се сведува на линеарна интерполација од точка до точка. GRBL може да се користи и за повеќе оски тука ќе биде разгледана интерполацијата од точка до точка за машина со 3 оски. На слика 3.6 е дадена линеарна интерполација од точката $P1(x_1, y_1, z_1)$ до точката $P2(x_2, y_2, z_2)$. Прво се пресметуваат растојанијата што треба да ги поминат сите оски за тековниот сегмент:

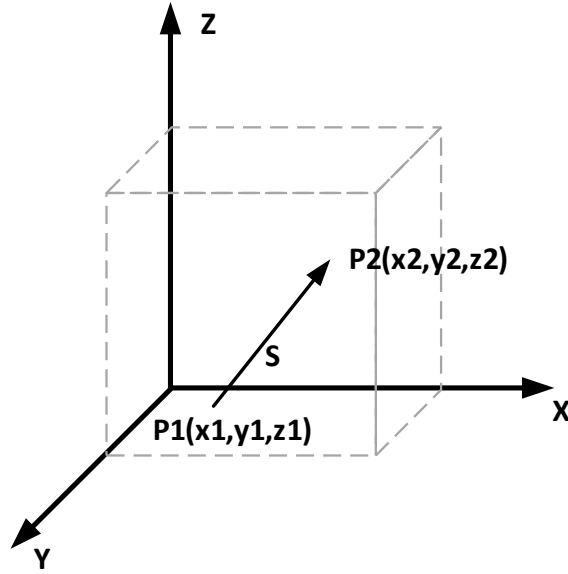
$$dx = x_2 - x_1, \quad dy = y_2 - y_1, \quad dz = z_2 - z_1 \quad (3.27)$$

Во зависност од знакот на пресметаните растојанија се ажурираат насоките на движење на секоја од оските. За да се добие бројот на чекори за секоја оска пресметаните растојанија се множат со константа која претставува бројот на чекори во еден милиметар за дадената машина (резолюција на машината):

$$stepsX = dx \times steps_per_mm \quad (3.28)$$

$$stepsY = dy \times steps_per_mm \quad (3.29)$$

$$stepsZ = dz \times steps_per_mm \quad (3.30)$$



Слика 3.6. Линеарна интерполација од точка P1 до точка P2.

Максимумот од бројот на чекори од трите оски се запишува во `step_event_count` променливата во баферот. Следно се пресметува растојанието во милиметри што треба да го помине главата на CNC-машината од почетната до дестинациската точка со линеарна интерполација:

$$milimeters = S = \sqrt{dx^2 + dy^2 + dz^2} \quad (3.31)$$

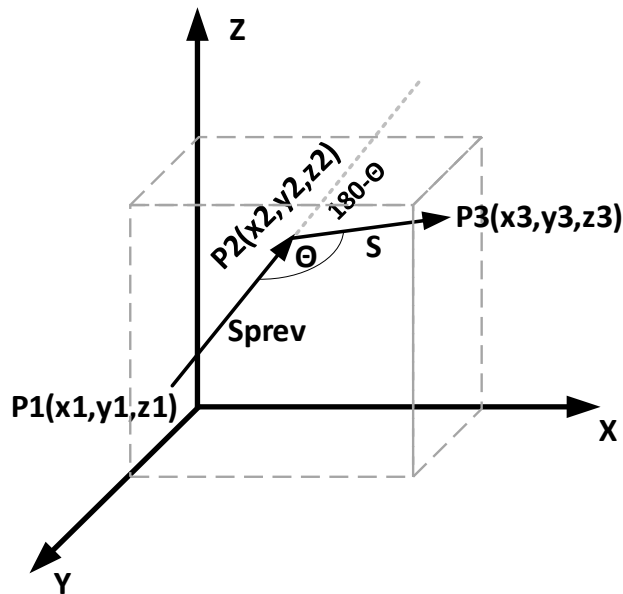
Имплементацијата на движењата би била едноставна ако брзината на движење е константна по должина на секој блок. Инерцијата на машината не го дозволува тоа, па мора да се користат различни профили на забрзување и успорување. Нека v е брзината која се добива од G-кодот за тековниот сегмент, а v_{xmax} , v_{ymax} и v_{zmax} се максималните дозволени брзини за секоја од оските дефинирани од механиката на системот и моторите кои се користат за нив. GRBL прави ограничување на номиналната брзина со цел да не се надмине максималната брзина на некоја од оските, па така пресметувањето на номиналната брзина е дадено во продолжение:

$$v_{nom} = \min\left(v, v_{xmax} \times \frac{S}{dx}, v_{ymax} \times \frac{S}{dy}, v_{zmax} \times \frac{S}{dz}\right), \text{ кога } dx, dy, dz \neq 0 \quad (3.32)$$

За појаснување на изразот (3.32) за да се добие брзина на движење v на главата на машината по патеката S оските треба да се движат со следните брзини:

$$v_x = v \frac{dx}{S}, \quad v_y = v \frac{dy}{S}, \quad v_z = v \frac{dz}{S} \quad (3.33)$$

На сличен начин се прави ограничување на забрзувањето по сите оски, со цел да не се надмине максималното можно забрзување на секоја од оските дефинирано од механичката конструкција и моторите на машината. Нека a_{xmax} , a_{ymax} и a_{zmax} се максималните можни забрзувања дефинирани за машината, GRBL го пресметува номиналното забрзување според следниот израз:



Слика 3.7. Одредување на аголот помеѓу два сегменти на движење.

$$a_{nom} = \min \left(a_{xmax} \times \frac{S}{dx}, a_{ymax} \times \frac{S}{dy}, a_{zmax} \times \frac{S}{dz} \right), \text{ кога } dx, dy, dz \neq 0 \quad (3.34)$$

Аголот помеѓу векторот на претходната патека и векторот на тековната патека е клучен при одредување на брзините на движење. Големата промена на правецот и насоката на движење на главата на машината ја нарушува инерцијата на системот, па брзината на премин треба да биде помала, односно соодветна со аголот помеѓу двата вектори. GRBL го пресметува косинусот од аголот помеѓу двата вектори како скаларен производ на нивните единечни вектори според следниот тригонометриски израз:

$$\cos \theta = - \frac{dx_{prev} \times dx + dy_{prev} \times dy + dz_{prev} \times dz}{S_{prev} \times S} \quad (3.35)$$

каде dx_{prev} , dy_{prev} и dz_{prev} се проекциите на претходниот сегмент на X , Y и Z -оските соодветно, а S_{prev} е должината на претходниот сегмент. На слика 3.7 се дадени два сегменти $\overrightarrow{P_1P_2}$ како претходен сегмент и $\overrightarrow{P_2P_3}$ како тековен сегмент. Во скаларниот производ на векторите првиот вектор се зема со спротивна насока со цел да се добие аголот θ , а не аголот $180^\circ - \theta$, оттука во изразот (3.35) се јавува знакот минус пред дробната црта.

Наоѓањето на аголот помеѓу сегментите е клучен дел за дефинирање на максималната можна брзина при премин помеѓу сегментите. Во GRBL се претпоставува дека во спојните точки на сегментите брзината е константна и тука се јавува само центрипетално забрзување. Во спојната точка се впишува кружница со радиус R , така што двата сегменти се тангенти на кружницата прикажано на слика 3.8. Кружниот сегмент кои ги спојува двата линеарни сегменти ја претставува апроксимираната патеката со константно центрипетално забрзување. Максималната брзина v_{max_j} со која може да се направи преминот помеѓу сегментите, а притоа да не се надмине максималното дозволено забрзување a_{max} за дадена CNC-машина е:

$$v_{\max_j} = \sqrt{a_{\max} \times R} \quad (3.36)$$

Кружниот сегмент креира виртуелна девијација δ која е дефинирана како најкраткото растојание помеѓу спојната точка на реалната патека и кружниот сегмент. Овој параметар се задава како константа од страна на корисникот, што индиректно го дефинира радиусот на кружницата, а оттаму и ја лимитира брзината на главата на машината во спојната точка. Од слика 3.8 може да видиме дека кружници со различни радиуси можат да бидат впишани помеѓу сегментите, но параметарот δ еднозначно ја определува кружницата. За ефикасно определување на радиусот на впишаната кружница се разгледува сивиот правоаголен триаголник означен на слика 3.8, па оттука се добива:

$$\sin \frac{\theta}{2} = \frac{R}{R + \delta} \quad (3.37)$$

оттука може да се изрази радиусот:

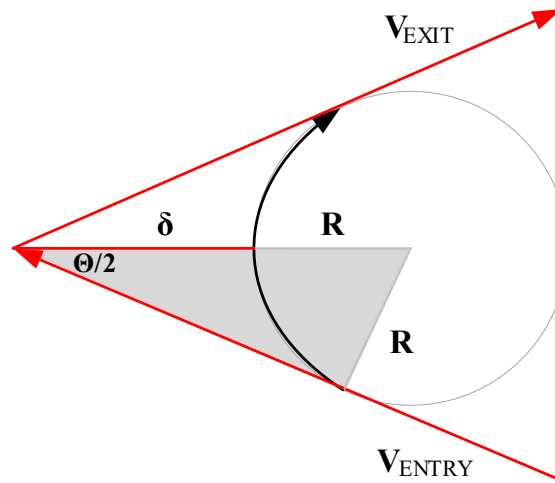
$$R = \delta \frac{\sin \frac{\theta}{2}}{1 - \sin \frac{\theta}{2}} \quad (3.38)$$

Со изразот (3.35) е најден $\cos \theta$, па за избегнување на сложените операции $\arccos$ и $\sin$ се користи тригонометрискиот идентитет:

$$\sin \frac{\theta}{2} = \pm \sqrt{\frac{1 - \cos \theta}{2}} \quad (3.39)$$

Ако (3.38) се замени во (3.36) се добива конечниот израз за квадратот на максималната брзина во спојните точки:

$$v_{\max_j}^2 = \frac{a_{\max} \times \delta \times \sin \frac{\theta}{2}}{1 - \sin \frac{\theta}{2}} \quad (3.40)$$



Слика 3.8. Одредување на максимална брзина при премин помеѓу два сегменти.

Овој пристап е оптимален во однос на минимизација на потребните ресурси, така што потребната процесирачка моќ е сведена на минимум, а во исто време се земаат во предвид нелинеарноста во однос на оштрината на аголот помеѓу соседните линеарни сегменти и според тоа се пресметува максималната можна брзина во секоја од спојните точки. Ова значи дека за прави агли помеѓу соседните сегменти и агли при кои соседните сегменти се движат во спротивна насока брзината во спојната точка е приближно нула. Додека за спојни точки во кои промената на правецот е многу мала или имаме приближно права линија, вака пресметаната максимална брзина е приближна со номиналната брзина. Тука мора да се напомене дека овој метод се базира на виртуелна девијација на патеката на главата на машината која се користи само за робусно пресметување на максималната брзина во спојните точки, но ова никако не е физичка девијација на патеката на движење на главата на машината.

Конечно, квадратот на максималната влезна брзина за тековниот сегмент се пресметува според следниот израз:

$$v_{\max_entry}^2 = \min(v_{\max_j}^2, v_{nom}^2, v_{nom_prev}^2) \quad (3.41)$$

како минимална брзина од квадратот на номиналната брзина на тековниот сегмент, на претходниот сегмент и квадратот на максималната можна брзина во точката на спојување на двата сегменти. Ова значи дека максималната влезна брзина на сегментот не може да биде поголема од номиналните брзини на тековниот и претходниот сегмент и пресметаната максимална брзина во спојната точка.

Вака пресметаната максимална влезна брзина е брзината со која може да се помине дадената спојна точка без да се надмине максималното можно забрзување дефинирано од механиката на машината и погонските мотори, но тука не се земени во предвид должините на соседните линеарни сегменти со кои се дефинира максималната брзина која може да ја постигне машината по нивната должина. Тоа значи дека брзината во спојните точки не зависи само од аголот меѓу соседните сегменти туку и од должината на сегментите. За да се земе во предвид и должината на соседните сегменти и да се пресмета оптималната (максималната) можна влезна брзина на секој сегмент мора да се разгледуваат сите сегменти во баферот.

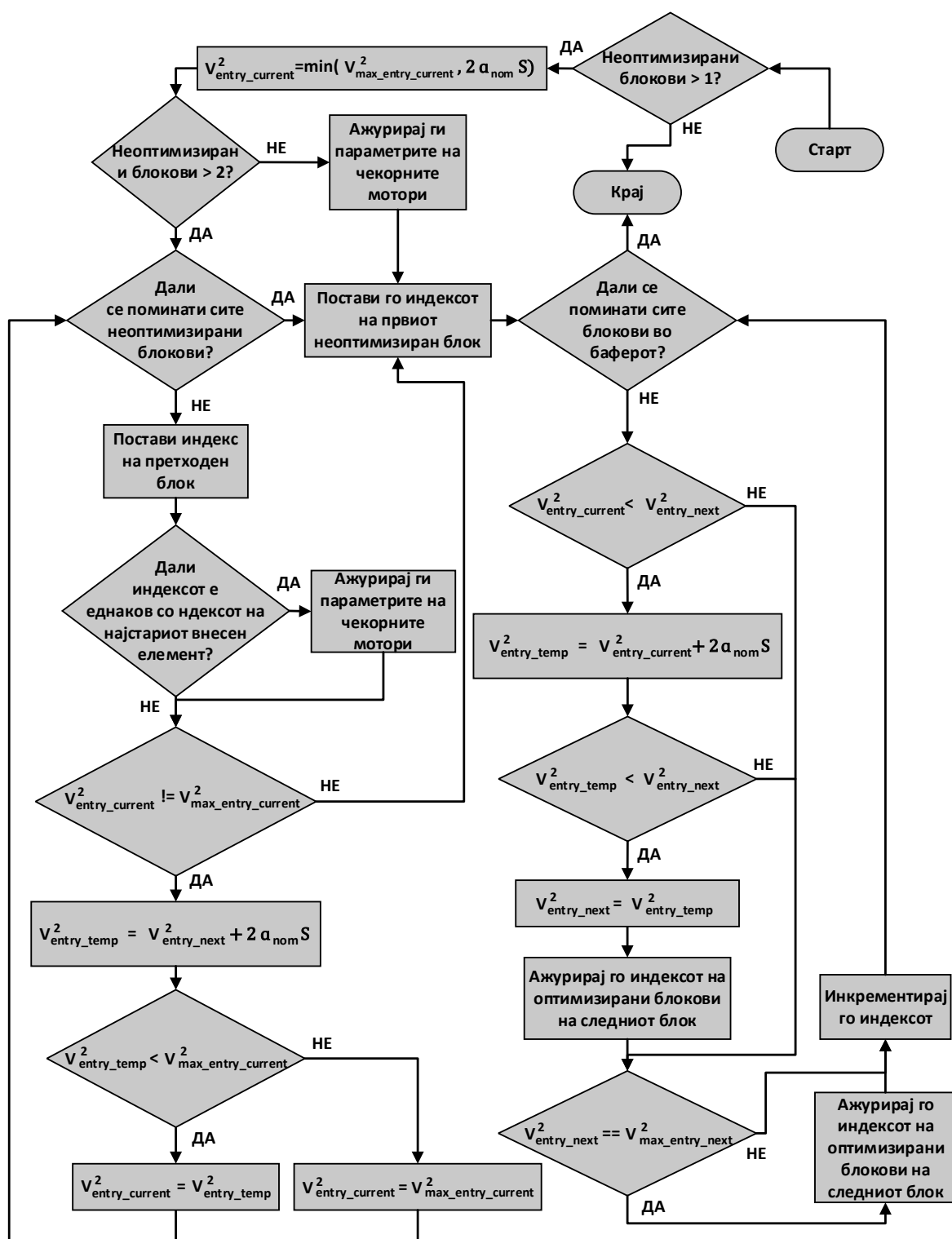
Алгоритамот според кој се пресметува влезната брзина на секој сегмент е даден на слика 3.9 и се извршува при секое додавање на нов елемент во баферот. Алгоритамот секогаш ја поставува крајната брзина на последниот сегмент на нула. Се изведуваат две итерации за поставување на оптималната влезна брзина. Првата итерација започнува од крајот на баферот и ги изминува сите блокови во обратен редослед од последниот кон првиот. Поставувањето на влезната брзина за секој блок се прави според изразот:

$$v_{entry_current}^2 = v_{entry_next}^2 + 2a_{nom}S \quad (3.42)$$

каде $v_{entry_current}$ е оптималната брзина на тековниот блок, v_{entry_next} е оптималната брзина претходно пресметана за следниот блок (итерацијата е од крај кон почеток), a_{nom} е номиналното забрзување за тековниот сегмент и S е должината на тековниот сегмент. Нормално секогаш се споредува оваа брзина да не биде поголема од брзината $v_{\max_entry_current}$ која зависи од аголот помеѓу соседните сегменти. Оваа итерација се однесува на успорувањето затоа се почнува од крајот кон почетокот со цел да може да се овозможи доволно растојание за главата на машината да забави со соодветното забрзување до нула на крајот од последниот блок. Втората итерација е од почетокот кон

крајот на баферот и при оваа итерација алгоритмот проверува дали влезната брзина на тековниот сегмент е помала во однос на влезната брзина на следниот сегмент (дали има забрзување во тековниот сегмент). Ако овој услов е исполнет се пресметува брзината на следниот сегмент според следниот израз:

$$v_{entry_next}^2 = v_{entry_current}^2 + 2a_{nom}S \quad (3.43)$$



Слика 3.9. Алгоритам за пресметка на оптималните брзини на блоковите во баферот.

Добиената брзина се споредува со веќе поставената брзина на следниот блок при првата итерација. Ако новата пресметана вредност е помала се става како оптимална влезна брзина на следниот сегмент/блок. Во овој случај се ажурира и индексот на оптимизирани блокови на следниот блок со цел да не се поминува повторно при нов внес на нов сегмент бидејќи нема што повеќе да се оптимизира, максималната влезна брзина е ограничена од должината на претходниот сегмент. Ажурирањето на индексот на оптимизирани блокови се прави и кога влезната брзина на тековниот блок е поголема во однос на влезната брзина на следниот блок и влезната брзина на следниот блок е еднаква на $v_{\max_entry_next}$.

Во планер баферот се пресметани само оптималните влезни брзини за секој блок, но овој алгоритам не ги пресметува внатрешните профили на брзините. Овие внатрешни профили на брзините се пресметуваат ад-хок од страна на алгоритмот за чекорните мотори и се состојат од само 7 можни типови и тоа:

- Блок со константна брзина;
- Блок со константна брзина и забавување;
- Блок со забрзување и константна брзина;
- Блок само со забрзување;
- Блок само со забавување;
- Трапезоиден блок (забрзување, константна брзина и забавување);
- Триаголен блок (забрзување и забавување без дел од сегментот со константна брзина).

Алгоритмот за одредување на профилот на брзина на блоковите е даден на слика 3.10. Овој алгоритам се базира на променливата $S_{intersect}$ која претставува растојанието од крајот на блокот во кое се случува транзиција од забрзување во забавување или од константна брзина во забавување. Оваа променлива се добива кога должината на целиот блок се дели на половина и на него се додава половина од растојанието потребно за да брзината се промени од влезна до излезна со даденото номинално забрзување за тој блок. Изразот за пресметка на оваа променлива е даден во продолжение:

$$S_{intersect} = \frac{S + \frac{(v_{entry}^2 - v_{exit}^2)}{2a}}{2} \quad (3.44)$$

Алгоритмот за секој блок прво ја пресметува оваа променлива и го споредува добиеното растојание со нула. Ако $S_{intersect}$ е помало од нула тоа значи дека потребното растојание да се стигне од влезната до излезната брзина со даденото номинално забрзување за тој блок е поголема од S . Ова значи дека блокот е само со забрзување, па во овој случај се поставува максималната брзина на блокот која ќе биде иста со излезната бидејќи се работи за блок само со забрзување.

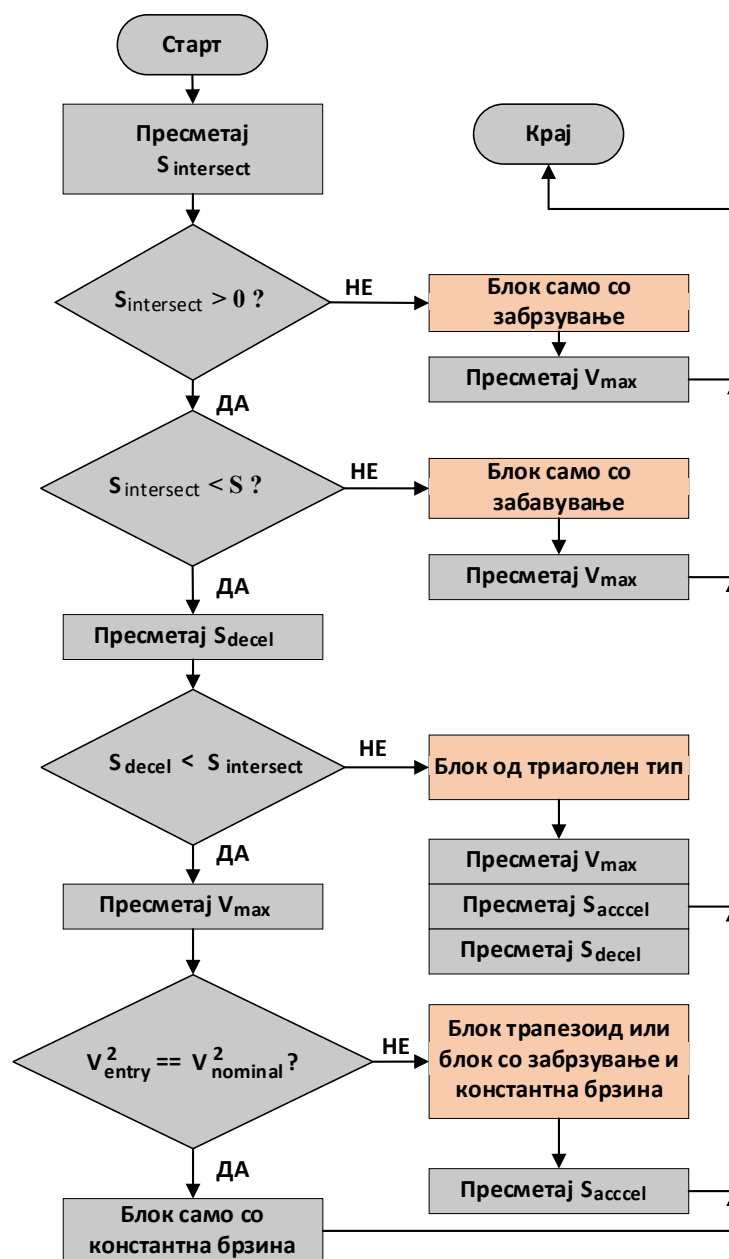
За вредности на променливата поголеми од нула т.е. растојанија поголеми од нула се прави споредба на одлучувачката променлива $S_{intersect}$ со должината на блокот S . Ако одлучувачката променлива т.е. растојанието на транзицијата е поголемо од должината на блокот, тоа значи дека блокот кој се разгледува е од тип само забавување и за него се поставува влезната брзина како максимална брзина. Ако пресекот на транзицијата е во рамките на должината на блокот тогаш типот на блокот е трапезоиден или триаголен. За таа цел прво се пресметува растојанието од крајот на блокот за кое машината ќе забави

од номиналната брзина до излезната брзина S_{decel} . Пресметката се прави според следниот израз:

$$S_{decel} = \frac{v_{nom}^2 - v_{exit}^2}{2a} \quad (3.45)$$

Во случај кога транзицискиот пресек е помал или еднаков во однос на потребното растојание за забавување од номиналната до излезната брзина (нема доволно должина за забавување) тогаш тековниот блок е од триаголен тип на профил на брзина. Во овој случај растојанијата за забрзување и забавување се поставуваат на транзицискиот пресек $S_{intersect}$, а максималната брзина на блокот се пресметува според следниот израз:

$$v_{max} = \sqrt{v_{exit}^2 + 2 \times a \times S_{intersect}} \quad (3.46)$$



Слика 3.10. Алгоритам за одредување на профилот на брзина на секој од блоковите.

Во спротивно кога транзицискиот пресек е поголем во однос на S_{decel} се прави споредба дали влезната брзина е еднаква со номиналната брзина. Ако влезната е иста со номиналната тогаш се работи за блок со константна брзина, во спротивно се работи за блок од трапезоиден тип. За трапезоиден тип на блок се пресметува растојанието од крајот на блокот каде главата на машината треба да заврши со забрзувањето според следниот израз:

$$S_{accel} = S - \frac{v_{nom}^2 - v_{entry}^2}{2a} \quad (3.47)$$

За подобро разбирање на тоа како се одредува типот на блокот на слика 11 дадени се два примери за триаголен и трапезоиден профил на брзина. За триаголниот профил на брзина познати се:

$$v_{entry} = \frac{2mm}{s}; v_{nom} = \frac{5mm}{s}; v_{exit} = \frac{3mm}{s}; a = \frac{2mm}{s^2}; S = 3mm \quad (3.48)$$

Алгоритамот прво го пресметува $S_{intersect}$ и според изразот (3.44) се добива:

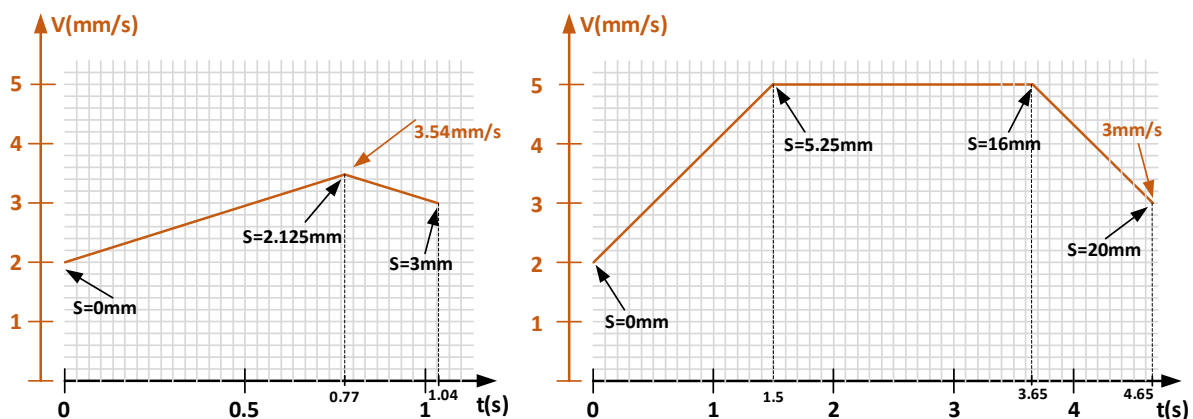
$$S_{intersect} = \frac{3 + \frac{(2^2 - 3^2)}{2 \times 2}}{2} = 0.875mm \quad (3.49)$$

Добиената вредност е поголема од нула и помала од должината на блокот, па се заклучува дека блокот е или трапезоиден или триаголен. За да се одреди на кој од овие два типови припаѓа блокот се пресметува потребното растојание за забавување од номиналната брзина до излезната брзина според изразот (3.45):

$$S_{decel} = \frac{5^2 - 3^2}{2 \times 2} = 4mm \quad (3.50)$$

Добиеното растојание е поголемо во однос на $S_{intersect}$, па се заклучува дека се работи за блок со триаголен профил на брзината. Алгоритамот ги поставува растојанијата за забрзување и забавување на $S_{intersect}$ што претставува растојанието од крајот на блокот каде се одвива транзицијата од забрзување во забавување. На слика 3.11 е прикажано растојанието од почетокот на блокот и тоа се добива како:

$$S_{accel} = S_{decel} = S - S_{intersect} = 2.125mm \quad (3.51)$$



Слика 3.11. Примери за триаголен и трапезоиден профил на брзина.

Максималната брзина која ќе се постигне во рамките на овој блок се пресметува според изразот (3.46):

$$v_{max} = \sqrt{3^2 + 2 \times 2 \times 0.875} = 3.54mm/s \quad (3.52)$$

За примерот со трапезоиден профил на брзина од слика 3.11 брзините и забрзувањето се исти како за триаголниот профил, со таа разлика што тука должината на блокот е: $S = 20mm$. На сличен начин како за претходниот пример се пресметуваат:

$$S_{intersect} = \frac{20 + \frac{(2^2 - 3^2)}{2 \times 2}}{2} = 9.375mm \quad (3.53)$$

$$S_{decel} = \frac{5^2 - 3^2}{2 \times 2} = 4mm \quad (3.54)$$

Тука растојанието од крајот на блокот каде има транзиција од константна брзина во забавување $S_{intersect}$ е поголемо во однос на потребното растојание за забавување од номиналната брзина до излезната брзина, па се заклучува дека се работи за трапезоиден профил на брзина. Максималната брзина е еднаква на номиналната за овој блок, а се пресметува растојанието од крајот на блокот каде треба да се заврши забрзувањето и да се премине во константна брзина според изразот (3.47):

$$S_{accel} = 20 - \frac{5^2 - 2^2}{2 \times 2} = 14.75mm \quad (3.55)$$

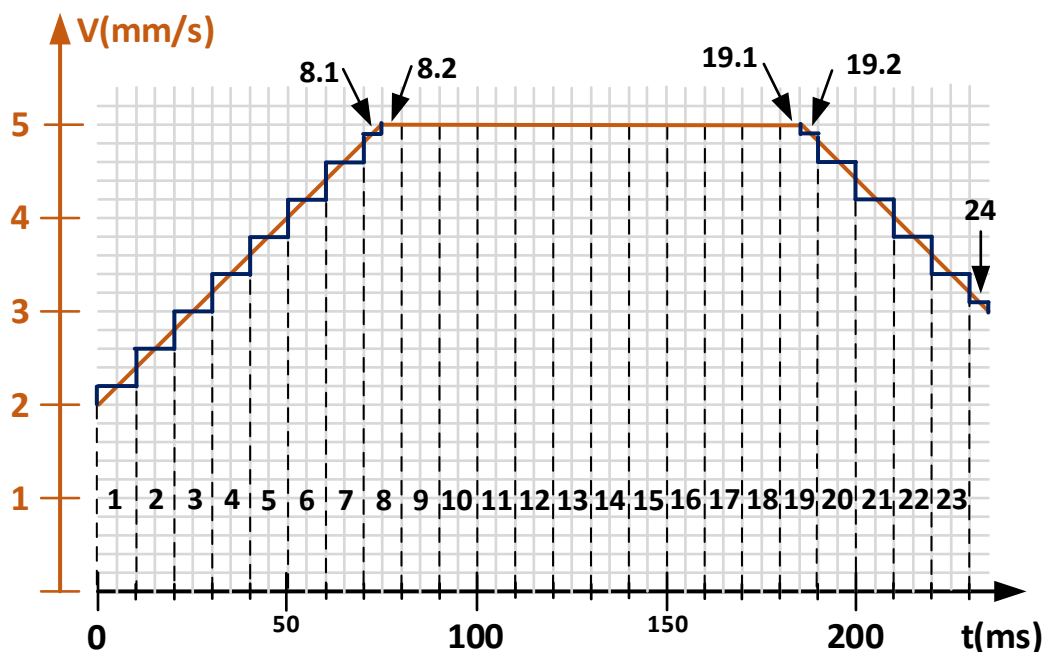
На слика 3.11 прикажани се растојанијата на премин од забрзување во константна брзина и преминот од константна во забавување од почетокот на блокот (5.25mm и 16mm соодветно). Со овој алгоритам се поставува типот на блокот и растојанијата од крајот на блокот до каде завршува забрзувањето, односно од каде почнува забавувањето.

Дефинираните блокови со профилот на брзина заедно со влезната брзина и излезната брзина за секој од нив, растојанијата од крајот на блокот на кој завршува забрзувањето, односно започнува забавувањето ги дефинираат сите потребни елементи за имплементација на движењата на CNC-машината. GRBL ги имплементира движењата на машината со сегментација на така дефинираните блокови на сегменти со константна брзина. Сегментацијата се реализира на константни временски интервали од $\Delta t = 10ms$. Препорачаните вредности за временскиот интервал на сегментација се движат во рамките на интервалот: 5ms – 20ms.

На слика 3.12 е даден пример за сегментација на еден блок од планерот со $\Delta t = 10ms$. Профилот на брзина е трапезоиден и дадени се:

$$v_{entry} = 2mm/s; v_{nom} = 5mm/s; v_{exit} = 3mm/s; a_{nom} = 40mm/s^2; \quad (3.56)$$

Со сегментацијата од 10ms добиени се 24 сегменти со тоа што сегментот 8 кој се протега и во делот забрзување и во делот константна брзина се дели на два потсегменти (8.1 и 8.2), истото се применува и за сегментот 19 (се дели на 19.1 и 19.2). Ова се прави со цел да нема сегменти кои припаѓаат на два различни профили на брзини (пр. забрзување и константна брзина, односно константна брзина и забавување). Со тоа вкупниот број на сегменти во примерот од слика 3.12 е 26.



Слика 3.12. Пример за сегментација на блок од планерот.

Во конкретниот пример сегментите се делат на два еднакви делови, но тоа не е правило, зависи колкав дел од сегментот Δt припаѓа на профилот на забрзување, а колкав дел на профилот константна брзина. Тоа значи дека сегментот 8.1 може да биде поголем, помал или еднаков со сегментот 8.2, истото важи за сегментите 19.1 и 19.2. Притоа, мора да се внимава да не се добие сегмент во кој нема ниту еден чекор. Слично 24-от сегмент е помал од $10ms$ (во конкретниот пример е $5ms$), бидејќи таму завршува блокот и должината е помала од Δt .

Пресметаната брзина од планерот е дадена со темно портокалова боја и има трапезоидна форма, но реалниот профил на брзина по која ќе се движи главата на машината е дадена со темно сина боја. Промената на брзината за секој сегмент од делот забрзување се пресметува со следниот израз:

$$\Delta v_i = a \times \Delta t_i \quad (3.57)$$

додека пак средната брзина на секој сегмент се пресметува според изразот:

$$v_{avg_i} = v_{entry_i} + \frac{\Delta v_i}{2} \quad (3.58)$$

каде v_{entry_i} е влезната брзина за i -от сегмент и се пресметува како:

$$v_{entry_{i+1}} = v_{entry_i} + \Delta v_i \quad (3.59)$$

Должината на i -от сегмент во делот забрзување се пресметува како:

$$S_i = v_{avg_i} \times \Delta t_i \quad (3.60)$$

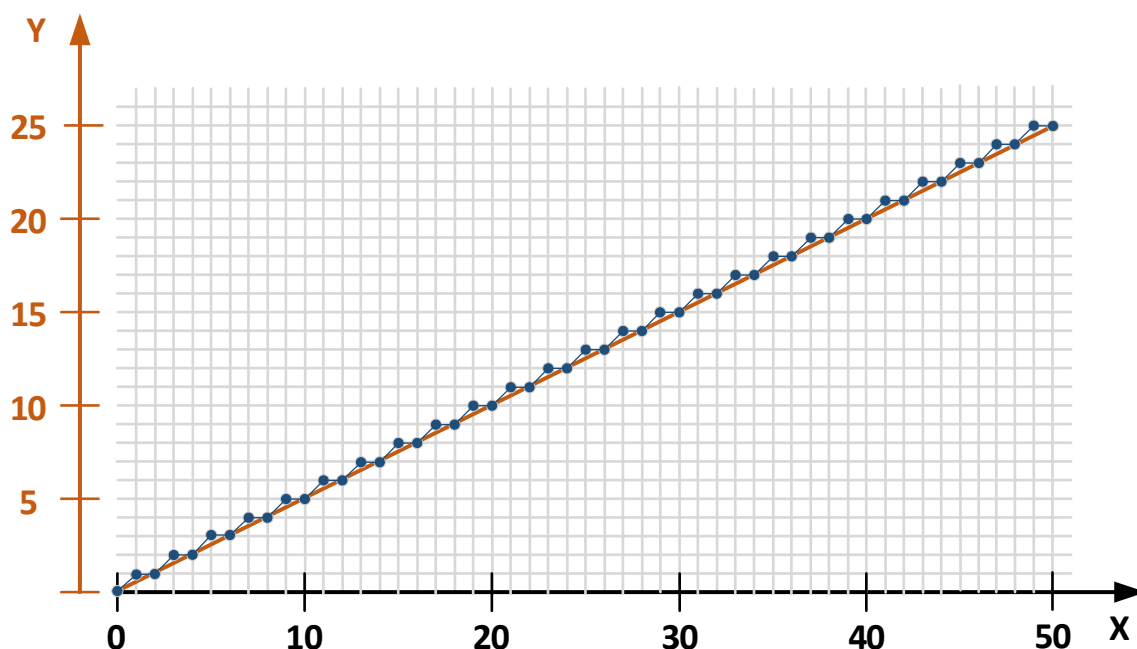
за делот забавување се користат истите равенки со тоа што изразите (3.58) и (3.59) се со знак минус. Во делот за константна брзина, брзината со која се движи главата на

машината е еднаква на номиналната брзина. Должината на секој сегмент во овој дел се пресметува според следниот израз:

$$S_i = v_{nom_i} \times \Delta t_i \quad (3.61)$$

Со сегментацијата патеките на движење на главата на машината се делат на прилично мали сегменти. Движењето на моторите за трите оски е со цел број на чекори, а пресметките на сегментите најчесто се со децимали и при секој сегмент се прави грешка на заокружување на цел број на чекори. Кумулативно, со секој сегмент оваа грешка расте и при релативно големи растојанија оваа грешка станува значителна и доведува до губење или додавање на нови чекори, што резултира со непрецизна изработка на дадениот 3D-модел. Затоа GRBL ја пресметува грешката која се прави при секое заокружување за секој сегмент и ја додава на следниот сегмент со што се имплементира корекција на грешката и се избегнува кумулативно собирање на истата. За секој сегмент се поставува бројот на циклуси (времето) за генерирање на прекините. Времето на прекините е директно поврзано со генерирањето на импулсите, односно со брзината на моторите.

За вака подготвените сегменти GRBL го користи Брезенхам алгоритмот за генерирање на линии од компјутерската графика со цел да ги трансформира овие сегменти во прецизно движење на чекорните мотори. Овој алгоритам обезбедува глатко и прецизно движење на повеќе оски со одлучување кога и кој мотор да изведи чекор за да се постигне прецизно движење по саканата патека. Секој чекор на моторите GRBL го третира како пиксел во примената на Брезенхам алгоритмот од компјутерската графика. Имплементацијата на овој алгоритам е изведена преку прекинска рутина на тајмер. За секој чекор на доминантната оска тајмерот генерира интерапт(прекин). Главни карактеристики на овој алгоритам се:



Слика 3.13. Пример за користење на Брезенхам алгоритмот за генерирање на линија од точка (0,0) до точка (50,25).

- **Прецизност:** алгоритмот гарантира дека ниту една оска не заостанува, без разлика дали се работи за праволиниски, дијагонални или кружни движења.
- **Адаптибилност на брзината:** Времето на чекорите динамички се управува од параметрите на секој сегмент, така што се обезбедува непречено, глатко забрзување и забавување.
- **Синхронизирано движење на повеќе оски:** Со инкрементирање и споредба на променливите за акумулираните грешки на секоја оска, алгоритмот гарантира синхронизирано движење на главата на машината.

Математичката репрезентација на Брезенхам алгоритмот имплементиран во GRBL е дадена во продолжение. Нека параметрите за сегментот ги означиме како во продолжение:

- S_x, S_y, S_z : број на чекори за тековниот сегмент за сите оски соодветно.
- E_x, E_y, E_z : акумулираните грешки за секоја оска соодветно.
- T : бројот на чекори на доминантната оска за тековниот сегмент (оската со најголем број на чекори).

Алгоритмот ги изведува следните операции при секое извршување на прекинската рутина:

1. Инкрементирање на променливите за акумулираните грешки на секоја оска:

$$E_x \leftarrow E_x + S_x, E_y \leftarrow E_y + S_y, E_z \leftarrow E_z + S_z \quad (3.62)$$

2. Ако акумулираната грешка го надмине прагот се извршува чекор на дадената оска:

$$\begin{aligned} &\text{Ако } E_x > T \text{ изврши чекор на } X, E_x \leftarrow E_x - T \\ &\text{повтори го истото за } Y \text{ и } Z \text{ оските.} \end{aligned} \quad (3.63)$$

На слика 3.13 е даден пример за користење на Брезенхам алгоритмот за генерирање на линија, односно чекори по кој ќе се движи главата на машината за линеарна интерполација со почетна точка $(0,0)$ и дестинациска точка $(50,25)$. Со портокалова линија е означена патеката по која теоретски треба да се движи главата на машината, а со темно сина боја е означена патеката по која реално ќе се движи поради конечната резолуција на чекорните мотори. Истата ситуација ќе се случи и кога би се користеле сервомотори поради конечната резолуција на енкодерите на самите мотори. Алгоритмот се користи во тридимензионален простор со вклучување и на Z-оската, но примерот е даден во 2D за полесно разбирање.

Дополнително ќе биде разгледан уште еден едноставен пример заедно со деталните пресметки за генерирање на точките со примена на Брезенхам алгоритмот. Даден е линеарниот сегмент со почетна точка $(x_0, y_0) = (0,0)$ и крајна (дестинациска) точка $(x_1, y_1) = (7,4)$, за овој сегмент треба да се генерираат точките од дискретната линија. Алгоритмот прво ги пресметува иницијалните параметри:

$$dx = x_1 - x_0 = 7; dy = y_1 - y_0 = 4 \quad (3.64)$$

бидејќи $dx > dy$, X-оската е доминантна оска за дадениот сегмент, па оваа вредност се зачувува во променливата steps:

$$steps = dx = 7 \quad (3.65)$$

Табела 3.2. Пример за користење на Брезенхам алгоритмот за генерирање на дискретна линија од точка (0,0) до точка (7,4).

Чекор	X	Y	Грешка пред ажурирање	Грешка по ажурирање	Инкрементирање на Y	Грешка по корекција
1	0	0	-3,5	-3,5+4=0,5	ДА	0,5-7=-6,5
2	1	1	-6,5	-6,5+4=-2,5	НЕ	-2,5
3	2	1	-2,5	-2,5+4=1,5	ДА	1,5-7=-5,5
4	3	2	-5,5	-5,5+4=-1,5	НЕ	-1,5
5	4	2	-1,5	-1,5+4=2,5	ДА	2,5-7=-4,5
6	5	3	-4,5	-4,5+4=-0,5	НЕ	-0,5
7	6	3	-0,5	-0,5+4=3,5	ДА	3,5-7=-3,5

Следно се одредува грешката која е половина од бројот на чекори на доминантната оска:

$$error = -\frac{dx}{2} = -3.5 \quad (3.66)$$

Насоката на двете оски е кон позитивниот дел на истите, па доминантната оска ќе се инкрементира на секое повикување на прекинската рутина, за континуирано движење по истата, додека пак кога акумулираната грешка ќе го премине прагот ќе се инкрементира и координатата на оската Y. Тука се користат само операции со цели броеви, затоа овој алгоритам е соодветен и за осум битни микроконтролери бидејќи не бара операции со децимални броеви. Затоа се користи и во првите верзии на GRBL наменети за осум битни микроконтролери. Меѓутоа и кај новите верзии се задржува овој алгоритам како ефикасно решение кое заштедува процесирачка моќ. Во табела 3.2 се дадени сите чекори за генерирање на сите точки од дадениот пример.

Главните недостатоци на Брезенхам алгоритмот применет во GRBL за генерирање на импулсите за синхроно движење на моторите се јавуваат при движења со недоминантни оски, односно сегменти кои имаат движења по оските со приближно ист број на чекори. Во вакви услови се јавува неглатко генерирање на импулсите, особено при ниски брзини на движење. Ова директно доведува до нарушена синхронизација на движењето, како и до појава на бучава и механички вибрации кај моторите. Состојбата дополнително се влошува при повеќеосно управување при мали брзини, најчесто во опсег на фреквенции на генерирање на чекорите од 0 до 5 kHz, што негативно се одразува врз квалитетот на изработените делови.

За делумно надминување на овој проблем, GRBL имплементира посебен алгоритам наречен Adaptive Multi-Axis Step Smoothing, со кој автоматски се менува ефективната резолуција на оските (се зголемува при ниски брзини и обратно), со цел да се подобрат перформансите и да се намалат вибрациите, бучавата и неглаткото движење. Дополнително, генерирањето на чекорите на овој начин при многу високи брзини станува критично поради користењето на само еден тајмер за синхронизација на сите мотори, при што времето на извршување на прекинската рутина може да стане поголемо од времето помеѓу два последователни чекори. Главниот структурен недостаток на овој начин на генерирање на импулси е тоа што чекорите за сите мотори мора да се генерираат исклучиво во ист временски момент со чекорите на доминантната оска, односно оската со најголем број на чекори.

3.1.2 Клипер напреден фирмвер наменет за контрола на 3D-печатари

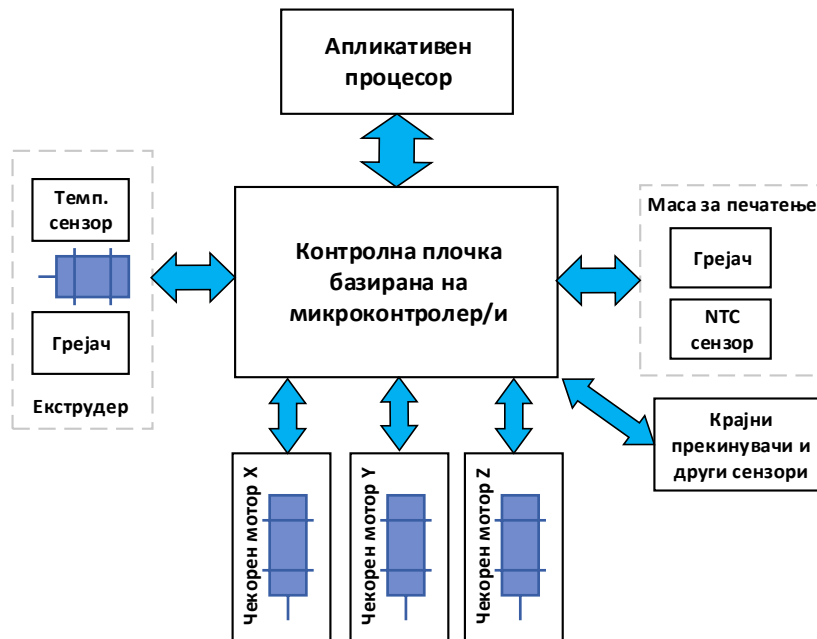
Клипер како напреден фирмвер првично наменет за 3D-печатари е креиран од Kevin O'Connor во 2014 година. Тој ја поставува првата верзија на Клипер на GitHub и набрзо други ентузијасти почнуваат да го користат и придонесуваат за неговото подобрување. Клипер популарноста ја стекнува кај заедницата на љубители на технологијата на 3D-печатење, особено кај оние што сакаат да експериментираат со различни хардверски и софтверски конфигурации. Еден од првите корисници на Клипер за своите печатари е Voxel уште со првиот дизајн на нивниот 3D-печатар во 2015 година. Основачот на Voxel е исто така поборник за идеологијата на отворен и слободен хардвер и софтвер, па нивните комплетни дизајни се отворени и слободни за користење од сите ентузијасти во областа. Во последните неколку години Клипер зема голем замав и голем дел од комерцијални 3D-печатари се префрлаат на овој фирмвер. Перформансите на 3D-печатарите кои го користат Клипер како фирмвер значително ги надминуваат перформансите на 3D-печатарите кои ги користат останатите фирмвери како: Marlin, GRBL, RepRap и останатите [80][80].

За постигнување на високи перформанси Клипер користи покомплексен хардвер во однос на останатите. Хардверот вклучува апликативен процесор (како што е Raspberry Pi) за пресметување на движењата и микроконтролер за нивно извршување. Апликативниот процесор ги пресметува времињата на сите чекори на секој мотор, ги компресира истите и ги праќа до микроконтролерот со посебен протокол, а микроконтролерот ги извршува сите чекори според добиените времиња од главниот процесор. Софтверот не користи кинематички апроксимации (како што е Брезенхам алгоритмот кај GRBL, Marlin итн.), наместо тоа ги пресметува прецизните времиња на секој чекор базирани на физиката на забрзување и кинематиката на механичката конструкција на печатарот или машината. Пресметаните времиња со поголема прецизност обезбедуваат потивки и постабилни движења на самата машина.

Воведувањето на апликативен процесор за изведување на сите пресметки му овозможува на Клипер да постигне високи фреквенции на импулсите кон моторите за сите микроконтролери. Така, на старите осум битни микроконтролери може да се постигне брзина од 175K чекори во секунда, додека пак на поновите микроконтролери може да се постигнат и до милион чекори во секунда. Дополнително, времето на генерирање на импулсите кон моторите е прецизно и за високи фреквенции, односно брзини.

На слика 3.14 е даден блок-дијаграм на хардверот на Клипер системот, специфично за него е тоа што поддржува печатари со повеќе микроконтролери. На пример, може да се користи еден микроконтролер за управување само на екструдерот, а посебен микроконтролер за управување на грејачите во системот и трет микроконтролер за управување на останатите делови од печатарот. Софтверот на Клипер имплементиран на хост апликативниот компјутер извршува синхронизација на тактот со цел да го анулира поместувањето на тактот кај различните микроконтролери во системот. За прецизно извршување на чекорите и командите во ваков систем со повеќе управувачки процесорски единици од клучни значење е синхронизацијата на тактот помеѓу сите модули на системот.

Клипер воведува неколку новини што не беа својствени за претходните фирмвери и кои придонесуваат за поголемата ефикасност и значително забрзување на процесот на 3D-печатење и квалитетот на изработените делови.



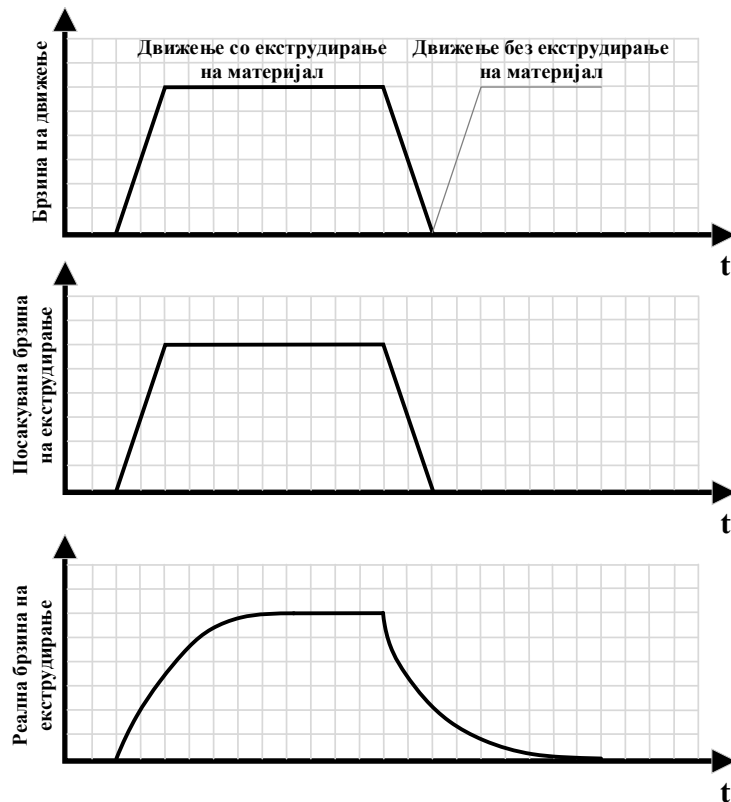
Слика 3.14. Блок-дијаграм на хардверот на Клипер системот.

Една од новините е филтрирање на влезниот сигнал за намалување на вибрациите (на англ. Input shaping), ова е управувачка техника со отворена јамка која креира командни сигнали до моторите од кои е филтрирана резонантната фреквенција која предизвикува вибрации на механичката конструкција на печатарите и CNC-машините. За да ова е возможно мора на почеток да се направи мерење и да се утврди резонантната фреквенција за механичката конструкција.

Најчесто кај комерцијалните 3D-печатари мерењето се прави при првото вклучување на печатарот и по потреба понатаму во текот на работата кога корисникот на истиот ќе процени дека има потреба од ново калибрирање. Мерењето се изведува со помош на акцелерометар кој е вграден во главата на 3D-печатарот, управувачката единица генерира предефинирани микродвижења на главата со различни фреквенции и ги мери вибрациите со што се одредува резонантната фреквенција.

Друга новина која ја вовеле Клипер во полето на 3D-печатењето е компензацијата на притисокот на повлекување на материјалот од страна на екструдерот. Теоретски во идеален случај при движење на главата на печатарот од тековната точка до дестинациската точка потребно е да се екструдира иста количина на материјал по целата должина и да нема екструдирање по завршувањето на движењето.

Тука се работи за материјал кој се топи и може да истекува и по слободен пад, па користењето на основната формула за еднакво екструдирање на материјал од почетната до дестинациската точка најчесто доведува до недоволно материјал на почетокот и повеќе екструдирани материјал на крајот од движењето и истекување на материјал по завршувањето на тековниот сегмент. На слика 3.15 се дадени временски дијаграми на брзината на главата на печатарот, посакуваниот временски облик на брзината на екструдирање на материјал и реалниот облик на брзината на екструдирање на материјалот кога се користи базичната формула за рамномерно екструдирање на материјал по целата должина на сегментот.



Слика 3.15. Споредба помеѓу брзината на движење на главата на печатарот и посакуваната и реалната брзина на екструдирање на материјал.

Во примерот се зема движење во кое има екструдирање на материјал, а по него има движење во кое нема екструдирање на материјал. Тоа е случајот во кој истекувањето на материјал прави најголеми проблеми. Моделот на компензација на притисокот на повлекување на материјалот е поразличен од базичниот модел на екструдерот. Наместо да се верува дека секој mm^3 филамент внесен во екструдерот ќе резултира со непосредно иста количина на материјал на излезот од екструдерот, Клипер користи поразличен пристап базиран на динамиката на притисокот во самиот екструдер. Филаментот во екструдерот се однесува слично како пружина, како што се зголемува количината на материјал внесен во екструдерот, така се зголемува и внатрешниот притисок како во законот на Нооке, а за ова потребно е време да се стопи материјалот и да се екстудира на излез. Брзината на течење на стопената пластика низ млазницата (дизната) е основно дефинирана од притисокот и дијаметарот на млазницата. Повисок притисок доведува до поголемо екструдирање, но промената на брзината на екструдирање не е веднаш во моментот кога моторот на екструдерот ќе се придвижи. Ова доаѓа како последица на законот на Poiseuille.

Главната идеја тука е дека врската помеѓу должината на филаментот (материјалот за печатење), притисокот и брзината на екструдирање може да се моделира со користење на линеарен коефициент:

$$E_{pa} = E_{nom} + K_{pa} \times v_{nom} \quad (3.67)$$

каде: E_{pa} е позицијата на екструдерот со употреба на методот на компензација на притисокот, E_{nom} е номиналната позиција на екструдерот без вклучена компензација,

v_{nom} е номиналната брзина на екструдирање и K е коефициентот на компензација со кој се дефинира интензитетот на компензација на притисокот.

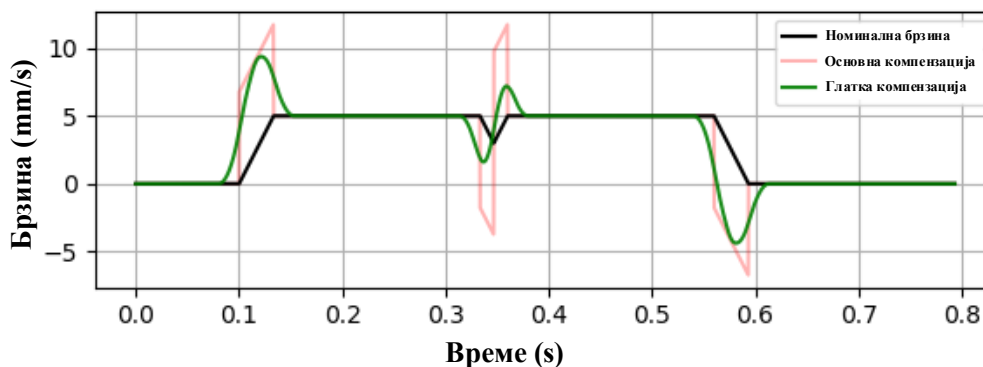
На слика 3.16 е даден пример за два последователни сегменти на екструдирање со брзина на екструдирање во точката на спој различна од нула. Како што може да се забележи методот со основна компензација на притисокот додава дополнителен филамент во споредба со номиналниот во делот за забрзување. Што поголема брзина на екструдирање, повеќе филамент/материјал треба да се турне во екструдерот за време на забрзувањето со цел да се компензира притисокот. Додека пак, за време на забавувањето се прави враќање назад на филаментот со цел израмнување на вишокот од делот со забрзување (екструдерот има негативна брзина). Основната формула на компензација дадена со изразот (3.68) и графикот од слика 3.16 може да предизвика нагли промени на брзината на екструдирање, што може да биде проблематично за чекорниот мотор на екструдерот и да предизвика губење или прескокнување на чекори, лош квалитет на екструдираниот материјал дури и механичко оштетување на запчаниците на екструдерот.

Затоа Клипер имплементира глатко порамнување на позицијата на екструдерот со користење на тежински просек на мали временски интервали, со што се осигурува дека моторот на екструдерот нема да прави нагли забрзувања и забавувања и ќе се движи глатко и континуирано. Формулата за глатка компензација на притисокот е дадена во продолжение:

$$E_{pa_smooth}(t) = \frac{\int_{t - \frac{t_{smooth}}{2}}^{t + \frac{t_{smooth}}{2}} E_{pa}(x) \times \left(\frac{t_{smooth}}{2} - |t - x| \right) dx}{\left(\frac{t_{smooth}}{2} \right)^2} \quad (3.68)$$

каде:

- Со интегралот се зема просекот на вредности од времето $t - \frac{t_{smooth}}{2}$ до $t + \frac{t_{smooth}}{2}$;
- Тежинската функција $\left(\frac{t_{smooth}}{2} - |t - x| \right)$ додава повеќе на значење на вредностите поблиски до тековниот момент и помалку на оние подалеку;
- Со именителот $\left(\frac{t_{smooth}}{2} \right)^2$ се прави нормализација на тежинската функција;



Слика 3.16. Различни модели на брзини на екструдирање и компензација на притисокот на два сегменти со брзина на екструдирање поголема од нула во точката на спој.

Слично како кај GRBL, кружните интерполации кај Клипер се имплементираат со делење на кружниот лак на мали линеарни сегменти. Пресметувањето на радиусот на лакот, аголот на кружниот лак во однос на центарот и должината на лакот се пресметуваат потполно идентично како кај GRBL со изразите (3.4), (3.8) и (3.14) соодветно. Одредувањето на бројот на сегменти на кои се дели лакот е различно од GRBL и Клипер го пресметува со претходно дефиниран параметар за должината на секој сегмент (MM_PER_ARC_SEGMENT). Тука должината на сегментот е фиксна, без разлика на радиусот на кругот и иницијалната вредност му е 1mm. Кај GRBL одредувањето на бројот на сегментите зависи од параметарот arc_tol (слика 3.4) што во пракса е подобро, бидејќи за кружни лакови со поголем радиус должината на сегментот ќе биде подолга, а за помали радиуси сегментите ќе бидат со помала должина.

Клипер прво ја пресметува должината на движење S_{travel} во зависност од тоа дали има само кружно движење или спирално движење (кружно движење во една рамнина и линеарно движење по третата оска):

$$S_{travel} = \theta \times R \quad (3.69)$$

за кружно движење (изведувањето дадено во делот за GRBL израз (3.14)) и

$$S_{travel} = \sqrt{(\theta \times R)^2 + S_{lin}^2} \quad (3.70)$$

за спирално движење каде $\theta \times R$ е должината на кружниот лак и S_{lin} е должината на патеката на линеарната оска. Тука се бара хипотенузата од должината на лакот и должината на линеарното движење по третата оска. Бројот на сегменти се добива според следниот израз:

$$segments = \frac{S_{travel}}{MM_PER_ARC_SEGMENT} \quad (3.71)$$

Должината на секој сегмент кај линеарната оска (нека биде оската Z), должината на материјалот кој треба да се екструдира за секој сегмент и аголот за секој сегмент се пресметуваат според следните изрази:

$$Z_{seg} = \frac{\Delta Z}{segments}; E_{seg} = \frac{\Delta E}{segments}; \theta_{seg} = \frac{\theta}{segments} \quad (3.72)$$

За пресметување на релативните координати на сегментите во однос на центарот на лакот, Клипер исто како GRBL ја користи трансформационата матрица (3.18), со таа разлика што Клипер го прави пресметувањето на следниот сегмент со примена на трансформационата матрица на иницијалниот радиус вектор со егзактно пресметување на тригонометриските функции (синус и косинус) за аголот $i \times \theta_{seg}$, дадено со следниот израз:

$$\begin{bmatrix} r_{x(i)} \\ r_{y(i)} \end{bmatrix} = \begin{bmatrix} -I \\ -J \end{bmatrix} \times \begin{bmatrix} \cos(\theta_{seg} \times i) & -\sin(\theta_{seg} \times i) \\ \sin(\theta_{seg} \times i) & \cos(\theta_{seg} \times i) \end{bmatrix} \quad (3.73)$$

Кај GRBL ова се прави само на секој 12-ти сегмент со цел корекција на сукцесивната грешка, а останатите сегменти се пресметуваат сукцесивно во однос на координатите на претходниот сегмент со едно пресметување на тригонометриските функции на аголот θ_{seg} . Клипер користејќи апликативен процесор може да ги

пресметува тригонометриските функции на секој сегмент без никакво оптоварување на процесорот и од тука може да се дозволи поедноставување во кодот. Координатите на материјалот и координатите на линеарната оска (нека биде оската Z) се пресметуваат според следните изрази:

$$Z_{new} = Z_{old} + Z_{seg}; E_{new} = E_{old} + E_{seg} \quad (3.74)$$

Апсолутните координати на сегментите од кружниот лак се добиваат според следните изрази:

$$X_i = X_C + r_{x(i)}; Y_i = Y_C + r_{y(i)} \quad (3.75)$$

каде X_C и Y_C се координатите на центарот на лакот. Овде се дадени пресметките за кружна интерполација на XY -рамнината со линеарно движење на Z -оската што заедно формираат спирално движење, но истите пресметки во Клипер се изведени и за XZ и YZ -рамнините. Вака генерираните сегменти Клипер ги праќа во баферот со G команди како $G1$ -команди.

Слично како кај GRBL и Клипер сите движења ги сведува на линеарни, дури Клипер ги претвора кружните движења во $G1$ -команди за линеарно движење, со што сите команди се сведуваат на линеарни уште при вчитувањето на G -кодот. Клипер бидејќи користи апликативен компјутер и голем дел од кодот е напишан во програмскиот јазик Пајтон користи објектно-ориентиран пристап, па движењата се претставени преку класата Move. На слика 3.17 се дадени елементите на класата Move. Оваа класа е задолжена за планирањето на движењата и генерирањето на траекториите. Основните елементи на оваа класа се почетната ($start_pos$) и крајната позиција (end_pos) кои тука содржат 4 координати вклучувајќи ја и координатата за количината на екструдираниот материјал. Растојанијата за секоја координата ($axes_d$, односно dX, dY, dZ, dE) кои се пресметуваат според следните изрази:

$$dX = X_e - X_s; dY = Y_e - Y_s; dZ = Z_e - Z_s; dE = E_e - E_s \quad (3.76)$$

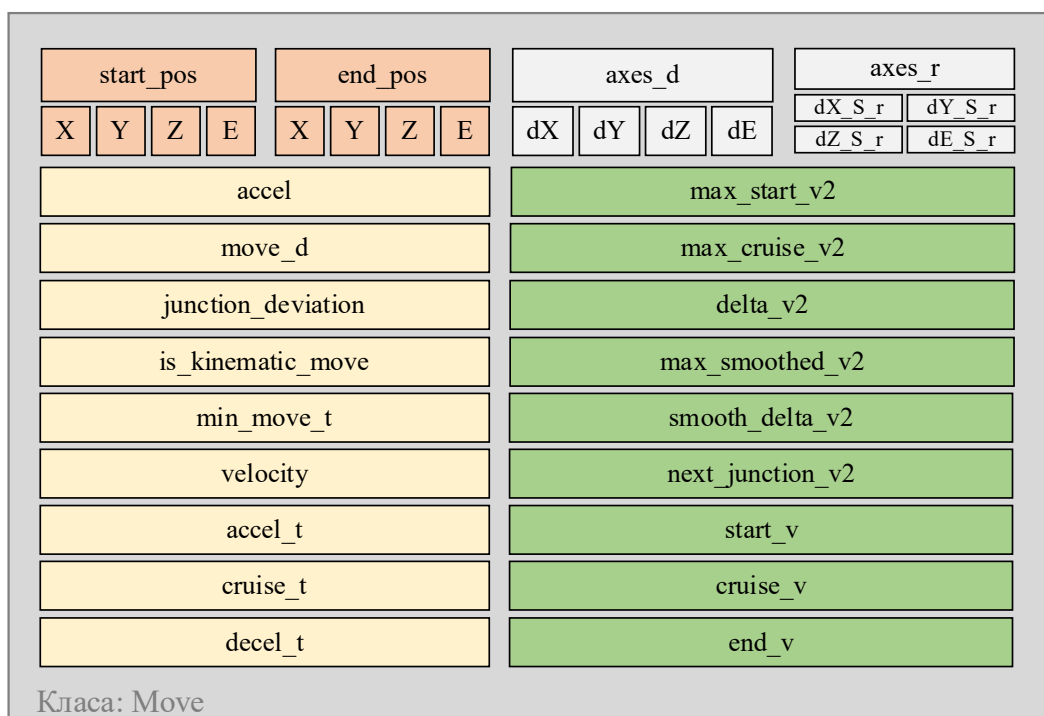
Разликата со GRBL е во дополнителната координата за материјалот која таму ја нема во основната верзија бидејќи таа е наменета исклучиво за CNC-машини, меѓутоа GRBL се користи како основа за сите денес познати отворени и слободни фирмвери за управување на 3D-печатари. Координатата за материјалот се вклучува и во пресметувањето на растојанието кое треба да го помине системот ($move_d$, односно S) за даденото движење:

$$S = \sqrt{dX^2 + dY^2 + dZ^2 + dE^2} \quad (3.77)$$

Како елементи во класата се чуваат и односите на растојанијата што треба да се поминат за секоја оска во однос на вкупното движење S дефинирано со изразот (11). Овие односи се зачувани во елементот $axes_r$ и пресметувањето е дадено во продолжение:

$$dX_S_r = \frac{dX}{S}; dY_S_r = \frac{dY}{S}; dZ_S_r = \frac{dZ}{S}; dE_S_r = \frac{dE}{S}; \quad (3.78)$$

Понатаму како елементи во класата се јавуваат виртуелната девијација δ дефинирана во GRBL која се користи со потполно истата намена и во Клипер само под името $junction_deviation$.



Слика 3.17. Елементи на класата Move.

Разликата со GRBL околу виртуелната девијација δ е во тоа што Клипер дефинира константа (параметар) максимална брзина при премин помеѓу два сегменти со агол помеѓу нив од 90 степени v_{90} . Ова е појасен параметар и полесно корисникот може да го разбере и дефинира за соодветната машина во споредба со виртуелната девијација. Иницијалната вредност на овој параметар е $5mm/s$, односно $300mm/min$. Овој параметар е воведен за олеснување на нагудувањето на параметрите поврзани со аголот помеѓу сегментите, но во пресметките понатаму се користи виртуелната девијација δ . Ако ја разгледаме сликата 3.8 од GRBL со агол $\theta = 90^\circ$, за синус од аголот $\frac{\theta}{2}$ се добива:

$$\sin 45^\circ = \frac{R}{R + \delta} \quad (3.79)$$

ако се земе дека $\sin 45^\circ = \frac{\sqrt{2}}{2}$ за δ се добива следниот израз:

$$\delta = \frac{R \times (2 - \sqrt{2})}{\sqrt{2}} \Rightarrow \delta = R \times (\sqrt{2} - 1) \quad (3.80)$$

брзината во моментот на спој слично како кај GRBL се зема да биде константна и единственото забрзување е центрипеталното во тој момент, па за брзината го имаме следниот израз:

$$v_{90}^2 = a \times R \Rightarrow R = \frac{v_{90}^2}{a} \quad (3.81)$$

ако (3.81) се замени во (3.80) се добива конечниот израз за виртуелната девијација:

$$\delta = \frac{v_{90}^2 \times (\sqrt{2} - 1)}{a} \quad (3.82)$$

Класата содржи и променлива која се користи за дефинирање/одредување дали се работи за кинематичко движење или некинематичко движење. Клипер ги класифицира сите движења на главата на машината како кинематички, а движењето само на екструдерот како некинематичко. Следно дефинирани се забрзувањето (accel) кое е ограничено од инерцијата на механичката конструкција на машината и брзината (velocity v) дефинирана како минимумот од добиената брзина за тековното движење од G-код командата и максималната брзина ограничена од инерцијата на механичката конструкција. Параметрите кои се дефинирани и тесно поврзани со механичката конструкција се однапред дадени. Во класата се чува и минималното време за тековното движење (min_move_t) кое се пресметува според следниот израз:

$$t_{min} = \frac{S}{v} \quad (3.83)$$

каде v претставува номиналната брзина која се чува како елемент во класата и тоа е максималната можна брзина за даденото движење. За полесно пресметување во класата се чуваат квадратите на специфичните брзини за тековното движење:

- Квадратот на максималната брзина (**max_cruise_v2**), што претставува номиналната брзина дадена од G-код командата ограничена од максимално можната брзина на конструкцијата.
- Квадратот на почетната максимална брзина (**max_start_v2**), што претставува максималната можна влезна брзина за тековниот сегмент. Иницијалната вредност се поставува на нула.
- Квадратот на промената на брзината што може да се постигне по должината на тековниот сегмент (**delta_v2**), пресметано според следниот израз:

$$dv^2 = 2 \times a \times S \quad (3.84)$$

- Квадратот на максималната влезна глатка брзина (**max_smoothed_v2**), која претставува максималната влезна брзина која се одредува на сличен начин како **max_start_v2** со таа разлика што за одредување на глатката брзина се користи dv_{smooth}^2 . Иницијалната вредност се поставува на нула.
- Квадратот на промената на брзината што може да се постигне по должината на тековниот сегмент (**smooth_delta_v2**), пресметано со користење на виртуелното забрзување $a_{smoothed}$. Пресметката се прави слично како во изразот (3.84) само се користи виртуелната вредност за забрзувањето:

$$dv_{smooth}^2 = 2 \times a_{smoothed} \times S \quad (3.85)$$

- Квадратот на брзината во точката на спој со следниот сегмент (**next_junction_v2**). Иницијалната вредност на овој параметар се поставува на невозможна голема вредност, за понатаму да се направи ограничување во зависност од аголот на свртување и максималните можни брзини по должина на соседните сегменти.

Клипер имплементира механизам за измазнување на кратките движења познати како „zigzag“. За таа цел Клипер ги користи глатките параметри на движењето кои се добиени од виртуелното забрзување:

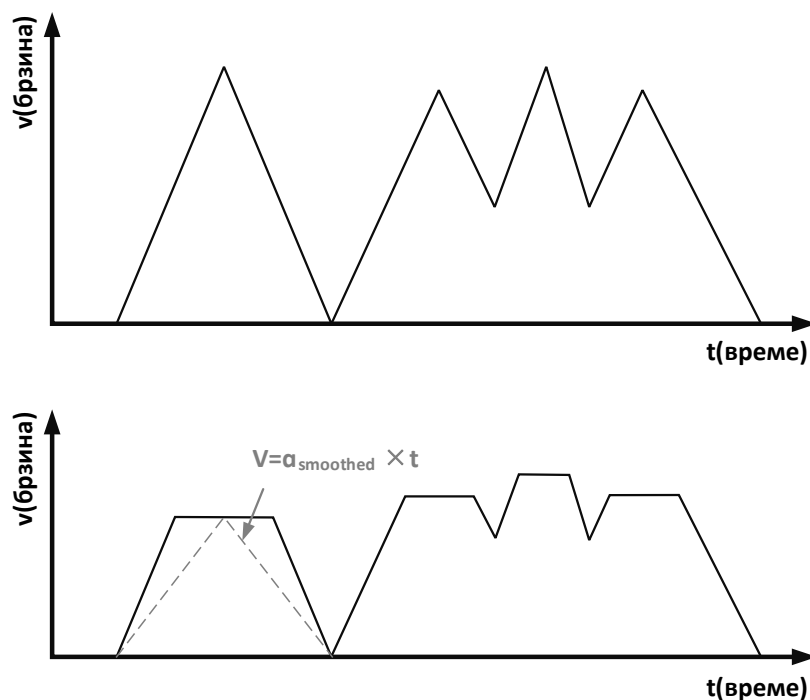
$$a_{smoothed} = K \times a \quad (3.86)$$

каде a е максималното забрзување кое реално се користи ограничено од конструкцијата, а K претставува коефициент кој го дефинира односот помеѓу овие две забрзувања и има иницијална вредност 0.5 што е и најчесто користената вредност. K може да прима вредности од 0 до 1. Во следните верзии на Клипер ќе се воведат нов параметар ***min_cruise_ratio*** кој ќе се користи за истата намена и неговата релација со $a_{smoothed}$ е дадена во продолжение:

$$a_{smoothed} = a \times (1 - \text{min_cruise_ratio}) \quad (3.87)$$

Примерот даден на слика 3.18 содржи движења со чести нагли менувања на профилите на брзината: непосредно менување од забрзување во забавување и обратно, односно триаголни профили на брзината кои предизвикуваат вибрации на машината, зголемен стрес на конструкцијата како и зголемена бучава. Клипер имплементира механизам кој ги претвора овие движења од триаголни во трапезоидни каде за секое движење помеѓу забрзувањето и забавувањето воведува движење со константна брзина. Ова се имплементира со намалување на максималната брзина на сегментот или секвенцата од сегменти со цел да се постигне минимална релативна дистанца за движење со константна редуцирана брзина во однос на деловите со забрзување и забавување.

За да се постигне ова Клипер ги пресметува параметрите на движењето и со регуларното забрзување a и со виртуелното забрзување $a_{smoothed}$. Попрецизно кажано, кодот пресметува какви би биле брзините за секој сегмент ако се лимитирани на виртуелното забрзување. Во примерот од слика 3.18 испрекинатите сиви линии го претставуваат обликот на брзината на првото движење со примена на виртуелното забрзување. Ако движењето не може да ја постигне максималната брзина (номиналната брзина дадена од G-кодот) со примена на виртуелното забрзување, тогаш максималната брзина за тој сегмент се редуцира на максимална вредност која може да се постигне со виртуелното забрзување.



Слика 3.18. Измазнување на кратките патеки со триаголни профили на брзини со користење на виртуелно забрзување.

За најголемиот број на движења/сегменти лимитот на брзината добиен на овој начин со користење на виртуелното забрзување ќе биде поголем или еднаков на номиналната брзина добиена од G-кодот за дадениот сегмент. Меѓутоа за кратките “zigzag” движења, максималната брзина ќе биде редуцирана, но треба да се напомене дека движењата сепак ќе го користат регуларното забрзување a до така редуцираната брзина. Со ова за сметка на малата промена на максималната брзина помеѓу забрзувањето и забавувањето се воведуваат сегменти со константна брзина со што значително се редуцираат вибрациите и бучавата на самата машина.

Максималната влезна брзина на секој сегмент кај Клипер исто како кај GRBL зависи од аголот помеѓу тековниот сегмент и претходниот сегмент (за кинематички движења). Поточно кажано, зависи од промената на брзината и правецот на движење на главата на машината помеѓу двата сегменти. Брзината на движење на главата на машината е тесно поврзана со брзината на екструдирање на материјалот, па затоа Клипер ја зема во предвид и промената на брзината на екструдирање на материјал и насоката (материјалот може и да се повлекува). За таа цел се дефинира диференцијалната разлика помеѓу односите на екструдираниот материјал и должините на сегментите за тековниот и претходниот сегмент:

$$dE_{S_r} = \frac{dE_{cur}}{S_{cur}} - \frac{dE_{prev}}{S_{prev}} \quad (3.88)$$

Во случај кога дефинираната диференцијална разлика dE_{S_r} е нула нема ограничување на брзината на движење помеѓу соседните сегменти од страна на движењето на екструдерот, бидејќи количините на екстудирани материјал по единица должина се исти за двата сегменти. Клипер дефинира и параметар кој претставува брзина со која може да се поминат и најострите агли $v_{instant_corner}$ и предефинираната вредност за овој параметар е $1mm/s$. Оваа брзина се однесува на движењето на главата на екструдерот, а не на брзината на екстудирање на материјалот. Овој параметар се скалира со диференцијалната разлика од количините на екстудирани материјал по единица должина за двата соседни сегменти според следниот израз:

$$v_{ext_lim}^2 = \left(\frac{v_{instant_corner}}{dE_{S_r}} \right)^2 \quad (3.89)$$

Добиената брзина v_{ext_lim} претставува безбедна брзина со која главата на машината може да го помине спојот помеѓу разгледуваните соседни сегменти без нагли промени во екстудирањето на материјалот што би предизвикале проблеми на моторот на екструдерот. Добиената брзина е ограничување само од динамиката на движење на екструдерот. Дополнително Клипер заедно со ова ограничување ги внесува и ограничувањата од максималната брзина на претходниот и тековниот сегмент, пресметаната брзина од претходниот сегмент за следната точка и максималната брзина која може да се постигне по должина на претходниот сегмент со дадената брзина за тој сегмент, па така ограничувањето на почетната брзина на тековниот сегмент е дадено со следниот израз:

$$v_{max_start}^2 = \min(v_{ext_lim}^2, v_{max_cur}^2, v_{max_prev}^2, v_{prev_next_j}^2, v_{prev_max_start}^2 + dv_{prev}^2) \quad (3.90)$$

Во продолжение ќе биде разгледано и ограничувањето од големината на аголот помеѓу двата сегменти. Пресметувањето на аголот помеѓу два соседни сегменти односно

$\cos \theta$ се прави комплетно исто како кај GRBL со изразот (3.35). На истиот начин се стига и до ограничувањето за брзината дефинирано од аголот, со таа разлика што Клипер за една спојна точка дефинира ограничувања на брзината во спојната точка и од тековниот и од претходниот сегмент:

$$v_{\max_j_cur}^2 = \frac{a_{\max_cur} \times \delta_{cur} \times \sin \frac{\theta}{2}}{1 - \sin \frac{\theta}{2}}; v_{\max_j_prev}^2 = \frac{a_{\max_prev} \times \delta_{prev} \times \sin \frac{\theta}{2}}{1 - \sin \frac{\theta}{2}} \quad (3.91)$$

Клипер дозволува за една спојна точка да има различно максимално дозволено забрзување и различна девијација за два соседни сегменти. Дополнително Клипер воведува и центрипетална брзина која се изведува од основниот израз за центрипетална брзина даден во продолжение:

$$v_{c_cur}^2 = a_{cur} \times R \quad (3.92)$$

Апроксимативната кружница впишана помеѓу двата сегменти треба да ги допира сегментите не подалеку од нивната средна точка. Во случај на кратки сегменти за даденото δ можно е овој услов да не е задоволен. Затоа се пресметува центрипеталната брзина и во случај кога кружницата ги допира сегментите во нивните средни точки, па така од слика 3.19 се добива:

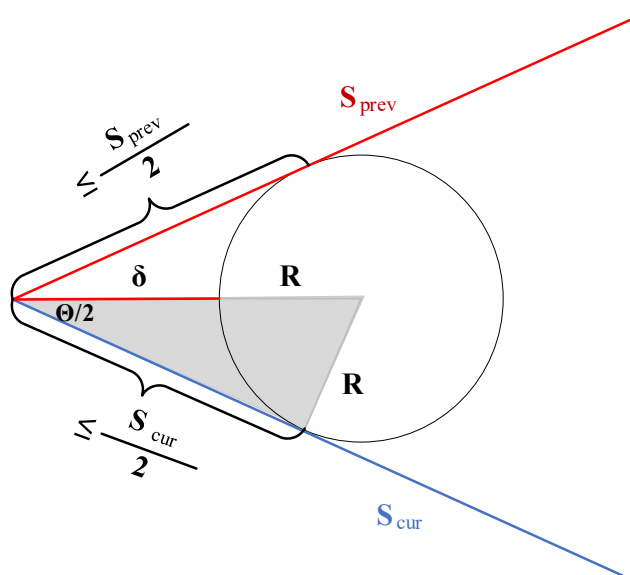
$$\operatorname{tg} \frac{\theta}{2} = \frac{R}{0.5 \times S_{cur}} \quad (3.93)$$

Следно од изразот (3.93) се добива R :

$$R = 0.5 \times S_{cur} \times \operatorname{tg} \frac{\theta}{2} \quad (3.94)$$

Ако изразот (3.94) го замениме во (3.92) за центрипеталната брзина се добива:

$$v_{c_cur}^2 = a_{cur} \times 0.5 \times S_{cur} \times \operatorname{tg} \frac{\theta}{2} \quad (3.95)$$



Слика 3.19. Агол помеѓу два сегменти и дефинирање на центрипетална брзина.

Ако во изразот (3.95) ја воведеме промената на брзината што може да се постигне по должина на тековниот сегмент дадена со изразот (3.84) се добива:

$$v_{c_cur}^2 = 0.25 \times 2 \times a_{cur} \times S_{cur} \times tg \frac{\theta}{2} \Rightarrow v_{c_cur}^2 = 0.25 \times dv_{cur}^2 \times tg \frac{\theta}{2} \quad (3.96)$$

На истиот начин се пресметува и центрипеталната брзина за претходниот сегмент, според следниот израз:

$$v_{c_prev}^2 = 0.25 \times dv_{prev}^2 \times tg \frac{\theta}{2} \quad (3.97)$$

Конечно, максималната влезна брзина за тековниот сегмент се поставува како минималната брзина од центрипеталната брзина и брзината во спојната точка за двата соседни сегменти земајќи ги во предвид ограничувањата од изразот (3.90):

$$v_{max_start}^2 = \min(v_{max_start}^2, v_{max_j_cur}^2, v_{max_j_prev}^2, v_{c_cur}^2, v_{c_prev}^2) \quad (3.98)$$

Дополнително се пресметува максималната глатка брзина според следниот израз:

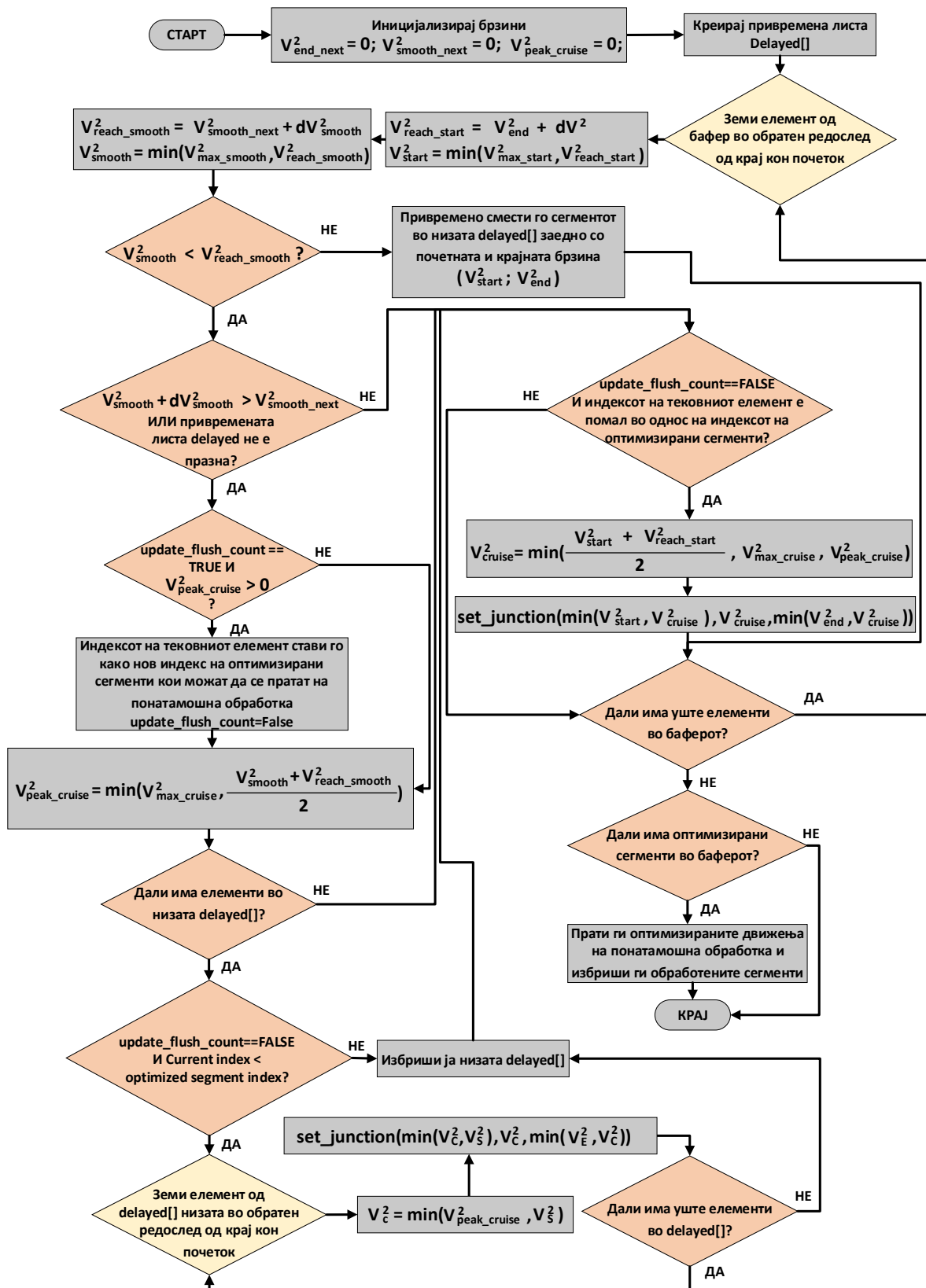
$$v_{max_smooth}^2 = \min(v_{start_max}^2, v_{max_smooth_prev}^2 + dv_{smooth_prev}^2) \quad (3.99)$$

Брзините добиени со изразите (3.98) и (3.99) се ограничени почетни брзини на секој сегмент од аголот помеѓу сегментите, од конструкцијата, од промената на насоката и брзината на екструдирање на материјал, од брзината добиена од G-кодот и дополнително од должината на сегментите. Последното ограничување претставува лимитирање на почетната брзина на тековниот сегмент на почетната брзина на претходниот сегмент, плус разликата на брзината која може да се постигне по должина на претходниот сегмент и ова се пресметува и за реалната брзина со номиналното забрзување и за глатката/виртуелната брзина со намаленото забрзување. Попрецизно кажано, реалната брзина и глатката/виртуелната брзина ги имаат сите претходно споменати ограничувања со таа разлика што ограничувањето од должината на претходниот сегмент се прави со различно забрзување, сите останати ограничувања се потполно исти.

Слично како кај GRBL и Клипер ги зачувува движењата во бафер и ги обработува однапред за одредување на оптималните брзини на секој сегмент. Обработката на движењата од баферот започнува кога во него ќе се соберат движења со времетраење од најмалку 2 секунди и за нив се извршени сите претходно наведени пресметки и ограничувања.

Со додавање на ново движење во баферот мора да се направи повторна пресметка на брзините на сите движења од баферот, но Клипер за оптимизација повторната пресметка ја прави кога во баферот се соберат нови движења со времетраење од најмалку 250ms. Овој алгоритам за секое движење (блок, сегмент) ги запишува во класата: почетната брзина, номиналната брзина која се постигнува, излезната брзина, времетраењето на делот од сегментот со забрзување, времетраењето на делот од сегментот со константна брзина и времетраењето на делот од сегментот со забавување. Алгоритмот што го извршува ова е даден на слика 3.20.

Одредувањето на оптималните брзини започнува со изминување на баферот со движења од крај кон почеток, со тоа што крајната брзина секогаш се поставува на нула. Ова се прави со цел машината глатко да застане во случај кога нема повеќе движења или од страна на корисникот е паузирана или стопирана работата на истата.



Слика 3.20. Алгоритам за пресметување на оптималните брзини по сегменти кај Клипер.

Следно се определуваат почетните брзини на сегментите кои можат да се постигнат од крајната брзина по должина на сегментот (ова е обратно од она што веќе е направено со изразите 3.90 и 3.99). Пресметката за секој сегмент се прави со следниот израз:

$$v_{reach_start}^2 = v_{end}^2 + dv^2 \quad (3.100)$$

истото се пресметува и за глатката/виртуелната брзина:

$$v_{reach_smooth}^2 = v_{smooth_next}^2 + dv_{smooth}^2 \quad (3.101)$$

Понатаму, се прави ограничување на овие брзини со брзините добиени од изразите (3.98) и (3.99) за да се вкалкулираат и сите претходно пресметани ограничувања:

$$v_{start}^2 = \min(v_{max_start}^2, v_{reach_start}^2) \quad (3.102)$$

истото се применува и за глатката/виртуелната брзина:

$$v_{smooth}^2 = \min(v_{max_smooth}^2, v_{reach_smooth}^2) \quad (3.103)$$

Од добиените вредности прво се споредува брзината v_{smooth}^2 со $v_{reach_smooth}^2$ и притоа од изразот (3.103) имаме дека виртуелната почетна брзина на тековниот сегмент v_{smooth}^2 може да биде или помала или еднаква на брзината која може да се постигне од крај на сегментот по должината на истиот $v_{reach_smooth}^2$. Во случајот кога овие две брзини се еднакви (опција 4 од слика 3.21) тоа значи дека крајната брзина може да се постигне само со забавување по должина на целиот сегмент почнувајќи од почетната брзина на сегментот. Поточно кажано, се работи за сегмент само со забавување иако алгоритмот ги разгледува виртуелните брзини. Сегменти од овој тип може да има последователно повеќе од еден и бидејќи се разгледуваат виртуелните брзини овие сегменти се ставаат во привремена низа (delayed[]) се додека не се стигне до сегмент кој има и забрзување (опција 1,2 и 3 од слика 3.21) со цел да се пресмета максималната номинална брзина за тој сегмент. Пресметаната максимална номинална брзина за тој сегмент може да влијае на следните кратки сегменти само со забавување дел од нив да се трансформираат во сегменти само со константна брзина, а можно е да има и сегмент со константна брзина и забавување. Во привремената низа (delayed[]) се зачувуваат движењата со квадратот на влезната брзина и квадратот на излезната брзина.

За сегментите, блоковите за кои важи дека $v_{smooth}^2 < v_{reach_smooth}^2$ (опција 1,2 и 3 од слика 3.21) што значи дека овие блокови имаат и забрзување, се врши дополнителна проверка на брзината која се постигнува од почетната виртуелна брзина по должина на целиот сегмент $v_{smooth}^2 + dv_{smooth}^2$ со брзината на крајот на сегментот, односно почетокот на следниот сегмент $v_{smooth_next}^2$. Кога брзината $v_{smooth}^2 + dv_{smooth}^2$ е поголема во однос на брзината $v_{smooth_next}^2$ (опција 1 и 3 од слика 3.21) се работи за триаголен или трапезоиден облик и тука се пресметува максималната номинална брзина која претставува минимална вредност од брзината која се добива од G-кодот и аритметичката средна од квадратите на брзините: v_{smooth}^2 и $v_{reach_smooth}^2$ според следниот израз:

$$v_{peak_cruise}^2 = \min\left(v_{max_cruise}^2, \frac{v_{smooth}^2 + v_{reach_smooth}^2}{2}\right) \quad (3.104)$$

Ако добро се анализира аритметичката средина на квадратите на брзините v_{smooth}^2 и $v_{reach_smooth}^2$ ќе видиме дека таа претставува максималната брзина (на квадрат) која може да се постигне за тековниот сегмент во случај на триаголен профил на брзината. Пресметките со виртуелни брзини, го прават и триаголниот профил на брзината да биде виртуелен (испрекината сива линија од слика 3.18) и тој само ќе ја лимитира максималната номинална брзина, но профилот со реалните забрзувања ќе биде трапезоиден бидејќи реалните забрзувања се поголеми од виртуелните.

Кога ќе се пресмета вредноста за максималната номинална брзина, се изминува привремената низа во обратен редослед и притоа се ограничуваат почетната брзина, максималната брзина на делот од сегментот со константна брзина и крајната брзина, сите на добиената максимална номинална брзина. Во привремената низа сегментите/блоковите се зачувани со почетна и крајна брзина и од нив заедно со максималната номинална брзина за долгиот сегмент кој претходи на оваа секвенца од кратки сегменти се воведува нова брзина според следниот израз:

$$v_c^2 = \min(v_{peak_cruise}^2, v_s^2) \quad (3.105)$$

каде v_s^2 е почетната брзина со која е зачуван сегментот во привремената низа, а $v_{peak_cruise}^2$ е добиената брзина од изразот (3.104). Следно за кратките сегменти од привремената низа се запишуваат влезната брзина, максималната брзина за константниот дел од сегментот и крајната брзина со помош на функцијата дадена во продолжение:

$$set_junction(\min(v_c^2, v_s^2), v_c^2, \min(v_c^2, v_s^2)) \quad (3.106)$$

Оваа функција ги пресметува брзините само со коренување на примените аргументи:

$$v_{start} = \sqrt{v_{start}^2}; v_{cruise} = \sqrt{v_{cruise}^2}; v_{end} = \sqrt{v_{end}^2}; \quad (3.107)$$

исто така и растојанијата за делот од сегментот со забрзување, константна брзина и забавување со следниве изрази:

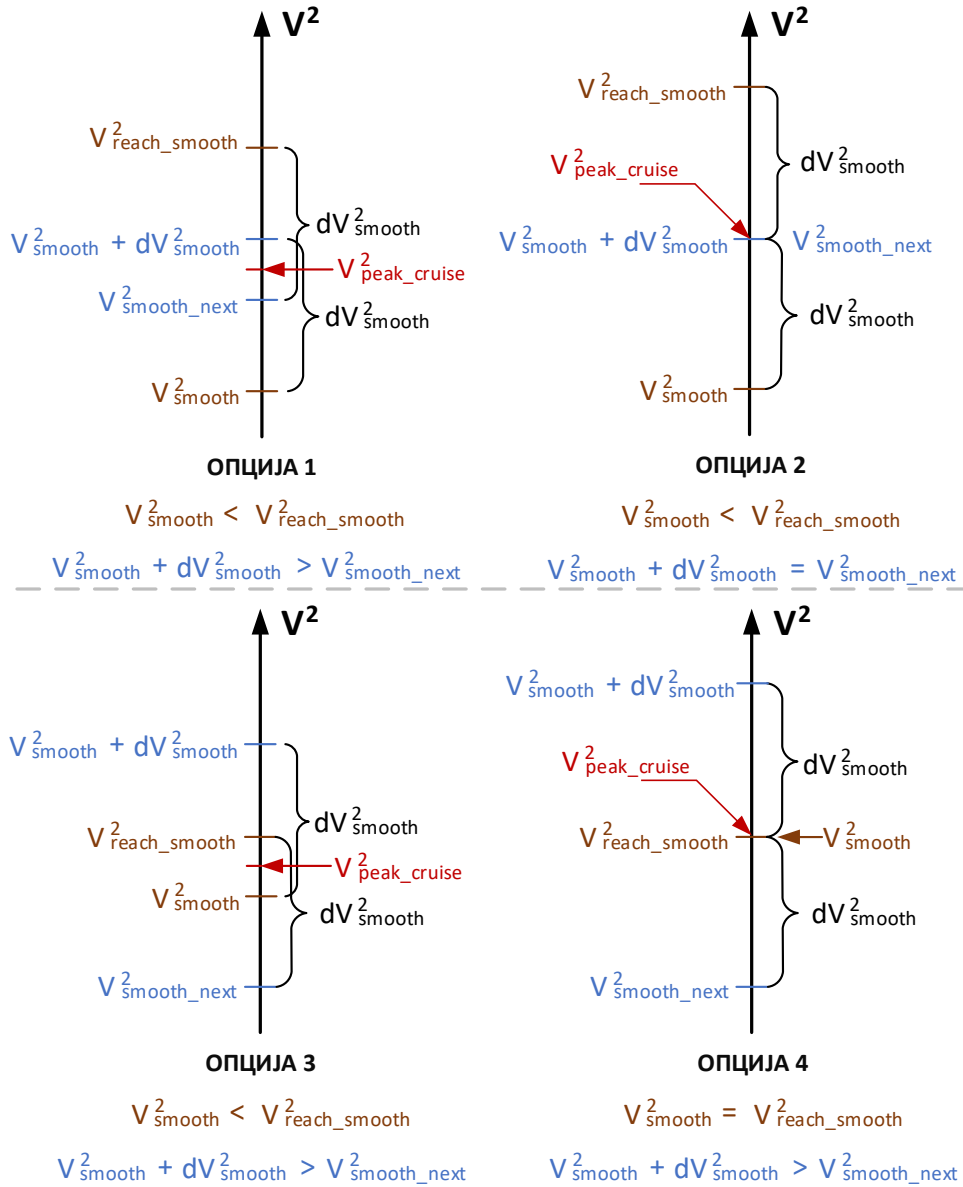
$$S_{acc} = \frac{v_{cruise}^2 - v_{start}^2}{2a}; S_{dec} = \frac{v_{cruise}^2 - v_{end}^2}{2a}; S_{cruise} = S - S_{acc} - S_{dec} \quad (3.108)$$

и на крај ги пресметува и времињата потребни за секој дел од сегментот:

$$t_{acc} = \frac{S_{acc}}{\frac{v_{start} + v_{cruise}}{2}}; t_{cruise} = \frac{S_{cruise}}{v_{cruise}}; t_{dec} = \frac{S_{dec}}{\frac{v_{end} + v_{cruise}}{2}}; \quad (3.109)$$

Во моментот кога сите движења од привремената листа се обработени, листата се бриши. Во случај кога листата нема елементи или е измината и избришана следно се пресметува v_{cruise}^2 брзината според следниот израз:

$$v_{cruise}^2 = \min\left(\frac{v_{start}^2 + v_{reach_start}^2}{2}, v_{max_cruise}^2, v_{peak_cruise}^2\right) \quad (3.110)$$



Слика 3.21. Споредба на виртуелните брзини кои ги користи алгоритмот за пресметка на оптимални брзини на блоковите.

Оваа брзина претставува реалната конечната максимална номинална брзина за сегментот (брзината за делот од сегментот со константна брзина) која ги зема во предвид и реалните брзини со реалното забрзување и виртуелните брзини со виртуелното забрзување преку брзината $v_{peak_cruise}^2$. Ова не се пресметува за кратките сегменти само со забавување, бидејќи за нив се поставени номиналните брзини и дефиниран е профилот на брзините. Првиот аргумент $\frac{v_{start}^2 + v_{reach_start}^2}{2}$ претставува максимална брзина (на квадрат) која може да се постигне по должина на тековниот сегмент при триаголен профил на брзината пресметано со реалното забрзување. Вториот аргумент $v_{max_cruise}^2$ претставува максималната брзина добиена од G-кодот ограничена од максималната брзина на механичката конструкција и моторите. Конечно третиот аргумент е виртуелната максимална номинална брзина која може да се постигне со виртуелното забрзување. Со ова брзината која се поставува ќе биде минимална од овие три брзини

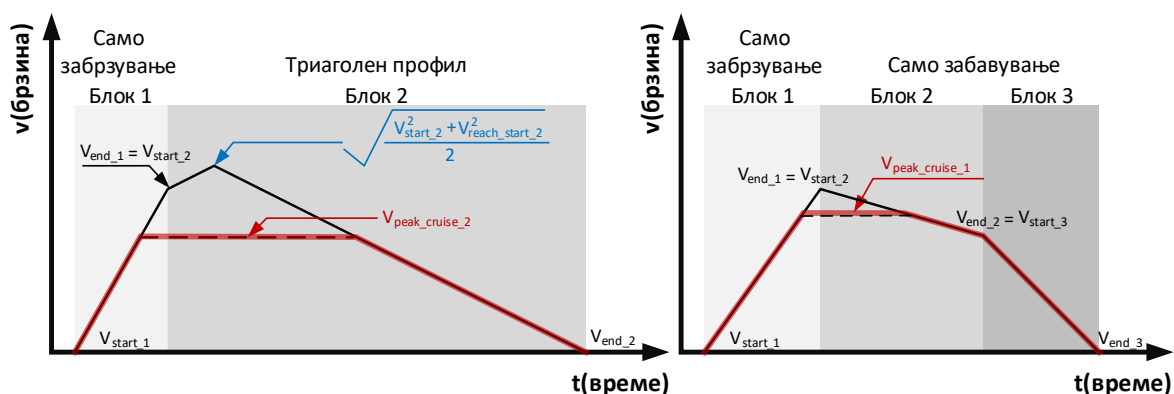
што ги зема во предвид сите ограничувања, па се повикува функцијата за поставување на брзините:

$$set_junction(\min(v_{start}^2, v_{cruise}^2), v_{cruise}^2, \min(v_{end}^2, v_{cruise}^2)) \quad (3.111)$$

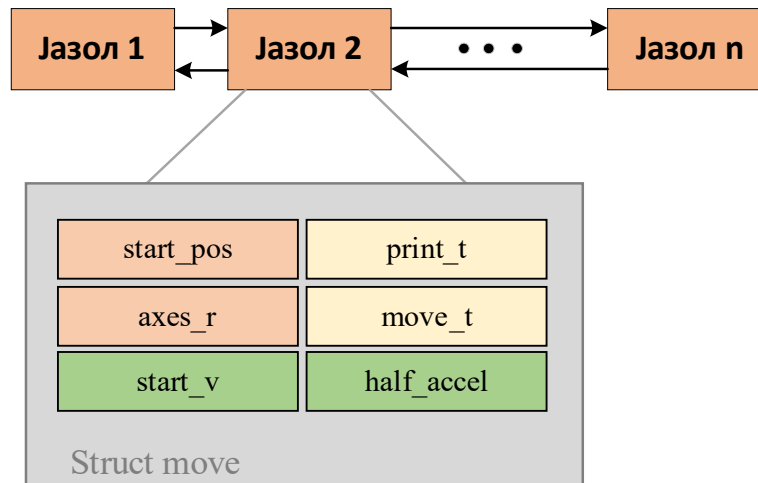
Оваа пресметка се прави за трапезоидните профили на брзини, триаголните профили кои со виртуелната брзина се трансформираат во трапезоидни, но и за сегментите кои имаат само забрзување. Сегментите само со забрзување се оние за кои важи $v_{smooth}^2 + dv_{smooth}^2 = v_{smooth_next}^2$, за овие сегменти не се пресметува $v_{peak_cruise}^2$ во случај кога привремената листа е празна и се користи пресметаната вредност од следниот сегмент (што е претходно обработен во баферот, бидејќи баферот се обработува во обратен редослед). Во ваков случај значи дека после тековниот сегмент со забрзување има сегмент триаголен, односно трапезоиден и тој веќе е обработен и за него е пресметана максималната номинална виртуелна брзина $v_{peak_cruise}^2$, па за тековниот сегмент само со забрзување потребно е ограничувањето да се постави на максималната номинална виртуелна брзина од следниот сегмент. На левиот график од слика 3.22 е прикажан еден ваков пример каде со црна линија се означени дадените брзини со иницијалните профили, а со црвена линија е означен новиот профил на брзина со кој се ограничува максималната брзина, но се внесува движење со константна брзина со што се избегнува нагло менување на брзината.

Случајот кога привремената низа има елементи и се појавува сегмент со само забрзување, тогаш се пресметува максималната номинална брзина за него и ова претставува сегмент со забрзување по кој следат сегмент/и само со забавување. Со примена на виртуелната брзина во спојот на сегментот/блокот само со забрзување и кратките сегменти само со забавување после него се додава дел со константна брзина. На слика 3.22 десно е претставен ваков пример каде со црна линија се означени дадените брзини со иницијалните профили, а со црвена линија е означен новиот профил на брзината со воведен дел со константна брзина.

Кога за движењата кои се добиени од G-кодот ќе се одредат оптималните брзини со примена на алгоритмот од слика 3.20 овие блокови/сегменти се праќаат на понатамошна обработка за да се пресметаат времињата на секој чекор од движењето. За таа цел кодот за пресметка и дообработка на веќе претходно припремените податоци е напишан во програмскиот јазик C за поголема ефикасност и брзина.



Слика 3.22. Примери за блокови само со забрзување проследени од триаголен блок (лев график) и мали кратки сегменти само со забавување (десен график).



Слика 3.23. Двојно поврзана листа во која дополнително се делат блоковите по сегменти со забрзување, забавување и константни брзини.

Добиените движења се делат на дополнителни сегменти и се зачувуваат во структурата Move. Во оваа структура претходно обработените блокови кои имаат триаголен или трапезоиден профил на брзината се делат на два или три елементи во оваа структура соодветно. Триаголните блокови се делат на сегмент со забрзување и сегмент со забавување, додека пак трапезоидните се делат на три сегменти: сегмент само со забрзување, сегмент со константна брзина и сегмент само со забавување. Во оваа структура секој елемент т.е. сегмент е од еден од трите основни типови, нема комбинирани блокови. Оваа структура ги содржи следните елементи:

- Start_pos – почетна позиција на сегментот;
- Axes_r – односот на растојанието на секоја оска за сегментот со должината на сегментот S добиен со изразот (3.77);
- Start_v – почетна брзина на сегментот;
- Print_t – глобално време на почеток на сегментот кое служи за комплетна синхронизација помеѓу целиот контролен систем;
- Move_t – времетраење на сегментот;
- Half_accel – забрзување за тековниот сегмент помножено со 0.5 за полесна пресметка понатаму.

Вака добиените сегменти се чуваат во двојно поврзана листа со динамичка алокација на меморија за поефикасно искористување на истата, прикажана на слика 3.23. Попрецизно кажано движењата се чуваат во структурата `trapq` во која има две двојно поврзани листи: едната со движењата кои допрва треба да се извршат и едната со историјата на веќе поминатите движења која се бриши до одреден период во минатото. Поделените сегменти се праќаат на понатамошна обработка за одредување на времињата на секој чекор.

Овој процес на пресметување на времињата на секој чекор се одвива во рамките на класата „Kinematic“. Надвор од оваа класа се е претставено преку милиметри, секунди и во Декартов координатен систем. Задачата на оваа класа е да направи конверзија од стандардниот координатен систем во координатниот систем специфичен за дадениот печатар. Кодот користи егзактни кинематички формули за пресметување на точните декартови координати на главата на машината во секој момент од времето и егзактни кинематички формули за изведување на точните позиции на секој од моторите за така

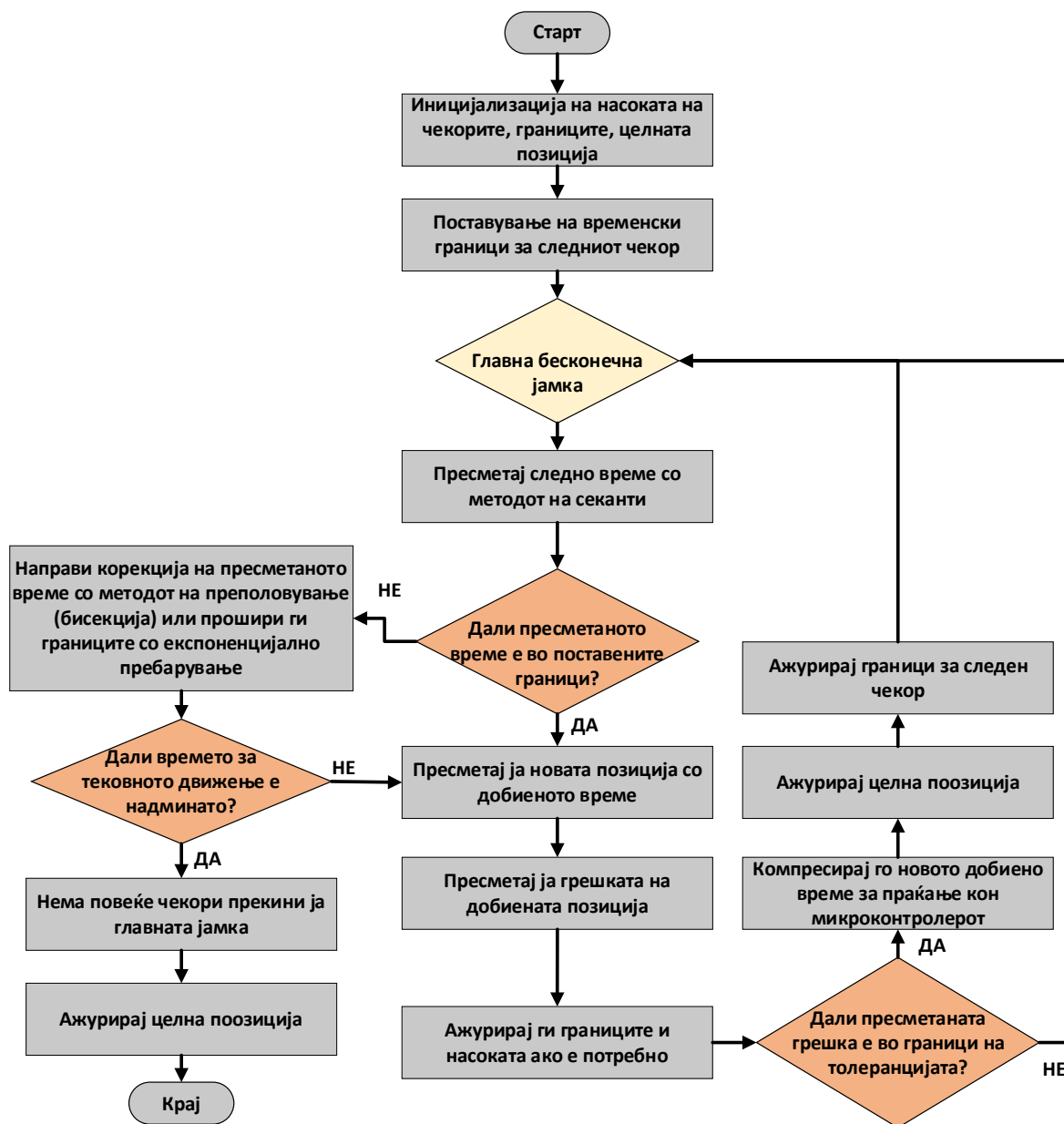
добиените координати. Додека пак за добивање на прецизните времиња на секој чекор за секој мотор Клипер користи нумерички методи имплементирани во програмскиот јазик C за постигнување на повисоки перформанси. За бараните времиња може да се работи и со точна формула кога имаме трапезоидни профили на брзини, но со нумеричките методи се контролира точноста на времето на секој чекор и е добра основа за понатамошно имплементирање на S профил на забрзување со кој се гарантира глаткост и на изводот на забрзувањето, а со тоа и уште поглатко движење на самата машина.

Без разлика дали се работи за чекорни мотори или сервомотори, еден импулс од микроконтролерот одговара на еден чекор на моторот, односно еден микрочекор. Резолуцијата и на сервомоторите, иако доста поголема во однос на чекорните мотори, сепак е ограничена со резолуцијата на енкодерите. Најчесто на опаѓачки раб на импулсот од микроконтролерот моторот се поместува за еден чекор/микрочекор. Затоа Клипер го пресметува времето кога соодветната оска треба да биде на средината помеѓу чекорите. Алгоритамот за пресметка на точните времиња на чекорите е даден на слика 3.24. За едно движење зачувано во двојно поврзаните листи Клипер пресметува чекори за секоја оска кај која има движење.

Алгоритамот почнува со иницијализација и поставување на целната позиција за која треба да се определи точното време и иницијални граници во кои треба да се наоѓа ова време. За временските граници се зема почетното време на чекорот и крајното време кое се добива како збир од почетното време и предефинирана константа која претставува времетраење на чекорот при најбавно можно движење на моторот. Понатаму со методата на секанти како главна нумеричка метода се прави итеративно пресметување на времето. Ако добиената вредност при тековната итерација не е во границите се повторува пресметувањето на времето со методот на преполовување (бисекција) или се прошируваат границите со методот на експоненцијално пребарување. При секоја итерација ново добиената вредност за времето се проверува дали го надминува предвиденото време за целиот сегмент, во таков случај тоа сигнализира дека нема повеќе чекори и дека пресметките на времињата за чекорите од дадениот сегмент се завршени, а со тоа завршува и алгоритамот.

Во случаи кога добиеното време е во рамките на предвидените граници или е искористен методот на преполовување или се ажурирани границите со помош на методот на експоненцијално пребарување се пресметува со егзактна формула новата позиција од добиеното време. Разликата од пресметаната позиција и целната позиција ја дава грешката на добиената позиција. Понатаму се ажурираат границите за следното погодување на времето и се проверува дали грешката е помала од зададениот праг. Прагот кој го користи Клипер е $10^{-9}m$. Во случај кога грешката е поголема, алгоритамот изведува нова итерација со нумеричките методи и со ново добиената вредност за времето се повторува целата претходно објаснета постапка се додека не се добие време со кое пресметаната позиција е со грешка помала од прагот. Добиеното време се праќа на компресија за испраќање кон микроконтролерот.

Пристапот со пресметување на точното време на секој импулс, односно чекор бара голема меморија што од страната на хост компјутерот не е проблем, меѓутоа од страната на микроконтролерите може да предизвика проблем со немање на доволно меморија и ресурси. Затоа Клипер дефинира структура `step_move` прикажана на слика 3.25 во која се пакуваат чекорите и во тој облик се испраќаат кон микроконтролерот или микроконтролерите задолжени за управување на чекорните мотори на машината.

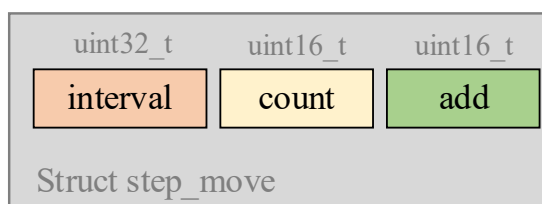


Слика 3.24. Алгоритам кој применува нумерички методи за пресметка на времетраењето на секој импулс/чекор на моторите.

Структурата има три елементи првиот елемент (*interval*) го претставува интервалот помеѓу чекорите, односно времето кое треба да помине од последниот чекор до тековниот чекор. Вториот елемент (*count*) го претставува бројот на чекори кои се спакувани во секвенцата и конечно третиот елемент (*add*) е квадратен коефициент на корекција со кој се застапува забрзувањето или забавувањето. Ова значи дека само со еден елемент од структурата кои содржи 3 променливи може да се спакуваат голем број на чекори.

Времето на *n*-тиот елемент се пресметува според следната формула:

$$t_n = t_0 + n \times interval + \frac{n(n-1)}{2} \times add \quad (3.112)$$



Слика 3.25. Структура во која се чуваат пакуваните чекори за праќање кон микроконтролерот.

Оваа формула овозможува секвенцата да репрезентира чекори со линеарна или квадратна зависност на брзината што е корисно за профили на движења со забрзување и забавување. Елементот t_0 го претставува почетното време на целата секвенца (што во конкретниот случај е нула), $n \times interval$ ги опфаќа чекорите со константна брзина и третиот елемент $\frac{n(n-1)}{2} \times add$ го претставува забрзувањето односно забавувањето. За да се најде времето на n -тиот чекор се зема почетното време на него се додава времето на сите чекори при константна брзина и се додава или одзема во зависност од тоа дали се работи за забрзување или забавување збирот на сите броеви од 1 до $n - 1$ помножен со коефициентот add . Коефициентот е позитивен за забрзување и негативен за забавување.

Генерирањето на импулсот од страна на контролерот во интервал помеѓу половина чекор пред и по точното време на тековниот чекор нема да предизвика промени во прецизноста на самата машина. Според ова Клипер дефинира максимални граници на грешка за секој чекор и во рамките на овие граници го прави пакувањето на чекорите со што е можно повеќе чекори во еден елемент од структурата за поефикасен пренос и минимизирање на потребната меморија за оваа цел.

На страната на микроконтролерот настаните, односно временските интервали за секој чекор се зачувуваат во соодветни приоритетни бафери. Клипер генерирањето на импулсите за чекорите го прави преку еден хардверски тајмер со висока резолуција како извор за синхронизација на сите оски. Според тоа сите чекори имаат единствено време во кое треба да се извршат синхронизирано со тајмерот. Организацијата на генерирање на чекорите е во вид на настани. За секој настан тајмерот генерира прекин, во чија рутина микроконтролерот го активира соодветниот пин за изведување на чекорот по соодветната оска, го ажурира времето за следниот настан во compare регистарот од тајмерот и ја враќа контролата на главната јамка. Импулсите кон драјверите од чекорните мотори мора да имаат и опаѓачка ивица која се дефинира како посебен настан што значи дека Клипер генерира два прекини за еден чекор. На овој начин Клипер изведува високо прецизното синхроно движење по сите оски, но бара два прекински настани за еден чекор што е проблематично за високи брзини, особено при движење на повеќе оски истовремено.

Управувањето со опаѓачката ивица на импулсите може да се направи и со посебен тајмер кој ќе се тригерира од главниот тајмер кога пулсот е активен на високо ниво и по фиксно време ќе генерира прекин и ќе го врати пинот на нула, но сепак ова нема да го надмине проблемот со два прекини за еден чекор, само ќе го намали товарот на главниот тајмер. Задачите кои треба да ги изведе прекинската рутина на главниот тајмер кога ќе настане прекин се: пристап до соодветниот пин и негова промена на состојбата, земање на времето за следниот настан/чекор од меморија и поставување во compare регистарот на тајмерот што и кај понапредните микроконтролери би зафатило на 150 такт циклуси на јадрото. Временските периоди на чекорите може да се многу блиски по различните

оски, па при високи брзини времето на извршување на самата сервисрутина може да го надмине времето во кое треба да се изврши следниот чекор од баферот со што воведува временски цитер (варијација во временскиот распоред на сигналите, односно отстапување од посакуваниот период на импулсите), односно задоцнето извршување на чекорите. Со моменталните брзини и резолуции на 3D-печатарите ваквото извршување на чекорите ги задоволува потребите, но во иднина, зголемувањата на брзината ќе доведе до потреба од нов пристап на извршување на чекорите и генерирање на импулсите.

Друго тесно грло кај Клипер е комуникацијата помеѓу процесорот и микроконтролерот која исто така со моменталните брзини и резолуции ги задоволува потребите, но во иднина ќе биде едно од главните ограничувања на овој фирмвер. Комуникациите кои се користат се најчесто: UART (сериска комуникација), CAN или USB-комуникација. Пристапот на Клипер иако направи револуција во 3D-печатењето и зголемувањето на брзината на оваа технологија, во блиска иднина ќе се постигнат максималните можности на Клипер, па ќе мора да се изнајде нов иновативен пристап што ќе овозможи уште поголеми брзини на печатење и дополнително подобрување на квалитетот, а ќе ги надмине наведените проблеми на овој фирмвер.

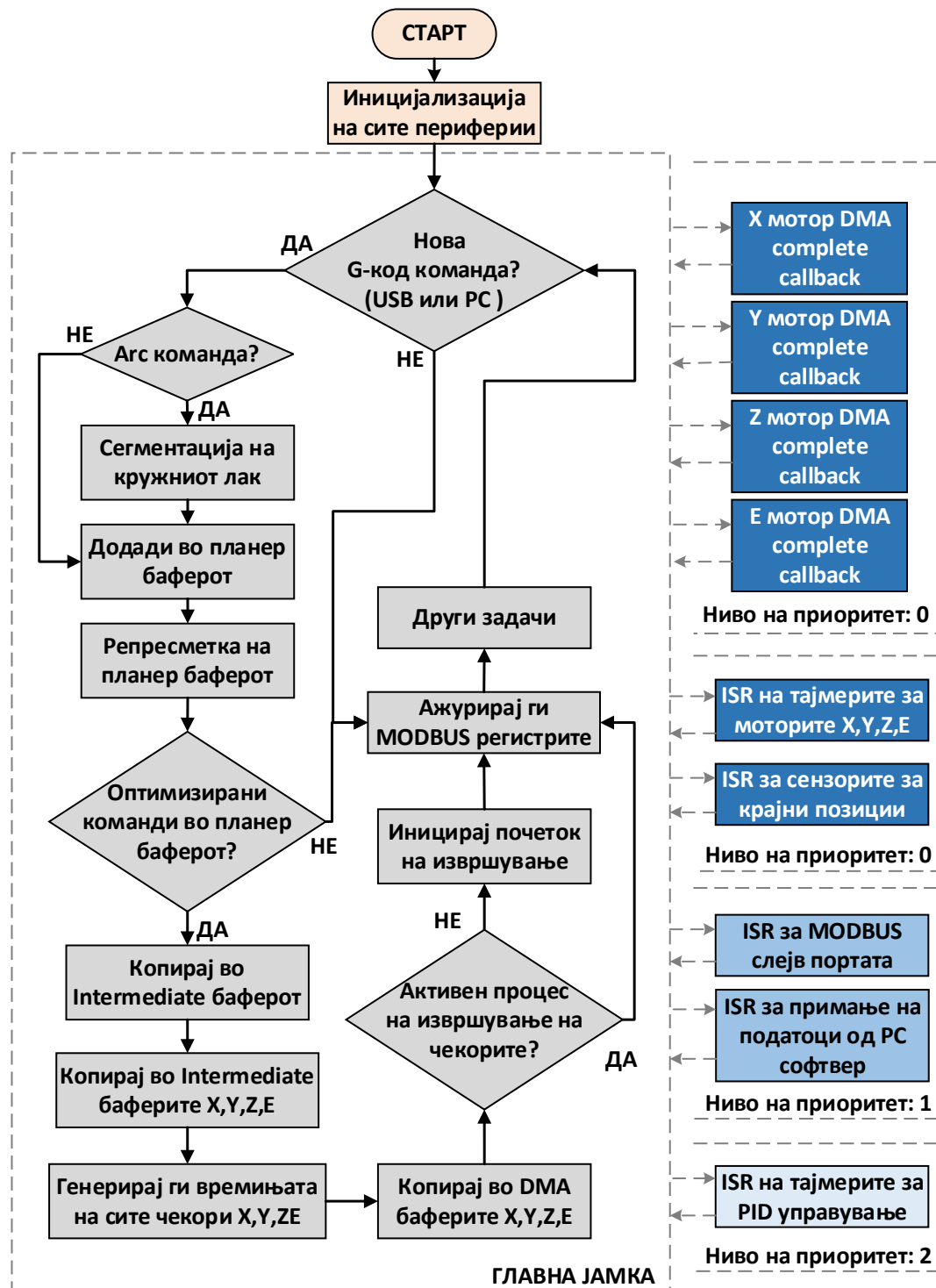
3.1.3 Архитектура на фирмверот на предложениот управувачки систем

Предложениот фирмвер во оваа докторска дисертација користи нов иновативен пристап кој ги комбинира едноставноста на GRBL-фирмверот со прецизниот временски модел на Клипер на хардверска платформа со ограничени ресурси базирана само на еден микроконтролер детално објаснета во делот 2. Системот е базиран комплетно на STM32F767-микроконтролер, кој обезбедува доволно процесирачка моќ за да управува и со планирањето на движењето и со реално-временското генерирање на чекорите без користење на надворешен процесор. Фирмверот лесно може да се прилагоди и на било кој микроконтролер со повисоки перформанси, а особени резултати можат да се постигнат со новите микроконтролери кои веќе се во фаза на тестирање и набрзо се очекува да бидат достапни на пазарот за широка потрошувачка. За разлика од GRBL, предложениот фирмвер ја елиминира сегментацијата на линеарните сегменти на мали потсегменти со константни брзини, наместо тоа извршува прецизна пресметка на времето за секоја точка од траекторијата на движење во реално време.

Ова се постигнува со максимално искористување на перифериите на микроконтролерот, особено хардверските тајмери и контролерот за директен мемориски пристап (DMA), со цел минимизирање на оптоварувањето на јадрото. Секоја оска од машината и екструдерот се управувани од посебен хардверски тајмер, што овозможува перфектна синхронизација и прецизна координација на сите оски. Фирмверот вклучува механизам за компензација на грешките настанати како резултат од заокружувањето на децималните броеви при кинематичките пресметки на времињата за секој чекор, со што се постигнуваат високи перформанси на управување со движењето на оските и нивна идеална синхронизација.

На слика 3.26 е претставена поедноставена верзија на управувачкиот алгоритам имплементиран во предложениот фирмвер, прикажувајќи го текот на процесите кога на машината е инициран процес на обработка или 3D-печатење. Истата структура на бафери и комплетниот управувачки алгоритам се користат и за обработка и за 3D-печатење, овозможувајќи унифициран пристап на планирање и извршување на движењата. Сепак, одделни функции/рутини се имплементирани за да се задоволат специфичните барања на секој процес. Како на пример, функциите за печатење

вклучуваат и дополнителни калкулации за количината на материјал кој треба да се екструдира и синхронизацијата со останатите оски, додека функциите за обработка се фокусираат на брзината на ротација на главата за обработка (спиндлата) и прецизното имплементирање на движењето на алатот за обработка. Како што беше наведено и претходно, моменталната верзија на овој алгоритам поддржува само секвенцијално хибридно производство, каде 3D-печатењето и обработката се изведуваат последователно, наместо симултано.



Слика 3.26. Поедноставена верзија на управувачкиот алгоритам на предложениот фирмвер.

Предложениот фирмвер е компатибилен со поголемиот дел од софтверските алатки достапни на пазарот кои вршат трансформација (раслојување) на 3D-модел во G-код за CNC-обработка или 3D-печатење. Во фирмверот се имплементирани повеќе начини за читање и извршување на G-код командите од генерираниот фајл. Едната можност овозможува директен пренос на фајлот преку USB-мемориски уред или SD-мемориска картичка, при што машината може да функционира самостојно, без потреба од поврзување со персонален компјутер. Алтернативно, управувањето може да се изврши преку специјализиран компјутерски софтвер развиен за оваа машина, при што G-код командите се испраќаат директно од персонален компјутер и се извршуваат во реално време. Без разлика на избраниот начин на пренос, процесот на извршување на G-кодот останува идентичен. При приемот, односно читањето на секоја нова команда, фирмверот проверува дали се работи за кружна интерполација, односно движење по кружен лак. Во случај на вакви команди, најпрво се извршува сегментација, односно делење на кружниот лак на серија мали линеарни сегменти. Концептот на ваква апроксимација на кружните движења (извршување на G2 и G3-командите) е идентичен кај комерцијалните, како и кај отворените и слободни фирмвери наменети за CNC-обработка и 3D-печатење.

Како што беше детално објаснето во деловите 3.1.1 и 3.1.2, фирмверите GRBL и Клипер користат речиси идентичен математички модел за сегментација на кружните движења во серија од мали линеарни сегменти. Сепак, постои суштинска разлика во начинот на дефинирање на бројот на сегменти на кои се дели кружниот лак. GRBL користи параметар `arc_tol`, кој ја претставува максимално дозволената геометриска грешка, односно отстапувањето од идеалниот кружен лак при неговата апроксимација со линеарни сегменти, како што е прикажано на слика 3.4. Овој параметар е дефиниран од страна на корисникот, а врз основа на неговата вредност GRBL автоматски го пресметува потребниот број на сегменти за даденото кружно движење. Од друга страна, Клипер користи поинаков пристап со дефинирање на фиксна должина на линеарниот сегмент, изразена во милиметри, независно од радиусот на кружниот лак. Како резултат на ваквиот пристап, геометриската грешка варира во зависност од радиусот на кружното движење, при што кај помали радиуси грешката е поголема, а кај поголеми радиуси се намалува. Поради овие причини, предложениот фирмвер го применува пристапот на GRBL, односно користи параметар за максимално дозволена грешка, со што се обезбедува контролирано и предвидливо отстапување од посакуваната патека при сегментацијата на кружните движења.

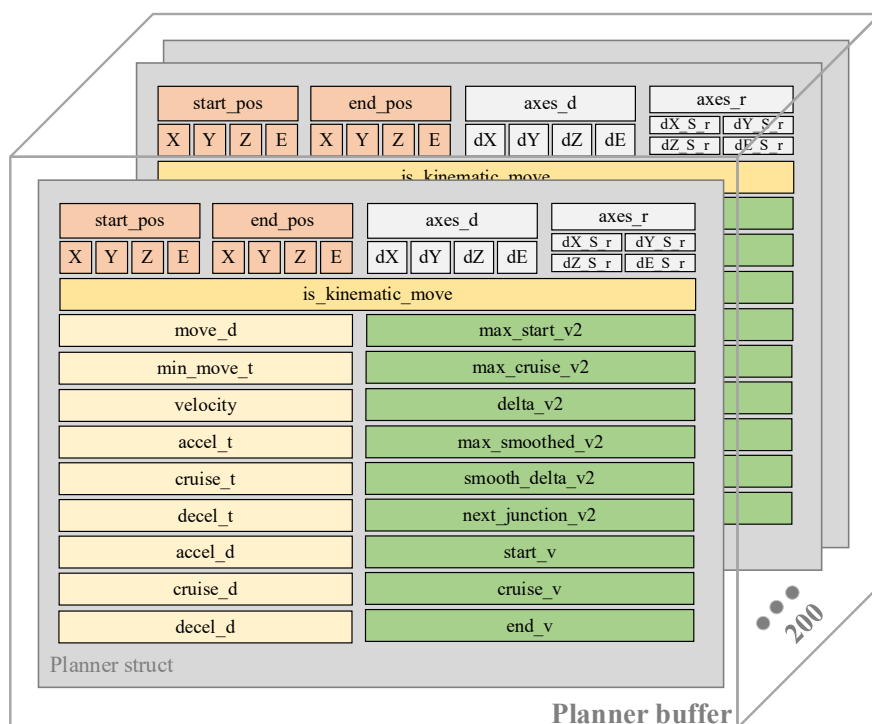
Дополнително, GRBL имплементира оптимизација на пресметувањето на тригонометриските функции со тоа што синусот и косинусот од аголот θ_{seg} се пресметуваат само еднаш, а потоа се користат рекурзивно во трансформационата матрица (3.21). Со цел да се ограничи акумулацијата на нумеричка грешка која настанува при ваквиот рекурзивен пристап, периодично се применува корекција, при што на секој дванаесетти сегмент повторно се пресметуваат тригонометриските функции врз основа на целосниот агол. Наспроти овој пристап, Клипер и предложениот фирмвер го пресметуваат секој сегмент независно, директно од иницијалната точка, со пресметување на тригонометриските функции за вкупниот агол на сегментот со трансформационата матрица (3.73). На овој начин се избегнува кумулативната нумеричка грешка и се постигнува повисока нумеричка прецизност при апроксимацијата на кружните движења.

Генерираните линеарни сегменти од кружните движења се сместуваат во привремен бафер. Овие сегменти се префрлуваат во планер баферот во зависност од

слободните места во истиот, бидејќи за разлика од Клипер предложениот фирмвер има ограничени ресурси (РАМ меморија), па планер баферот е со ограничена и фиксна должина од 200 елементи. Во планер баферот се пресметуваат профилите на брзините за сите сегменти. Во случај на линеарни сегменти добиени како G1 команди, сегментите се испраќаат директно во планер баферот (привремениот бафер се користи само за кружни сегменти), што значи дека планерот содржи само линеарни сегменти, без разлика дали тие потекнуваат од G1, G2 или G3-команди.

Процесот на додавање нови елементи во планер баферот, како и пресметката на брзините во спојните точки, почетните и крајните брзини кај предложениот фирмвер, се изведуваат на концептуално сличен начин како кај Клипер, но со дополнителни оптимизации насочени кон ефикасно извршување на платформа базирана само на еден микроконтролер. Дополнително, предложениот фирмвер го користи идентичниот look-ahead алгоритам (алгоритам кој зема во предвид поголем број на последователни сегменти со цел пресметување на оптималните брзини на премин помеѓу нив) и основните концепти развиени во Клипер, но имплементирани во поедноставена, детерминистичка форма, напишана во програмскиот јазик C и прилагодена за вградливи микрокомпјутерски системи со ограничени ресурси.

На слика 3.27 е прикажана структурата на планер баферот, која претставува основа на look-ahead алгоритмот. Елементите во оваа структура се концептуално идентични со елементите од класата *Move* кај Клипер, со таа разлика што во предложениот фирмвер дополнително се чуваат три параметри кои ги опишуваат должините на сегментите со забрзување, забавување и константна брзина на кои е поделен секој сегмент, односно блок. Овој бафер содржи фиксно 200 елементи, што е директна последица на ограничените ресурси на микроконтролерот, за разлика од Клипер кој користи динамичка алокација на меморија и пресметките ги врши на високо ниво на процесорот. Секој елемент во планерот зафаќа приближно 272 бајти, што резултира со вкупна мемориска побарувачка од околу 54.4 kB за целиот бафер од 200 елементи.

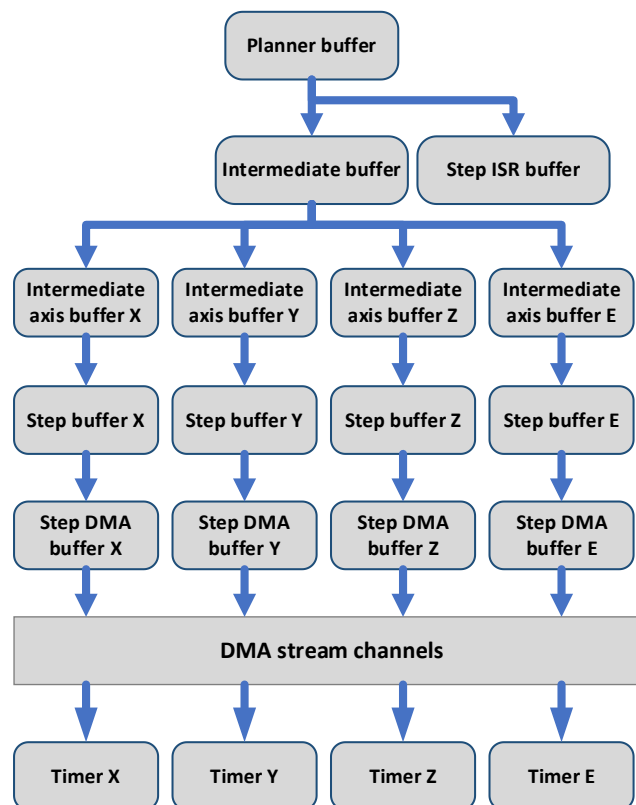


Слика 3.27. Структура на планер баферот која се користи кај предложениот фирмвер.

Математичкиот модел и постапката за пресметка на оптималните брзини во предложениот алгоритам се имплементирани на концептуално идентичен начин како кај Клипер, со користење на look-ahead алгоритмот. Дополнително, во предложениот алгоритам е имплементиран и механизам за измазнување на профилите на брзините воведен од Клипер, со кој триаголните профили на брзина се трансформираат во трапезоидни, преку воведување на виртуелно забрзување a_{smoothed} , како што е прикажано на слика 3.18.

Откако во планер баферот се пресметани оптималните брзини за секој сегмент: влезната брзина, максималната константна брзина и излезната брзина со примена на look-ahead алгоритмот, следниот чекор претставува прецизна пресметка на временските интервали на секој поединечен чекор и нивното извршување во реално време. Овој дел претставува клучна разлика помеѓу предложениот управувачки алгоритам и постоечките решенија, бидејќи имплементацијата на генерирањето на импулсите за чекорите и синхронизацијата помеѓу оските фундаментално се разликува од пристапите што се користат кај отворените и комерцијалните фирмвери за 3D-печатари и CNC-машини.

Предложениот управувачки фирмвер користи комплексен, мемориски бафер систем, наменет за високо прецизна пресметка на временските интервали на сите чекори и за синхроно управување со повеќе оски. Како што е прикажано на слика 3.28, управувачкиот систем се потпира на повеќеслојна мемориска архитектура на бафери која го претставува јадрото на целокупниот управувачки алгоритам. Овие бафери овозможуваат ефикасен и детерминистички трансфер на податоци помеѓу планерот и реално-временскиот алгоритам за пресметка и генерирање на импулсите кон драјверите за моторите.



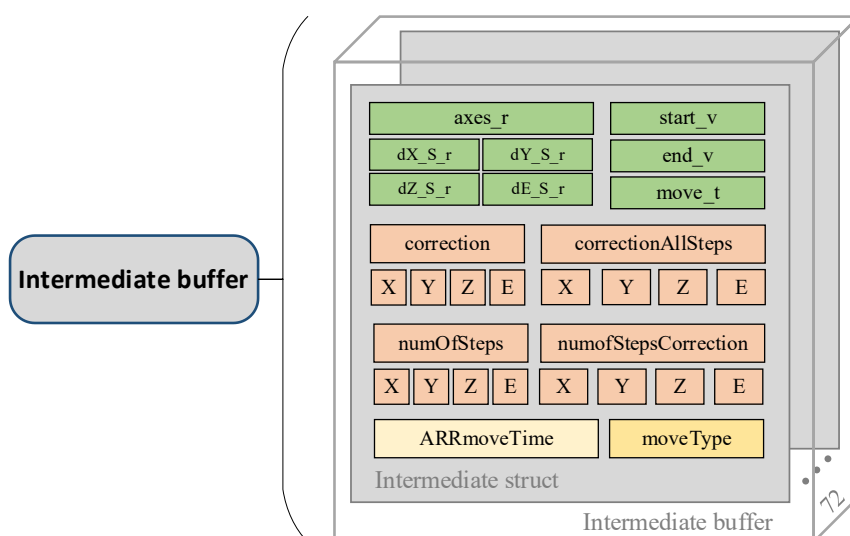
Слика 3.28. Хиерархиска структура на мемориските бафери кај предложениот фирмвер.

Оптимизираниите сегменти во планер баферот се делат на максимум три потсегменти: забрзување, константна брзина и забавување. Оваа поделба се изведува со цел новите сегменти да бидат исклучиво од еден тип на движење (забрзување, константна брзина или забавување). Новите сегменти се трансферираат во Intermediate-баферот, кој чува значајни кинематички параметри за движењето дадени на слика 3.29 како што се: почетната и крајната брзина на сегментот, типот на сегмент (забрзување, константна брзина и забавување) и односите на должините на секоја оска во однос на должината на резултантното поместување S пресметани со изразот (3.78). Дополнително, вкупниот број на чекори за секоја оска за дадениот сегмент е пресметан и зачуван во овој бафер.

За постигнување на висока прецизност кај предложениот фирмвер, имплементиран е механизам за компензација на грешки со цел анулирање на грешките настанати поради заокружување на децималните броеви при пресметка на кинематичките параметри, како и грешките предизвикани од ограничената резолуција на тајмерите. Овој механизам е клучен за постигнување на прецизна и детерминистичка синхронизација помеѓу оските, без која не е можно да се обезбеди висок степен на прецизност на самата машина. Имплементацијата на овој механизам е реализирана преку Intermediate-баферот. Времетраењето на секој сегмент, пресметано во планер баферот, се конвертира од секунди во број на такт циклуси на тајмерот. Ова времетраење е идентично за сите оски кои учествуваат во движењето за дадениот сегмент и се зачувува во 64-битна целобројна променлива (t_{ARR} , односно $ARRmoveTime$ во имплементацијата на фирмверот) во Intermediate-баферот, при што се дефинира како:

$$t_{ARR} = t_{move} \times XYZE_M_NUM_OF_TICKS_PER_SEC \quad (3.113)$$

каде што t_{move} го претставува времетраењето на сегментот определено од look-ahead алгоритмот, а $XYZE_M_NUM_OF_TICKS_PER_SEC$ го претставува бројот на такт циклуси на тајмерот во една секунда, односно ефективната временска резолуцијата на тајмерите кај STM32F767 микроконтролерот. За сите оски се користи максималната достапна резолуција од 108 MHz, со цел обезбедување на униформна и високо прецизна временска база.



Слика 3.29. Структура на Intermediate-баферот кај предложениот фирмвер, со приказ на зачуваните кинематички параметри, податоците за бројот на чекори и променливите за компензација на нумеричките грешки.

За пресметување на параметрите за компензација на грешките кај сегментите од типот забрзување, најпрво се определува времето на извршување на последниот чекор од сегментот во однос на неговиот почеток. Ова време се пресметува со користење на следниот израз:

$$t_{accelX} = \frac{-v_{startX} + \sqrt{v_{startX}^2 + 2a_x dx_{\mu m}}}{a_x} \times XYZ_M_NUM_OF_TICKS_PER_SEC \quad (3.114)$$

каде:

$$v_{startX} = v_{start} \times dX_S_r \quad (3.115)$$

$$a_x = a \times dX_S_r \quad (3.116)$$

$$dx_{\mu m} = X_{numOfSteps} \times XYZ_STEP_RES_{\mu m} \quad (3.117)$$

Со изразите (3.115) – (3.117) се дефинира трансформацијата на кинематичките параметри на главата на машината во кинематички параметри на поединечните оски. Во овој случај анализата се однесува на X-оската, при што пресметките за останатите оски (Y, Z и E) се изведуваат на целосно идентичен начин.

Параметрите v_{startX} и a_x ја претставуваат почетната брзина и забрзувањето на X-оската, добиени преку скалирање на вкупното движење по дадениот сегмент со односот на правецот за X-оската dX_S_r . Со $XYZ_STEP_RES_{\mu m}$ се дефинира резолуцијата на машината по дадената оска. Кај предложената машина, резолуцијата по оските X, Y и Z е идентична и изнесува $1 \mu m$.

Добиената вредност од изразот (3.114) се заокружува и се зачувува во 64-битна целобројна променлива t_{accelX_long} . Разликата помеѓу вкупното времетраење на сегментот t_{ARR} и пресметаното време за последниот чекор t_{accel_long} го дефинира бројот на такт циклуси на тајмерот што треба да се искористат за корекција со цел усогласување на времетраењето помеѓу сите оски:

$$t_{correctionX} = t_{ARR} - t_{accelX_long} \quad (3.118)$$

Добиената вредност ги компензира грешките што произлегуваат од дискретизацијата со ограничената резолуција на тајмерите, како и грешките настанати при заокружување на децималните кинематички параметри. Оваа корекциска вредност $t_{correctionX}$ (аналогно и за Y, Z и E оската) понатаму се трансформира во три параметри кои се зачувуваат во Intermediate-баферот:

- **correctionAllSteps** – го претставува бројот на такт циклуси на тајмерот што се додаваат на секој чекор од сегментот;
- **numOfStepsCorrection** – го претставува бројот на чекори од почетокот на сегментот кај кои се применува дополнителна корекција;
- **correction** – го претставува бројот на дополнителни такт циклуси на тајмерот што се додава на секој од првите $numOfStepsCorrection$ чекори.

Комбинираната примена на овие параметри овозможува високо прецизна временска синхронизација по должина на сите активни оски. Со примената на предложениот модел за корекција, сите оски имаат идентичен временски интервал

(односно идентичен број на циклуси на тајмерот) за дадениот сегмент, што резултира со симултан почеток и крај на движењето.

Дополнително, кумулативните грешки при пресметката на времето за секој чекор ефективно се елиминираат преку пресметување на апсолутното време на секој чекор во однос на почетокот на сегментот, наместо преку инкрементално додавање од претходниот чекор.

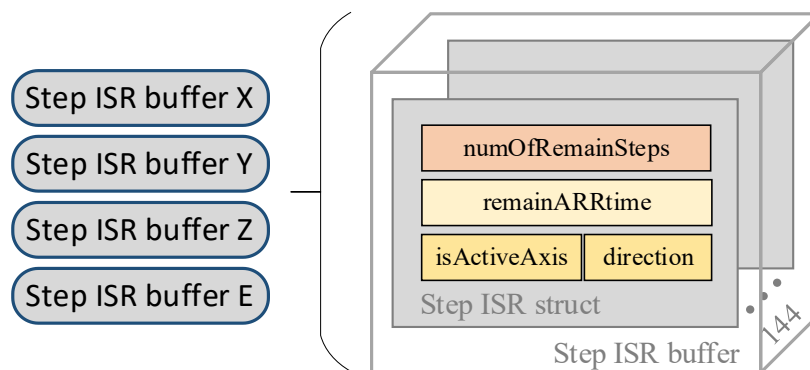
Слична стратегија за компензација на грешките, имплементирана на идентичен начин, се применува и кај сегментите од типот забавување, со цел прецизно определување на параметрите за корекција и постигнување идеална временска синхронизација помеѓу сите активни оски. Кај сегментите со константна брзина, пресметувањето на овие параметри е поедноставно: вкупното времетраење на сегментот, изразено во такт циклуси на тајмерот (t_{ARR}) се дели со вкупниот број на чекори на сегментот. Доколку резултатот од ова делење е цел број, нема потреба од корекција. Меѓутоа, кога остатокот од делењето е различен од нула, истиот се распределува неуниформно од почетокот на сегментот, при што на времетраењето на поединечните чекори се додава или одзема по еден такт циклус на тајмерот се додека остатокот не биде целосно компензиран. Овој пристап овозможува глатка и континуирана синхронизација на движењето по сите активни оски, без појава на кумулативни или позициони грешки.

Предложената архитектура со повеќеслоен мемориски бафер систем, имплементиран механизам за компензација на грешките и максимална временска синхронизација на оските, во комбинација со користење на посебен хардверски тајмер за секоја оска, ја формира основата на предложениот високо-прецизен управувачки фирмвер. Оваа архитектура овозможува прецизно генерирање и извршување на чекорите, минимална акумулација на грешки и висока усогласеност помеѓу зададената траекторија на движење и реалното поместување на главата на машината.

За управување со насоката на движење на моторите, која исто така мора да биде строго синхронизирана со генерирањето на чекорите, предложениот фирмвер користи посебен мемориски бафер наречен StepISR-бафер, чија структура е прикажана на слика 3.30. Овој бафер работи паралелно со Intermediate-баферот и ги добива потребните податоци директно од планер баферот. Потребата од воведување на овој дополнителен бафер произлегува од самата архитектура на системот, која се базира на претпроцесирани временски интервали за секој чекор, зачувани во мемориски бафери, за чиј трансфер кон регистрите на соодветните тајмери се користи контролерот за директен мемориски пристап (DMA).

Генерирањето на временските импулси кон моторите од страна на хардверските тајмери зависи од три главни регистри: prescaler, capture/compare и auto-reload, кои за постигнување на максимални перформанси се баферирани и во самите тајмери преку користење на т.н. shadow регистри. Заедно, овие елементи го дефинираат импулсно-ширински модулираниот сигнал на излезот, кој е задолжен за прецизното управување со сите оски на машината.

Бидејќи предложениот систем, кој се базира на трансфер на пресметани податоци преку DMA-контролерот директно кон регистрите на тајмерите, е оптимизиран за континуирано, прецизно и баферирано ажурирање на временските интервали, тој по својата природа не овозможува директно и детерминистичко вметнување на промени на насоката на движење на моторите. Промените на насоката се дефинираат на ниво на оптимизирани сегменти на движење во планер баферот, односно пред нивната поделба



Слика 3.30. Структура на StepISR баферот.

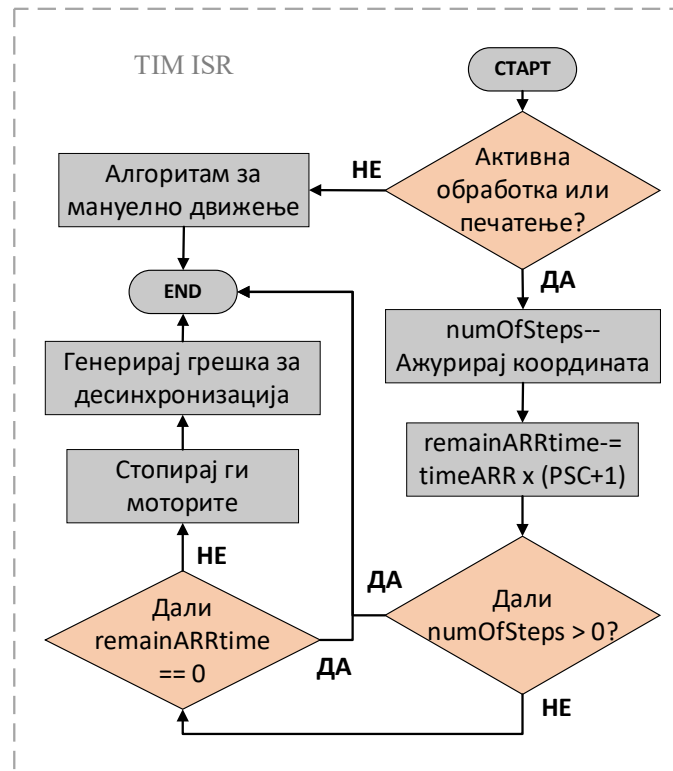
на потсегменти со забрзување, константна брзина и забавување во Intermediate-баферот. Поради тоа, неопходен е дополнителен механизам за синхронизација кој гарантира дека промената на насоката ќе се изврши во точно определениот временски момент што одговара на планираната траекторија на движење. Дури и минимално временско отстапување во овој процес може да резултира со кумулативни позициони грешки, со што се намалува точноста на обработката и се нарушува квалитетот на 3D-печатените делови.

Затоа преку елементите на StepISR-баферот се реализира прецизно управување со насоката на движење за секоја поединечна оска. Овој бафер ги содржи следните клучни параметри: бројот на преостанати чекори до крајот на сегментот (`numberOfRemainSteps`), насоката на движење и статусот на активност на оската, кој индицира дали оската моментално учествува во движењето. Дополнително, баферот вклучува и механизам за детекција на десинхронизација помеѓу оските, кој овозможува проверка во реално време на временската усогласеност при извршувањето на чекорите.

Секој извршен чекор активира прекинска сервисна рутина која го намалува параметарот `numberOfRemainSteps`, ја ажурира координатата на оската и го одзема изминатото време од зачуваното преостанато време (`remainARRtime`). Кога вредноста на `numberOfRemainSteps` ќе достигне нула, насоката на движење безбедно се менува за да одговара на следниот сегмент на движење во баферот. Потоа, механизмот за детекција на десинхронизација проверува дали параметарот `remainARRtime` исто така достигнал нула, доколку тоа не е случај, се детектира синхронизациска грешка, што укажува на можно временско отстапување помеѓу оските. Деталниот алгоритам на оваа прекинска сервисна рутина за една оска, кој е идентичен по структура и логика за сите останати оски, е прикажан на слика 3.31.

За да се обезбеди непрекината и синхронизирана работа на сите оски за време на извршувањето на движењето, во архитектурата е воведен дополнителен слој на бафери, за секоја оска по еден посебен бафер, наречени Intermediate axis-бафери. Со оваа архитектура, системот содржи еден главен Intermediate-бафер и четири Intermediate axis-бафери, по еден за секоја оска (X, Y, Z и E).

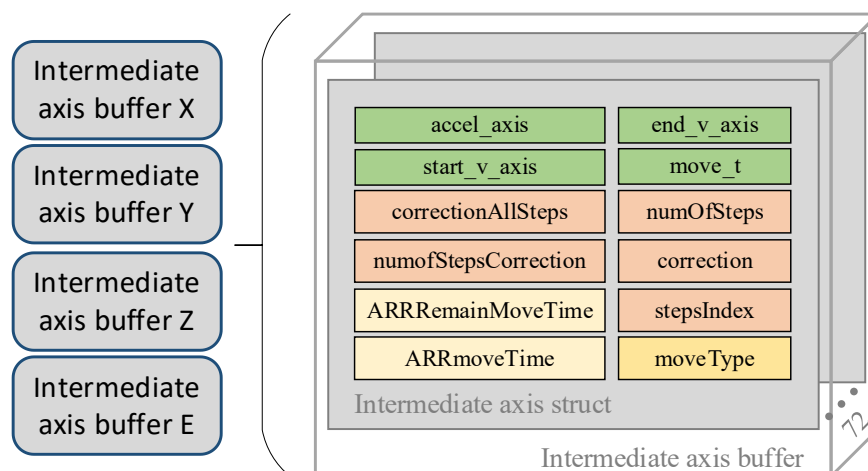
Овие бафери имаат слична структура како главниот Intermediate-бафер, но кинематичките параметри во нив се трансформирани во параметри специфични за соодветната оска. На пример, почетната и крајната брзина на главата на машината за даден сегмент, зачувани во главниот бафер, се конвертираат во индивидуални брзини на секоја оска (v_{startX} , v_{endX} , и сл.). Како дополнување на основните параметри, секој Intermediate axis-бафер содржи индексна променлива преку која се следи прогресот на



Слика 3.31. Алгоритам кој се извршува во прекинската рутина на тајмерите за една оска (идентичен за сите оски).

пресметката на временските интервали на чекорите за соодветната оска. Структурата и елементите на овие бафери се прикажани на слика 3.32.

Воведувањето на овие осно-специфични бафери решава критичен проблем со синхронизацијата што се јавува при директен пренос на податоци од Intermediate-баферот кон Step-баферите. Поточно, кога постои голема разлика во бројот на чекори помеѓу оските, односно кога една оска извршува значително повеќе чекори од друга, оската со помал број чекори може да нема пресметано доволен број временски интервали за чекорите, неопходни за континуиран DMA-пренос. Ваквата состојба може да доведе до прекин на извршувањето и целосна десинхронизација на системот.



Слика 3.32. Структура на Intermediate axis-баферите.

Со имплементацијата на независните Intermediate-axis бафери, секоја оска автономно ги презема и обработува податоците од главниот Intermediate-бафер. Овој дизајн ја елиминира меѓусебната зависност помеѓу оските при подготовката на податоците, овозможува непрекината работа на DMA-механизмот и обезбедува конзистентна синхронизација и глатко извршување на движењето кај сите активни оски.

Надоврзувајќи се на претходно опишаната архитектура на бафери, завршната фаза на управување со движењето на машината опфаќа генерирање на временските интервали на чекорите, при што кинематичките параметри на секој сегмент се трансформираат во прецизни вредности за тајмерите, подготвени за извршување. Step-баферите се прикажани на слика 3.33. За секоја оска на движење (X, Y, Z и E) е резервиран посебен бафер, со што се обезбедува независна временска контрола за секоја оска.

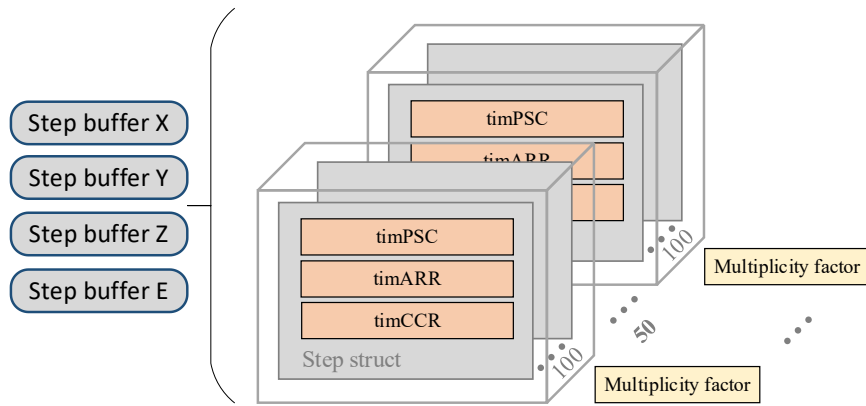
Секој Step-бафер е имплементиран како матрица со димензии 50×100 , при што секој ред од 100 елементи претставува блок на податоци кој последователно се пренесува до хардверските тајмери со користење на контролерот за директен мемориски пристап (DMA). За време на извршувањето, временските интервали на чекорите се пресметуваат во циклуси од по 100 елементи за секоја оска. Во секоја итерација од главната управувачка јамка се генерираат по 100 нови временски интервали за оските X, Y, Z и E (кога машината работи во режим на 3D-печатење).

Секој елемент во Step-баферите содржи четири основни компоненти:

- **PSC** - вредност на прескалерот, со која се дели влезната фреквенција на тајмерот, односно се користи за да се продолжи периодот на тајмерот при пониски брзини на моторите;
- **ARR** - вредност на auto-reload регистарот, која го дефинира периодот на излезниот импулсно-ширински модулиран (PWM) сигнал, а со тоа и фреквенцијата на чекорите;
- **CCR** - вредност на capture/compare регистарот, која ја одредува ширината на импулсот на PWM сигналот, моментот во кој ќе се изврши поместувањето на оската за еден чекор;
- **Фактор на мултиплицирање** - се чува во паралелна низа и го определува бројот на повторувања на соодветниот блок од сто елементи во баферот.

Овој фактор е особено ефикасен за сегментите со константна брзина (cruise сегментите), каде повеќе последователни блокови имаат идентични временски интервали. Со кодирање на овие повторливи блокови од по 100 елементи како единечен запис со придружен фактор на мултиплицирање, системот значително ја намалува потребната меморија, притоа задржувајќи ја високата временска резолуција и прецизноста на управувањето. Така сегментите со илјадници чекори (сегментите со константна брзина се најдолги имаат најголем број на чекори) се претставуваат само со стотина елементи во меморија. Ова е исто така еден од клучните механизми што ги користи предложениот фирмвер без која не е можно да се изведе ваков тип на управување. Со овој механизам пресметката и баферирањето на временските интервали за сегментите со константна брзина се изведува многу брзо и ефикасно.

Педесетте редови од матрицата одговараат на 50 последователни сегментни елементи, при што секој ред содржи до 100 однапред пресметани временски интервали на чекорите. Постапката за пресметка е прикажана за X-оската, но таа е идентична по структура и логика и за останатите оски.



Слика 3.33. Структура и димензии на Step-баферите.

Пресметката на временските интервали на чекорите започнува со определување на линеарното поместување помеѓу почетокот на сегментот и тековниот чекор:

$$dx = 0.001 + i \times X_Y_Z_STEP_RES \quad (3.119)$$

каде што i е индексот на тековниот чекор ($i = 0$ за првиот чекор), а $X_Y_Z_STEP_RES$ е резолуцијата на оските X, Y и Z во милиметри. Времето од почетокот на сегментот до тековниот чекор се пресметува како:

$$t_{step}[i] = \frac{-v_{startX} + \sqrt{v_{startX}^2 + 2a_x dx}}{a_x} \quad (3.120)$$

Бидејќи овој израз го дава кумулативното време од почетокот на сегментот, временскиот интервал помеѓу последователните чекори се добива како:

$$xPeriod = (t_{step}[i] - t_{step_prev_ARR}) \times XYZE_M_NUM_OF_TICKS_PER_SEC \quad (3.121)$$

Кај 32-битните тајмери, пресметаниот период вообичаено може директно да се постигне со големиот опсег на тајмерите. Во повеќето случаи, дури и кај 16-битни тајмери, употребата на прескалерот не е потребна, бидејќи добиениот период се наоѓа во дозволеният опсег на бројачот. Само во ретки случаи, поврзани со екстремно бавно движење, прескалерот станува неопходен за да може да се постигне ниската фреквенција на импулсите кон моторите, па прескалерот се пресметува со следниот израз:

$$xPrescaler = \frac{xPeriod}{X_MOT_CNT_RANGE} \quad (3.122)$$

каде што $X_MOT_CNT_RANGE$ ја претставува максималната вредност на бројачот на тајмерот. По пресметката за временскиот интервал на тековниот чекор, се ажурира променливата која го чува времето на последниот чекор според следниот израз:

$$t_{step_prev_ARR} += \frac{\left[\text{round} \left(\frac{xPeriod}{xPrescaler + 1} \right) \times (xPrescaler + 1) \right]}{XYZE_M_NUM_OF_TICKS_PER_SEC} \quad (3.123)$$

Следно ширината на импулсот се дефинира во средината на периодот на чекорот според следниот израз:

$$xImpulse = \frac{xPeriod}{2} \quad (3.124)$$

Интервалот на последниот чекор од сегментот се пресметува како остатокот од вкупното време на сегментот:

$$t_{step}[last] = t_{ARR} - t_{step}[last - 1] \quad (3.125)$$

Пред зачувување на пресметаните вредности во Step баферите, сите резултати добиени со двојна прецизност соодветно се заокружуваат на целобројни вредности, со цел да се обезбеди компатибилност со хардверските тајмери.

Кај сегментите со константна брзина, пресметката е значително поедноставена, бидејќи временските интервали на чекорите се униформни и се пресметуваат според следниот израз:

$$xPeriod = \frac{t_{ARR}}{numOfSteps} \quad (3.126)$$

Процедурата за сегментите од типот забавување е идентична на онаа за забрзување, со таа разлика што брзината се намалува со негативно забрзување, што резултира со модифициран израз за пресметување на времето на i-тиот чекор:

$$t_{step}[i] = \frac{v_{startX} - \sqrt{v_{startX}^2 - 2a_x dx}}{a_x} \quad (3.127)$$

Ограничената флексибилност на контролерот за директен мемориски пристап (DMA) кај микроконтролерот STM32F767 претставуваше дополнителен предизвик при имплементацијата на предложената архитектура. Имено, DMA-модулот кај овој микроконтролер поддржува паралелен пренос на повеќе податоци кон тајмерите само во точно определен секвенцијален редослед, даден во продолжение:

$$PSC \rightarrow ARR \rightarrow RCR \rightarrow CCR_x (CCR1, CCR2, CCR3, CCR4) \quad (3.128)$$

Во предложениот сервобазиран управувачки систем, во реално време е потребно да се ажурираат само вредностите на регистрите PSC (прескалер), ARR (auto-reload регистарот) и еден од CCR регистрите, во зависност од тоа кој канал на соодветниот тајмер се користи за дадената оска. Хардверската имплементација на системот наложува секоја оска да користи различен канал на тајмерот, кој не секогаш е првиот канал. На пример, за X-оската се користи каналот 3 од тајмерот 2 (TIM2), што подразбира дека DMA-трансферот мора задолжително да ги вклучи и вредностите за регистрите кои реално не се користат (RCR, CCR1 и CCR2). Овие регистри се пополнуваат со нулта вредност.

Ова ограничување на архитектурата не беше земено во предвид при иницијалниот дизајн на хардверот, односно на електронските управувачки плочки, поради што не сите тајмери го користат каналот 1. Доколку кај сите тајмери се користеше каналот 1, единствениот непотребен регистар во DMA-секвенцата на пренос би бил регистарот RCR, за кој, поради архитектурата на DMA-модулот, не постои можност за елиминација. За разлика од тоа, понапредните микроконтролери од серијата STM32H7 располагаат со значително пофлексибилен DMA-модул, кој овозможува директно мапирање исклучиво на потребните регистри, со што целосно се елиминира потребата од ажурирање на непотребни регистарски полиња.

Позначајниот проблем што произлегува од ова ограничување на микроконтролерот е зголемувањето на мемориските побарувања, бидејќи Step-баферите би морале да користат структури кои содржат и елементи што не се функционално потребни. Основната идеја на предложениот систем е, преку користење на доволно големи бафери за временските интервали на чекорите, да се овозможи рамномерна распределба на процесирачкиот товар на јадрото на микроконтролерот. Поголемите бафери овозможуваат пресметување на нови временски интервали во периоди кога јадрото е најмалку оптоварено, односно за време на сегментите со константна брзина, додека при влез во сегментите со поголеми процесирачки барања, како што се сегментите со забрзување и забавување, да постојат доволен број однапред пресметани вредности зачувани во баферите.

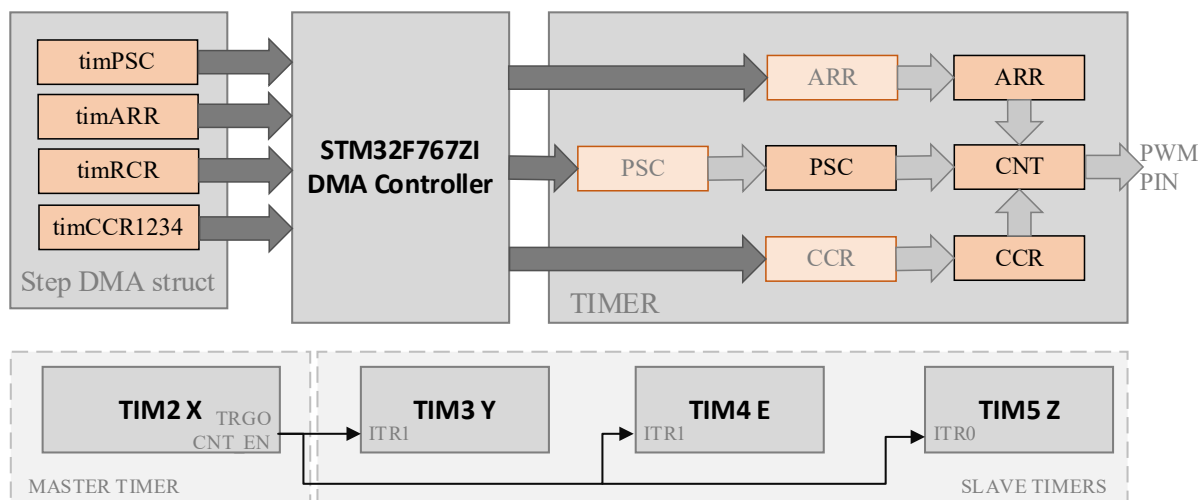
Со цел да се избегне непотребното зголемување на мемориските побарувања, во предложената архитектура е воведен дополнителен слој на бафери со помали димензии. Во примарните Step-бафери се зачувуваат исклучиво вредностите неопходни за пресметување и генерирање на временските интервали на чекорите, додека посебните StepDMA-бафери ги содржат сите регистарски вредности потребни за непречен DMA-трансфер, вклучувајќи ги и дополнителните регистри кои се пополнуваат со нулти вредности.

Во продолжение е прикажан пример за структурата на StepDMA-баферот за X-оската, која го користи каналот 3 од тајмерот 2 (TIM2) за генерирање на управувачките импулси кон моторот, односно неговиот драјвер:

$$PSC, ARR, RCR(0), CCR1(0), CCR2(0), CCR3(\text{вред. за шир. на имп.}) \quad (3.129)$$

Проширувањето на примарниот Step-бафер со овие дополнителни регистри би резултирало со значително зголемување на мемориските побарувања (на пример, 50 сегменти $\times$ 100 вредности по сегмент $\times$ бројот на неискористени регистарски полиња). Поради тоа, StepDMA-баферот е имплементиран како компактна структура составена од 4×100 елементи, по еден блок за секоја оска, во која динамички се копираат вредностите од Step-баферот секогаш кога ќе се ослободи простор за следниот блок од 100 елементи. Со овој пристап се обезбедува континуиран DMA-пренос во реално време, со оптимална употреба на RAM меморијата.

Прецизната синхронизација помеѓу сите оски бара сите тајмери да се вклучат со броење во егзактно исто време. Предложениот фирмвер ги вклучува тајмерите само еднаш, во случај кога извршувањето на програмата е иницирано преку HMI-дисплејот или компјутерскиот софтверот и тајмерите мора да работат идеално синхронизирани до завршувањето на тековната програма. За да се овозможи детерминистички старт на сите тајмери во исто време, тие се конфигурирани во мастер-слејв топологија, каде тајмерот 2 (TIM2) задолжен за моторот на X-оската работи како мастер, а сите останати тајмери се конфигурирани како слејв тајмери. Тригер сигнал се генерира од мастерот кон сите останати тајмер само во случај кога се прави овозможување на истиот, тригерот од мастер тајмерот се пренесува на останатите тајмери и овозможувањето на тајмерите е во истиот момент (во ист момент се стартуваат сите тајмери). Оваа синхронизација на хардверско ниво е прикажана на слика 3.34.

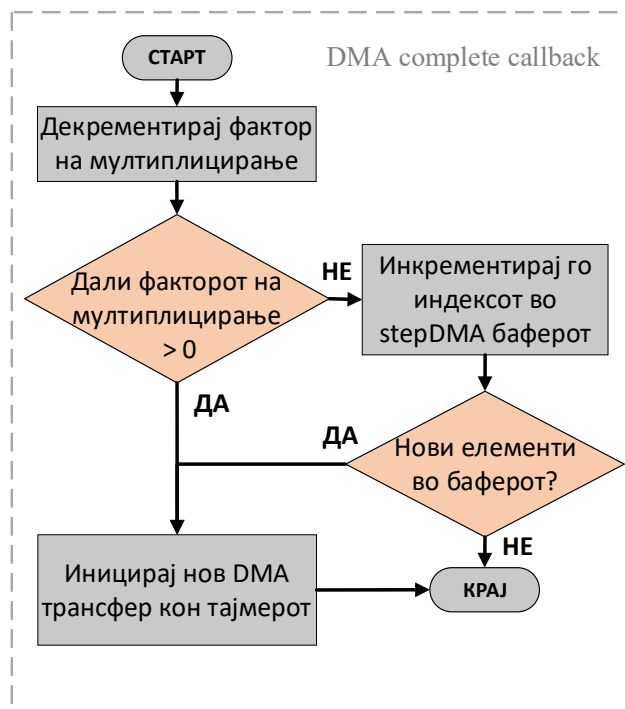


Слика 3.34. Хардверска мастер-слејв топологија на синхронизација помеѓу тајмерите кои ги управуваат моторите на оските на машината и екструдерот и блок-дијаграм на преносот на податоците од StepDMA-баферот кон баферираните регистри на тајмерите (shadow регистрите).

За време на извршувањето на движењето, за одредени сегменти поместувањето по некои од оските може да биде нула. Со цел да се зачува глобалната временска синхронизација помеѓу сите оски, тајмерите задолжени за оските со нулто движење продолжуваат со броењето, иако механичко поместување по овие оски не се изведува. Излезните импулси кај овие оски се оневозможуваат со поставување на условот $CCR > ARR$, со што во PWM2 излезен режим тајмерот го задржува излезниот пин на постојано ниско логичко ниво. На овој начин се одржува целосна временска усогласеност на сите оски, без да се предизвика несакано механичко движење кај неактивните оски. Неактивните оски се означуваат со соодветен флег во рамките на StepISR-баферот, кој е задолжен за синхрона промена на насоката на движење, со што се овозможува соодветно управување и филтрирање на прекините генерирани од тајмерите за овие оски (ако оската е означена како неактивна нема ажурирање на координатите на истата при генерираните прекини).

Сите ажурирања на регистри на тајмерите се изведуваат преку баферираните (shadow) регистри, со што се обезбедува детерминистичко и целосно синхронизирано однесување на системот, како што е прикажано на слика 3.34. Регистрите PSC, ARR и CCR се имплементирани како двојни регистри (реални и бафер регистри), при што вредностите од бафер регистри се префрлаат во реалните регистри исклучиво при генерирање на настан (update event), кој се генерира кога бројачот на тајмерот ќе ја достигне вредноста запишана во ARR-регистарот. Непосредно по генерирањето на овој настан, DMA-контролерот автоматски префрла нови вредности од StepDMA-баферот во бафер регистри на тајмерот. Со ова се обезбедува во секој момент во регистри на тајмерот да има по една однапред подготвена вредност за секој функционален регистар, што овозможува континуирано и непрекинато ажурирање на реалните вредности, а со тоа и комплетна синхрона работа на сите тајмери.

Контролерот за директен мемориски пристап (DMA) генерира прекин по завршувањето на трансферот на блок од 100 елементи од StepDMA-баферот кон регистри на тајмерите. Во тој момент, во реалните регистри на тајмерот се наоѓаат тековно активните вредности за PSC, ARR и CCR, додека во нивните бафер регистри е веќе запишана последната вредност од последниот блок.



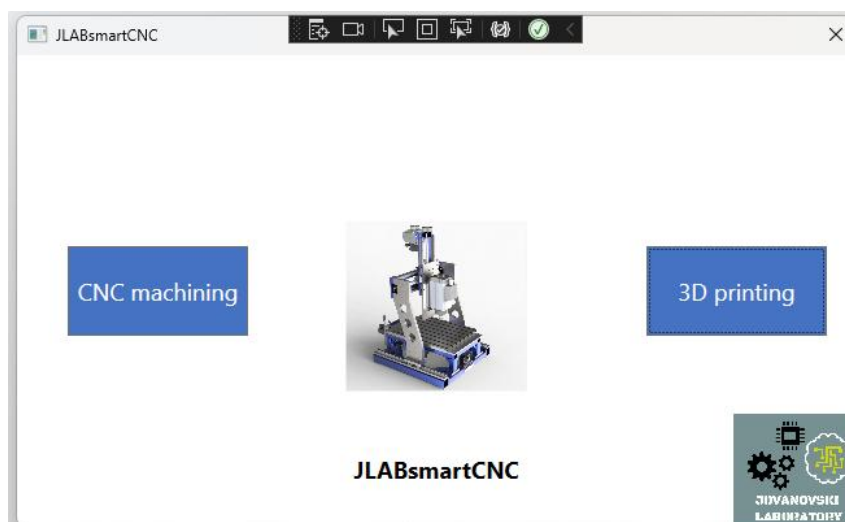
Слика 3.35. Алгоритам кој се изведува во прекинската рутина при завршување на тековниот DMA-трансфер и иницирање на нов. За секоја оска се користи посебна прекинска рутина со истиот алгоритам имплементиран за истата.

Во рамките на DMA-сервисна рутина, факторот на мултиплицирање се декрементира. Доколку неговата вредност е поголема од нула, се иницира нов DMA-трансфер на истиот блок од 100 елементи, што е карактеристично за сегментите со константна брзина. Кога факторот на мултиплицирање ќе достигне нула, се иницира DMA-трансфер на следниот блок од 100 елементи од StepDMA-баферот, со што се обезбедува континуиран пренос на податоци кон тајмерите.

Прекилот и иницирањето на новите DMA-трансфери се изведуваат во момент кога во тајмерот се префрлени последните две вредности од претходниот блок, што значи дека новиот трансфер се иницира пред целосното исцрпување на тековниот бафер. Ова однесување е овозможено благодарение на постоењето на бафер регистрите (shadow регистрите) кај тајмерите и обезбедува загарантирано време за иницирање на новите трансфери, а со тоа и сигурна, непрекината и детерминистичка работа на целиот систем. Деталниот алгоритам на прекинската рутина за иницирање на нов DMA-трансфер е прикажан на слика 3.35.

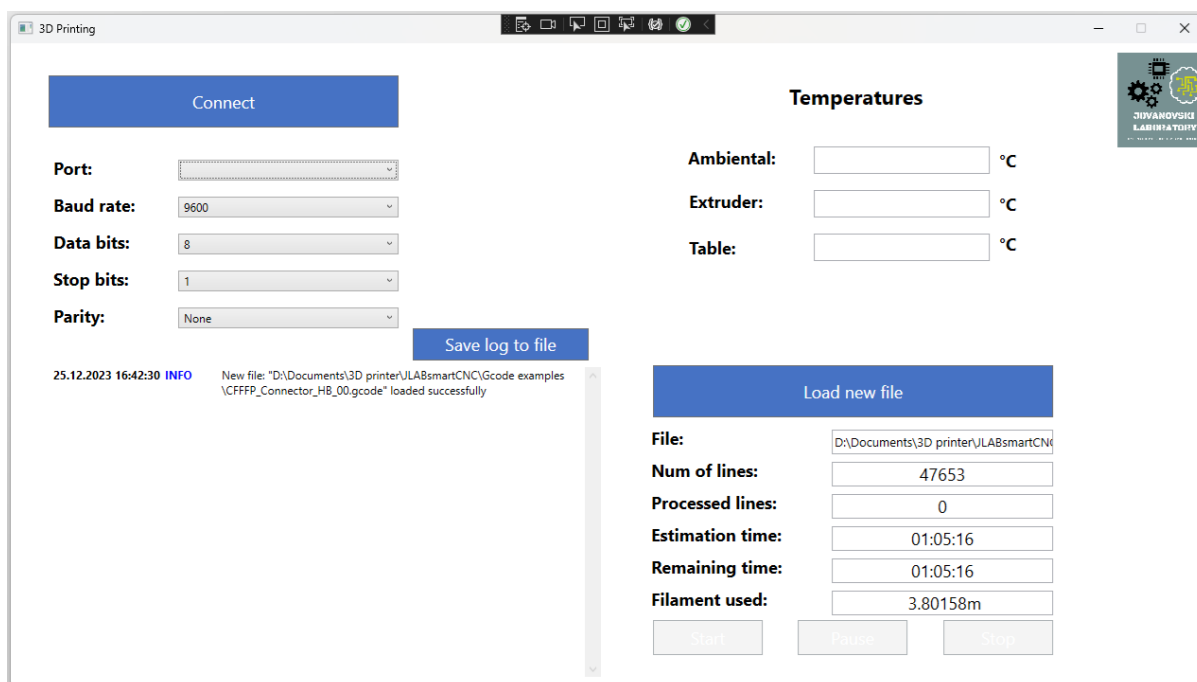
3.2 Имплементација на компјутерски софтвер за комуникација со предложената машина

Основниот интерфејс преку кој корисникот управува со целата машина е HMI-дисплејот. Преку овој интерфејс се овозможени сите функционалности на машината, при што фајловите со G-кодови за CNC-обработка или 3D-печатење се вчитуваат преку USB-мемориски уред или SD-мемориска картичка директно на самата машина. Како дополнителна опција, развиен е и едноставен компјутерски софтвер кој овозможува комуникација со машината преку двонасочна RS232 сервиска врска. Главната цел на овој софтвер е испраќање на G-код команди директно од компјутер, како и подесување и читање на основните параметри на машината.

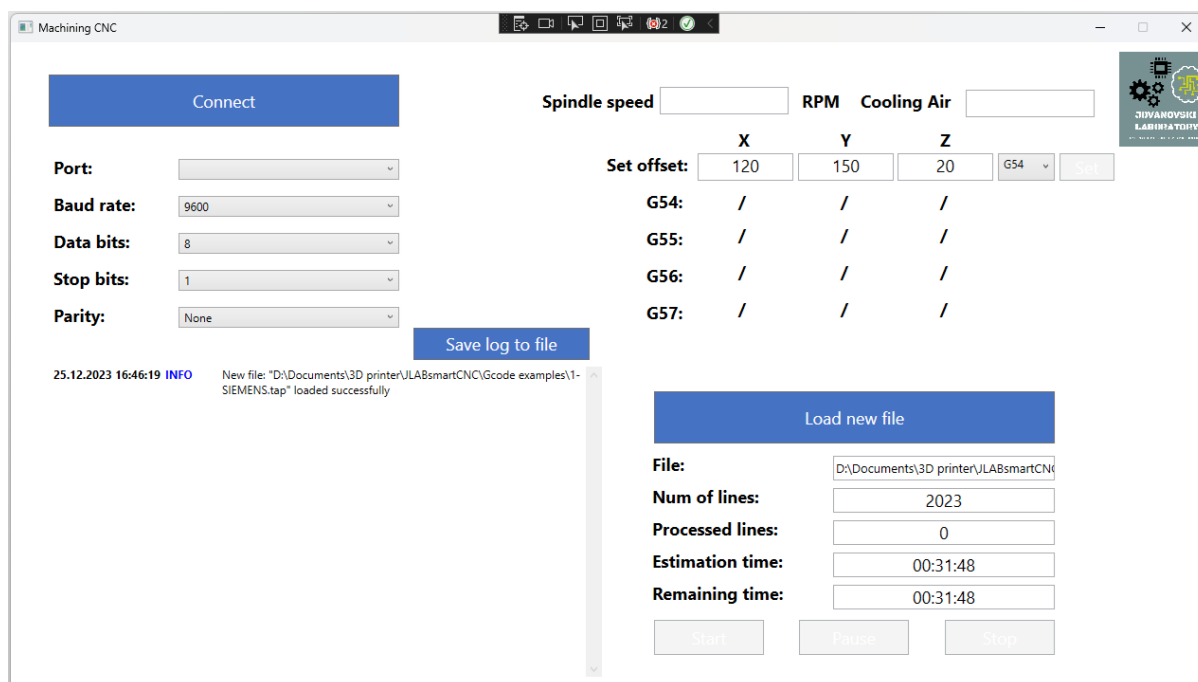


Слика 3.36. Изглед на основното мени на компјутерскиот софтвер каде се избира дали машината ќе се користи за обработка или 3D-печатење.

На слика 3.36 е прикажано основното мени на машината, во кое корисникот избира дали машината ќе се користи за CNC-обработка или за 3D-печатење. Во менито за 3D-печатење, прикажано на слика 3.37, во горниот дел од левата страна се избираат параметрите за сериската порта преку која е поврзана машината, имплементирано е и копче за воспоставување на комуникацијата. Во долниот дел од левата страна е имплементирана информативна лабела во која се прикажуваат сите релевантни информации поврзани со машината и софтверот, односно успешно испратени команди, настанати грешки при комуникацијата и информации добиени од самата машина. Дополнително, овозможена е и опција за зачувување на сите овие информации во надворешен фајл за понатамошна анализа.



Слика 3.37. Мени од компјутерскиот софтвер наменето за 3D-печатење.



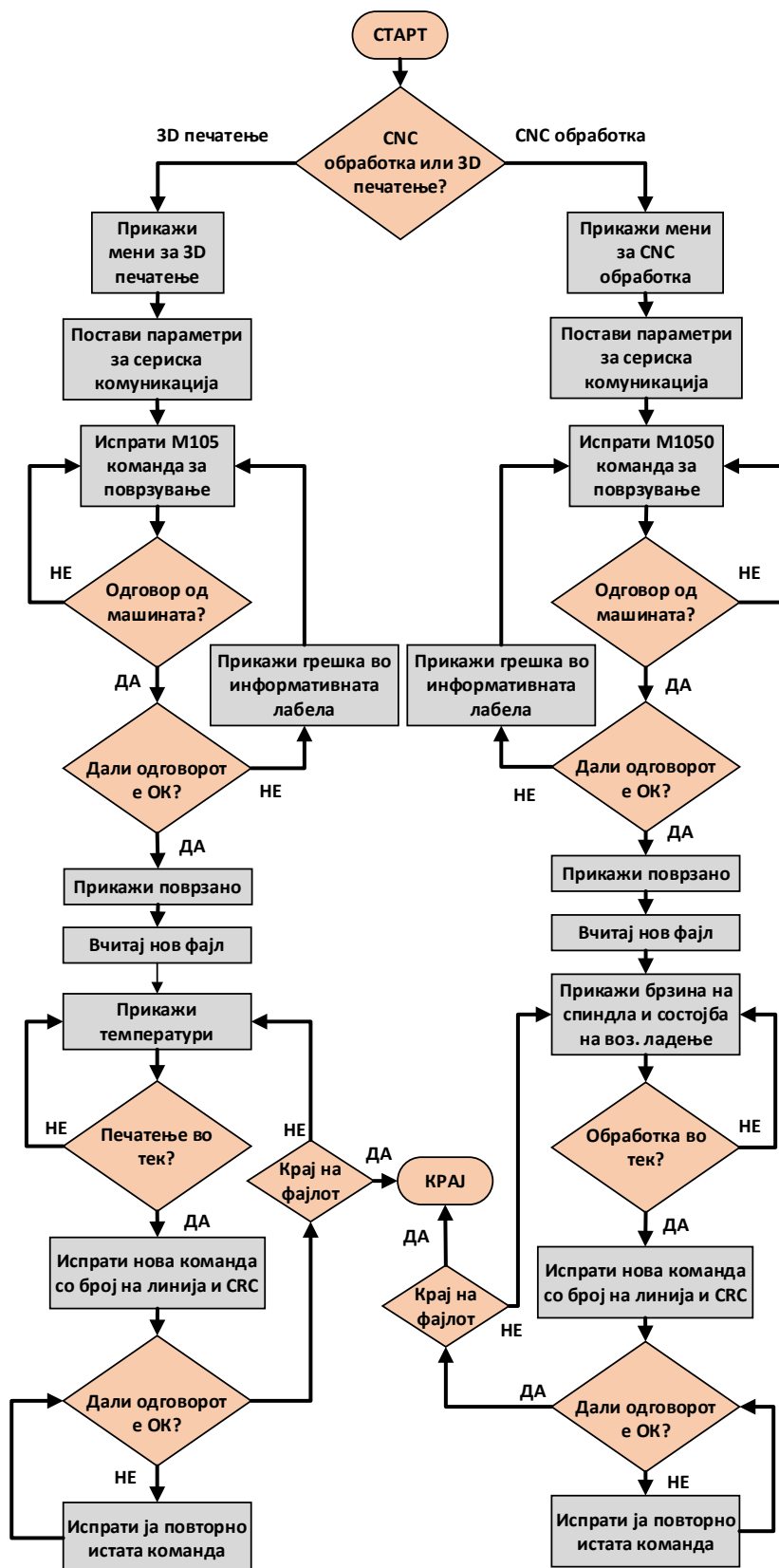
Слика 3.38. Мени од компјутерскиот софтвер наменето за CNC-обработка.

Горниот дел од десната страна на менито е наменет за прикажување на моменталните температури на екструдерот, масата за печатење и амбиенталната температура, кои се клучни параметри за правилно и стабилно извршување на процесот на 3D-печатење. Долниот дел од десната страна е наменет за вчитување на G-код фајлот кој ги содржи сите команди потребни за печатење на избраниот 3D-модел. Во овој дел се прикажуваат основните информации за моделот што се изработува, а корисникот од тука може да го започне, паузира или прекине целиот процес на печатење.

Многу слична структура има и менито за CNC-обработка, прикажано на слика 3.38. Во горниот десен дел од ова мени се прикажуваат моменталната брзина на спиндлата и статусот на системот за ладење (воздушно ладење на парчето при обработка). Дополнително, во овој дел е имплементирана и можност за задавање на работните координатни системи на парчето кое се обработува преку стандардните G-команди: G54, G55, G56 и G57. Останатите функционалности и распоредот на елементите се еквивалентни со менито за 3D-печатење.

За имплементација на комуникацијата помеѓу компјутерскиот софтвер и машината развиен е посебен комуникациски протокол, прикажан на слика 3.39. При воспоставување на конекцијата се испраќаат две специфични G-код команди: M105 за режим на 3D-печатење и M1050 за режим на CNC-обработка. Доколку машината е достапна на сериската порта, таа одговара со моменталните вредности на температурите во случај на 3D-печатење, односно со вредностите за брзината на спиндлата и состојбата на системот за воздушно ладење при CNC-обработка. Овие команди периодично се испраќаат од компјутерскиот софтвер со цел континуирано ажурирање на параметрите кои се прикажуваат при активна конекција со машината. Процесот на испраќање на G-код командите се изведува линија по линија, при што на секоја команда ѝ се додава реден број на линија за идентификација, како и CRC-код за проверка на интегритетот на податоците од приемната страна. Машината ги потврдува успешно примените команди со враќање на соодветен одговор придружен со бројот на линија за кој се однесува. Во случај на настанати грешки при комуникацијата, машината враќа соодветен код за

грешка, кој компјутерскиот софтвер го обработува и го прикажува на информативната лабела, овозможувајќи навремена детекција и дијагностика на проблемите во комуникацијата.



Слика 3.39. Протокол на комуникацијата помеѓу компјутерскиот софтвер и машината.

4 Експериментална евалуација на предложениот управувачки фирмвер

4.1 Искористеност на RAM-меморијата кај предложениот управувачки фирмвер

Како што беше разгледано во претходното поглавје, предложениот управувачки систем се базира на комплексна мемориска архитектура со повеќеслојни бафери, која бара значителна алокација на RAM-меморија. Првиот чекор при експерименталната евалуација на перформансите на системот е утврдувањето на мемориските барања на сите бафери кои се суштински за правилно и детерминистичко функционирање на предложениот управувачки алгоритам. За таа цел, направена е детална и прецизна анализа на мемориските побарувања на секој поединечен бафер, со цел да се определи вкупната меморија потребна за реализација на ваквата архитектура.

Системот користи исклучиво статичка алокација на меморија за сите бафери. Овој пристап овозможува баферите да бидат постојано достапни за алгоритмот и континуирано пополнети со доволен број на елементи, со што се обезбедува непрекинато извршување на секвенците од чекори во реално време. Динамичката алокација на меморија е намерно избегната, бидејќи операциите на алоцирање и ослободување на меморија за време на извршувањето би вовеле дополнителни временски доцнења и недетерминистичко однесување, што би претставувало дополнително оптоварување за системот и потенцијално би ја компромитирало временската предвидливост и стабилноста на управувачкиот процес.

Во табела 4.1 е прикажана големината на секој од баферите во системот што ја зафаќаат во меморија. Вкупната меморија потребна за предложениот напреден управувачки алгоритам изнесува приближно 310 KB, што претставува околу 60% од вкупно достапните 512 KB RAM-меморија кај STM32F767 микроконтролерот. Преостанатите приближно 200 KB се доволни за поддршка на останатите функции на машината, како што се комуникацијата, управувањето со периферии и обработката на сензорски податоци.

Анализата на искористувањето на меморијата при компајлирање, извршена со помош на STM32CubeIDE, покажува дека вкупната RAM-меморија резервирана од целата апликација изнесува околу 401 KB, што одговара на приближно 78% од

Табела 4.1. Големина на секој од баферите во RAM-меморијата.

Име на бафер	X оска(B)	Y оска(B)	Z оска(B)	E оска(B)	Опис
Planner			56,000		Главен бафер за планирање на движењата
StepISR	3,456	3,456	3,456	3,456	Бафер за менаџирање со насоките на движење
Intermediate			9,792		Бафер во кој се делат комбинирани сегменти
Intermediate Axis	5,760	5,760	5,760	5,760	Бафер во кој се вршат пресметките за времињата на секој чекор
Step	60,000	60,000	30,000	30,000	Бафер кој ги содржи времињата за чекорите
StepDMA	9,600	11,200	3,200	3,200	Бафер со форматирани времиња за DMA трансфер

расположливата меморија. Останатиот мемориски простор се користи како stack и heap меморија, со што се обезбедува доволна меморија за извршување на функциите, сервисирање на прекините и сместување на привремените податоци за време на извршувањето на апликацијата. Во текот на експерименталната работа не беше забележан ниту еден случај на преполнување на stack-меморијата, што потврдува дека конфигурираните мемориски маргини се соодветни и за најлошите сценарија на работа.

Оваа мемориска архитектура има големи мемориски побарувања од аспект на микроконтролер, но може дополнително да се оптимизира. Вредностите за прескалерот (PSC) може целосно да се изостават од баферите. Кај STM32F767 микроконтролерот, PSC не е неопходен за 32-битните тајмери како што се TIM2 и TIM5 (кои се користат за X и Y-оските), а ретко е потребен и кај останатите 16-битни тајмери, освен при екстремно бавни движења. Во такви случаи, повеќе циклуси на тајмерот можат да се комбинираат за да се генерира еден единствен чекорен импулс, со што целосно се елиминира потребата од користење на PSC. Со отстранување на PSC, големината на баферите кои ги чуваат временските интервали на чекорите би се намалила за повеќе од 30%, што би резултирало со значително подобрување на мемориската ефикасност.

Дополнително, користењето на микроконтролери од повисока класа, како што е серијата STM32H7, кои располагаат со 1 MB RAM-меморија, овозможуваат имплементација на значително поголеми бафери без мемориски ограничувања. Ова овозможува истиот алгоритам да се скалира на четири, пет или повеќе оски, што го прави пристапот погоден за покомплексни CNC-системи, при што се задржува детерминистичкото однесување во реално време.

За разлика од дистрибуираните архитектури за управување, како што е Клипер, кои се потпираат на микропроцесорски (хост) системи (single-board компјутери) со RAM-меморија од ранг на гигабајти за планирање на траектории и баферирање, предложениот пристап постигнува споредливи резултати во управувањето на движењата во реално време користејќи само еден микроконтролер со ограничена вградена меморија. Целото извршување на траекториите, синхронизацијата на оските и генерирањето на временските интервали на чекорите се реализираат целосно на микроконтролерско ниво, без потреба од надворешен процесор или хост платформа. Со ваквиот пристап значително се намалува системската комплексност и се елиминираат комуникациските тесни грла карактеристични за хост базирани архитектури, при што истовремено се задржува целосно детерминистичко, реалновременско однесување и покрај повеќекратно помалата достапна меморија во споредба со системи од типот на Клипер.

4.2 Анализа на алгоритамот за пресметка на временските интервали за чекорите во реално време

Предложениот управувачки алгоритам има релативно високи побарувања од аспект на процесирачката моќ, бидејќи егзактно ги пресметува временските интервали за сите чекори на сите оски при движење на главата на машината со многу висока резолуција. Поради тоа, во овој дел е направена детална анализа на пресметковните побарувања и извршена е евалуација на имплементацијата на алгоритамот на предложената платформа со ограничени ресурси, базирана исклучиво на еден микроконтролер. Пресметките што се изведуваат на ниво на сегментите на движење, како оние во *Planner* и *Intermediate* баферот, имаат значително помали побарувања од аспект на процесирачката моќ, бидејќи се однесуваат на цели сегменти, а не на поединечни чекори. Затоа тие не претставуваат критични реално-временски ограничувања за јадрото на микроконтролерот. Критичните временски побарувања се

јавуваат при пресметувањето на временските интервали за секој поединечен чекор, особено во сегментите со забрзување и забавување, каде што бројот на чекори за кои е потребна пресметка е значително поголем во споредба со сегментите со константна брзина.

За да се одреди искористеноста на микроконтролерот, најпрво измерено е времето потребно за пресметка на временскиот интервал на еден чекор и негово зачувување во *Step*-баферите, користејќи го хардверскиот бројач на такт циклуси DWT (Data watchpoint and Trace). Мерењата се извршени на сегменти со забрзување бидејќи тие претставуваат најлош случај поради континуираната променлива брзина. Фазата на забрзување е симетрична со фазата на забавување, за која исто се изведени мерења, но резултатите се идентични, па тука се прикажани само резултатите за сегментите со забрзување. Без користење на оптимизација на компајлерот, просечното време на извршување изнесува приближно 480 такт циклуси за еден чекор (пресметување и зачувување во бафер). Со вклучени оптимизации *O3* и *ffast-math* времето се редуцира на 288 циклуси. Доминантна операција која зазема најмногу такт циклуси е квадратниот корен која се користи при пресметка на брзината. За оптимизација на операцијата квадратен корен изведени се и дополнителни експерименти со алтернативни апроксимации базирани на методот на Њутн-Рафсон, но користењето на овој метод и мануелната оптимизација покажаа полоши резултати во споредба со оптимизацијата на компајлерот.

За дадена фреквенција на микроконтролерот од 216 MHz, времето потребно за пресметка и зачувување на временскиот интервал за еден чекор изнесува:

$$t_{step} = \frac{N_{cycles}}{f_{clk}} = \frac{288}{216000000} = 1.33 \mu s \quad (4.1)$$

Максималната брзина на сервомоторите кои се користат во предложениот систем изнесува 3000 rpm (вртежи во минута), тие управуваат навојни вретена кои при една ротација се поместуваат за 5 mm, што резултира со максимална линеарна брзина од:

$$v_{max} = \frac{3000}{60} \times 5 = 250 \frac{mm}{s} \quad (4.2)$$

Резолуцијата на оските на предложената машина изнесува 1 μm , па соодветната фреквенција на чекорите изнесува:

$$f_{step} = \frac{250\,000 steps}{1s} = 250\,kHz, t_{required} = 4 \mu s \quad (4.3)$$

Во реални услови на работа никогаш трите оски не работат симултано со максимална брзина. Ограничувањето на максималната брзина на движење на главата на машината е на лимитот што го диктираат моторите 250 mm/s за во случај на движење по само една оска да не се пречекорат физичките можности на истите, па кога повеќето оски се движат заедно резултантната брзина мора да остане во зададената граница. Така во реално сценарио на работа на машината брзината на движење на оските е помала од 60% од максималната брзина. Најлошото сценарио кога микроконтролерот би бил најмногу оптоварен се случува кога сите мотори се движат со брзина 60% од максималната, иако реалната оптовареност би била помала. Под овие услови, достапното време за пресметки помеѓу чекорите се зголемува на:

$$t_{available} \approx 6.7 \mu s \quad (4.4)$$

За времето потребно за пресметување и сместување во бафер за сите три оски се добива:

$$t_{compute,3axes} = 3 \times 1.33 \mu s \approx 3.9 \mu s \quad (4.5)$$

Дополнително предложениот алгоритам генерира периодични прекини за секој чекор. Измереното време на прекинските рутина за чекорите вклучувајќи ја и латенцијата [81-82] изнесува 110 такт циклуси, па оттука се добива времето потребно за извршување на една прекинска рутина:

$$t_{ISR} = \frac{N_{ISRcycles}}{f_{clk}} = \frac{110}{216000000} = 0.51 \mu s \quad (4.6)$$

Со ова искористеноста на микроконтролерот може да се пресмета според следниот израз:

$$MCU_{utilization} = \frac{3 \times 1.33 \mu s + 3 \times 0.51 \mu s}{6.7 \mu s} \approx 82 \% \quad (4.7)$$

Кога машината се користи како 3D-печатар брзината на екструдерот е стотина пати помала во споредба со брзината на оските, па така генерирањето на импулсите за екструдерот воведува незначително оптоварување на јадрото на микроконтролерот.

Пресметувањето на временските интервали на чекорите од сегментите со константна брзина е околу 120 такт циклуси по чекор. Сепак, благодарение на факторот на мултиплицирање, истите временски вредности се користат за повеќе последователни чекори, така што само за многу мал дел од чекорите реално е потребна пресметка. Како резултат на тоа, оптоварувањето на јадрото на микроконтролерот не зависи од должината на сегментите со константна брзина и вкупната оптовареност од овие сегменти е многу мала. Дополнително, предложениот систем користи големи бафери што овозможува максималното оптоварување на јадрото на микроконтролерот за сегментите со забрзување и забавување да се распредели порамномерно, така што голем дел од него да се прерфрли во сегментите со константна брзина.

Од добиените експериментални резултати може да се заклучи дека предложениот управувачки систем успешно функционира на избраната микроконтролерска платформа, при што значителен дел од достапната процесирачка моќ на микроконтролерот е искористена. Сепак, идентификувани се повеќе можности за дополнителна оптимизација на алгоритмот и целокупниот систем, со цел постигнување повисоки перформанси, како и обезбедување на скалабилност, односно поддршка за управување со поголем број оски.

Еден од пристапите за понатамошно подобрување на перформансите и намалување на оптовареноста на микроконтролерското јадро е елиминацијата на прекините кои се генерираат за секој поединечен чекор и нивна замена со посебен тајмер за управување со насоката на движење, кој би генерирал прекини исклучиво при премин помеѓу сегменти во кои е потребна промена на насоката.

Дополнителна можност претставува миграцијата кон понови и помоќни фамилии на микроконтролери, како што е серијата STM32H7, која овозможува работни фреквенции до 450 MHz, двојадрена архитектура и до 1 MB вградена RAM-меморија. Понатаму, најновата серија STM32V8 која набргу се очекува да биде достапна за

Табела 4.2. Првите 16 координати добиени од G-кодот кои се користат за евалуација на алгоритмот за генерирање на временските интервали на чекорите.

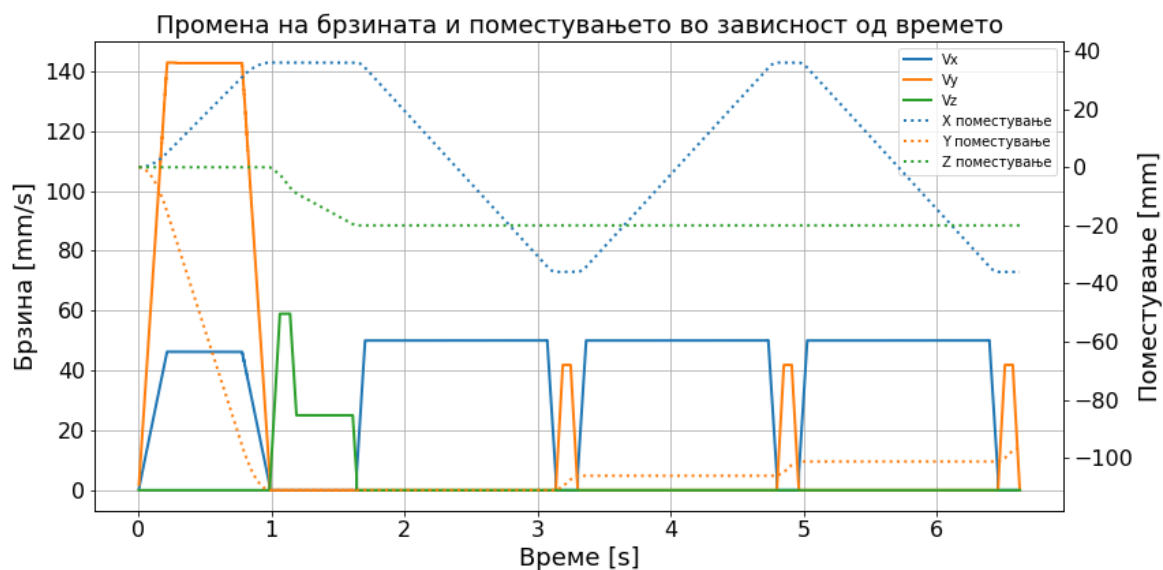
X	Y	Z
0	0	0
36.000	-111.064	-9.000
36.000	-111.064	-20.000
36.000	-111.064	-20.000
31.000	-111.064	-20.000
-31.000	-111.064	-20.000
-36.000	-111.064	-20.000
-36.000	-106.128	-20.000
-31.000	-106.128	-20.000
31.000	-106.128	-20.000
36.000	-106.128	-20.000
36.000	-101.192	-20.000
31.000	-101.192	-20.000
-31.000	-101.192	-20.000
-36.000	-101.192	-20.000
-36.000	-96.255	-20.000

комерцијална употреба, со работни фреквенции до 800 MHz, може да обезбеди значително зголемување на пресметувачките перформанси и дополнителен простор за проширување на алгоритмот.

Конечно, дополнително намалување на оптоварувањето на јадрото на микроконтролерот може да се постигне со директна пресметка на временските интервали на секои 2 до 4 чекори и нивна интерполација, при што влијанието врз точноста на управувањето би било занемарливо мало.

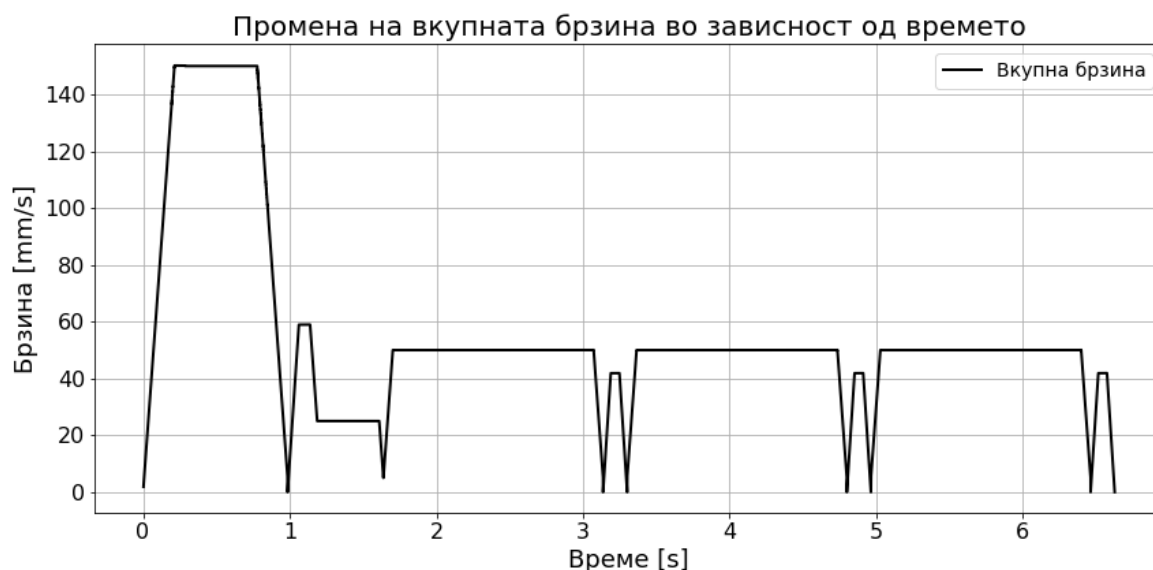
За да се евалуира алгоритам за генерирање на временските интервали на чекорите, изведен е експеримент на самиот микроконтролер користејќи го предложениот систем. За да се изведе овој експеримент во самиот фирмвер беше додаден софтверски модул кој ги праќа пресметаните вредности за временските интервали на чекорите (PSC, ARR, CCR и насоката за секоја оска) до компјутер преку сервиска комуникација. Зачуваните податоци беа обработени преку скрипти напишани во Пајтон за да се добијат граfiците за евалуација. За имплементација на овој експеримент беа користени првите 16 точки од G-код за обработка на основни геометриски фигури на дрво. Првите точки одговараат на првите сегменти од рамнењето на дрвото. Координатите на точките се дадени во табела 4.2.

На графикот на слика 4.1 е прикажана зависноста на брзината и поместувањето од времето за сите оски. Од овој график може да се заклучи дека профилите на брзината не се преклопуваат и се комплетно синхронизирани. Бидејќи се работи за висока резолуција (од 1 μm) бројот на генерирани точки е многу голем и грешката од еден такт циклус на тајмерот при генерирање на чекорите кумулативно може да предизвика голема десинхронизација која ќе биде видлива уште при првите сегменти, односно ќе се појави преклопување на профилите на брзината. Од граfiците за поместувањето може да се види точноста на позиционирањето на секоја од оските. За да се утврди брзината со која се движи главата на машината генериран е графикот на слика 4.2 кој ја дава зависноста на вкупната брзина од времето.

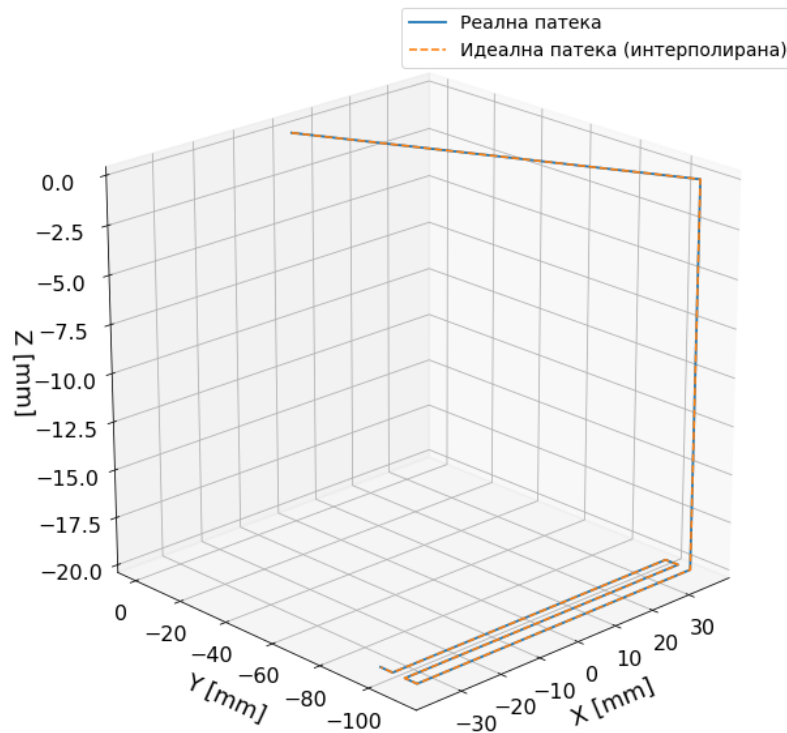


Слика 4.1. Промена на брзината и поместувањето по сите оски во зависност од времето за првите 16 точки од G-кодот.

На слика 4.3 е дадена споредбата на траекторијата генерирана од предложениот алгоритам според временските интервали на чекорите и идеалниот пат добиен математички со линеарна интерполација помеѓу точките од G-кодот во 3D-просторот. Од овој график визуелно не може да се забележи отстапување на генерираната патека означена со сина линија на графикот од математички интерполираната означена со портокалова испрекината линија. Ова означува дека отстапувањето е многу мало, па за попрецизно да се утврди разликата, дополнителна скрипта е направена која ја пресметува грешката на генерираната патека од математички добиената за секоја оска посебно. Овој график е даден на слика 4.4 и како што може да се види на графикот максималната грешка не надминува 0,005 mm. Оваа мала грешка се добива првично од

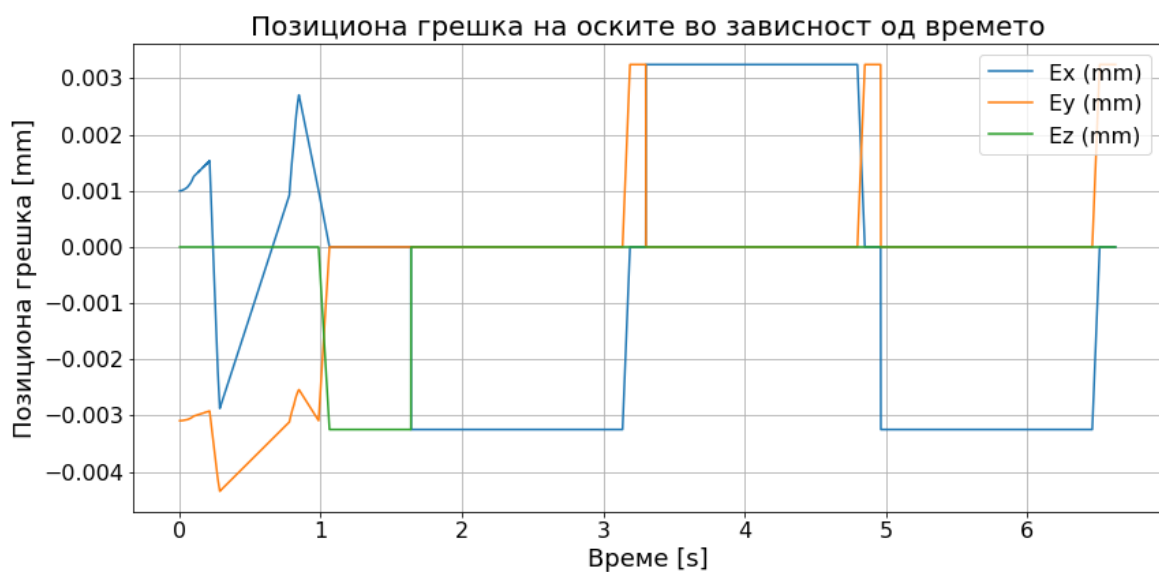


Слика 4.2. Промена на вкупната брзина (брзината со која се движи главата на машината) во зависност од времето.

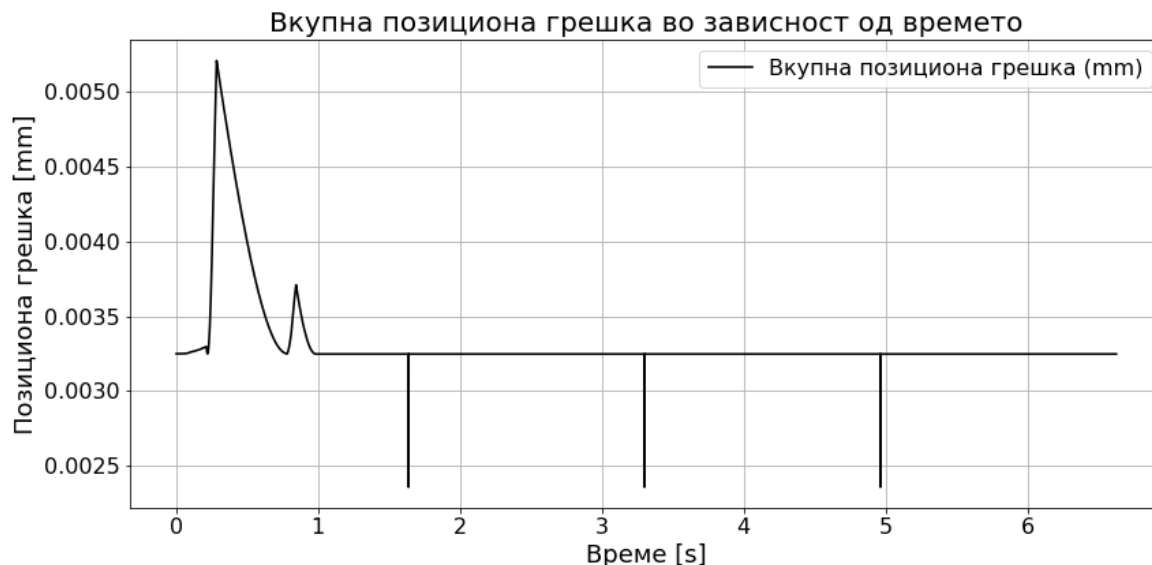


Слика 4.3. Споредба помеѓу генерираниот пат од страна на предложениот алгоритам и идеалниот пат добиен со интерполација во 3D-просторот.

конверзијата на пресметаните вредности за временските интервали во конечни целобројни вредности со ограничена резолуција на тајмерите. Со други зборови, малата грешка која се јавува е резултат на заокружувањето на пресметаните вредности и ограничената резолуција на тајмерите, а не поради непрецизност на предложениот алгоритам. Ако подобро се анализира графикот од сликата 4.4 може да се забележи дека



Слика 4.4. Позициона грешка на оските во зависност од времето.



Слика 4.5. Вкупна позициона грешка на главата на машината во зависност од времето.

отстапување од егзактно пресметаната патека се јавува само при движење на оските, грешката при позиционирањето на истите е нула. Според овие резултати ефективната резолуција на позиционирање на предложениот управувачки систем изнесува 0,001 mm, независно од механичките толеранции на системот.

Конечно, на слика 4.5 е дадено заедничкото отстапување на сите оски во зависност од времето. Грешката (отстапувањето) е пресметано како Евклидово растојание помеѓу генерираната и идеалната траекторија пресметано со следниот израз:

$$E_{3D}(t) = \sqrt{E_X(t)^2 + E_Y(t)^2 + E_Z(t)^2} \quad (4.8)$$

Резултатите од графикот на слика 4.5 покажуваат дека максималната грешка на реалната траекторија на движење на главата на машината не надминува 0.006 mm која се јавува на високи брзини каде конечната резолуција на тајмерите доаѓа повеќе до израз и се генерира поголема грешка.

Покрај тоа што овој експеримент е направен на ограничен број на сегменти, измерената грешка претставува реална потврда на перформансите на предложениот контролер, особено од аспект на точност на позиционирање и синхронизација помеѓу оските.

4.3 Експериментална валидација на временската точност на генерираните чекори преку споредба со пресметаните вредности и анализа на временските отстапувања

Предложената архитектура за извршување на движењата користи независни хардверски тајмери за секоја оска, синхронизирани преку заеднички механизам за синхронизација и DMA-пренос на однапред пресметаните временски параметри. Временските интервали помеѓу чекорите се пресметуваат однапред и се запишуваат во мемориските бафери, по што DMA-контролерот автоматски ги пренесува кон регистрите на тајмерите. Со ова се овозможува високо детерминистичко однесување на системот и рамномерна распределба на оптовареноста на јадрото на микроконтролерот.

Со цел експериментално да се верифицира временската точност на генерираните импулси, избран е тест пример со две синхронизирани оски (X и Y), при што се извршуваат две последователни линеарни движења со различни зададени брзини. За таа цел е употребена следната секвенца од G-код команди:

- G01 X40 Y35 F9000
- G01 X55 Y60 F3000

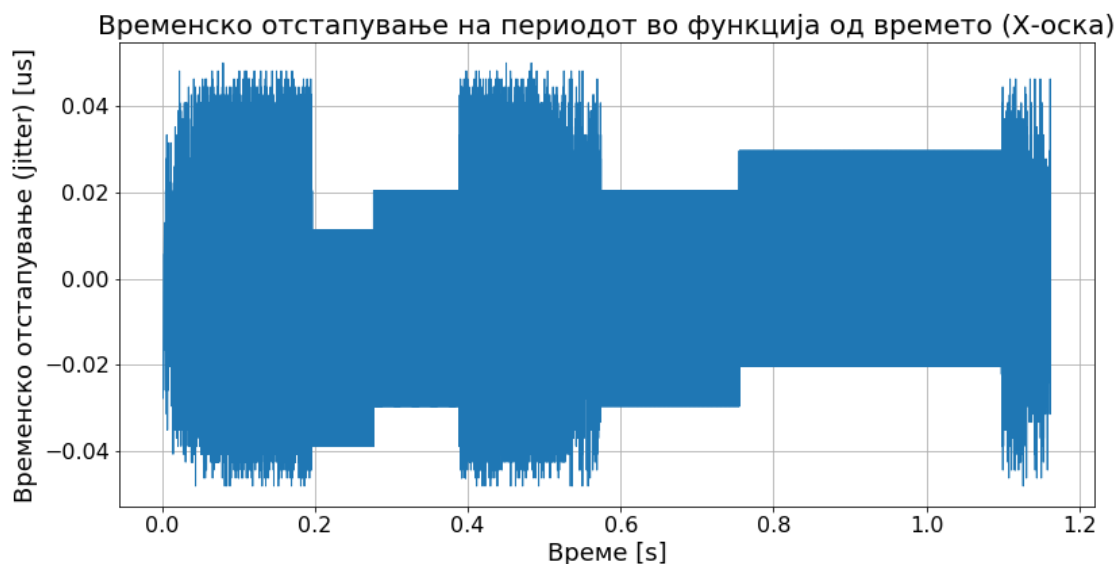
За време на извршувањето на движењето, временските интервали пресметани од алгоритмот се испраќаат до персонален компјутер преку сериска комуникација, додека пак генерираните сигнали кон моторите се мерат директно на влезовите на драјверите на моторите со користење на дигитален осцилоскоп Rigol MSO5354. Од добиените мерења од осцилоскопот се издвоени временските моменти на секој раб од сигналот и се пресметани реалните временски интервали помеѓу последователните чекори.

Потоа извршена е споредба помеѓу пресметаните и измерените временски интервали. На слика 4.6 е прикажано временското отстапување (jitter) за X-оската во функција од времето. Анализата е извршена врз 55 000 последователни чекори.

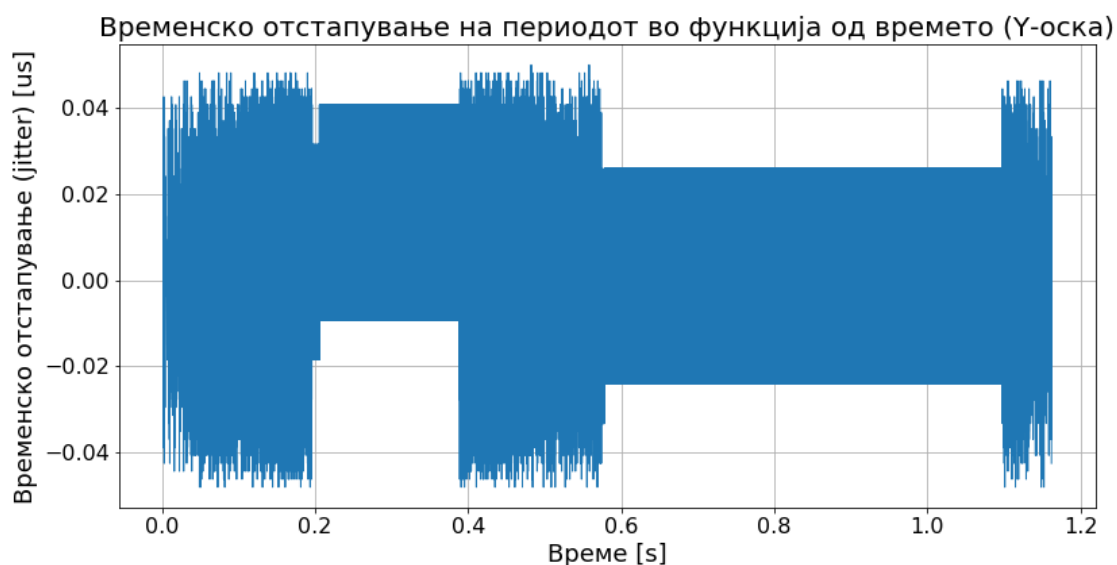
Добиени се следните резултати:

- Интервал на семплирање на осцилоскопот: 0.050 μ s.
- Максимално апсолутно временско отстапување: 0.050 μ s.
- RMS вредност на временското отстапување: 0.022 μ s.
- Средна вредност на временското отстапување: 0.000 μ s.
- Стандардна девијација: 0.022 μ s.

Од добиените резултати може да се забележи дека максималното измерено временско отстапување е ограничено од резолуцијата на самиот мерен инструмент.



Слика 4.6. Временско отстапување на периодот (jitter) во функција од времето за X-оската



Слика 4.7. Временско отстапување на периодот (jitter) во функција од времето за Y-оската.

Средната вредност на грешката е еднаква на нула, што укажува дека не постои систематско временско отстапување помеѓу пресметаните и измерените интервали. Поради тоа, преостанатите варијации од приближно $\pm 0.05 \mu\text{s}$ може да се препишат на процесот на дискретизација на осцилоскопот, а не на реално недетерминистичко однесување на предложениот систем.

На слика 4.7 е прикажана аналогната анализа за Y-оската, извршена врз 60 000 последователни чекори. Резултатите се конзистентни со резултатите добиени за X-оската и ја потврдуваат временската стабилност на системот при синхронизирано движење на повеќе оски.



Слика 4.8. Хистограм на измерените временски отстапувања на периодот за X-оската. Сите измерени вредности се наоѓаат во границите од $\pm 0.05 \mu\text{s}$, што одговара на интервалот на семплирање на осцилоскопот.

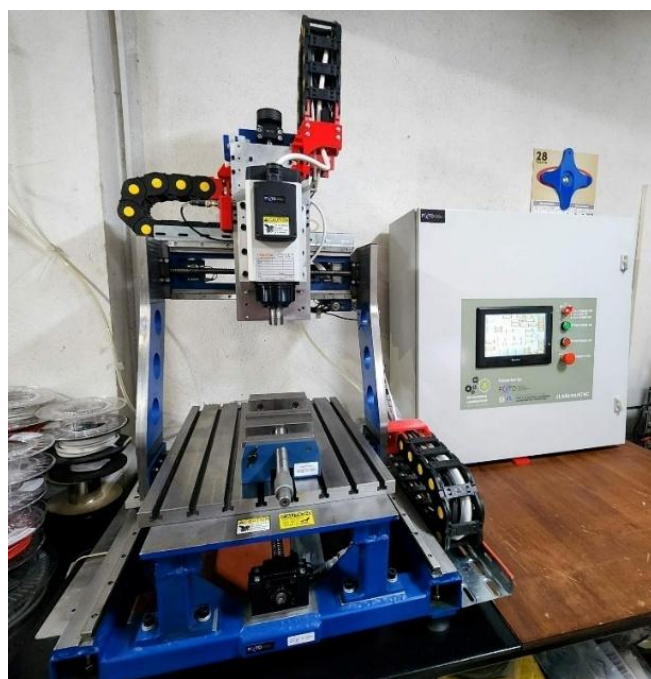
Дополнително, на слика 4.8 е прикажан хистограмот на измерените временски отстапувања за X-оската. Сите измерени вредности се наоѓаат во границите од $\pm 0.05 \mu s$, што повторно одговара на резолуцијата на осцилоскопот. Оттука може да се заклучи дека измерените варијации се ограничени од мерниот процес и не претставуваат реални временски флуктуации на предложениот управувачки систем.

Добиените резултати експериментално потврдуваат дека предложената архитектура обезбедува исклучително висока временска точност при генерирањето на чекорните импулси и детерминистичко однесување при синхронизирано управување со повеќе оски.

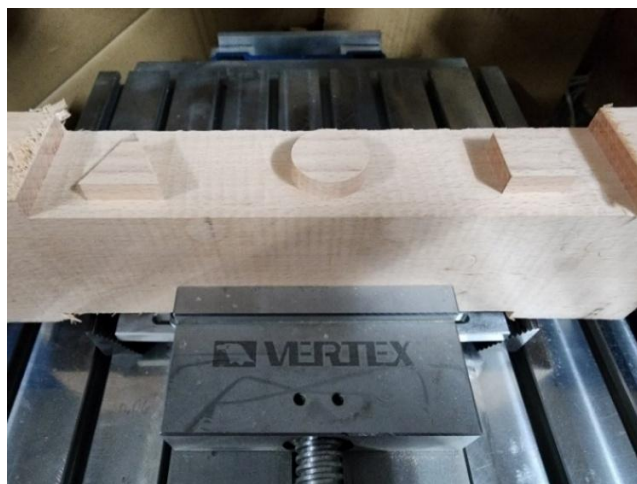
4.4 Практична валидација на предложениот систем преку експериментални обработки со различни алати изведени на самата машина

Со цел дополнителна валидација на перформансите на предложениот систем во реални услови, изведени се повеќе експерименти со реална обработка на самата CNC-машина, при што се користени различни материјали, димензии на алати и различни форми на траектории. Комплетната конфигурација и изгледот на изработената машина на која се изведени сите експерименти е прикажана на слика 4.9. Основната цел на овие експерименти е да се утврди дали прецизноста на следење на траекторијата, добиена во поглавјето 4.2, се пренесува и во процесот на обработка во реални услови.

На слика 4.10 е прикажана обработка во меко дрво на едноставни геометриски форми (кружна, триаголна и квадратна) со користење на двоперно глодало со димензија 12 mm. Покрај релативно големиот дијаметар на алатот и хетерогената структура на материјалот, измерените отстапувања на димензиите на изработените форми се под 0.02 mm. Површинската грубост на изработените форми во најголем дел се должи на големината на алатот и карактеристиките на материјалот, а не на неточности во управувањето со движењето на оските, што укажува на стабилна и конзистентна синхронизација на оските за време на процесот на обработка.



Слика 4.9. Изглед на изработената CNC-машина користена во сите експерименти.



Слика 4.10. Машинска CNC-обработка на основни геометриски форми на меко дрво со користење на двоперно глодало со дијаметар од 12 mm.

На слика 4.11 е прикажана обработка на лого со фини детали и форми на квалитетно сушено ореово дрво, изработено со троперно глодало со дијаметар од 4 mm. Посложените форми на логото во комбинација со помалиот алат бараат построга прецизност во следењето на траекторијата. Измерената грешка на контурите во овој случај е под 0,01 mm, што ја потврдува способноста на управувачкиот алгоритам да одржи висока прецизност на траекторијата при сложени динамични движења со чести промени на брзината и насоката.

Прототипна мала електронска плочка со димензии од 13×16 mm изработена на предложената CNC-машина со прецизен конусен алат за оваа намена е прикажана на слика 4.12. Кај изработената плочка репродуцирањето на минималната ширина и меѓусебното растојание помеѓу водовите е конзистентно со отстапувања помали од 0.01 mm. Поради ограничувањата на мерната опрема која се користеше за овие експерименти, вредности под дефинираниот праг не можат да бидат измерени, па затоа тие се прикажани како < 0,01 mm. Овие резултати ги потврдуваат перформансите на предложениот алгоритам за управување на CNC-машини и за високо прецизни



Слика 4.11. Машинска CNC-обработка во висококвалитетно суво дрво од орев со користење на троперно глодало со дијаметар од 4 mm.



Слика 4.12. Машинска CNC-обработка на прототип на мала електронска плочка со димензии 13×16 mm со користење на прецизен алат за гравирање.

апликации, како што е CNC-обработката на печатени кола, каде временската точност и синхронизацијата на оските директно влијаат врз геометриската точност.

Добиените експериментални резултати недвосмислено потврдуваат дека ниските грешки утврдени преку аналитичката евалуација на генерираните временски интервали на чекорите во секција 4.2 директно се рефлектираат во висока геометриска точност при реална CNC-обработка. Оваа корелација укажува дека предложениот систем за управување со движењето на CNC-машини овозможува стабилна, повторлива и високо прецизна точност на позиционирање во широк опсег на производствени сценарија, почнувајќи од груба обработка на материјал, па сè до апликации со висока резолуција како што е финото гравирање.

4.5 Перформанси на предложениот систем во конфигурација на 3D-печатење во хибридна и одвоена варијанта

Покрај опцијата за CNC-обработка, предложениот фирмвер овозможува и високопрецизно адитивно производство. Експерименталните мерења на 3D-печатените делови покажаа отстапувања помали од 0,01 mm по сите оски, што покажува висока прецизност и синхронизација на оските и при процесот на 3D-печатење.

Масивноста на главата за обработка која е фиксна во оваа хибридна конфигурација на машината има голема инерција што ги ограничува забрзувањето и максималната брзина на истата. За разлика од комерцијалните 3D-печатари кои се механички оптимизирани и подвижниот дел е само екструдерот кој е многу лесен, хибридниот дизајн мора да ја интегрира и главата за обработка, што резултира со многу поголема

инерција и намалени динамички карактеристики. Меѓутоа, ваквата архитектура овозможува мултифункционалност која не може да се постигне кај системи наменети исклучиво за 3D-печатење.

Предложениот фирмвер може да се користи и во наменска конфигурација само за адитивно производство кај 3D-печатарите. Во таквата конфигурација кај 3D-печатарите со многу помала инерција на подвижната глава, овој фирмвер може да постигне драстично повисоки брзини и забрзувања, достигнувајќи нивоа споредливи, а во одредени случаи и повисоки од оние кај современите високобрзински комерцијални 3D-печатари. Дополнителна предност на предложениот пристап е неговата хардверска ефикасност: целиот контролен систем е имплементиран исклучиво на еден микроконтролер, со што се елиминира потребата од процесорски платформи од повисока класа, кои најчесто се користат кај новите генерации на печатари со фирмвер како што е Клипер. Со тоа се намалуваат комплексноста на управувачкиот хардвер и цената на чинење, а истовремено се добиваат подобрени перформанси во однос на достапните системи и комплетно детерминистичко однесување во реално време.

Споредбата со типична механичка конфигурација на 3D-печатар, дополнително го нагласува потенцијалот на предложениот систем. Голем дел од денешните напредни 3D-печатари користат преносни ремени од типот GT2 и ременици со 20 заби. Кај овој тип на стандардни ремени распонот помеѓу два заби е 2 mm, па со една ротација на ременицата, се добива поместување од 40 mm на оската. Стандардните чекорни мотори кои се користат имаат по 200 чекори во една револуција, а напредните драјвери за истите со користење на техника на микростепинг ги делат основните чекори на моторот на 16 микрочекори, што резултира со ефективна резолуција од 12,5 μm добиена според следниот израз:

$$Step_{resolution} = \frac{40\text{mm}}{200 \times 16} = 12.5 \mu\text{m} \quad (4.9)$$

Ова е околу 12,5 пати помала резолуција во однос на резолуцијата на предложената машина. Максималната брзина на предложениот систем е 250 mm/s во конфигурација со резолуција од 1 μm , што значи дека во еквивалентна конфигурација на комерцијалните 3D-печатари, предложениот систем би постигнал брзини до ≈ 3125 mm/s, што претставува брзина околу 5 пати поголема во однос на брзината на денешните напредни комерцијални 3D-печатари, која изнесува околу 600 mm/s. Овие резултати покажуваат дека предложениот управувачки систем во конфигурација само за 3D-печатење има потенцијал да поддржи исклучително високи брзини на 3D-печатење, притоа задржувајќи ги супериорната прецизност и точност.

5 Заклучок

Во рамките на оваа докторска дисертација беше развиен, имплементиран и експериментално верификуван напреден систем за управување со хибридни CNC-машини. Акцентот беше ставен на високата временска и геометриска прецизност, детерминистичкото однесување во реално време и ефикасната имплементација на системот на хардвер со ограничени ресурси. За таа цел, најпрво беше дизајнирана и реализирана целосната механичка конструкција на машината, на која беа интегрирани погонските мотори, мерните сетила и грејачите. Електричното поврзување на сите компоненти беше внимателно изведено со цел да се обезбеди висока отпорност на системот кон надворешни пречки и електромагнетни влијанија, како и да се минимизира

меѓусебната интерференција помеѓу проводниците за напојување и сигналните проводници.

Следно, беше дизајнирана, изработена и интегрирана електронската управувачка единица, врз основа на чијашто архитектура беше формулиран математичкиот модел за напредно синхронизирано управување со движењата на оските на машината. Моделот беше ефикасно имплементиран на зададената хардверска платформа со ограничени ресурси, при што беа зачувани детерминистичкото реално-временско однесување и високата прецизност на управувањето. Реализираната машина овозможи систематска и објективна експериментална евалуација на перформансите на предложениот управувачки систем во реални работни услови.

Во експерименталната фаза за евалуација на математичкиот модел беа реализирани симулациски тестови врз реално генерираните временски импулси за чекорите, при што резултатите беа анализирани со нивно графичко претставување во форма на релевантни временски и кинематички зависности. Добиените резултати јасно покажуваат дека прецизното генерирање на временските интервали на чекорите, заедно со контролираното управување со забрзувањето и брзината, директно придонесуваат кон висока точност на позиционирање и стабилност на движењето. Експерименталната валидација потврди дека идентификуваните ниски временски грешки кои се јавуваат поради ограничената резолуција на тајмерите, не доведуваат до значајни акумулирани геометриски отстапувања, што резултира со повторлива прецизност при реална CNC-обработка и 3D-печатење.

Дополнително, во оваа докторска дисертација е демонстрирана применливоста на предложениот управувачки систем и во хибридни и во наменски конфигурации. Покажано е дека истата хардверска архитектура може да се користи и за CNC-машини и за 3D-печатари со што и во двете конфигурации се постигнува пониска цена на чинење во однос на комерцијално достапните верзии, а се задржува високата прецизност, брзината и квалитетот на изработените делови. Со тоа се потврдува дека предложеното решение претставува флексибилна и скалабилна платформа со значителен потенцијал за примена во широк опсег на индустриски и истражувачки апликации.

5.1 Придонеси во областа на истражувањето

Главните научни и инженерски придонеси на оваа докторска дисертација може да се сумираат во следните точки:

- Дизајнирана и изработена е хибридна CNC-машина која ги интегрира двете основни производствени технологии: адитивно и супстрактивно производство. Со оглед дека хибридното производство претставува клучен тренд во развојот на современите производствени системи, предложената машина обезбедува соодветна експериментална и истражувачка платформа за понатамошни научни и применети истражувања во оваа област.
- Развиен е оригинален систем за управување со движењето на хибридни CNC-машини, кој може да се користи и како управувачки систем со високи перформанси за 3D-печатари и високо прецизен управувачки систем за CNC-машини.
- Развиени се напредни оптимизирани алгоритми наменети за работа само на еден микроконтролер, без употреба на процесорски платформи или оперативни системи, што претставува значаен придонес во насока на намалување на хардверската комплексност, трошоците и недетерминистичкото однесување на

ваквите системи. Перформансите кои ги постигнува предложениот систем се споредливи со перформансите на најдобрите комерцијални системи кој користат сложени хардверски архитектури, а во некои параметри и подобри од нив.

- Предложен е аналитички и симулациски модел за евалуација на временските грешки при генерирање на чекорните импулси, со што е овозможено квантитативно поврзување помеѓу временската точност на управувањето и конечната геометриска точност на изведеното движење.
- Предложениот алгоритам може да придонесе за подобрување на перформансите и кај сложените архитектури со хост процесори каде сите пресметки ќе останат како задача на процесорот, но нивната имплементација на самиот микроконтролер и генерирањето на импулсите за чекорите може да се реализира преку предложениот алгоритам со што значително ќе се подобри временската нестабилност при генерирањето на чекорите.

5.2 Насоки за идни истражувања

Иако предложениот систем демонстрира високи перформанси и јасна практична применливост, резултатите од оваа докторска дисертација отвораат повеќе релевантни насоки за понатамошни истражувања и унапредувања. Идната работа може да се фокусира на проширување и дополнителна оптимизација на предложениот алгоритам, во согласност со заклучоците и ограничувањата идентификувани во текот на спроведеното истражување, со цел да се постигне целосно искористување на неговиот потенцијал и дополнително подобрување на системските перформанси.

Една од клучните насоки за идно унапредување е оптимизацијата на мемориските ресурси потребни за имплементација на предложениот алгоритам, чии побарувања се релативно високи кога се користи хардвер со ограничени ресурси базиран исклучиво на микроконтролер. Намалувањето на мемориските побарувања може да се постигне со елиминација на вредностите за прескалерите (PSC) од баферите во кои се складираат временските интервали за чекорите, при што нивната функционалност би се реализирала преку комбинирање на повеќе циклуси на тајмерите за движења со многу ниски брзини. Со ваков пристап се очекува намалување на потребната меморија за приближно 30 %, со што би се овозможила употреба на поголеми чекорни бафери и поддршка за дополнителни оски, односно проширување на системот кон посложени CNC-конфигурации.

Дополнителна насока за унапредување е намалувањето на оптоварувањето на јадрото на микроконтролерот, кое произлегува од генерирањето на прекин за секој чекор и за секој мотор, особено при високи работни брзини кога фреквенцијата на прекините е исклучително висока. Овој проблем може да се адресира со воведување на дополнителен тајмер во архитектурата на системот, целосно синхронизиран со останатите тајмери, кој би генерирал прекини само при премини помеѓу сегментите каде што е потребна промена на насоката на движење. Со ваква организација би се постигнало значително намалување на бројот на прекини и подобра искористеност на процесорските ресурси. Овие две подобрувања претставуваат логичен чекор во идната доработка на системот и се очекува значително да придонесат кон понатамошно зголемување на неговите перформанси.

Високата искористеност на ресурсите на микроконтролерот кај предложениот систем исто така отвора можност за негово тестирање и имплементација на помоќни микроконтролерски платформи со повисоки перформанси, особено кај архитектури со две процесорски јадра. Архитектурата на предложениот систем овозможува ефикасна

адаптација на вакви платформи, при што јадрото со повисоки перформанси (најчесто Cortex-M7) би се користело за имплементација на математичкиот модел и за пресметка на временските интервали за управување со тајмерите, додека второто јадро (најчесто Cortex-M4) би било задолжено за манипулација со чекорните бафери, иницирање на DMA-трансфери и извршување на останатите задачи на машината, како што се PID-управување со температурите во системот, обработка на сензорските податоци и мануелна контрола преку HMI-интерфејсот. Комуникацијата помеѓу двете јадра би се реализирала преку споделената RAM-меморија. Овој пристап претставува значајна насока за идно истражување и според досегашните резултати, би овозможил реализација на исклучително ефикасен и детерминистички управувачки систем.

Имплементацијата на предложениот систем на напредни двојадрени микроконтролери дополнително би овозможила проширување кон управување со CNC-машини со четири или пет оски, што претставува особено значајно поле за примена на управувачки решенија со ограничени хардверски ресурси. Потврдувањето на стабилноста и точноста на ваков систем кај сложени повеќеосни машини би имало значителен научен и практичен придонес за областа.

Конечно, дополнителна насока за идни истражувања претставува имплементацијата и тестирањето на предложениот систем кај 3D-печатари, каде што подвижната глава има мала маса и е составена исклучиво од екструдер. Ваквите конфигурации би овозможиле експериментална проверка на теоретски добиените резултати поврзани со високите брзини и динамичките перформанси кои може да ги постигне системот.

Сумирано, наведените насоки претставуваат логично продолжение на спроведеното истражување и јасно укажуваат дека предложениот управувачки систем поседува значителен потенцијал за понатамошен развој, унапредување и широка практична примена.

Список на користена литература

- [1] Dafflon, B., Moalla, N. & Ouzrout, Y. The challenges, approaches, and used techniques of CPS for manufacturing in Industry 4.0: a literature review. *Int J Adv Manuf Technol* 113, 2395–2412 (2021).
<https://doi.org/10.1007/s00170-020-06572-4>
- [2] M Eswaran, M V A Raju Bahubalendruni, Challenges and opportunities on AR/VR technologies for manufacturing systems in the context of industry 4.0: A state of the art review, *Journal of Manufacturing Systems*, Volume 65, 2022, Pages 260-278, ISSN 0278-6125.
<https://doi.org/10.1016/j.jmsy.2022.09.016>
- [3] Saurabh Vaidya, Prashant Ambad, Santosh Bhosle, Industry 4.0 – A Glimpse, *Procedia Manufacturing*, Volume 20, 2018, Pages 233-238, ISSN 2351-9789.
<https://doi.org/10.1016/j.promfg.2018.02.034>.
- [4] Lucas Santos Dalenogare, Guilherme Brittes Benitez, Néstor Fabián Ayala, Alejandro Germán Frank, The expected contribution of Industry 4.0 technologies for industrial performance, *International Journal of Production Economics*, Volume 204, 2018, Pages 383-394, ISSN 0925-5273.
<https://doi.org/10.1016/j.ijpe.2018.08.019>.
- [5] Lim, JM., Song, W., Lee, JS. et al. Recent Advances in CNC Technology: Toward Autonomous and Sustainable Manufacturing. *Int. J. Precis. Eng. Manuf.* 26, 2311–2344 (2025).
<https://doi.org/10.1007/s12541-025-01334-2>
- [6] Abdulhameed O, Al-Ahmari A, Ameen W, Mian SH. Additive manufacturing: Challenges, trends, and applications, *Advances in Mechanical Engineering*, Vol. 11(2), 2019.
<https://doi.org/10.1177/1687814018822880>
- [7] Zhu, Z., Dhokia, V. G., Nassehi, A., & Newman, S. T. (2013). A review of hybrid manufacturing processes – state of the art and future perspectives, *International Journal of Computer Integrated Manufacturing*, 26(7), 596–615.
<https://doi.org/10.1080/0951192X.2012.749530>
- [8] Marcel Müller, ElmarWings, An Architecture for Hybrid Manufacturing Combining 3D Printing and CNC Machining, *International Journal of Manufacturing Engineering*, Vol. 2016, Article ID 8609108.
<https://doi.org/10.1155/2016/8609108>
- [9] A.A. Krimpenis, D.M. Iordanidis, Design and Analysis of a Desktop Multi-Axis Hybrid Milling-Filament Extrusion CNC Machine Tool for Non-Metallic Materials, *Machines* Vol. 11, 637, 2023.
<https://doi.org/10.3390/machines11060637>
- [10] L. V. T. Martinez, J-F. Witz, D. Najjar, X. Boidin, F. Lesaffre, V. Martin, S. Badin, E. Berte, Hybrid FFF/CNC: An open source hardware & software system, *HardwareX* Vol. 18, E00536, June 2024.
<https://doi.org/10.1016/j.ohx.2024.e00536>
- [11] Taku Yamazaki, Development of A Hybrid Multi-tasking Machine Tool: Integration of Additive Manufacturing Technology with CNC Machining, *Procedia CIRP*, Vol. 42, 2016, pp. 81-86.
<https://doi.org/10.1016/j.procir.2016.02.193>

- [12] J. Wolf, K.T. Werkle, H-C. Mohring, Study on Dynamic Behaviour in FFF 3D-printing with Crossed Gantry Kinematic, *Procedia CIRP* Vol. 121 (2024), pp. 162–167.
<https://doi.org/10.1016/j.procir.2023.09.244>
- [13] K. Yu, Z. Zhang, Z. Zhou and M. Dai, Application of the Five-phase S-curve Velocity Model on FDM Three-dimensional Printer, 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2020, pp. 1365-1371.
<https://doi.org/10.1109/ITOEC49072.2020.9141587>
- [14] R. Dorado-Vicente, Medina-Sánchez, García-Collado, D. Carou, M. Pérez, How to use and compare interpolation schemes in Fused Deposition Modeling, *Procedia Manufacturing* Vol. 41 (2019), pp. 343–350.
<https://doi.org/10.1016/j.promfg.2019.09.018>
- [15] D. Ahlers, F. Wasserfall, J. Hörber, J. Zhang, Automatic in-situ error correction for 3D printed electronics, *Additive Manufacturing Letters*, Vol. 7, 2023, 100164.
<https://doi.org/10.1016/j.addlet.2023.100164>
- [16] E. Kopets, A. Karimov, L. Scalera, D. Butusov, Estimating Natural Frequencies of Cartesian 3D Printer Based on Kinematic Scheme *Applied Science*, 2022, Vol. 12, 4514.
<https://doi.org/10.3390/app12094514>
- [17] Murr, L.E. A Metallographic Review of 3D Printing/Additive Manufacturing of Metal and Alloy Products and Components. *Metallography, Microstructure, and Analysis*, 7, 103–132 (2018).
<https://doi.org/10.1007/s13632-018-0433-6>
- [18] Thomas Duda, L. Venkat Raghavan, 3D Metal Printing Technology, *IFAC-PapersOnLine*, Volume 49, Issue 29, 2016, Pages 103-110, ISSN 2405-8963,
<https://doi.org/10.1016/j.ifacol.2016.11.111>.
- [19] Chahal, V., & Taylor, R. M. (2020). A review of geometric sensitivities in laser metal 3D printing. *Virtual and Physical Prototyping*, 15(2), 227–241.
<https://doi.org/10.1080/17452759.2019.1709255>
- [20] Basavraj Gadagi, Ramesh Lekurwale, A review on advances in 3D metal printing, *Materials Today: Proceedings*, Volume 45, Part 1, 2021, Pages 277-283, ISSN 2214-7853,
<https://doi.org/10.1016/j.matpr.2020.10.436>.
- [21] H. Giberti, L. Sbaglia, M. Silvestri, Mechatronic Design for an Extrusion-Based Additive Manufacturing Machine, *Machines*, Vol. 5, 29, 2017.
<https://doi.org/10.3390/machines5040029>
- [22] Munish Kumar, P Khatak, HC Garg, Advancements in CNC interpolators for linear and parametric curve approximation and command, National Conference on Trends and Advances in Mechanical Engineering, MCA University of Science and Technology, Haryana, 2017.
- [23] R. Ward, B. Sencer, G. Panoutsos, E. Ozturk, On-the-fly cnc interpolation using frequency-domain fft-based filtering, *Procedia CIRP* Vol. 107 (2022), pp. 1571–1576.
<https://doi.org/10.1016/J.PROCIR.2022.05.193>.
- [24] D. Lyu, Q. Liu, H. Liu, W. Zhao, Dynamic error of CNC machine tools: a state-of-the-art review. *International Journal of Advanced Manufacturing Technology*, Vol. 106, pp. 1869–1891 (2020).
<https://doi.org/10.1007/s00170-019-04732-9>

- [25] B. Sencer, K. Ishizaki, E. Shamoto, A curvature optimal sharp corner smoothing algorithm for high-speed feed motion generation of NC systems along linear tool paths, *International Journal of Advanced Manufacturing Technology*. Vol. 76, pp. 1977-1992.
<http://dx.doi.org/10.1007/s00170-014-6386-2>
- [26] X. Du, J. Huang, L-M. Zhu, H. Ding, An error-bounded B-spline curve approximation scheme using dominant points for CNC interpolation of micro-line toolpath, *Robotics and Computer-Integrated Manufacturing*, Vol. 64, 2020, 101930.
<https://doi.org/10.1016/j.rcim.2019.101930>
- [27] Qitao Tan, Mohd Ariffanan Mohd Basri, Hardware–Software Embedded System for Real-Time Trajectory Planning of Multi-Axis Machine Using B-Spline Curve Interpolation Algorithm, *Machines*, Vol. 11, 1043, 2023.
<https://doi.org/10.3390/machines11121043>
- [28] Q-B. Xiaoa, M. Wan, Y. Liua, X-B. Qina, W-H. Zhanga, Space corner smoothing of CNC machine tools through developing 3D general clothoid, *Robotics and Computer-Integrated Manufacturing*, Vol. 64, 2020, 101949.
<https://doi.org/10.1016/j.rcim.2020.101949>
- [29] D-K. Baek, T-J. Ko, S-H. Yang, Fast and precision NURBS interpolator for CNC systems, *International Journal of Precision Engineering and Manufacturing* Vol. 13, No. 6, pp. 955-961.
<https://doi.org/10.1007/s12541-012-0124-1>
- [30] Shijun Ji, Lianggen Lei, Ji Zhao, Xiuquan Lu, Hong Gao, An adaptive real-time NURBS curve interpolation for 4-axis polishing machine tool, *Robotics and Computer-Integrated Manufacturing*, Vol. 67, 2021, 102025.
<https://doi.org/10.1016/j.rcim.2020.102025>
- [31] Jiangang Li, Qian Wang, Ganggang Zhong, Planar tool radius compensation for CNC systems based on NURBS interpolation, *Mechanics & Industry* Vol. 21, 107 (2020).
<https://doi.org/10.1051/meca/2019044>
- [32] Valere Huypens, Constant speed lines—curves—NURBS reference pulse IPOs (part I), *International Journal of Advanced Manufacturing Technology* Vol. 111, pp. 1247–1275 (2020).
<https://doi.org/10.1007/s00170-020-05339-1>
- [33] P. Guo, Y. Wu, G. Yang, Z. Shen, H. Zhang, P. Zhang, F. Lou, H. Li, A Feedrate Planning Method for the NURBS Curve in CNC Machining Based on the Critical Constraint Curve, *Applied Science*, 2021, Vol. 11, 4959.
<https://doi.org/10.3390/app11114959>
- [34] Chih-Hsing Chu, Hsiang-Yun Chen, Chun-Hao Chang, Continuity-preserving tool path generation for minimizing machining errors in five-axis CNC flank milling of ruled surfaces, *Journal of Manufacturing Systems*, Vol. 55, 2020, pp. 171-178.
<https://doi.org/10.1016/j.jmsy.2020.03.004>
- [35] Y. Jin, Q. Bi, Y. Wang, Dual-Bézier path smoothing and interpolation for five-axis linear tool path in workpiece coordinate system. *Advances in Mechanical Engineering*. 2015, Vol. 7(7).
<https://doi.org/10.1177/1687814015595211>

- [36] Shujie Sun, Yusuf Altintas, A G3 continuous tool path smoothing method for 5-axis CNC machining, *CIRP Journal of Manufacturing Science and Technology*, Vol. 32, pp. 529-549, 2021.
<https://doi.org/10.1016/j.cirpj.2020.11.002>
- [37] Ming Yang, Jixiang Yang, Limin Zhu, Xi Yu, A novel curvature circle iterative algorithm for contour error control of multi-axis CNC machine tools, *Precision Engineering*, Vol. 65, pp. 23-31, 2020.
<https://doi.org/10.1016/j.precisioneng.2020.05.005>
- [38] Burak Sencer, Yasuhiro Kakinuma, Yuki Yamada, Linear Interpolation of machining tool-paths with robust vibration avoidance and contouring error control, *Precision Engineering*, Vol. 66, pp. 269-281, 2020.
<https://doi.org/10.1016/j.precisioneng.2020.04.007>
- [39] Weixin Wang, Chuxiong Hu, Kai Zhou, Suqin He, Limin Zhu, Local asymmetrical corner trajectory smoothing with bidirectional planning and adjusting algorithm for CNC machining, *Robotics and Computer-Integrated Manufacturing*, Vol. 68, 102058, 2021.
<https://doi.org/10.1016/j.rcim.2020.102058>
- [40] R. Ward, B. Sencer, B. Jones, E. Ozturk, Accurate prediction of machining feedrate and cycle times considering interpolator dynamics, *International Journal of Advanced Manufacturing Technology* Vol. 116, pp. 417–438 (2021).
<https://doi.org/10.1007/s00170-021-07211-2>
- [41] Noor Hatem, Yusri Yusof, Aini Zuhra A. Kadir, Kamran Latif, M.A Mohammed, A novel integrating between tool path optimization using an ACO algorithm and interpreter for open architecture CNC system, *Expert Systems with Applications*, Vol. 178, 2021, 114988.
<https://doi.org/10.1016/j.eswa.2021.114988>
- [42] Cristina Martínez-Ruedas, Francisco José Adame-Rodríguez, Juan Manuel Díaz-Cabrera, Integrating and interconnecting of older SINUMERIK CNC machines with industry 4.0 using a plug-and-play system, *Journal of Industrial Information Integration*, Vol. 38, 100583, 2024.
<https://doi.org/10.1016/j.jii.2024.100583>
- [43] ISO 6983-1:2009, Automation systems and integration - Numerical control of machines - Program format and definition of address words.
- [44] R. Ward, B. Sencer, Accurate TCP Position and Orientation Trajectory Generation in 6DOF Robotic Manipulators and CNC Machine Tools using FIR Filtering and Haversine Synchronisation, *Procedia CIRP* Vol. 120 (2023), pp. 27–32.
<https://doi.org/10.1016/j.procir.2023.08.006>
- [45] R. Zhao, S. Ratchev, On-line trajectory planning with time-variant motion constraints for industrial robot manipulators, 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 3748-3753.
<https://doi.org/10.1109/ICRA.2017.7989432>
- [46] Koulamas, C.; Lazarescu, M.T. Real-Time Embedded Systems: Present and Future. *Electronics* 2018, 7, 205.
<https://doi.org/10.3390/electronics7090205>
- [47] Olmedo-García, L.F.; García-Martínez, J.R.; Cruz-Miguel, E.E.; Barra-Vázquez, O.A.; González-Lee, M.; Martínez-Sánchez, T. Real-Time Embedded System-Based Approach for Sensing Power Consumption on Motion Profiles. *Electronics* 2023, 12, 3853.

<https://doi.org/10.3390/electronics12183853>

- [48] Alonso, S.; Muguira, L.; Garate, J.I.; Cuadrado, C.; Bidarte, U. Interrupt Latency Accurate Measurement in Multiprocessing Embedded Systems by Means of a Dedicated Circuit. *Electronics* 2024, 13, 1626.
<https://doi.org/10.3390/electronics13091626>
- [49] Rubattu, C.; Ledda, A.; Ratto, F.; Jugade, C.; Goswami, D.; Palumbo, F. FPGA-based Implementation for Industrial Motion Control System. In *Proceedings of the 2024 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, San Francisco, CA, USA, 27–31 May 2024; pp. 141–147.
<https://doi.org/10.1109/IPDPSW63119.2024.00039>
- [50] Feng, Q.; Wang, L. FPGA-based acceleration and deceleration control for CNC machine tools. In *Proceedings of the 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*, Shenyang, China, 20–22 December 2013; pp. 210–214.
<https://doi.org/10.1109/MEC.2013.6885074>
- [51] Hidayatullah, M.; Hariadi, F.I.; Sasongko, A. Development of interface and coordination module of FPGA-based controller for CNC PCB milling and drilling machine. In *Proceedings of the 2017 International Symposium on Electronics and Smart Devices (ISESD)*, Yogyakarta, Indonesia, 17–19 October 2017; pp. 112–117.
<https://doi.org/10.1109/ISESD.2017.8253315>
- [52] Li, X.; Wang, D.; Luan, X. FPGA-based PSO function design for CNC systems. In *Proceedings of the 2023 IEEE 18th Conference on Industrial Electronics and Applications (ICIEA)*, Ningbo, China, 18–22 August 2023; pp. 141–145.
<https://doi.org/10.1109/ICIEA58696.2023.10241472>
- [53] Tao, Y.; Lin, H.; Hu, Y.; Zhang, X.; Wang, Z. Efficient implementation of CNC Position Controller using FPGA. In *Proceedings of the 2008 6th IEEE International Conference on Industrial Informatics*, Daejeon, Republic of Korea, 13–16 July 2008; pp. 1177–1182.
<https://doi.org/10.1109/INDIN.2008.4618281>
- [54] Li, Y.-F.; Xu, Z.-W.; Zhang, Y.-J.; Wu, Y.-B.; Chuang, L. On the Design of Simultaneous Motion Controller for the Four-axis SCARA System. In *Proceedings of the 2023 IEEE 14th International Conference on Power Electronics and Drive Systems (PEDS)*, Montreal, QC, Canada, 7–10 August 2023; pp. 1–6.
<https://doi.org/10.1109/PEDS57185.2023.10246556>
- [55] Tran, H.G.; Le, K.S.; Bui, T.L.; Nguyen, H.H. An Improved Error Interpolator of Milling CNC Controller Based on FPGA. In *Proceedings of the 2020 International Conference on Advanced Mechatronic Systems (ICAMechS)*, Hanoi, Vietnam, 10–13 December 2020; pp. 104–107.
<https://doi.org/10.1109/ICAMechS49982.2020.9310160>
- [56] Przybył, A. FPGA-Based Optimization of Industrial Numerical Machine Tool Servo Drives. *Electronics* 2023, 12, 3585.
<https://doi.org/10.3390/electronics12173585>
- [57] Jia, Z.; Cui, G.; Li, X.; Shen, M.; Liu, X.; Dong, S.; He, F. A DMA-Based High-Precision Control Method for Attitude Control Flywheels. In *Proceedings of the 2025 12th International Forum on Electrical Engineering and Automation (IFEEA)*, Xi'an, China, 7–9 November 2025; pp. 59–65.
<https://doi.org/10.1109/IFEEA66847.2025.11388194>

- [58] Da Rocha, P.A.S.; De Oliveira, W.D.; De Lima Tostes, M.E. An Embedded System-Based Snap Constrained Trajectory Planning Method for 3D Motion Systems. *IEEE Access* 2019, 7, 125188–125204.
<https://doi.org/10.1109/ACCESS.2019.2939116>
- [59] Ramezani, H.; Torbensen, R.; Duggen, L.; Jouffroy, J. An Accurate Programmable Pulse Generator for Stepper Actuated Real-Time Control Systems. In *Proceedings of the 2019 IEEE International Conference on Mechatronics (ICM)*, Ilmenau, Germany, 18–20 March 2019; pp. 358–363.
<https://doi.org/10.1109/ICMECH.2019.8722832>
- [60] Wang, T.; Liu, Q.; Wang, L. An RTOS-based embedded CNC system. In *Proceedings of the 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering*, Changchun, China, 24–26 August 2010; pp. 33–36.
<https://doi.org/10.1109/CMCE.2010.5609563>
- [61] Rocha, P.; Tostes, E. Development of an Embedded CNC Control System. In *Proceedings of the 2018 13th IEEE International Conference on Industry Applications (INDUSCON)*, Sao Paulo, Brazil, 12–14 November 2018; pp. 407–412.
<https://doi.org/10.1109/INDUSCON.2018.8627296>
- [62] Liu, J.; Fu, Y.; Han, Z.; Fu, H. Design of an industrial Ethernet based embedded open architecture CNC system. In *Proceedings of the 2015 International Conference on Estimation, Detection and Information Fusion (ICEDIF)*, Harbin, China, 10–11 January 2015; pp. 413–417.
<https://doi.org/10.1109/ICEDIF.2015.7280154>
- [63] Jiankang, L.; Yunzhong, F.; Hongya, F. On-demand real-time scheduling strategy for CNC system based on embedded multi-core processor. In *Proceedings of the 2016 International Symposium on Flexible Automation (ISFA)*, Cleveland, OH, USA, 1–3 August 2016; pp. 202–205.
<https://doi.org/10.1109/ISFA.2016.7790161>
- [64] Gunawan, H.; Sugiono, D.; Tu, R.-Q.; Jong, W.-R.; Mufarrih, A. Enhanced Computer Numeric Controller Milling Efficiency via Air-Cutting Minimization Using Logic-Based Benders Decomposition Method. *Electronics* 2025, 14, 2613.
<https://doi.org/10.3390/electronics14132613>
- [65] Liu, B.; Zhang, H.; Liu, Y.; Lu, M. A Feedrate Planning Method in CNC System Based on Servo Response Error Model. *Electronics* 2023, 12, 3150.
<https://doi.org/10.3390/electronics12143150>
- [66] Grbl Developers. GRBL: CNC Motion Control Firmware [Software]. 2024. Available online: <https://github.com/grbl/grbl> (accessed on 20 December 2025).
- [67] Grbl Developers. grblHAL: CNC Motion Control Firmware [Software]. 2024. Available online: <https://github.com/grblHAL> (accessed on 10 November 2025).
- [68] Klipper Developers. Klipper 3D Printer Firmware [Software]. 2024. Available online: <https://www.klipper3d.org/> (accessed on 20 November 2025).
- [69] LinuxCNC Community. LinuxCNC Documentation [Software/Documentation]. Available online: <https://linuxcnc.org> (accessed on 5 December 2025).
- [70] Modbus Organization, MODBUS Application Protocol Specification V1.1b3, 2012. [Online]. Available:
<https://www.modbus.org/file/secure/modbusprotocolspecification.pdf>

- [71] H. W. Ott, Electromagnetic Compatibility Engineering, 1st ed. Hoboken, NJ, USA: Wiley, 2009. [Online]. Available: <https://onlinelibrary.wiley.com/doi/book/10.1002/9780470508510>
- [72] IEC 61800-3:2004, Adjustable speed electrical power drive systems – Part 3: EMC requirements and specific test methods, International Electrotechnical Commission, 2004. [Online]. Available: <https://webstore.iec.ch/en/publication/65056>
- [73] Delta Electronics, “ASDA-B2 Series AC Servo Drive User Manual,” Delta AC Drives, Taiwan, 2020. [Online]. Available: <https://deltaacdrives.com/Delta-ASDA-B2-User-Manual.pdf>
- [74] Yaskawa Electric, “GA500 Technical Manual,” Yaskawa Electric Corporation, Kitakyushu, Japan, 2019. [Online]. Available: https://ga500.yaskawa-perform.de/wp-content/uploads/2019/04/Documentation/GA500_Technical_Manual.pdf
- [75] Toshiba Electronic Devices & Storage Corporation, TB6600HG Stepper Motor Driver IC Datasheet, Revision 2016-06-10. Available online: https://toshiba.semicon-storage.com/info/TB6600HG_datasheet_en_20160610.pdf (accessed on 12 June 2025).
- [76] T. Hopkins, “Stepper motor driving,” STMicroelectronics Application Note AN235, Nov. 2012. [Online]. Available: https://www.st.com/resource/en/application_note/an235-stepper-motor-driving-stmicroelectronics.pdf
- [77] STMicroelectronics, “UM1974 – STM32 Nucleo-144 Boards (MB1137) User Manual,” STMicroelectronics, 2019. [Online]. Available: https://www.st.com/resource/en/user_manual/um1974-stm32-nucleo144-boards-mb1137-stmicroelectronics.pdf
- [78] STMicroelectronics, “RM0410 – STM32F76xxx and STM32F77xxx Advanced Arm-based 32-bit MCUs Reference Manual,” STMicroelectronics, 2023. [Online]. Available: https://www.st.com/resource/en/reference_manual/rm0410-stm32f76xxx-and-stm32f77xxx-advanced-armbased-32bit-mcus-stmicroelectronics.pdf
- [79] GNU General Public License, “GNU General Public License, version 3,” Available: <https://www.gnu.org/licenses/gpl-3.0.en.html>
- [80] Kingroon, “Best Klipper 3D Printers (in 2024) – A Detailed Overview,” Available: <https://kingroon.com/blogs/3d-print-101/best-klipper-3d-printers> (Accessed: Feb. 26, 2025).
- [81] ARM, Technical Reference Manual, Arm Cortex-M7 Processor, DDI 0489F (ID121118), 2018
- [82] ARM, Technical Reference Manual, Arm Cortex-M4 Processor, 100166_0001_04_en, 2020