

# Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: informatica

## **Masterthesis**

### **Automatisatie van object toevoer in vrije val met behulp van etherCAT**

#### **Lowie Nijssen**

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: informatica

#### **PROMOTOR :**

Prof. dr. Nick MICHIELS

#### **PROMOTOR :**

ir. Ruben PRAETS

Gezamenlijke opleiding UHasselt en KU Leuven



Universiteit Hasselt | Campus Diepenbeek | Faculteit Industriële Ingenieurswetenschappen | Agoralaan Gebouw H - Gebouw B | BE 3590 Diepenbeek

Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE 3590 Diepenbeek  
Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE 3500 Hasselt



**2022**  
**2023**

# Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: informatica

## ***Masterthesis***

### ***Automatisatie van object toevoer in vrije val met behulp van etherCAT***

#### **Lowie Nijssen**

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: informatica

#### **PROMOTOR :**

Prof. dr. Nick MICHIELS

#### **PROMOTOR :**

ir. Ruben PRAETS



**KU LEUVEN**



# Woord vooraf

Met deze masterproef sluit ik mijn opleiding tot industrieel ingenieur af. Dit zou echter niet mogelijk geweest zijn zonder de steun van verschillende mensen. Vandaar een woord van dank.

Om te beginnen bij ir. Praets Ruben en prof. dr. Michiels Nick voor hun begeleiding en steun bij deze proef. Hierbij mag ook zeker Qcify niet vergeten worden, alle apparatuur benodigd voor deze proef werd namelijk door hen zonder fout voorzien. Ook kon ik altijd kennis (van andere vakgebieden) opdoen bij verschillende werknemers van het bedrijf om meer (en beter) inzicht over het globale systeem te verkrijgen.

Vervolgens wil ik ook mijn teamgenoten van verschillende onderwaterhockey clubs (Orca Bilzen, Koraal Genk, Astacus Maastricht) als ook mijn vrienden bedanken voor zowel de mentale steun als de nodige afleiding en alle prachtige momenten die de drukte verdraagbaar maakte.

Ten slotte wil ik ook mijn familie bedanken voor hun steun met voornamelijk mijn ouders voor de kans om deze opleiding te kunnen volgen.

Buiten dit woord van dank wil ik ook melden dat bepaalde termen in deze masterproef origineel in het engels zijn en zo behouden zijn omdat hier of geen goede vertaling voor is of de vertaling zelden in de context gebruikt wordt.



# Inhoudsopgave

Voorwoord	i
Lijst met tabellen	v
Lijst met figuren	vii
Abstract	ix
Abstract in English	xi
<b>1 Inleiding</b>	<b>1</b>
<b>2 Datatransmissie en aansturing van de actuator</b>	<b>3</b>
2.1 De proefopstelling . . . . .	3
2.2 EtherCAT . . . . .	3
2.2.1 Opbouw van een etherCAT netwerk . . . . .	4
2.2.2 Communicatie via etherCAT . . . . .	5
2.2.3 EtherCAT state machine . . . . .	6
2.3 Simple open etherCAT master (SOEM) . . . . .	7
2.3.1 Netwerk configuratie en communicatie met SOEM . . . . .	7
2.3.2 Privileges van SOEM . . . . .	8
2.3.3 Toevoegen van SOEM in de Qcify codebase . . . . .	9
2.4 Data transmissie met SOEM . . . . .	10
2.5 Data transmissie tussen de controller en randapparatuur . . . . .	12
2.6 Resultaten bij een constante stuurspanning . . . . .	13
<b>3 Signaal regeling op basis van snelheid</b>	<b>17</b>
3.1 Bepalen van de snelheid . . . . .	17
3.1.1 Moving average . . . . .	18
3.2 De PID regelaar . . . . .	19
3.2.1 De proportionele Regelaar (P-regelaar) . . . . .	20
3.2.2 De integrerende regelaar (I-regelaar) . . . . .	20
3.2.3 De differentiërende actie (D-actie) . . . . .	20
3.3 Implementatie van de PID regelaar . . . . .	21
3.4 Resultaten van de actuator gecontroleerd door een PID regelaar . . . . .	23
<b>4 Inclusie van de buffer in de regelaar</b>	<b>27</b>
4.1 Bepalen van het foutsignaal . . . . .	27

4.2 Resultaten van de aangepaste PID regelaar . . . . .	28
<b>5 Conclusie</b>	<b>31</b>
<b>Literatuurlijst</b>	<b>35</b>
<b>A Technische data van de shaker</b>	<b>37</b>
<b>B Signaal convertor irg1-s</b>	<b>41</b>
<b>C EtherCAT modules: technische data</b>	<b>47</b>

# Lijst van tabellen

2.1	Working counter increments . . . . .	5
3.1	Parameters PID regelaar . . . . .	24
4.1	Parameters PID regelaar met aangepast feedback signaal . . . . .	28





# Lijst van figuren

2.1	baan voor vallende objecten . . . . .	4
2.2	Schematische voorstelling van de actuator sturing . . . . .	4
2.3	EtherCAT group logo . . . . .	5
2.4	EtherCAT on the fly . . . . .	5
2.5	EtherCAT state machine . . . . .	6
2.6	Find adapter code snippet . . . . .	7
2.7	Set modules state code snippet . . . . .	8
2.8	Config modules code snippet . . . . .	8
2.9	Build and run script . . . . .	9
2.10	voorbeeld vcpkg.json voor het gebruik van soem . . . . .	9
2.11	CMake list . . . . .	10
2.12	Module info log . . . . .	11
2.13	Controller schema voor een constante sturing . . . . .	11
2.14	EL4001 sturing naar een voltmeter . . . . .	13
2.15	EL4001 sturing naar een voltmeter . . . . .	13
2.16	Shaker gevuld met macadamia noten . . . . .	14
2.17	Histogram van interval van amandelen bij 10v sturing . . . . .	14
2.18	Frequentie in functie van de spanning voor amandelen . . . . .	15
3.1	Moving average . . . . .	19
3.2	Riemanns som . . . . .	22
3.3	Schema na implementatie van de regelaar . . . . .	23
3.4	Spanning verloop van de PID regelaar bij nieuwe observaties . . . . .	24
3.5	Spanning en snelheid verloop van de PID regelaar bij vast interval . . . . .	25
3.6	Spanning en snelheid verloop van de PID regelaar bij vast interval . . . . .	25
4.1	Spanning en snelheid verloop van de actuator bij de aangepaste PID regelaar . . . . .	29
4.2	Aantal observaties in de buffer in functie van de tijd . . . . .	29
4.3	Foutsignalen in functie van de tijd . . . . .	30



# Abstract

Qcify ontwikkelt machines om de kwaliteit van noten te garanderen. Een van deze machines is een batchcontrolesysteem waarvan de actuator herzien wordt. De actuator is verantwoordelijk voor de afzonderlijke toevoer van noten aan een observator die observaties opslaat in een buffer. EtherCAT wordt gebruikt om communicatie tussen de controller en de actuator mogelijk te maken in real-time. Vanwege de beperkte grootte van de buffer en de afzonderlijke toevoer van noten wordt de toevoersnelheid vertraagd tot de snelheid waarmee observaties verwerkt worden. Het stuursignaal kan niet constant zijn vanwege de variabele aard van de toevoersnelheid die onder andere veroorzaakt wordt door de inconsistente afstand tussen noten. Om het stuursignaal te corrigeren wordt een proportionele, integrerende en differentiërende regelaar geïmplementeerd die de verwerkingssnelheid volgt. Ondanks dat deze regelaar de toevoersnelheid de gewenste snelheid behaalt zal er vertraging optreden in dit proces. Dit resulteert potentieel in een overvolle buffer of systeemvertragingen. Deze effecten ontstaan door onvoldoende feedback over de bufferstatus. Om deze effecten te elimineren wordt de regelaar bijgesteld zodat ook de marge van de buffer gemaximaliseerd wordt. Dit resulteert in een regelaar die een balans tussen de gewenste snelheid en de gewenste buffermarge vindt. Ruis blijft invloedrijk in de regelaar. Doordat het foutsignaal uit verschillende signalen samengesteld is, zijn kostoptimalisatiealgoritmes een alternatief voor de toegepaste regelaar.



# Abstract in English

Qcify develops machines to assure the quality of nuts. Among these machines is a batch control system of which the actuator is being revised. The actuator is responsible for feeding nuts separately to an observer which stores observations in a buffer. EtherCAT is used to achieve communication between the controller and the actuator with real-time latency. Due to the limited size of the buffer and the separation of nuts, the feeding speed is reduced to match the processing speed. It is however not possible to have a constant input signal for the actuator. This is caused by the inconsistent feeding speed. To correct the input signal, a three term regulator is implemented which matches the feeding speed to the processing speed of the system. Despite the regulator reaching the desired processing speed, there is latency between the wanted and current state. This latency causes the system to stall or overflow the buffer as there is no accounting for the buffer state. In order to eliminate these effects, the regulator is changed to maximize the margin for error by using the current buffer state in addition to the speed. This change results in the regulator balancing between the buffer margin and the desired speed. However, noise remains impactful on the regulator. Further research on noise reduction could optimize the regulator further. The regulator uses a combination of different error signals. Cost optimization algorithms provide an alternative approach to the regulator.



# Hoofdstuk 1

## Inleiding

Binnen de voedingsindustrie zijn kwaliteitsnormen vastgelegd die behaald moeten worden vooraleer een product op de markt gebracht mag worden. Qcify ontwikkelt machines om producten te analyseren en slechte producten te verwijderen specifiek voor verschillende soorten noten.

Binnen Qcify wordt een nieuwe versie van een batchcontrolesysteem (QIS) ontwikkeld. Deze machine is bestemd voor het uitvoeren van steekproeven en het leveren van gedetailleerde info over de objecten. De snelheid van het systeem wordt gelimiteerd door de dataverwerking. Geobserveerde objecten moeten bij gevolg opgeslagen worden om dataverlies te vermijden. Een van de wijzigingen van de nieuwe machine is dat observaties in het geheugen bijgehouden worden in plaats van dit in niet vluchtig geheugen zoals een harde schijf op te slaan. Deze aanpassing zorgt ervoor dat onverwerkte data sneller toegankelijk is voor het systeem. Het geheugen is echter beperkt in grootte, bij een overvloed van data leidt dit tot dataverlies en bijgevolg een foutieve analyse. Het is eveneens ongewenst om geen onverwerkte data te hebben. In deze situatie zouden er geen observaties zijn om te verwerken en zal de verwerking niet de limiterende component zijn voor het systeem. Idealiter is de verwerkingssnelheid gelijk aan de observatiesnelheid zodat het systeem altijd beelden te verwerken heeft zonder dataverlies te veroorzaken.

Deze thesis onderzoekt hoe de objecttoevoer van het systeem aangestuurd kan worden om deze problemen te elimineren. Voor het aansturen van de componenten wordt voor etherCAT geselecteerd. Dit protocol bevat real-time communicatiecapaciteiten, bovendien is het protocol open-source. Om de gewenste snelheid te bereiken wordt een proportionele, integrerende en differentiërende regelaar ontwikkeld om de toevoer snelheid zodanig te wijzigen dat deze overeenstemt met de verwerkingssnelheid. De toevoersnelheid is echter onderhevig aan ruis die verschillende oorzaken heeft waaronder de variatie tussen objecten en de onderlinge afstand. Dit geeft aanleiding tot fouten op de regelaar wat bijgevolg de problemen van de buffer niet uitsluit. Om deze fouten te elimineren wordt de buffer als een bijkomend foutsignaal gebruikt. Dit veroorzaakt een regelaar die de gewenste snelheid volgt als ook een marge voorziet in de buffer om potentiële problemen te voorkomen.





# Hoofdstuk 2

## Datatransmissie en aansturing van de actuator

### 2.1 De proefopstelling

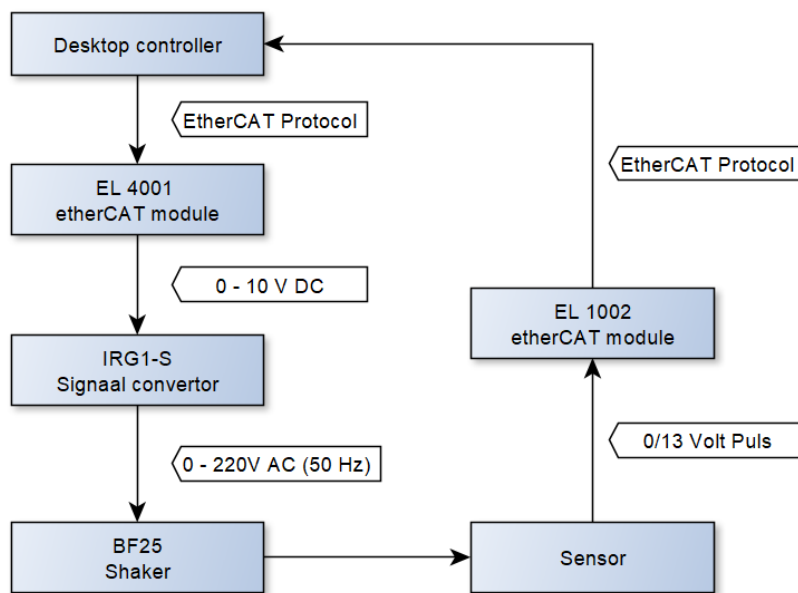
Het batchcontrolesysteem bestaat uit vier delen. Als eerste is er de toevoer, dit wordt door een observator gevolgd die op zijn beurt de verwerking voedt. Het geheel wordt aangestuurd vanuit de controller, dit is een ingebouwde desktop. Deze thesis onderzoekt de eerste component van dit systeem als ook hoe dit aangestuurd wordt, namelijk de toevoer. In de toevoer wordt een BF-25 bowlfeeder gebruikt als actuator gecombineerd met een IRG1-S signaalomvormer. Bijlage A en B bevatten respectievelijk de technische gegevens van deze componenten [1, 2]. De systeemtoevoer bevat ook een lasersensor die objecten detecteert die de observator ingaan. Deze sensor volgt niet direct op de bowlfeeder maar staat gepositioneerd op het einde van een baan die vallende noten begeleidt vanaf de bowlfeeder tot de observator. Figuur 2.1 toont deze begeleider. De gebruikte componenten in de toevoer kunnen niet rechtstreeks op de controller aangesloten worden. Om communicatie tussen de desktop en de componenten te voorzien wordt een etherCAT netwerk opgebouwd. Hierin communiceren de componenten met etherCAT modules die op hun beurt aangestuurd worden door de desktop. Het systeem maakt gebruik van een EL4001 module voor de bediening van de bowlfeeder. De sensor wordt aan een EL1002 module gekoppeld. Figuur 2.2 toont een schematische voorstelling van de opstelling. De modules en de controller zijn in een lijn topologie verbonden. De modules worden geproduceerd door Beckhoff. De technische gegevens van deze modules kunnen in bijlage C teruggevonden worden [3].

### 2.2 EtherCAT

”EtherCAT of Ethernet for control automation technology is een protocol dat de kracht en flexibiliteit van ethernet naar automatisatie, controle en data acquisitie systemen brengt [4]”. EtherCAT maakt real-time communicatie mogelijk te maken tussen een controller en randapparatuur zoals de actuator. Het protocol biedt ook een antwoord op verschillende problemen die typerend zijn aan protocollen gebouwd op ethernet. Dit zijn: het gebruik van de bandbreedte, stack delays en switch delays. EtherCAT is een open standaard met als doel dat verschillende bedrijven modules ontwikkelen en om het gebruik van etherCAT compatibele apparatuur bijgevolg goedkoper aan te bieden.



Figuur 2.1: baan voor vallende objecten



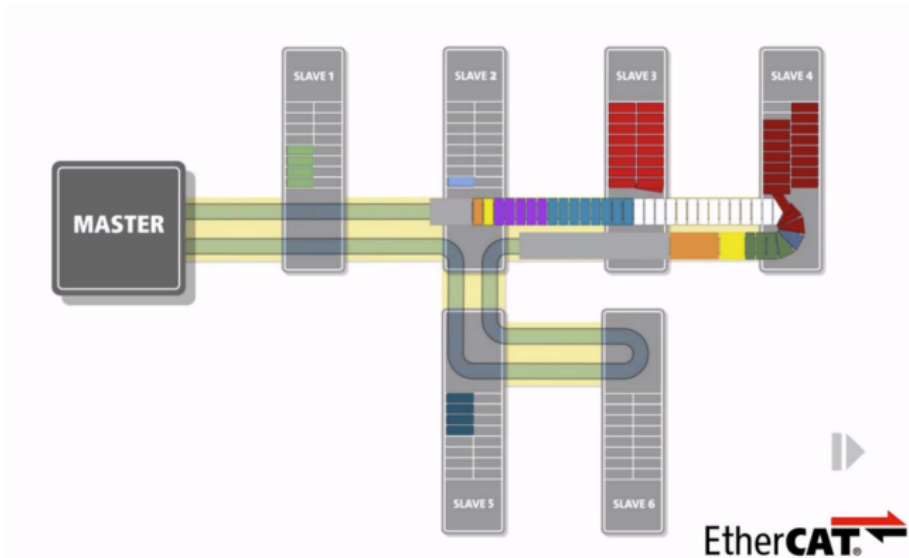
Figuur 2.2: Schematische voorstelling van de actuator sturing

### 2.2.1 Opbouw van een etherCAT netwerk

Een etherCAT netwerk staat een flexibele topologie toe. Zoals een lijn, ring, boom of ster-topologie, ook een combinatie van de verschillende topologieën is mogelijk. EtherCAT herleid het netwerk tijdens de configuratiefase tot een ringtopologie door gebruik te maken van de bi-directionele capaciteiten van conventionele ethernetkabels. Dit zorgt ervoor dat pakketten die verstuurd worden door het netwerk altijd hetzelfde pad volgen. Doordat dit pad vastligt zijn er geen switches meer nodig in het werk en bijgevolg is ook geen switch delay meer aanwezig in het netwerk. Figuur 2.4 toont het pad van een flexibele topologie na de configuratie van het netwerk.

Het configureren van een etherCAT netwerk gebeurt bij het opstarten van de controller, optioneel kan de configuratie geverifieerd en gewijzigd worden op ieder moment na de opstartfase. Deze optie maakt het mogelijk om het netwerk te herconfigureren indien een wijziging gebeurt en op deze manier alle componenten die bereikbaar zijn op te nemen in de configuratie. Dit bevordert de redundantie van het netwerk aangezien componenten aanspreekbaar blijven zolang er minstens één pad naartoe is.

Figuur 2.3: EtherCAT group logo [5]



Figuur 2.4: EtherCAT on the fly [6]

## 2.2.2 Communicatie via etherCAT

Communicatie in een etherCAT netwerk maakt gebruik van het etherCAT protocol. Dit gebruikt het standaard ethernet frame, gespecificeerd in de IEEE 802.3 standaard. Het ethertype van etherCAT bevat de waarde 0x88A4 [5]. Een enkel pakket kan tot 15 datagrammen bevatten die voor verschillende modules die zich in het netwerk bevinden, bestemd zijn. Ieder datagram bevat buiten data die uitgewisseld wordt ook een working counter. Dit bevat informatie over de uitvoering van de communicatie, bv. of dit succesvol verlopen is of niet. Deze informatie kan gebruikt worden om fouten in de communicatie te herstellen.

Tabel 2.1: Working counter increments [7, Tab. 3.2]

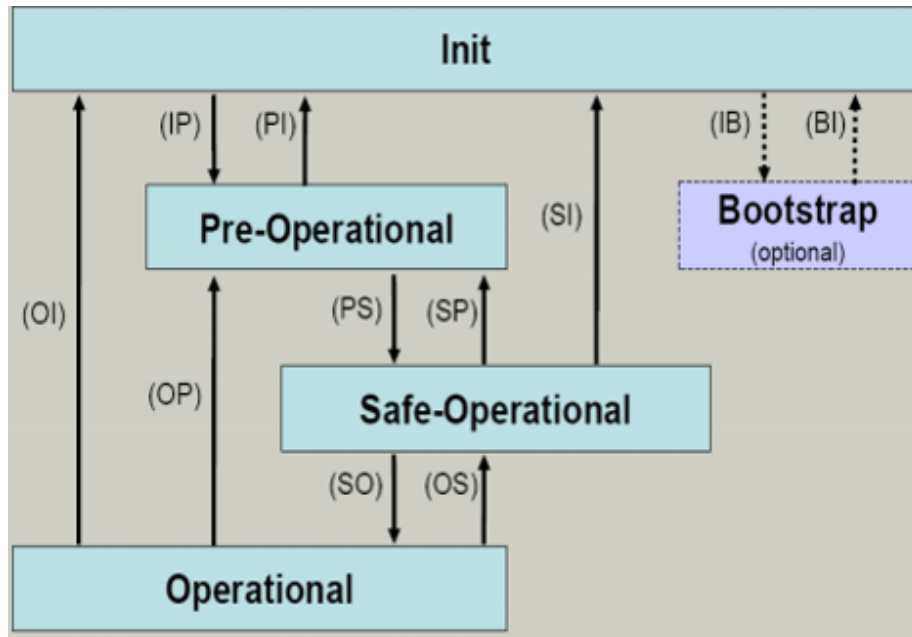
Command	Result	Increment
Read command	No succes	0
	successful	1
Write command	No succes	0
	successful	1
Read/Write command	No succes	0
	successful Read	1
	successful Write	2
	successful Read/Write	3

Een etherCAT pakket wordt altijd bij de controller (ook wel de master) gegenereerd. Modules in het netwerk zijn enkel in staat om een pakket te wijzigen en/of door te sturen. Het gegenereerde pakket wordt vervolgens in de volgorde die tijdens de configuratie van het netwerk vastgelegd is gestuurd. In tegenstelling tot conventionele netwerk switches sturen de modules het pakket direct verder naar de volgende module en passen het pakket indien nodig aan tijdens dit proces. Deze

wijzigingen zijn afhankelijk van de operatie die dient te gebeuren bv. Bij een lees commando zal de module in kwestie procesdata in het pakket toevoegen.

### 2.2.3 EtherCAT state machine

EtherCAT modules gedragen zich als state machines. De staat bepaalt welke communicatie mogelijk is met de module. Tijdens het wijzigen van de staat worden ook interne controles uitgevoerd om een correcte functionaliteit te garanderen. Figuur 2.5 toont een alle staten die een module kan aannemen als ook de paden die genomen kunnen worden om van staat te wisselen.



Figuur 2.5: EtherCAT state machine (esm) [8]

Ten eerste is er de “init” staat, dit de standaard staat waar een module zich in verkeert indien hij opgestart wordt. Deze staat is bestemd voor de configuratie van het netwerk. In deze staat kan de module geen procesdata uitwisselen via het etherCAT netwerk. De init staat wordt meestal opgevolgd door de pre-operational status. Het is echter mogelijk dat de init staat gevolgd wordt door de bootstrap staat, die gebruikt wordt voor updates. Het dient opgemerkt te worden dat de bootstrap staat niet altijd aanwezig is in een etherCAT module.

De “Pre-Operational” staat wordt gebruikt om interne componenten van de module te configureren zoals de Fieldbus memory management unit (FMMU) en sync managers. Deze configuratie stap is noodzakelijk om een correcte transmissie en translatie van data. Modules bevatten ook standaardinstellingen zodat tijdens het configureren enkel de instellingen die niet hiertoe behoren aangepast moeten worden. Bij het overgaan naar een andere staat worden deze instellingen gecontroleerd om te garanderen dat er geen configuratiefouten aanwezig zijn in de module.

Vervolgens komt de module in de “Safe-Operational”(SafeOP) staat terecht. Indien de module zich in deze staat verkeert, dan zal de module geen uitgangspoorten aansturen. Ingangspoorten kunnen wel uitgelezen worden. In deze staat kan de module ook procesdata uitwisselen met de master. Dit geeft de module bijkomend tijd om te synchroniseren met andere componenten binnen het netwerk.

Ten slotte is er de “Operational” staat. Deze staat wordt bereikt nadat er geldige data verzonden

is terwijl de module in zich in de safeOP staat verkeerde. Dit laat communicatie van procesdata toe en zal dit ook direct synchroniseren met de in -en output van de module. Dit is de meest aangewezen staat voor het aansturen van componenten [7–9].

## 2.3 Simple open etherCAT master (SOEM)

Deel 2.1 bespreekt de connectie tussen etherCAT modules en de controller. De pakketten die over het netwerk gestuurd worden moeten echter gegenereerd worden. Dit kan onder andere door het gebruik van een library die geïntegreerd wordt met de rest van de software die het systeem bedient. Er bestaan verschillende opties die dit mogelijk maken. De meest bekende opties zijn TwinCAT3, SOEM en IgH Master. Binnen dit onderzoek wordt geopteerd voor SOEM als library omdat dit op meerdere besturingssystemen beschikbaar is (Linux en Windows), Toegang geeft tot alle niveau's van het etherCAT protocol en tegelijkertijd een volledige library is met optionele functies zoals Ethernet over EtherCAT of CANopen over etherCAT [10, 11].

Andere conventionele libraries zoals TwinCAT3 en IgH master bevatten gelijkaardige functionaliteiten, maar zijn beperkt tot een enkel besturingssysteem en hebben een minder gunstige licentie respectievelijk. In het geval van SOEM bestaan er libraries die gebaseerd zijn op SOEM zoals FastCAT en SOEM interface. Deze libraries zijn een minder goede keuze omdat ze bestemd zijn naar een aantal specifieke modules die niet gebruikt worden binnen dit onderzoek en/of functies bevatten waar geen functionaliteit achter zit wat onverwachte artefacten veroorzaakt. Het dient opgemerkt te worden dat er een rust wrapper bestaat voor IgH master die momenteel nog geen volledige functionaliteit ondersteunt. Naar de toekomst toe kan dit echter een betere keuze worden dan SOEM [12–15].

### 2.3.1 Netwerk configuratie en communicatie met SOEM

```
1 void find_adapter()
2 {
3     ec_adaptert * adapters = ec_find_adapters();
4     while (adapters != nullptr){
5         if (configModules(adapters->name)){
6             //connection was made to the modules
7             break;
8         }else {
9             adapters = adapters->next;
10        }
11    }
12    ec_free_adapters(adapters);
13 }
```

Figuur 2.6: Find adapter code snippet

Vooraleer het netwerk geconfigureerd kan worden, moet bekend zijn op welke netwerkinterface het etherCAT netwerk zich bevindt. Indien dit niet op voorhand geweten is biedt SOEM de mogelijkheid om alle beschikbare poorten op te lijsten. Deze lijst wordt vervolgens gebruikt om het etherCAT netwerk te vinden door op iedere poort te zoeken achter een netwerk. Een code voorbeeld hiervan wordt getoond door figuur 2.6. Het dient opgemerkt te worden dat het `ec_free_adapters` commando memory leaks voorkomt.

```

1   boolean setModuleState(const ec_state &mode)
2   {
3       int expectedwkc = 1;
4       ec_slave[slaveIndex].state = mode;
5       int wkc = ec_writestate(slaveIndex);
6       if (wkc != expectedwkc){
7           //workcount incorrect, potential error
8       }
9       return ( ec_statecheck(slaveIndex,mode,EC_TIMEOUTSTATE *3) == mode);
10  }

```

Figuur 2.7: Set modules state code snippet

```

1   bool configModules(const char *ifname){
2   char IOmap[4096];
3   if (ec_init(ifname)){
4       if (ec_config(FALSE,&IOmap)>0){
5           //find and auto-config slaves
6           ec_configdc();
7           if (ec_slavecount <= 0){
8               return false;
9           }
10          int expectedWorkingCounter = (ec_group[0].outputsWKC * 2) + ec_group
              [0].inputsWKC;
11
12          ec_statecheck(0, EC_STATE_SAFE_OP, EC_TIMEOUTSTATE * 3);
13          if (ec_slave[0].state != EC_STATE_SAFE_OP ){
14              ec_readstate(); //can be used to check the state of all modules
15          }
16          return true;
17      }
18  }
19  return false;
20 }

```

Figuur 2.8: Config modules code snippet [16]

Na het configureren van het netwerk is via SOEM een lijst van de netwerk modules beschikbaar die geordend zijn in volgorde waarin het pakket door het netwerk zal propageren. De weergegeven modules komen overeen met de modules in het netwerk maar ze worden niet automatisch gesynchroniseerd met elkaar. SOEM voorziet lees en schrijf functies die wijzigingen van in -en outputs opneemt in een pakket en dit vervolgens verstuurt. Figuur 2.12 toont een gedeeltelijk overzicht van de beschikbare data over een module [17].

## 2.3.2 Privileges van SOEM

Zoals aangehaald in deel 2.3.1 worden pakketten gegenereerd door middel van SOEM. Dit wordt bij het gebruik van standaardinstellingen geblokkeerd door het besturingssysteem om de veiligheid van het systeem te garanderen. Om toch etherCAT pakketten te genereren moet SOEM raw socket privileges verkrijgen. Dit geeft toegang tot alle trafiek die door de netwerk interface verstuurd en ontvangen wordt als ook volledige vrijheid over opbouw van pakketten die hierover verstuurd worden [18].

Deze privilege kan op verschillende manieren verkregen worden. De eerste methode is door het

programma als superuser uit te voeren. Dit zorgt ervoor dat er geen restricties meer zijn voor de applicatie, dit is ook de minst veilige optie omdat aanvallen die in staat zijn om commando's uit te voeren op het besturingssysteem (operating system command injection) rechtstreeks dezelfde privileges hebben [19].

Een alternatief op het gebruik van superuser rechten is om specifieke privileges toe te kennen na de compilatie van het programma. Dit heeft het nadeel dat bij iedere update deze privileges verwijderd worden en bijgevolg opnieuw toegekend moeten worden. Dit proces kan geautomatiseerd worden door middel van een script. Figuur 2.9 toont een script om deze privileges toe te kennen. Deze methode zorgt ervoor dat pakketten gegenereerd kunnen worden met minimale privileges, wat ook de veiligheid bevordert [20, 21].

```
1 #Build
2 /snap/clion/235/bin/cmake/linux/x64/bin/cmake --build /home/qcify/Desktop/SOEM
   /SOEM/cmake-build-debug --target slaveinfo -j 10
3 #Set privileges
4 sudo setcap cap_net_raw=eip /home/qcify/Desktop/SOEM/SOEM/cmake-build-debug/
   test/linux/slaveinfo/slaveinfo
5 #Run
6 /home/qcify/Desktop/SOEM/SOEM/cmake-build-debug/test/linux/slaveinfo/slaveinfo
```

Figuur 2.9: Build and run script

### 2.3.3 Toevoegen van SOEM in de Qcify codebase

SOEM maakt deel uit van de controller. Deze library moet gelinkt worden met de rest van het programma dat op de machine uitgevoerd wordt. Dit vereist een aantal stappen die in dit deel beschreven worden.

De eerste stap is het toevoegen van de afhankelijkheid van de library. Dit gebeurt in het bestand genaamd "vcpkg.json". Binnen het deel dependencies moet SOEM toegevoegd worden. De versie wordt gespecificeerd in het deel overrides. Figuur 2.10 toont hoe dit eruit ziet. Het dient opgemerkt te worden dat afhankelijkheden van andere libraries verwijderd zijn en dat bepaalde irrelevante variabelen gewijzigd zijn.

```
1 {
2   "name": "machinecode",
3   "version": "1.0.0",
4   "dependencies": [
5     "soem"
6   ],
7   "builtin-baseline": "XXXXXXXXXXXXXXXX",
8   "overrides": [
9     { "name": "soem", "version": "1.4.0#4" }
10  ]
11 }
```

Figuur 2.10: voorbeeld vcpkg.json voor het gebruik van soem

Vervolgens kan SOEM toegevoegd worden in CMake. Dit maakt het mogelijk om het project te bouwen en te testen [22]. Het toevoegen van SOEM gebeurt in een CMakeLists.txt bestand. Dit gebeurt door middel van de eerste twee regels van figuur 2.11. Ook de verschillende bestanden



die onderdeel uitmaken van het project worden toegevoegd. Enkel de bestanden die betrekking hebben op het onderzoek zijn opgenomen in deze figuur. Indien deze aanpassingen gedaan zijn kan SOEM gebruikt worden door de regel ”`#include < soem/ethercat.h >`” toe te voegen in de bestanden die de library aanspreken.

```
1 find_path(LIBSOEM_INCLUDE_DIR NAMES soem/ethercat.h)
2 find_library(LIBSOEM_LIBRARY NAMES soem REQUIRED)
3
4
5 add_library(soem_ethercat OBJECT
6     src/actuator/soemEthercat_connection.cpp
7     src/actuator/soemethercat_slave_modules/soemEthercat_EL4001_slave.cpp
8     src/actuator/soemethercat_slave_modules/soemEtherCAT_EL1014_slave.cpp
9     src/actuator/soemethercat_slave_modules/soemEthercat_EL1002_slave.cpp
10    src/actuator/soemEthercat_slave.cpp
11    src/actuator/shaker_test.cpp
12    src/actuator/shaker_speed_regulator.cpp
13    )
14
15
16 target_include_directories(soem_ethercat PUBLIC ${LIBSOEM_INCLUDE_DIR})
17 target_link_libraries(soem_ethercat PRIVATE comm_lib ethercat ${
    LIBSOEM_LIBRARY})
```

Figuur 2.11: CMake list

## 2.4 Data transmissie met SOEM

Om de output van de EL4001 te wijzigen dient de output variabelen van deze module gewijzigd te worden terwijl de module zich in de operational staat bevindt. Deze combinatie zorgt bij een schrijf commando dat deze wijziging ook toegepast wordt. Variabelen worden in SOEM als een pointer bijgehouden. De pointer verwijst naar een reeks bits die gezamenlijk de waarde vormen van de in -of output. De EL4001 module bevat een 16 bit output die zowel een positieve als negatieve waarde kan voorstellen. Deze module heeft een maximale uitgangsspanning van 10 volt. Door gebruik te maken van de 15 minst significante bits kan een positieve spanning voorgesteld worden en dus 32767 verschillende instelbare spanningen. Dit resulteert in een resolutie van 0,3 mV per verdeling.

Het input signaal van de EL1002 module is beschikbaar in de input variabelen van deze module en geeft slechts een enkele bit als input per kanaal. Deze bit geeft weer of een object gedetecteerd wordt of niet. Bijlage C bevat de technische data van de vernoemde etherCAT modules.

Een alternatief voor de EL1002 module is de EL1502 module. Deze module bevat een teller zodat het aantal pulsen die gebeuren bijgehouden worden in de module. Dit kan dataverlies tegengaan indien de module niet snel genoeg uitgelezen kan worden om impulsen te detecteren.

## Output of slaveinfo.c

ec\_init on enp60s0 succeeded.  
5 slaves found and configured.  
Calculated workcounter 6

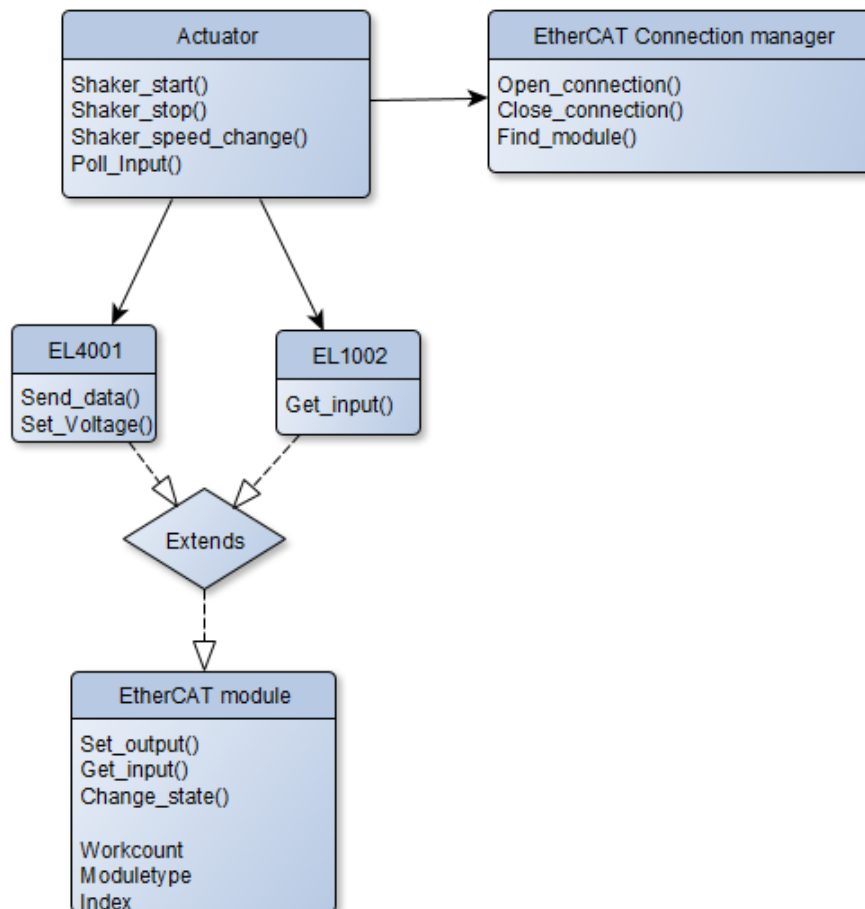
Slave:1

Name:EK1100  
Output size: 0bits  
Input size: 0bits  
State: 4  
Delay: 0[ns]  
Has DC: 1  
DCParentport:0  
Activeports:1.1.0.0  
Configured address: 1001  
Man: 00000002 ID: 044c2c52 Rev: 00120000  
FMMUfunc 0:0 1:0 2:0 3:0  
MBX length wr: 0 rd: 0 MBX protocols : 00  
CoE details: 00 FoE details: 00 EoE details: 00 SoE details: 00  
Ebus current: -2000[mA]  
only LRD/LWR:0

Slave:2

Name:EL1002

Figuur 2.12: Module info log.



Figuur 2.13: Controller schema voor een constante sturing.

## 2.5 Data transmissie tussen de controller en randapparatuur

Om de data in de controller en de modules consistent up-to-date te houden wordt voor input modules geopteerd voor polling. Dit zorgt ervoor dat de staat van deze modules regelmatig gelezen en geüpdatet worden in de controller. Dit is noodzakelijk omdat de controller die het systeem controleert geen data van de module zal verkrijgen indien dit niet opgevraagd wordt. De staat van output modules daarentegen is wel bekend voor de controller en vereist bijgevolg geen polling. Communicatie met de output modules is enkel noodzakelijk wanneer er een wijziging optreedt in de controller.

Om de controller ook controle te geven over de etherCAT modules en dus ook de randapparatuur worden deze geïmplementeerd als klassen die de functionaliteit van de gebruikte modules bepalen. Figuur 2.13 toont een overzicht van deze klassen. Bij de implementatie dienen twee zaken opgemerkt te worden.

1) de klasse 'EtherCAT module' voorziet algemene functionaliteit voor alle geïmplementeerde modules. Bij verschillende typen modulen kunnen dezelfde functionaliteiten bestaan. Er kunnen geen instanties van deze klasse bestaan, enkel van de specifieke modules die deze klasse uitbreiden. Deze aanpak voorkomt dat functies zoals het aanpassen van de module staat meermaals geïmplementeerd moeten worden.

2) `Shaker_start()` begint het pollen van de shaker, dit gebeurt niet op dezelfde thread als de thread die dit proces start. Het starten en stoppen van de actuator gebeurt door andere componenten binnen het programma. Indien het pollen niet op een aparte thread zou gebeuren zou de thread van een andere component die de shaker start gebruikt worden om de input te pollen. Volgens [23] kan een thread aangemaakt worden en bijgehouden worden in de klasse en op een later moment ook weer samengevoegd worden in een andere functie van dezelfde instantie. In deze situatie kan van een 'multiple instruction multiple data' (MIMD) systeem gesproken worden [24].

Een van de objectieven is om de snelheid van de toevoer aan te passen aan de verwerkingssnelheid. Om dit te verwezenlijken moet de snelheid van de toevoer gedefinieerd worden. Via de EL1002 module worden pulsen verkregen op de momenten dat een object gedetecteerd wordt. Door een tijdmeting uit te voeren kan het interval tussen twee pulsen bepaald worden. Hieruit kan het aantal noten die geobserveerd worden per seconden bepaald worden. Formule 2.1 toont hoe deze frequentie bepaald kan worden uit de tijdmetingen.

$$f = \frac{1}{t_n - t_{n-1}} \quad (2.1)$$

waarbij:

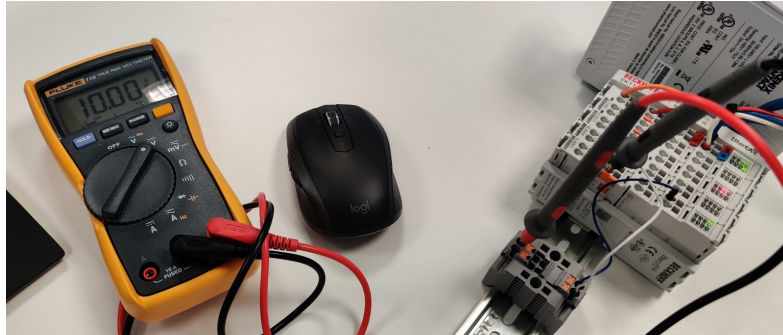
$f$  = het aantal objecten toegevoerd per seconde [1/s]

$t_n$  = tijdstip van puls n [s]

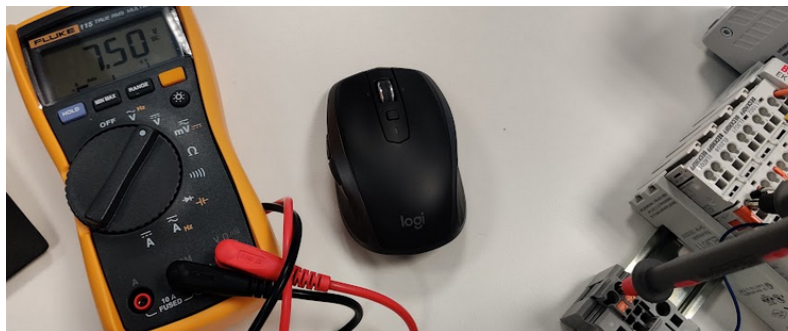
$t_{n-1}$  = tijdstip van de voorgaande puls [s]

## 2.6 Resultaten bij een constante stuurspanning

Figuren 2.14 en 2.15 tonen de gemeten output op de EL4001 module bij verschillende ingestelde spanningen. De gemeten spanning komt overeen met de spanning die ingesteld wordt via de controller met een afwijking van 0,01 volt. Deze afwijking komt overeen met de foutmarge van de gebruikte voltmeter. Dit bevestigt dat de EL4001 module functioneel is en de mogelijkheid biedt om de actuator te bedienen.



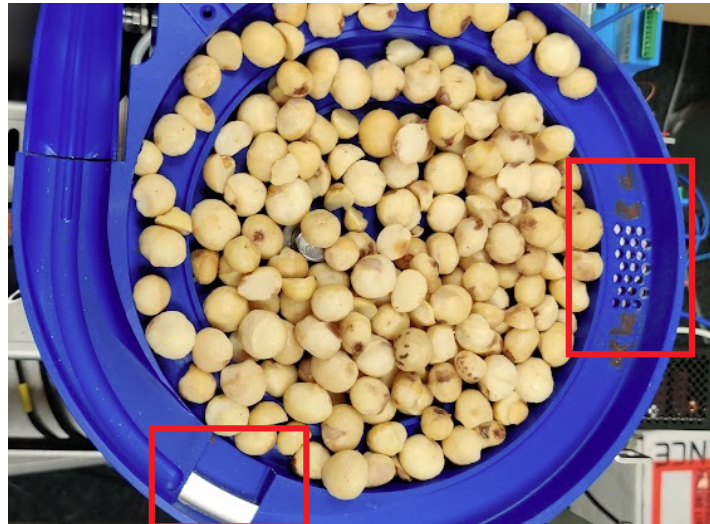
Figuur 2.14: EL4001 sturing van 10 volt



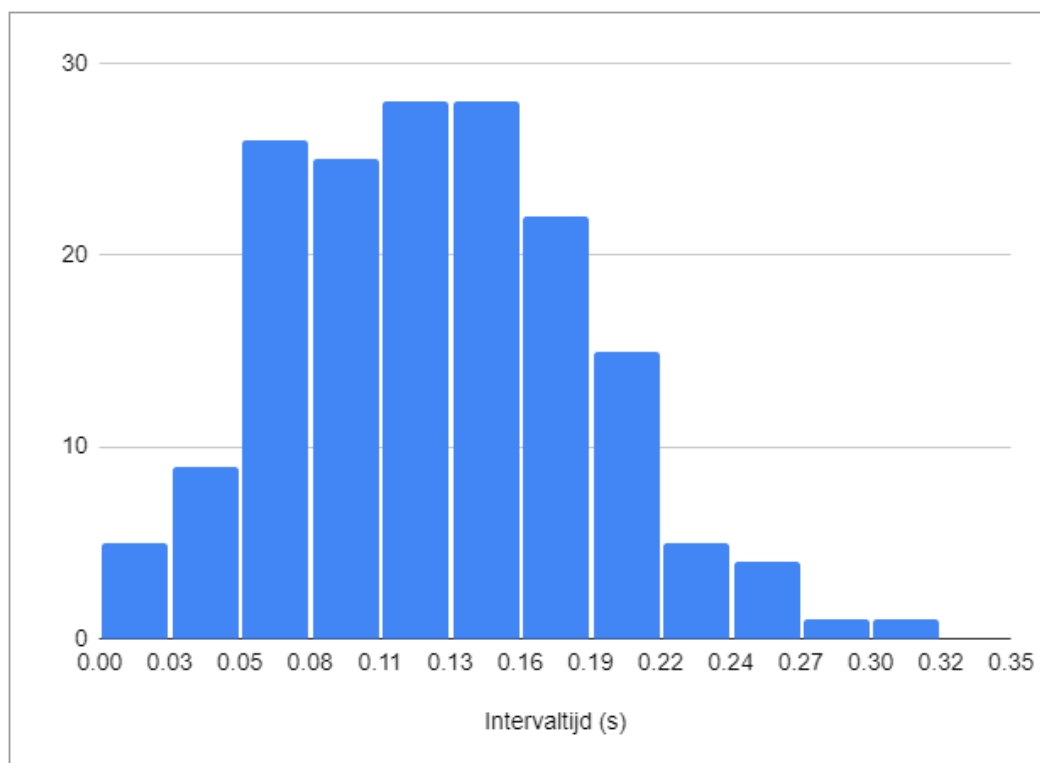
Figuur 2.15: EL4001 sturing van 7,5 volt

Figuur 2.17 toont een histogram van de interval metingen bij een constante sturing van 10 volt voor amandelen. Het dient opgemerkt te worden dat de metingen een grote variatie bevatten. Dit wordt veroorzaakt door verschillen tussen objecten en de afstand ertussen. Ook de meting leidt tot een afmeting omdat er voor polling geopteerd is. Bijkomend zijn er ook oneffenheden in de actuator die ook aanleiding kunnen geven tot verschil in afstand tussen noten. Twee van deze oneffenheden worden getoond door de figuur 2.16. De meting bevat een bijkomende fout, aangezien er gebruik wordt gemaakt van polling. Dit zorgt ervoor dat het verkrijgen van een puls niet consistent is.

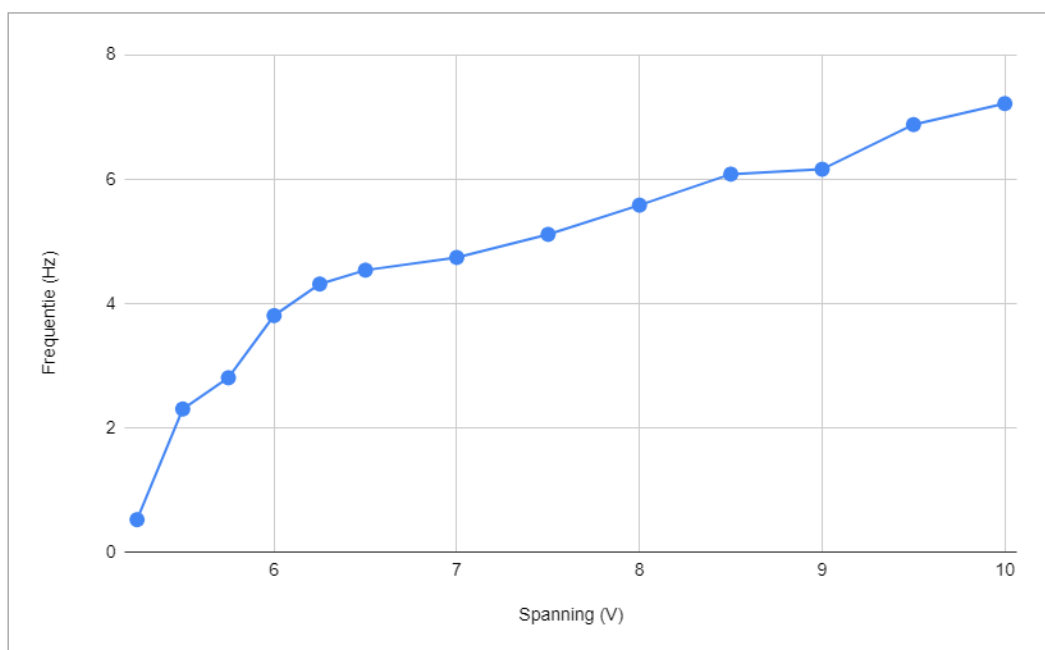
Ten slotte toont figuur 2.18 de gemiddelde frequentie bij verschillende stuurspanningen voor amandelen. Bij spanningen van 6 volt en lager vloeien noten niet meer door het systeem. Bij spanningen hoger dan 6 volt is er een lineair verband tussen de frequentie en de spanning.



Figuur 2.16: Shaker gevuld met macadamia noten



Figuur 2.17: Histogram van interval van amandelen bij 10v sturing



Figuur 2.18: Frequentie in functie van de spanning voor amandelen



# Hoofdstuk 3

## Signaal regeling op basis van snelheid

Hoofdstuk 2 bespreekt hoe communicatie tussen de controller en de randapparatuur fungeert. Dit biedt geen antwoord op hoe het stuursignaal moet veranderen om de verwerkingssnelheid te volgen. Dit hoofdstuk gaat in op hoe het stuursignaal geregeld wordt om de toevoer snelheid naar de verwerkingssnelheid te doen streven.

### 3.1 Bepalen van de snelheid

Vooraleer de signaalregeling besproken wordt, moet de snelheidsbepaling vastgelegd worden. Zoals reeds besproken in deel 2.6 is de snelheidsmeting van de toevoer onderhevig aan variatie. Door deze variatie op het signaal kan dit niet rechtstreeks gebruikt worden om een regelaar op te bouwen. Indien een gemiddelde genomen wordt van alle intervallen dan zal hier de snelheid bij een stuursignaal wel uit afgeleid kunnen worden. Om een indicatie te verkrijgen van deze waarde wordt de trend van het signaal gezocht. Dit gebeurt door middel van een techniek genaamd "moving average". Deel 3.1.1 gaat dieper in op deze techniek. Om de snelheid van de toevoer te bepalen wordt geopteerd voor exponentiële moving average omdat de geïnduceerde vertraging klein is ten opzichte van lineaire moving average terwijl het doel van de trend te achterhalen mogelijk blijft.

Een tweede wijziging van de snelheidsbepaling is door niet het interval tussen twee objecten te bepalen maar in plaats daarvan te bepalen hoeveel noten in de laatste seconde voorbij zijn gekomen.

Deze bepaling gebeurt door op het moment dat een noot gedetecteerd wordt het tijdstip op te slaan in een queue. Een queue data structuur kan namelijk als een linked list geïmplementeerd worden om een tijdscomplexiteit van  $O(1)$  bij het toevoegen en verwijderen van tijdstippen te verkrijgen [25]. Een check wordt uitgevoerd om te bepalen of meer dan één seconde verstreken is indien de snelheid opgevraagd wordt. Indien dit het geval is wordt het eerste object verwijderd en wordt dezelfde check uitgevoerd op het volgende object. Indien er geen object meer verwijderd wordt dan geeft de lengte van de queue aan hoeveel objecten in de laatste seconde geobserveerd zijn.

De snelheid wordt bepaald door een combinatie van beide technieken door een moving average te bepalen van het aantal objecten in de queue. Dit resulteert in de trend van het aantal noten



geobserveerd in de laatste seconde. De queue wordt gebruikt om kleine fluctuaties tussen noten op te vangen, terwijl de moving average fluctuaties van de queue opvangt.

### 3.1.1 Moving average

Moving average is een techniek die onder andere gebruikt wordt als indicator voor investeerders [26]. Het doel van een moving average is het verwijderen van korte termijn fluctuaties. Het resultaat hiervan wordt gebruikt om de trend van een signaal te bepalen. Een nadeel van de techniek is dat deze altijd achter loopt op de huidige situatie. Dit effect ontstaat doordat het gemiddelde wordt genomen van de meest recente waarden. Hoe meer waarden meegenomen worden in de berekening hoe groter deze achterstand. In een poging om moving average responsiever te maken voor recente veranderingen bestaat de exponentiële moving average. Deze methode herverdeelt de aandelen zodat recentere waarden zwaarder doorwegen in het resultaat [27]. Formule 3.1 bepaalt de lineaire moving average. Formule 3.2 geeft aan hoe het exponentiële moving average bepaald wordt. Figuur 3.1 toont een lineaire moving average van 5 gemeten intervallen en een exponentiële moving average met  $s = 0.2$  en  $n = 3$  toegepast op een van de metingen.

$$MA = \frac{T_1 + T_2 + \dots + T_n}{n} \quad (3.1)$$

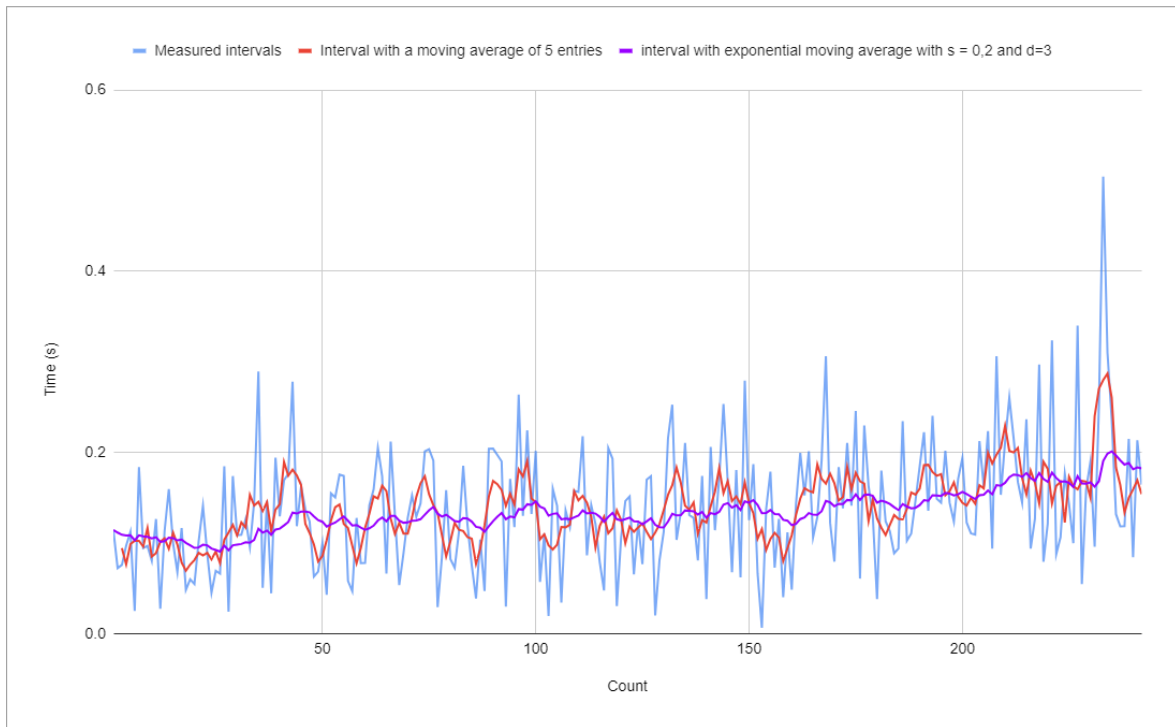
waarbij:

- $MA$  = Moving average in (s)
- $T$  = Interval tijd in (s)
- $n$  = Het aantal intervallen

$$MA := \left(T \times \frac{s}{1+n}\right) + MA \times \left(1 - \frac{s}{1+n}\right) \quad (3.2)$$

waarbij:

- $MA$  = Moving average in (s)
- $T$  = Interval tijd in (s)
- $n$  = Het aantal intervallen
- $s$  = een smoothing ratio tussen 0 en 1



Figuur 3.1: Voorbeeld van moving average (lineair: rood, exponentieel: paars) op interval meting bij statisch signaal

## 3.2 De PID regelaar

De PID of drie term regelaar is een combinatie van de proportionele (P), integrerende (I) -en differentiërende (D) regelaar. Het combineren van deze regelaars zorgt ervoor dat zowel de voor als de nadelen van deze regelaars gecombineerd worden. Formule 3.3 beschrijft de transferfunctie van de parallelle PID regelaar. Het dient opgemerkt te worden dat er ook een seriële PID regelaar bestaat die analoog werkt aan de parallelle regelaar maar een andere samenstelling van de componenten bevat [28]. Er bestaan alternatieven voor de PID regelaars, voornamelijk regelaars bestemd voor kostenoptimalisatie omdat het foutsignaal evenzeer als een kostfunctie aanschouwd kan worden. Binnen dit onderzoek wordt hier niet voor geopteerd omdat de PID regelaar beter aansluit bij de snelheidsregeling en het lineaire verband tussen de spanning en snelheid.

$$TF_{PID_p} = Kr \left( 1 + \frac{1}{\tau_i p} + \tau_d p \right) \quad (3.3)$$

waarbij:

- $Kr$  = de propagatie constante
- $\tau_d$  = de differentiatie tijdsconstante [s]
- $\tau_i$  = de integratie tijdsconstante [s]
- $p$  = de variabele van Laplace
- $TF_{PID_p}$  = De transfertfunctie van de parallelle PID regelaar

### 3.2.1 De proportionele Regelaar (P-regelaar)

de P regelaar versterkt of verzwakt het foutsignaal met een constante factor. De P-regelaar kan het systeem sneller maken en de stand fout verkleinen. Ook kan het voor ruisonderdrukking zorgen. Bij een te grote versterking wordt het systeem instabiel [28, 29].

$$TF_P = Kr \quad (3.4)$$

$$u(t) = Kr \times e(t) \quad (3.5)$$

waarbij:

$TF_P$  = De transfertfunctie van de proportionele regelaar

$Kr$  = een constant reëel getal

$u(t)$  = de waarde op de uitgang in functie van de tijd (t)

$e(t)$  = de fout/ingang van de regelaar in functie van de tijd (t)

### 3.2.2 De integrerende regelaar (I-regelaar)

De integrerende regelaar of integrator integreert het signaal aan de ingang. Verder wordt het signaal gedeelt door de integratie tijdsconstante ( $\tau_i$ ). Door de tijdsconstante te verhogen zal de reactie sneller verlopen. Dit veroorzaakt een minder stabiel of instabiel systeem. De integrerende regelaar integreert de ingang en vermenigvuldigt het signaal vervolgens met  $\frac{1}{\tau_i}$  [28].

$$TF_{PI} = Kr(1 + \frac{1}{\tau_i p}) \quad (3.6)$$

$$u(t) = \frac{1}{\tau_i} \int_0^t e(t) dt \quad (3.7)$$

waarbij:

$TF_{PI}$  = De transfertfunctie van de proportionele integrerende regelaar.

$Kr$  = een constant reëel getal.

$\tau_i$  = de integratie tijdsconstante [s]

$u(t)$  = de waarde op de uitgang in functie van de tijd (t)

$e(t)$  = de fout/ingang van de regelaar in functie van de tijd (t)

De I-regelaar elimineert de statische fouten die optreden. Deze fout wordt ook wel de standfout genoemd. Dit gedrag wordt veroorzaakt doordat het signaal blijft integreren en dus de sturing blijft bijwerken tot er geen fout resteert. Meestal wordt in praktijk de I-regelaar in combinatie met een P-regelaar gebruikt. Dit combineert de voor en nadelen van beide regelaars [28, 29].

### 3.2.3 De differentiërende actie (D-actie)

De differentiërende actie of differentiator geeft als uitgangssignaal de afgeleide van hetingangssignaal. Hier wordt niet gesproken van een D-regelaar maar van een D-actie omdat de differentiator onderdeel uitmaakt van een groter geheel. Een alleenstaande differentiator komt in tegenstelling tot de P-regelaar of I-regelaar in uitzonderlijke gevallen voor in de praktijk. De D-actie heeft een stabiliserend effect op de regelkring [28, 29].

$$TF_D = \tau_d p \quad (3.8)$$

$$u_t = \tau_d \frac{d}{dt} e(t) \quad (3.9)$$

waarbij:

$TF_D$  = De transferfunctie van de D-actie

$Kr$  = een constant reëel getal

$\tau_d$  = de differentiatie tijdsconstante [s]

$u(t)$  = de waarde op de uitgang in functie van de tijd (t)

$e(t)$  = de fout/ingang van de regelaar in functie van de tijd (t)

### 3.3 Implementatie van de PID regelaar

In het systeem wordt een PID regelaar geïmplementeerd. Deze is gelijkaardig aan de transferfunctie van formule 3.3. Er wordt geopteerd om een tweede constante te gebruiken. Het foutsignaal wordt gedefinieerd door het verschil tussen de gewenste en gemeten snelheid.

$$u(t) = K(Kr \times e(t) + \frac{1}{\tau_i} \int_0^t e(t) dt + \tau_d \frac{d}{dt} e(t)) \quad (3.10)$$

$$e(t) = f_{gewenst} - f_{gemeten} \quad (3.11)$$

waarbij:

$u(t)$  = het uitgangssignaal in V

$Kr$  = de propagatie constante

$K$  = een constante reëel getal

$\tau_d$  = de differentiatie tijdsconstante [s]

$\tau_i$  = de integratie tijdsconstante [s]

$e(t)$  = de fout/ingang van de regelaar in functie van de tijd (t) [1/s]

$f_{gewenst}$  = De snelheid waarmee beelden verwerkt worden [1/s]

$f_{gemeten}$  = De snelheid waarmee beelden worden toegevoerd [1/s]

De implementatie maakt geen rechtstreeks gebruik van de integraal en differentiaal besproken in deel 3.2. Dit is omdat het in praktijk niet haalbaar is om een oneindig klein tijdsinterval te bereiken. Om deze functionaliteiten toch te integreren in de regelaar wordt geopteerd voor een benadering van de integraal en de differentiaal. Voor de integraal wordt een riemannsom geïmplementeerd. Formule 3.12 beschrijft hoe deze som berekend wordt. De riemannsom ligt aan de basis van de integraal, dit kan bekomen worden door  $\Delta t$  oneindig klein te veronderstellen. Ook de differentiërende actie wordt vervangen, er wordt geopteerd voor het verschil in het foutsignaal over het tijdsverschil. Dit kan berekend worden door middel van formule 3.13. Analoog aan de riemannsom zal een oneindig kleine  $\Delta t$  tot de differentiaal leiden.

$$E_i = \sum_{n=0}^t e(t) \times \Delta t \quad (3.12)$$

$E_i$  = De oppervlakte onder de curve

$t$  = Het huidige tijdstip

$e(t)$  = de fout/ingang van de regelaar in functie van de tijd ( $t$ ) [1/s]

$\Delta t$  = het verschil in tijd waarmee de actuator geupdate wordt [t]

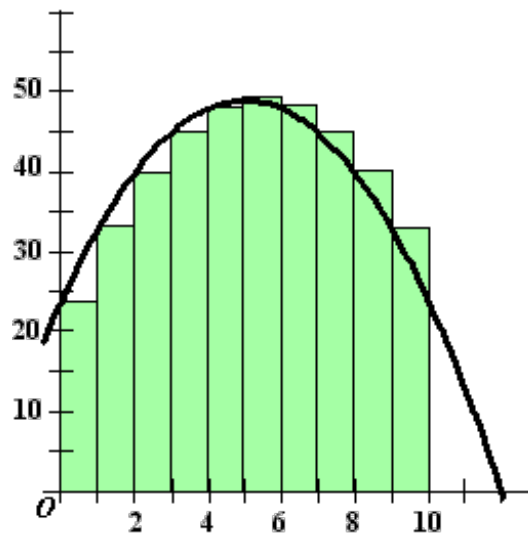
$$E_d = \frac{e(t_1) - e(t_0)}{t_1 - t_0} = \frac{\Delta e(t)}{\Delta t} \quad (3.13)$$

$E_d$  = Het foutverschil per tijdseenheid

$t_0$  = Het vorige tijdstip

$t_1$  = Het huidige tijdstip

$e(t)$  = de fout/ingang van de regelaar in functie van de tijd ( $t$ ) [1/s]



Figuur 3.2: Riemanns som [30]

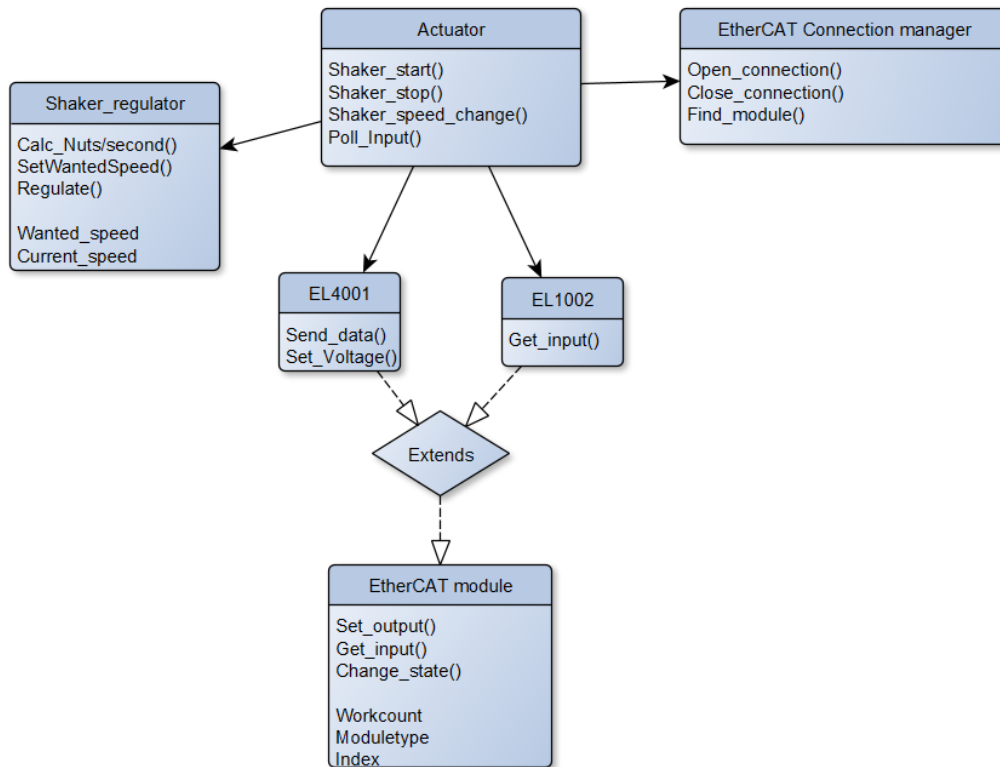
Het implementeren van een regelaar brengt aanpassingen in de controller software met zich mee. Figuur 3.3 toont het herwerkte klasse diagram met regelaar. De regelaar wordt afgezonderd van de actuator om het regelsysteem herbruikbaar te maken voor andere componenten van dezelfde of een andere machine. De bekomen uitkomst van de regelaar is niet het nieuwe stuursignaal. De uitkomst wordt gesommeerd met de huidige spanning gelimiteerd tot de capaciteiten van de etherCAT modules. De gewijzigde spanning wordt gebruikt als nieuw stuursignaal van de actuator.

In zowel formule 3.13 als 3.12 wordt van een tijdsverschil gesproken tussen updates. Ook werd vermeld dat dit niet oneindig klein kan zijn. Het tijdsverschil kan op verschillende manieren vastgelegd worden. Binnen dit onderzoek worden twee methoden onderzocht:

- 1) Bij het detecteren van een nieuwe observatie.
- 2) Bij een vast interval onafhankelijk van veranderingen.

De methode van updates bij nieuwe observaties voorziet ook een timer om de spanning te verhogen indien een ingestelde tijdsperiode verstreken is.

De parameters van de regelaar worden door het geleidelijk aan wijzigen van de parameters en vervolgens het analyseren van de bekomen resultaat bijgesteld tot een optimaal resultaat bereikt wordt.



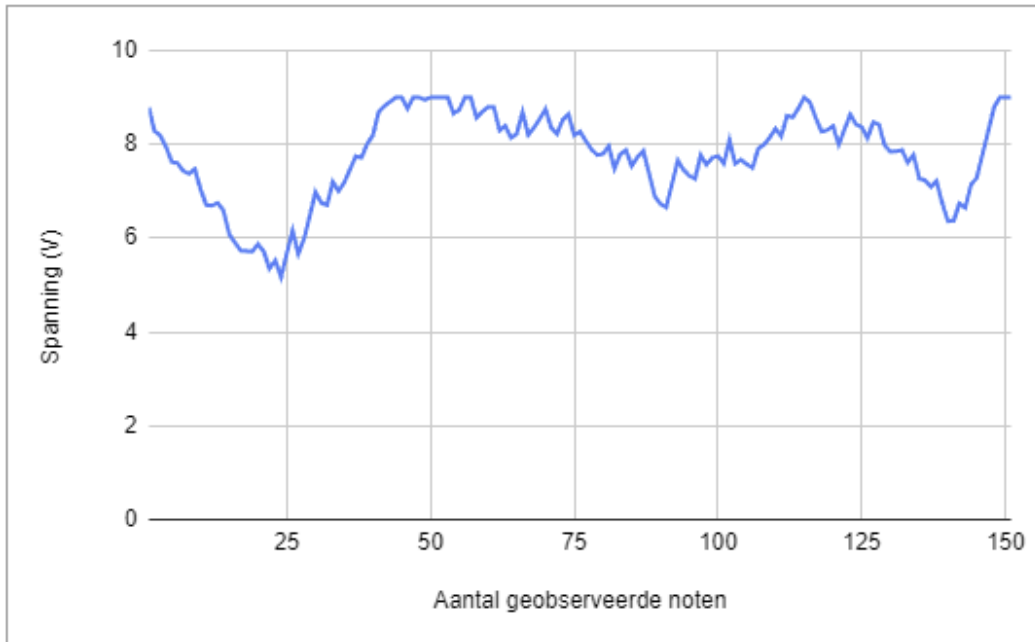
Figuur 3.3: Schema na implementatie van de regelaar

De parameters van de regelaar worden bepaald door iteratief kleine wijzigingen te brengen en de resultaten te analyseren tot een optimaal resultaat bereikt is. Binnen de systeemtheorie is het mogelijk om deze parameters te berekenen. Dit kan fungeren als een schatting voor de parameters. Dit kan niet als een finale methode voor het bepalen van de parameters gebruikt worden vanwege de ruis op het signaal. De grootte van de ruis blijft namelijk onbepaald binnen dit onderzoek met als gevolg dat dit niet meegenomen wordt in een berekening wat tot een afwijking leidt ten opzichte van het werkelijke systeem.

### 3.4 Resultaten van de actuator gecontroleerd door een PID regelaar

Figuur 3.4 toont het verloop van de spanning indien er updates gebeuren bij het detecteren van noten. In dit verloop zijn momenten waarop de spanning terugvalt zichtbaar, dit wordt veroorzaakt door de afstand tussen noten. Dit veroorzaakte dat noten tijdelijk niet meer door de observator gaan. Door de updates door een te groot tijdsverschil wordt dit gedrag hersteld. Dit effect is echter niet gewenst en leidt tot instabiel gedrag van de regeling.

Bij een vaste update snelheid is ditzelfde effect niet zichtbaar. Figuur 3.5 toont het effect van updates bij een vast interval van 0,4 seconden. De parameters van de regelaar worden weergegeven in tabel 3.1. Het gedrag van deze regelaar is veel vlakker en regelt de snelheid geleidelijk aan naar de gewenste snelheid van vier noten per seconde. Deze regelaar zorgt voor een stabiel systeem waarbij de gewenste snelheid behaald wordt. De exacte snelheid zal echter niet consistent aangehouden worden, dit effect ontstaat door de ruis in het systeem. Dit kan niet volledig weggefilterd worden. Figuur 3.6 toont het gedrag indien pinda noten toegevoerd worden in

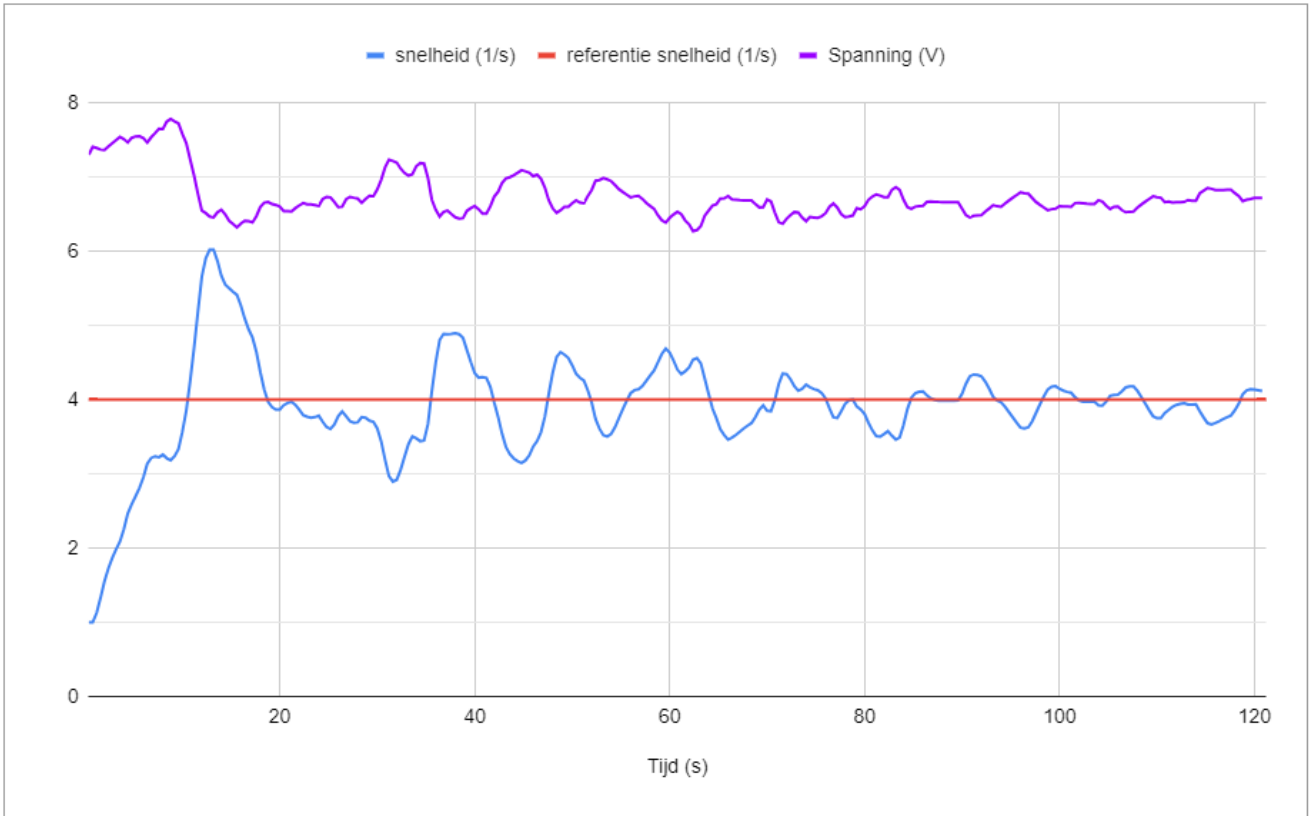


Figuur 3.4: Spanning verloop van de PID regelaar bij nieuwe observaties bij de toevoer van amandelen

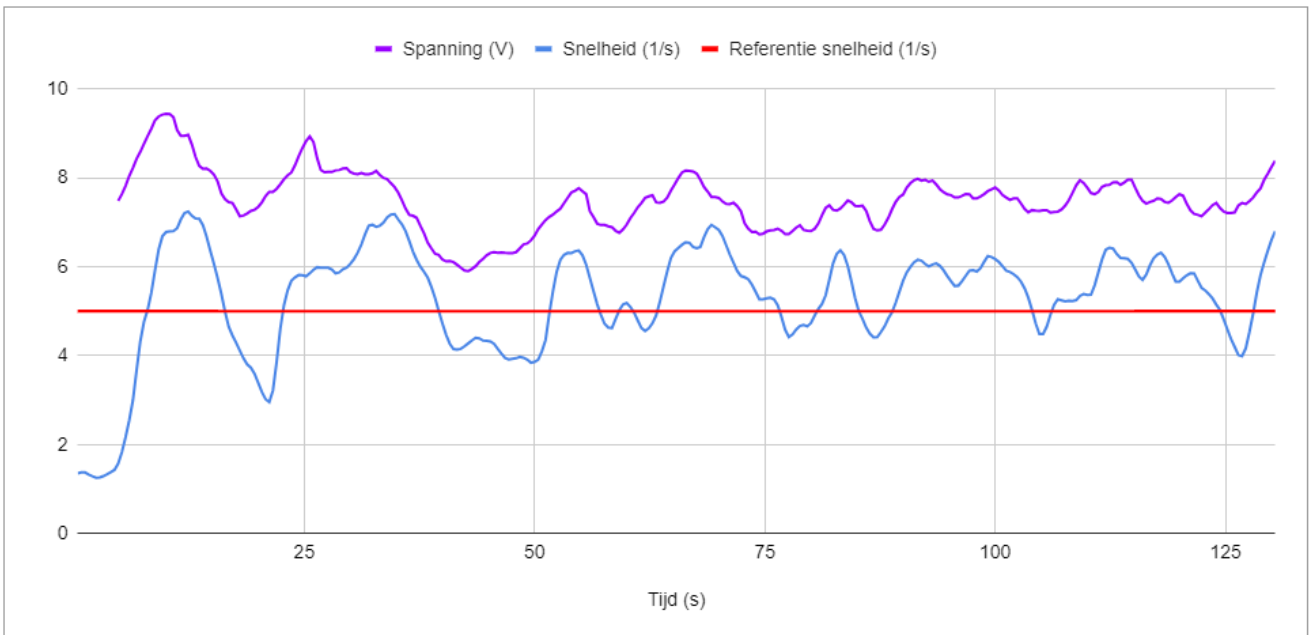
Tabel 3.1: Parameters PID regelaar

Parameter	Value
$K_r$	1,2
$\tau_i$	20
$\tau_d$	0,9
$K$	1/37
Interval tijd	0,4 s

plaats van amandelen. Dit geeft een minder stabiel gedrag veroorzaakt door de aard van het materiaal. Pindanoten zijn lichter dan amandelen en bovendien ook kleiner. Dit zorgt ervoor dat het snelheidsverschil van pinda's bij een gelijk verschil in de trilfrequentie groter is dan bij amandelen. Een aanpassing van de spanning zal bijgevolg niet dezelfde verandering in de snelheid geven bij ieder type noot. Een globale regelaar die optimaal functioneert voor alle soorten noten is bijgevolg niet mogelijk. Indien dit toch geïmplementeerd wordt zal de regeling te traag zijn voor zwaardere objecten. De regeling voor lichte objecten is in deze situatie potentieel onstabiel. Ten slotte zal het systeem bij het gebruik van een regelaar die enkel de snelheid als feedback gebruikt op bepaalde tijdstippen, geen objecten hebben om te verwerken. Dit introduceert stalling en de actuator zal bijgevolg de limiterende component zijn voor de totale systemsnelheid. Een bijkomende factor die marge op het aantal geobserveerde noten voorziet is noodzakelijk om te zorgen dat de actuator niet limiterend is voor het systeem.



Figuur 3.5: Spanning en snelheid verloop van de PID regelaar bij vast interval (0,4s) bij de toevoer van amandelen



Figuur 3.6: Spanning en snelheid verloop van de PID regelaar bij vast interval (0,4s) bij de toevoer van pinda 's





# Hoofdstuk 4

## Inclusie van de buffer in de regelaar

Hoofdstuk 3 geeft aan dat een regelaar die enkel de snelheid als feedback gebruikt, leidt tot stalling in het systeem. Dit is ongewenst voor het systeem omdat dit veroorzaakt dat het volledige proces vertraagd en gelimiteerd wordt door de toevoer. Om dit probleem op te lossen moet de verwerkingsmodule van het systeem consistent observaties hebben om te verwerken. Zoals beschreven in hoofdstuk 1 wordt deze gevoed door een buffer in het geheugen van de controller. De verwerkingsmodule zal dus observaties hebben om te verwerken indien deze buffer niet leeg is. Binnen dit onderzoek gebruikt de regelaar niet de feedback van de buffer in het systeem, het gedrag ervan wordt gesimuleerd. Deze simulatie gebeurt door een teller die omhoog gaat bij het detecteren van objecten en omlaag zal gaan bij vaste intervallen die de verwerkingstijd nabootsen.

### 4.1 Bepalen van het foutsignaal

Het foutsignaal van de PID regelaar dient aangepast te worden om de staat van de buffer in acht te nemen. Dit foutsignaal bevat hetzelfde signaal als in formule 3.11.

Het foutsignaal van de buffer kan op verschillende methodieken bepaald worden. Het eerste is door de fout te doen afnemen naarmate de buffer meer observaties bevat. Een tweede is door een optimaal punt te definiëren en de fout te doen toenemen naarmate de huidige staat hier verder van afwijkt.

De buffer in kwestie heeft een beperkte grootte en kan bijgevolg ook overvol geraken. Dit effect is evenzeer ongewenst, de eerste methode beperkt deze mogelijkheid niet. De tweede methode heeft hier wel preventie tegen indien het optimum gekozen wordt zodat zowel voor stalling als een overvolle buffer de fout groot zal zijn.

Voor het bepalen van de fout op de buffer wordt voor het optimum indien de buffer half gevuld is geopteerd, binnen het systeem is dit bij 10 observaties. De buffer fout wordt bepaald door middel van formule 4.1.

$$e(n) = n_{\text{gewenst}} - n_{\text{gemeten}} \quad (4.1)$$

waarbij:

$e(n)$  = de fout op het aantal objecten in de buffer  
 $n_{gewenst}$  = Het optimale aantal objecten in de buffer  
 $n_{gemeten}$  = Het huidige aantal objecten in de buffer

De totale fout kan op verschillende manieren samengesteld worden uit de individuele foutsignalen. Deze paper opteert voor dezelfde methodiek die in lineaire regressie gebruikt wordt om een kostfunctie op te stellen. Namelijk door een sommatie van de foutsignalen met een constante factor  $\theta$  zoals in formule 4.2 beschreven [31].

$$e_t = \theta_1 \times e(t) + \theta_2 \times e(n) \quad (4.2)$$

waarbij:

$e_t$  = De totale fout  
 $\theta_1$  = Het gewicht van de snelheid fout  
 $\theta_2$  = Het gewicht van de buffer fout  
 $e(t)$  = de snelheid fout  
 $e(n)$  = de buffer fout

Aangezien het toevoegen van een nieuwe bron van feedback ook een wijziging brengt in de aard van het foutsignaal dienen ook de parameters van de regelaar herzien te worden. Tabel 4.1 toont de parameters van de regelaar voor het nieuwe foutsignaal. De integratie tijdsconstante wordt vergroot om de invloed van het verleden te beperken. Indien deze factor een te grote invloed zou hebben, zal het systeem te traag reageren op het leeg of overlopen van de buffer. De parameters worden op analoge wijze als in hoofdstuk 3 bepaald.

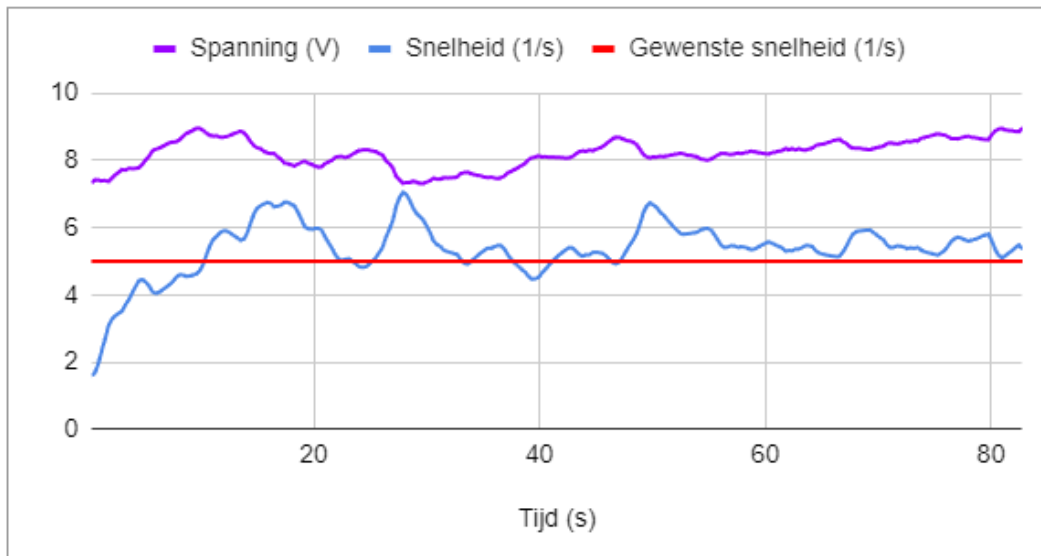
Tabel 4.1: Parameters PID regelaar met aangepast feedback signaal

Parameter	Value
Kr	0.8
$\tau_i$	150
$\tau_d$	0,25
K	1/23
Interval tijd	0,3 s

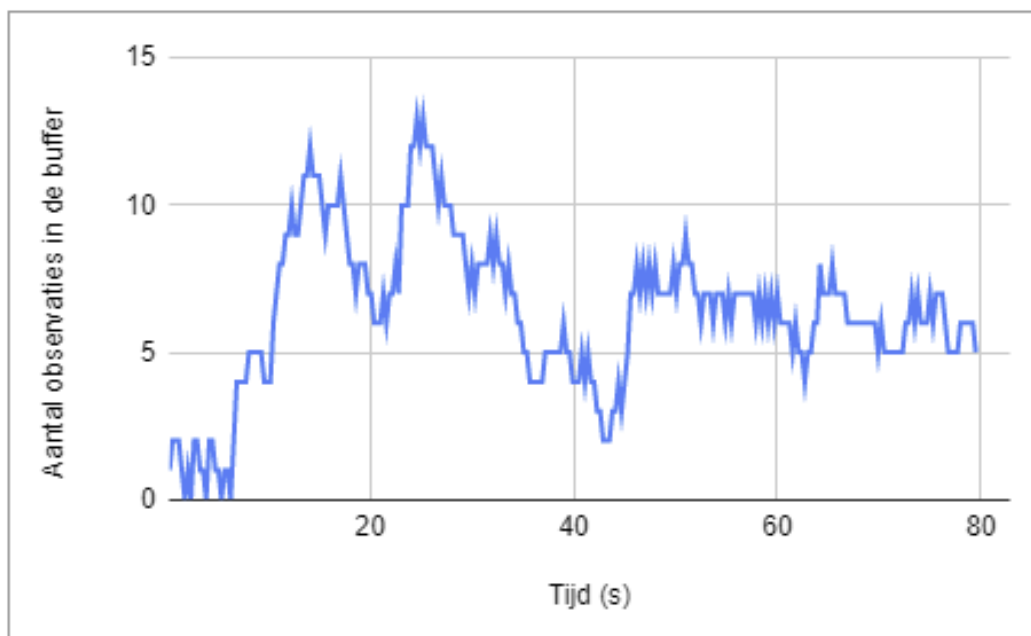
## 4.2 Resultaten van de aangepaste PID regelaar

Figuur 4.1 toont de spanning en snelheid in functie van de tijd. Een eerste observatie is dat in tegenstelling tot de resultaten in hoofdstuk 3 de snelheid groter is dan de gewenste snelheid. Bijkomend toont figuur 4.2 het aantal objecten in de buffer. De buffer bevat consistent minder objecten dan als optimaal beschouwd wordt (10 objecten). Beide signalen bevatten bijgevolg een fout die dient gecorrigeerd te worden door de regelaar. Het teken van de foutsignalen is omgekeerd, bijgevolg wordt de totale fout kleiner en zal de regelaar minder bijsturen. De regelaar zal dus niet de ideale snelheid en ideale bufferstaat bereiken maar proberen de fout te minimaliseren ongeacht de waarde van de individuele foutsignalen. Deze compromis is niet problematisch voor het systeem aangezien er consistent te verwerken objecten zijn in de buffer met uitzondering van de opstart van het systeem waarin nog geen beelden beschikbaar zijn en de actuator nog

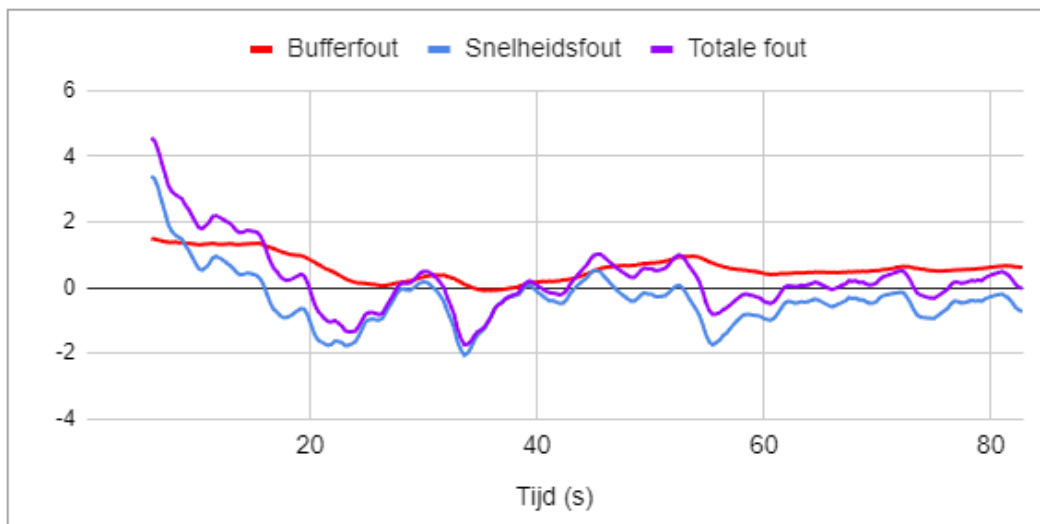
moet versnellen. De balans tussen de fouten kan gewijzigd worden door de gewichten in de foutbepaling aan te passen. De gewichten geven aan hoeveel invloed individuele signalen hebben op de totale fout. Een hoger gewicht geeft meer voorkeur aan het bijbehorende signaal om het optimaal te bereiken. Het dient opgemerkt te worden dat de foutsignalen steeds van richting zullen verschillen indien dit soort compromis ontstaat aangezien de signalen destructief werken op elkaar in deze situatie. Indien de richting gelijk is zullen de fouten elkaar versterken. Figuur 4.3 toont de foutsignalen in functie van de tijd.



Figuur 4.1: Spanning en snelheid verloop van de actuator bij de aangepaste PID regelaar bij toevoer van amandelen



Figuur 4.2: Aantal observaties in de buffer in functie van de tijd bij toevoer van amandelen



Figuur 4.3: Foutsignalen in functie van de tijd bij toevoer van amandelen

# Hoofdstuk 5

## Conclusie

EtherCAT staat de desktop controller toe om met randapparatuur te communiceren. Er blijft wel een nood aan polling met de modules vanwege de aard van het protocol. De object toevoer blijft ook onderhevig aan ruis vanwege natuurlijke verschillen tussen producten en oneffenheden in de actuator. De metingen blijven daarbuiten onderhevig aan ruis. Door middel van het uitmiddelen van de feedback signalen wordt een PID regelaar ontwikkeld die in staat is om de toevoersnelheid bij te sturen zodat dit de verwerking kan volgen. De regelaar is optimaal voor slechts een type noot vanwege de verandering in de systeem response door het gewicht van verschillende type noten. Een regelaar die enkel op snelheid regelt is niet ideaal aangezien deze stalling zal veroorzaken in het verwerkingsproces. Dit wordt opgelost door de toevoeging van feedback over de hoeveelheid te verwerken data. Dit zorgt ervoor dat er steeds te verwerken data is terwijl de snelheid aangrenzend blijft aan de gewenste toevoer snelheid. Met constante evoluties in de verwerkingssnelheid zal op termijn de gebruikte hardware niet meer in staat zijn om deze module te volgen. Er dient dus vervolgonderzoek gedaan te worden naar een nieuw systeem om deze te vervangen. Ook vervolgonderzoek over andere typen regelaars kan leiden tot een verbetering in het regelsysteem. Zo werd een kostfunctie opgesteld om stalling te vermijden. Binnen machine learning worden technieken gebruikt om deze te optimaliseren, wat ook de regelaar kan optimaliseren. Verder dient de regelaar uitgebreid te worden om ook met andere soorten noten om te kunnen gaan.



# Literatuurlijst

- [1] “Bowl feeder bf20 / bf25 / bf30 / bf35 / bf40 / bf50,” Afag, 2023. [Online]. Available: [https://www.afag.com/fileadmin/user\\_upload/afag/Produkte/ZTK/PDF-Files/BF20-50\\_OI\\_EN\\_01.pdf](https://www.afag.com/fileadmin/user_upload/afag/Produkte/ZTK/PDF-Files/BF20-50_OI_EN_01.pdf) [geopend 6 Juni 2023]
- [2] “Controller irg 1-s,” Afag, 2022. [Online]. Available: [https://www.afag.com/fileadmin/user\\_upload/afag/Produkte/ZTK/PDF-Files/BA\\_IRG1-S\\_R02.4\\_EN.pdf](https://www.afag.com/fileadmin/user_upload/afag/Produkte/ZTK/PDF-Files/BA_IRG1-S_R02.4_EN.pdf) [geopend 6 Juni 2023]
- [3] “El40xx analog output terminals (12 bit),” Beckhoff, 2022. [Online]. Available: <https://download.beckhoff.com/download/Document/io/ethercat-terminals/EL40xxen.pdf> [geopend 6 Juni 2023]
- [4] G. M. Smith, “What is ethercat protocol and how does it work?” Dewesoft, 10 Dec. 2021. [Online]. Available: <https://dewesoft.com/blog/what-is-ethercat-protocol> [geopend 17 February 2023]
- [5] “Ethercat - the ethernet fieldbus,” Ethercat Technology group. [Online]. Available: <https://www.ethercat.org/en/technology.html> [geopend 17 February 2023]
- [6] EtherCATGroup, “Ethercat in 20 minutes,” 2019. [Online]. Available: <https://www.youtube.com/watch?v=k4KufZR6XYs> [geopend 8 Maart 2023]
- [7] P. Behaeghel, “Ethercat: Studie en uitwerking didactische opstelling,” Master’s thesis, Universiteit Gent, 2014. [Online]. Available: <https://lib.ugent.be/catalog/rug01:002385458>
- [8] “Ethercat state machine,” Beckhoff, 22 Feb. 2022. [Online]. Available: [https://infosys.beckhoff.com/english.php?content=../content/1033/ax5000\\_usermanual/html/Bt\\_EcBasics\\_EcStateMachine.htm](https://infosys.beckhoff.com/english.php?content=../content/1033/ax5000_usermanual/html/Bt_EcBasics_EcStateMachine.htm) [geopend 10 Mei 2023]
- [9] H. Xing, H. Jia, and L. Yanqianga, “Motion control system using sercos over ethercat,” *Procedia Engineering*, vol. 24, pp. 749–753, 2011.
- [10] “Soem,” OpenEtherCATsociety, 2022. [Online]. Available: <https://github.com/OpenEtherCATsociety/SOEM> [geopend 12 Mei 2023]
- [11] “Soem - documentation,” openethercatsociety, 2019. [Online]. Available: <https://openethercatsociety.github.io/doc/soem/index.html> [geopend 12 Mei 2023]
- [12] “C++ samples,” Beckhoff, 2022. [Online]. Available: [https://infosys.beckhoff.com/english.php?content=../content/1033/tc3\\_c/2407126539.html](https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_c/2407126539.html) [geopend 5 Juni 2023]



- [13] “Igh ethercat master for linux,” Ingenieurgesellschaft IgH, 2021. [Online]. Available: <https://etherlab.org/en/ethercat/> [geopend 5 Juni 2023]
- [14] “fastcat,” NASA Jet Propulsion Laboratory, 2023. [Online]. Available: <https://github.com/nasa-jpl/fastcat> [geopend 5 Juni 2023]
- [15] “Soem interface,” Robotic Systems Lab - ETH Zurich, 2021. [Online]. Available: [https://github.com/leggedrobotics/soem\\_interface](https://github.com/leggedrobotics/soem_interface) [geopend 5 Juni 2023]
- [16] “Soem - slaveinfo.c,” OpenEtherCATsociety, 2022. [Online]. Available: <https://github.com/OpenEtherCATsociety/SOEM/blob/master/test/linux/slaveinfo/slaveinfo.c> [geopend 5 Juni 2023]
- [17] “Soem - tutorial,” rt-labs, 2020. [Online]. Available: <https://rt-labs.com/docs/soem/tutorial.html> [geopend 12 Mei 2023]
- [18] “Cyber security attacks packet crafting,” Valency networks, 2008. [Online]. Available: <https://www.valencynetworks.com/articles/cyber-security-attacks-packet-crafting.html> [geopend 12 Mei 2023]
- [19] “Os command injection,” PortSwigger, 2023. [Online]. Available: <https://portswigger.net/web-security/os-command-injection> [geopend 12 Mei 2023]
- [20] “setcap(8) — linux manual page,” Michael Kerrisk, 2022. [Online]. Available: <https://man7.org/linux/man-pages/man8/setcap.8.html> [geopend 12 Mei 2023]
- [21] “capabilities(7) — linux manual page,” Michael Kerrisk, 2022. [Online]. Available: <https://man7.org/linux/man-pages/man8/setcap.8.html> [geopend 12 Mei 2023]
- [22] “Cmake,” Kitware, 2023. [Online]. Available: <https://cmake.org/> [geopend 6 Juni 2023]
- [23] “std::thread,” cplusplus.com, 2023. [Online]. Available: <https://cplusplus.com/reference/thread/thread/> [geopend 12 Mei 2023]
- [24] P. S. Pacheco and M. Malensek, *An Introduction to Parallel Programming (Second Edition)*. Philadelphia: Morgan Kaufmann, 2022.
- [25] J. Wengrow and B. MacDonald, *A common-sense guide to data structures and algorithms: Level up your core programming skills*. The Pragmatic Bookshelf, 2020.
- [26] “Moving average (ma): Purpose, uses, formula, and examples,” Investopedia, 31 Mar. 2023. [Online]. Available: <https://www.investopedia.com/terms/m/movingaverage.asp> [geopend 2 April 2023]
- [27] “Moving averages - simple and exponential,” StockCharts. [Online]. Available: [https://school.stockcharts.com/doku.php?id=technical\\_indicators:moving\\_averages](https://school.stockcharts.com/doku.php?id=technical_indicators:moving_averages) [geopend 2 April 2023]
- [28] J. Baeten, *REG: Regeltechniek: syllabus, oefeningenbundel [cursus]*. Diepenbeek: Faculteit Industriële Ingenieurswetenschappen KU Leuven & UHasselt, 2020.
- [29] T. van Waterschoot, *Meet- en Regeltechniek Les 4: De klassieke regelaars [cursus]*. Faculteit Industriële Ingenieurswetenschappen ESAT Departement Elektrotechniek KU Leuven, Belgium, 2015.

- [30] “De oppervlakte onder een grafiek,” H.hofstede, 2022. [Online]. Available: <https://www.hhofstede.nl/modules/riemann.htm> [geopend 14 Mei 2023]
- [31] A. NG, A. Bagul, E. Shyu, and G. Ladwig, *Supervised Machine Learning: Regression and Classification [cursus]*. Coursera, 2012.

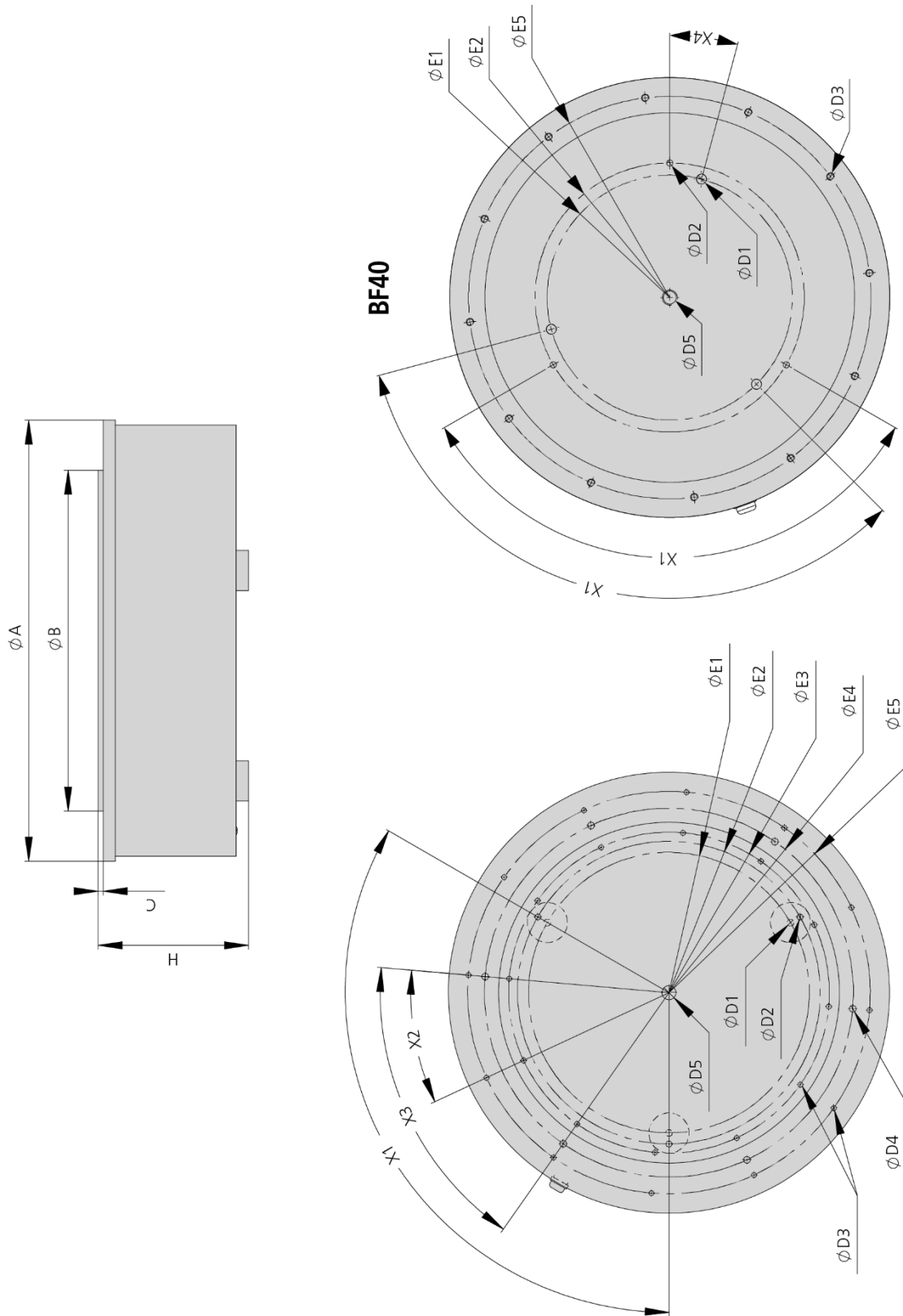


# Bijlage A

## Technische data van de shaker

## 2.4 Technical data

Figure 3: Dimension sheet for BF08-15



**Table 1: Technical data**

Models		Units	BF 20	BF 25	BF 30	BF 35	BF 40	BF 50
Dimension	A	[mm]	180	224	265	305	345	440
	B	[mm]	142	194	194	294	294	340
	C	[mm]	2	2	3	3	3	5
	D1	-	M8	M10	M10	M10	M12	M16
	D2	-	---	---	4 x M6	4 x M6	4 x M6	24 x M6
	D3	[mm]	3 x 4H7	3 x 5H7	3 x 5H7	3 x 5H7	3 x 5H7	---
	D4	[mm]	---	3 x 5H7	3 x 5H7	3 x 5H7	---	---
	D5	-	3 x M6	3 x M6	3 x M8	3 x M8	3 x M8	3 x M8
	E1	[mm]	87	100	140	162	214	---
	E2	[mm]	---	116	162	214	---	---
	E3	[mm]	120	120	195	195	195	280
	F	[mm]	---	---	186	186	320	320 / 400
	H	[mm]	90	93	100	103	112	150
	X1	[°]	3 x 120°	3 x 120°	3 x 120°	3 x 120°	3 x 120°	3 x 120°
	X2	[°]	4 x 90°	4 x 90°	4 x 90°	4 x 90°	4 x 90°	12 x 30°
	X3	[°]	---	---	---	---	---	6 x 30°
X4	[°]	---	---	---	---	3 x 15°	---	
Max. mass moment of inertia	[kg*dm <sup>2</sup> ]	1,0 ±10%	2,2 ±10%	4,5 ±10%	8,8 ±10%	17,0 ±10%	65,0 ±10%	
Max. bowl weight	[kg]	2,0 ±30%	3,0 ±30%	4,0 ±30%	6,0 ±30%	9,0 ±30%	20,0 ±30%	
Weight	[kg]	7,3	12,0	15,4	23,3	34,5	56,6	
Rating	[VA]	69	69	135	135	180	950	
Protection type	IP	54						
Vibration force compensation	-	yes						
Electric version	[V / Hz]	230 / 50 or 115 / 60						
Vibration frequency	[1/min]	6000 / 7200						
Direction of rotation	-	right / left						
Leaf spring angle	[°]	12° / 18°						
Environmental conditions for operation: Temperature range	[°C]	+10 to +45						
Noise emission: Continuous noise pressure level (without transported material)	[dB]	<70						
Measuring height/measuring distance	[m]	1,6 / 1						
Measurement direction with respect to the noise source	[°]	90						
Measurement method	-	A evaluation						



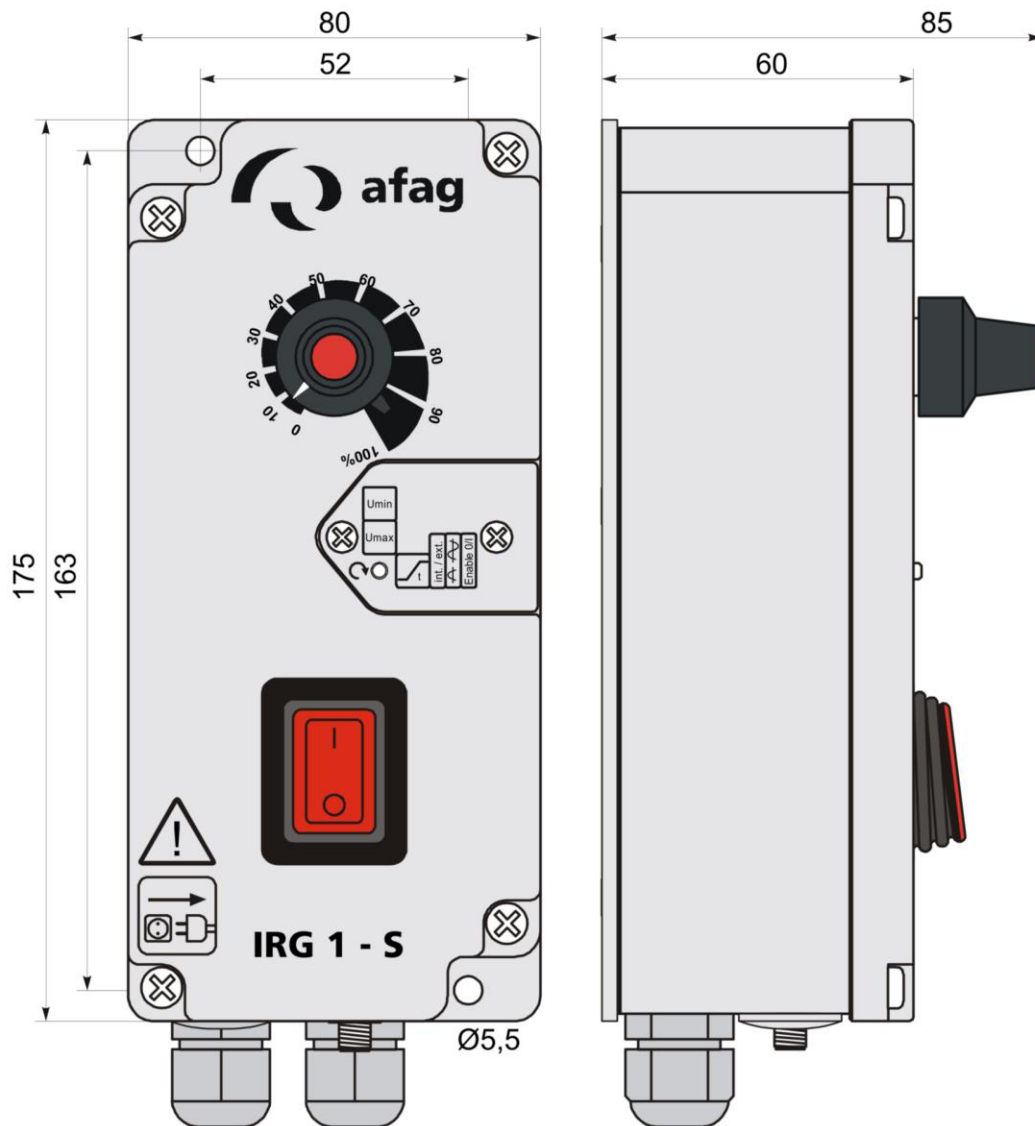
# Bijlage B

## Signaal convertor irg1-s



## 2.2 Technical data

Figure 1: IRG 1-S



Fixing dimensions: 163 x 52 mm

**Table 1: Technical data**

Type	Units	IRG 1-S	
Operating voltage	[VAC]	230 ±10%	115 / ± 10%
Operating frequency	[Hz]	50 / 60	
Oscillation frequency (Full- / half-wave)	[Hz]	50 / 100 with 50 Hz supply frequency 60 / 120 with 60 Hz supply frequency	
Output voltage	[VAC]	40 - 220	20 – 105
Output current	[A]	0 - 6	
Type of protection	IP	IP54	
Fuses	---	1 x 6,3 A	
Mains connector	---	2m with moulded Schuko angle plug	
Connector to conveyor	---	2m cable 3 x 1 mm <sup>2</sup> with Hirschmann connector STAK 20	
Dimensions (l x d x h) ca.	[mm]	175 x 80 x 60	
Control input (Optocoupler input)	---	Contact or +24 V DC external voltage	
Soft start	[s]	0 - 4	
Environmental conditions for operation: Temperature range	[°C]	0 to +45	

### 3 Assembly instructions

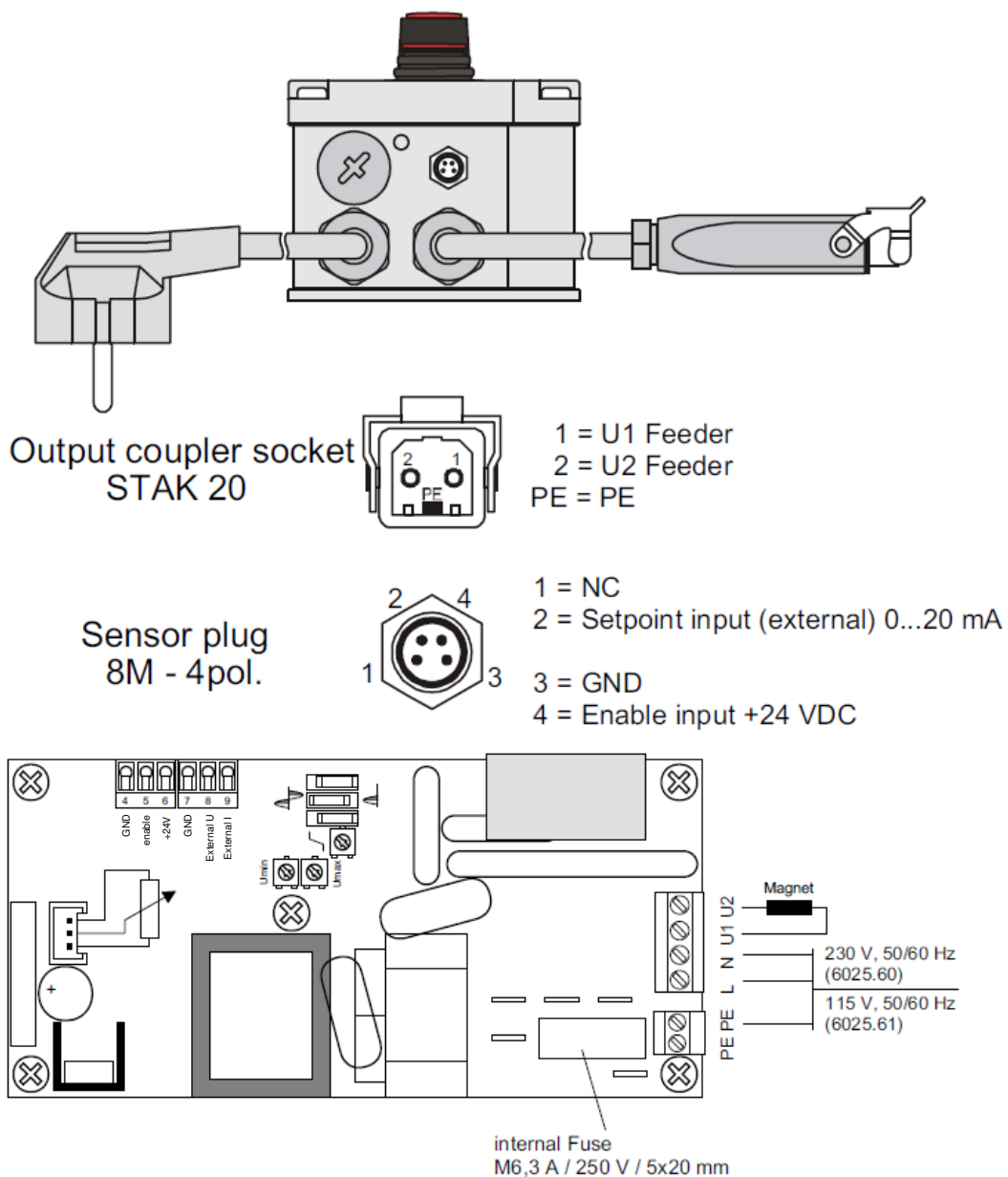
#### 3.1 Installing the unit

There are two holes on the underside for mounting the controller. The holes are separated from the interior of the housing. (See Figure 1)

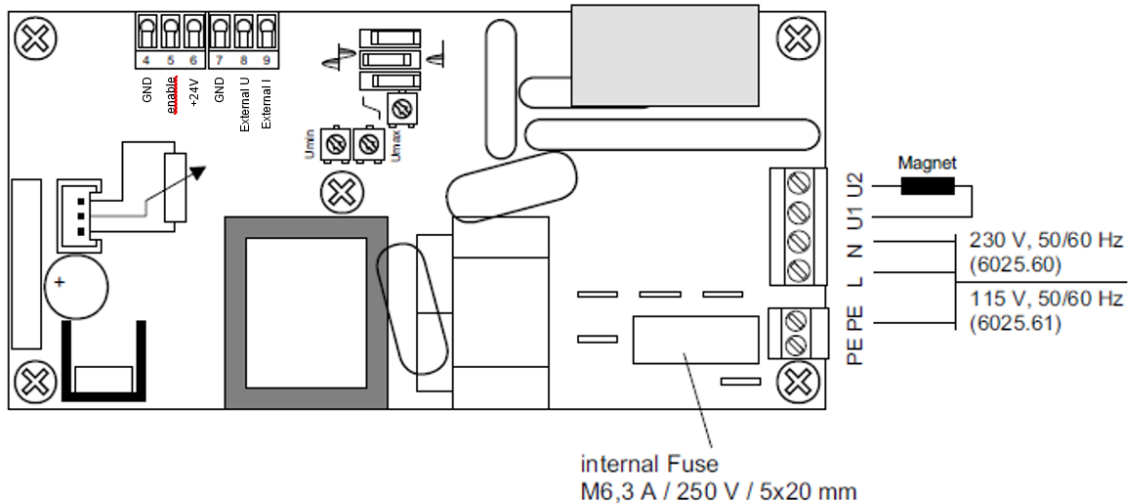
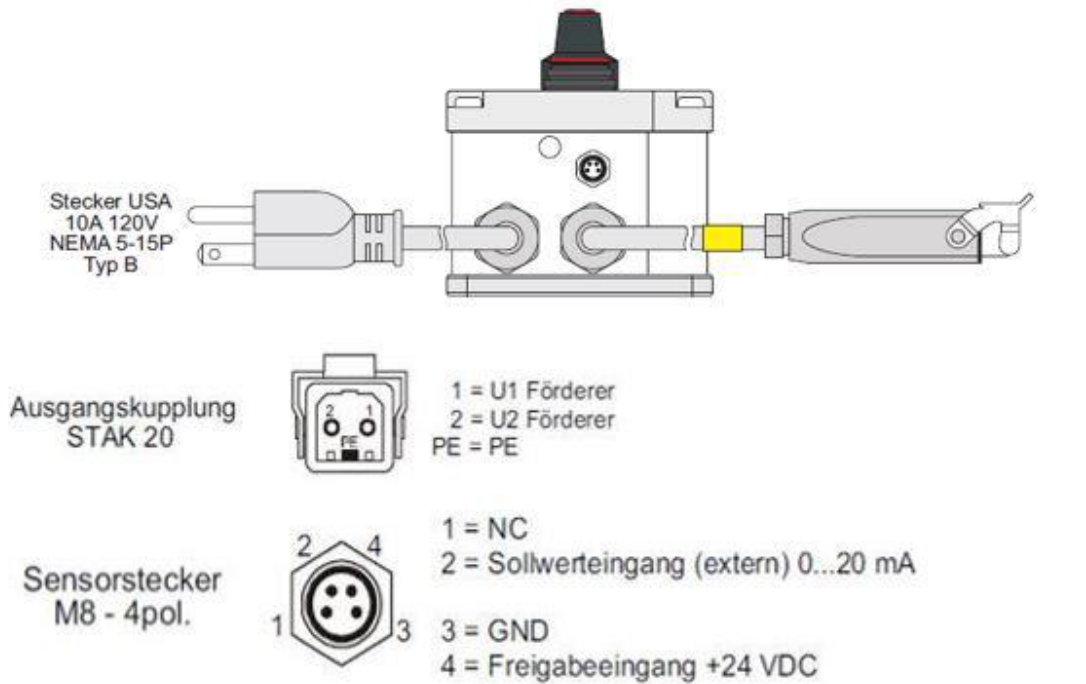
#### 3.2 Connection possibilities

Type 230V/50Hz

Figure 2: Connection possibilities



## Type 115V/60Hz



### CAUTION

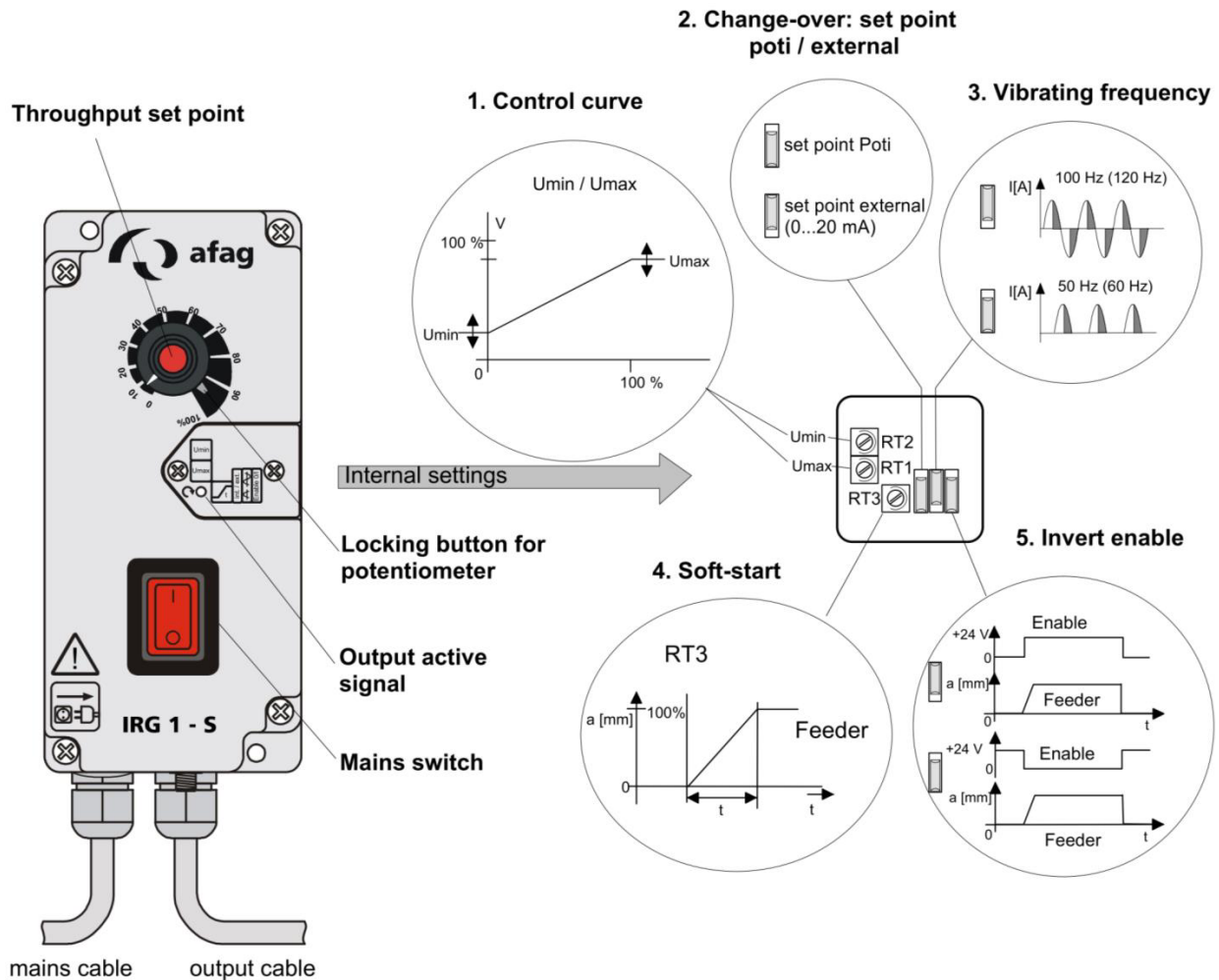


Jumpers may only be inserted for the respective application, otherwise this may lead to a malfunction of or damage to the p.c.b.

## 4 Operating instructions

Standard settings can be made without removing the front-panel. The setting up components are accessed by removing the small cover on the right hand side of the front-panel.

Figure 3: Settings



# Bijlage C

## **EtherCAT modules: technische data**

## 2.1.2 EL400x - Technical data

Technical data	EL4001	EL4002	EL4004	EL4008
Number of outputs	1	2	4	8
Power supply	24 V DC via the power contacts			
Signal voltage	0..10 V			
Load	> 5 kΩ (short-circuit-proof)			
Measuring error	< ± 0.1% (at 0 °C ... +55 °C, relative to the full scale value) < ± 0.2% (when the extended temperature range is used)			
Resolution	12 bit			
Sampling type	multiplex			
Ground reference	single ended			
Conversion time	~ 100 μs	~ 150 μs	~ 250 μs	~ 400 μs
Power supply for electronics	via the E-bus			
Distributed Clocks	yes			
Current consumption via E-bus	typ. 140 mA	typ. 140 mA	typ. 140 mA	typ. 100 mA
Current consumption from the power contacts (24V)	typ. 40 mA at full output			
Electrical isolation	500 V (E-bus/field voltage)			
Bit width in process image	1 x 16-bit AO output	2 x 16-bit AO output	4 x 16-bit AO output	8 x 16-bit AO output
Configuration	via TwinCAT System Manager			
Weight	approx. 60 g			
Permissible ambient temperature range during operation	-25 °C ... +60 °C (extended temperature range)			
Permissible ambient temperature range during storage	-40 °C ... +85 °C			
Permissible relative humidity	95%, no condensation			
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)			
<a href="#">Mounting</a> [▶ 85]	on 35 mm mounting rail conforms to EN 60715			
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also <a href="#">installation instructions</a> [▶ 88] for enhanced mechanical load capacity			
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4			
Protection class	IP20			
Installation position	variable			
Marking / Approval <sup>*)</sup>	CE, UKCA, EAC <a href="#">ATEX</a> [▶ 79], <a href="#">cULus</a> [▶ 84]			

\*) Real applicable approvals/markings see type plate on the side (product marking).

### Ex marking

Standard	Marking
ATEX	II 3 G Ex nA IIC T4 Gc

## 2.1.4 EL1002, EL1004-00x0, EL1008 - Technical data

Technical data	EL1002	EL1004	EL1004-0020	EL1008
Number of inputs	2	4		8
Number of simultaneously controllable inputs, depending on the ambient temperature	2 (> +55°C)	4 (-25°C ... +55°C) 2 (> +55°C)		8 (-25°C ... +55°C) 4 (> +55°C) (aligned in <a href="#">horizontal installation position</a> ) [► 63]
Nominal voltage of the inputs	24 V <sub>DC</sub> (-15% / +20%)			
Signal voltage "0"	-3 V ... 5 V (EN 61131-2, type 1/3)			
Signal voltage "1"	11 V ... 30 V (EN 61131-2, type 1/3)			
Input filter	3 ms			
Input current	typically 3 mA (EN 61131-2, type 1/3)			
Current consumption power contacts	typ. 2 mA + load			
Current consumption via E-bus	typ. 90 mA			
Electrical isolation	500 V (E-bus/field voltage)		2500 V functional isolation (test voltage E-bus / field voltage, production test).	500 V (E-bus/field voltage)
Bit width in the process image	2 input bits	4 input bits		8 input bits
Configuration	no address setting, configuration via TwinCAT System Manager			
Weight	approx. 55 g			
Permissible ambient temperature range during operation	-25°C ... +60°C (extended temperature range)		0°C ... +55°C	-25°C ... +60°C (extended temperature range, aligned in <a href="#">horizontal installation position</a> ) [► 63] -25°C ... +45°C (all other installation positions [► 63])
Permissible ambient temperature range during storage	-40°C ... +85°C		-25°C ... +85°C	-40°C ... +85°C
Permissible relative humidity	95%, no condensation			
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)			
Mounting [► 51]	on 35 mm mounting rail conforms to EN 60715			
Vibration/shock resistance	according to EN 60068-2-6/EN 60068-2-27, see also <a href="#">Installation instructions for terminals with increased mechanical load capacity</a> [► 65]		according to EN 60068-2-6/EN 60068-2-27	according to EN 60068-2-6/EN 60068-2-27, see also <a href="#">Installation instructions for terminals with increased mechanical load capacity</a> [► 65]
EMC resistance burst/ESD	conforms to EN 61000-6-2 / EN 61000-6-4			
Protection class	IP20			
Installation position	variable			see note [► 63]
Approval	CE, cULus [► 58], ATEX [► 53], IECEx [► 54], cFMus [► 56]		CE, cULus [► 58]	CE, cULus [► 58], ATEX [► 53], IECEx [► 54], cFMus [► 56]

### Ex markings

Standard	Marking
ATEX	II 3 G Ex nA IIC T4 Gc II 3 D Ex tc IIIC T135 °C Dc
IECEX	Ex nA IIC T4 Gc Ex tc IIIC T135 °C Dc
cFMus	Class I, Division 2, Groups A, B, C, D Class I, Zone 2, AEx/Ex ec IIC T4 Gc



## 2.2 Technical data

Technical data	EL1502
Number of counters	1 or 2
Rated voltage	24 V <sub>DC</sub> (-15%/+20%)
Signal voltage "0"	-3 V ... 5 V (EN 61131-2, type 1)
Signal voltage "1"	15 V ... 30 V (EN 61131-2, type 1)
Counting frequency	100 kHz
Counter depth	32 bit
Input current	typ. 5 mA (EN 61131-2, type 1)
Output current (per channel)	max. 0.5 A (short-circuit-proof)
Distributed Clocks (DC)	yes (from FW 02)
Current consumption power contacts	typ. 14 mA + load
Current consumption via E-bus	typ. 130 mA
Electrical isolation	500 V (E-bus/field voltage)
Supports NoCoeStorage [► 24] function	yes
Configuration	via TwinCAT System Manager
Weight	approx. 50 g
Permissible ambient temperature range during operation	-25°C ... +60°C (extended temperature range)
Permissible ambient temperature range during storage	-40°C ... +85°C
Permissible relative humidity	95%, no condensation
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm
Mounting [► 33]	on 35 mm mounting rail conforms to EN 60715
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also <a href="#">installation instructions [► 36]</a> for enhanced mechanical load capacity
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approvals/markings	CE, EAC, UKCA, ATEX [► 31] cULus [► 32]

\*) Real applicable approvals/markings see type plate on the side (product marking).

### Ex marking

Standard	Marking
ATEX	II 3 G Ex nA IIC T4 Gc

## 2.3 Start

For commissioning:

- mount the EL15xx as described in the chapter [Mounting and wiring \[► 30\]](#)
- configure the EL15xx in TwinCAT as described in the chapter [Commissioning \[► 120\]](#).