

2013•2014  
FACULTEIT INDUSTRIËLE INGENIEURSWETENSCHAPPEN  
*master in de industriële wetenschappen: chemie*

## Masterproef

Simuleren en ontwerpen van klassieke en geavanceerde regelschema's in de chemische industrie

Promotor :  
Prof. dr. ir. Leen THOMASSEN

Promotor :  
Prof. dr. ir. FILIP LOGIST

Gerd Decramer

*Proefschrift ingediend tot het behalen van de graad van master in de industriële wetenschappen: chemie*

Gezamenlijke opleiding Universiteit Hasselt en KU Leuven

2013•2014  
Faculteit Industriële  
ingenieurswetenschappen  
*master in de industriële wetenschappen: chemie*

## Masterproef

Simuleren en ontwerpen van klassieke en geavanceerde  
regelschema's in de chemische industrie

Promotor :  
Prof. dr. ir. Leen THOMASSEN

Promotor :  
Prof. dr. ir. FILIP LOGIST

Gerd Decramer

*Proefschrift ingediend tot het behalen van de graad van master in de industriële  
wetenschappen: chemie*

## Voorwoord

Als student van het schakeljaar kreeg ik vorig jaar de kans om dit onderwerp te kiezen. Hierdoor zou ik mijn ervaring als oud-meetkameroperator van een chemisch bedrijf op de Chemelot-site in Geleen kunnen gebruiken in combinatie met wat ik de laatste 2 jaren geleerd heb. Het aangeboden onderwerp leek mij onmiddellijk het meest interessante van de mogelijke keuzes en ben dankbaar voor het feit dat ik deze masterproef heb mogen uitvoeren. Het feit dat chemie en programmeren in MATLAB gecombineerd werden vond ik boeiend. Ook heb ik dankzij deze masterproef mij kunnen verdiepen in een onderwerp dat ik heel interessant vind, namelijk de regeltechniek.

Deze masterproef is in de eerste plaats bestemd als achtergrond voor studenten die les krijgen rond PID-regeling en MPC. Principes in de regeltechniek worden aangehaald en methoden van o.a. instellen van een PID-regelaar worden uitgelegd. Daarna worden deze toegepast op concrete voorbeelden die gradueel complexer worden doorheen het verhaal. Door gebruik te maken van MATLAB en Simulink is het ook voor studenten eenvoudiger om de zaken te visualiseren. De bedoeling is dat de gerealiseerde modellen in een educatieve context gebruikt kunnen worden waarbij multidisciplinair gewerkt wordt. Zo kan een toekomstig ingenieur automatisatie in contact komen met een chemisch proces en kan een toekomstig ingenieur chemie via een chemisch proces in contact komen met automatisatie.

De belangrijkste persoon tijdens mijn masterproef was prof.dr.ir. Filip Logist. Hij was mijn promotor bij de onderzoeksinstelling waar ik de masterproef heb uitgevoerd bij het departement chemische ingenieurstechnieken van de KULeuven. Tijdens de masterproef heeft hij kennis omtrent de regeltechniek aangereikt. Dankzij het vele computerwerk was een flexibel en efficiënt werkschema mogelijk wat financieel en praktisch gezien ook wel een factor die mijn keuze voor een deel mee bepaald heeft. Hem zou ik willen bedanken voor alle tijd en moeite die hij mee geïnvesteerd heeft in deze masterproef. Daarnaast zou ik prof.dr.ir Leen Thomassen willen bedanken. Zij was mijn schoolpromotor en heeft in die hoedanigheid veel tips gegeven en geholpen door teksten na te lezen.



# Inhoudsopgave

Voorwoord.....	1
Lijst van tabellen .....	5
Lijst van figuren.....	7
Abstract.....	9
Abstract in English.....	11
1 Inleiding .....	13
1.1 Situering .....	13
1.2 Probleemstelling.....	14
1.3 Doelstellingen.....	14
2 Literatuurstudie.....	17
2.1 Regeltechniek.....	17
2.2 Klassieke regeling: PID-regelaar .....	19
2.2.1 Wat is een PID-regelaar?.....	19
2.2.2 Eigenschappen van een PID-regelaar .....	19
2.2.3 Instellen van een PID-regelaar .....	21
2.2.4 Root Locus.....	24
2.3 Modelgebaseerde regeling: MPC-regelaar .....	25
2.3.1 Wat is een MPC-regelaar?.....	26
2.3.2 Eigenschappen van een MPC-regelaar .....	32
3 Materialen en methode.....	35
3.1 Software.....	35
3.1.1 MATLAB.....	35
3.1.2 Simulink.....	37
3.2 Opstellen van procesmodellen.....	40
3.2.1 Lineaire modellen in continue tijd.....	40
3.2.2 Lineaire modellen in discrete tijd .....	41
3.2.3 Niet-lineaire modellen in continue tijd.....	42
3.2.4 Opstellen van modellen voor procesregeling.....	43
3.3 Gevalstudies.....	44
3.3.1 Lineaire tank.....	44
3.3.2 Niet-lineaire tank .....	46
3.3.3 CSTR.....	47
4 Resultaten: Regeling van het niveau in een opslagvat.....	57
4.1 Lineaire tank.....	57
4.1.1 Geen regeling, open lus.....	57
4.1.2 Klassieke regeling: PID .....	57
4.1.3 Geavanceerde procescontrole: MPC.....	66
4.2 Niet-lineaire tank.....	72
4.2.1 Klassieke regeling: PID .....	72
4.2.2 Geavanceerde procescontrole: MPC.....	77
5 Resultaten: Regeling van een wandgekoelde chemische reactor.....	83

5.1	Klassieke regeling: PID .....	83
5.2	Geavanceerde procescontrole: MPC .....	94
6	Conclusie en blik vooruit.....	109
7	Bibliografie.....	111
8	Bijlagen .....	113
8.1	Simulink lineaire tank.....	113
8.2	ODE lineaire tank.....	113
8.3	S-functie lineaire tank.....	114
8.4	MPC-model lineaire tank.....	116

## Lijst van tabellen

Tabel 1: PID-instelparameters op basis van gesloten lus.....	22
Tabel 2: PID-instelparameters op basis van relais.....	22
Tabel 3: Instelparameters op basis van een open lus systeem.....	24
Tabel 4: PID-instellingen lineaire tank op basis van open lus.....	61
Tabel 5: Instellingen PID op basis van relais.....	64
Tabel 6: Instellingen MPC simulaties lineaire tank.....	68
Tabel 7: Overzicht gevallen MPC lineaire tank.....	68
Tabel 8: PID-instellingen niet-lineaire tank op basis van open lus.....	72
Tabel 9: Instellingen MPC simulaties niet-lineaire tank.....	77
Tabel 10: Overzicht gevallen MPC niet-lineaire tank.....	78
Tabel 11: Totaaloverzicht casussen CSTR: PID.....	85
Tabel 12: PID-instellingen CSTR op basis van open lus.....	85
Tabel 13: PID-instellingen CSTR 330K op basis van gesloten lus.....	86
Tabel 14: Schematisch overzicht van de casussen die worden uitgevoerd voor wat betreft externe verstoringen.....	86
Tabel 15: PID-instellingen CSTR op basis van gesloten lus.....	90
Tabel 16: PID-instellingen CSTR na trial and error.....	91
Tabel 17: Totaaloverzicht casussen CSTR: PID.....	95





## Lijst van figuren

Figuur 1: Schematisch overzicht van de elementen van een regelkring.....	18
Figuur 2: Uitleg PID: P-actie.....	20
Figuur 3: Uitleg PID: I-actie.....	20
Figuur 4: Uitleg PID: D-actie.....	21
Figuur 5: Schema van een regelkring met relais.....	23
Figuur 6: open lus respons: constructie om doortijd en processnelheid te achterhalen, naar (Liptak, 2006).....	23
Figuur 7: Root locus plot van een CSTR met reactortemperatuur 330K beschouwd voor 2 conversies (Luyben, 2007).....	25
Figuur 8: Schematische voorstelling van een tank.....	28
Figuur 9: MPC: uitleg voorbeeld: stabiel proces.....	29
Figuur 10: MPC: uitleg voorbeeld: aanleggen van stap op $k+3t$ .....	30
Figuur 11: MPC: uitleg voorbeeld: eerste regelactie aan proces.....	31
Figuur 12: MPC: uitleg voorbeeld: stap nieuwe ingestelde waarde.....	32
Figuur 13: Overzicht MATLAB 2014.....	36
Figuur 14: Overzicht Simulink.....	39
Figuur 15: Schematische weergave van de tank.....	45
Figuur 16: Schema van de CSTR.....	47
Figuur 17: Schematische weergave van de temperatuursensor.....	53
Figuur 18: Respons open lus: geen regelaar.....	57
Figuur 19: Schema simulink: simulatie PID.....	58
Figuur 20: Respons met PID zonder initiële integrator.....	58
Figuur 21: Respons met PID met initiële integrator.....	59
Figuur 22: Instellingen van de PID in Simulink.....	60
Figuur 23: Respons lineaire tank open lus met constructie voor instelregels.....	61
Figuur 24: Respons lineaire tank: enkel P-actie op basis van open lus.....	62
Figuur 25: Respons lineaire tank: PI-actie op basis van open lus.....	62
Figuur 26: Blokschema relaistuning in Simulink.....	63
Figuur 27: Instellingen Relais.....	63
Figuur 28: Regelactie bij de lineaire tank door relais.....	64
Figuur 29: Respons systeem: PID-instelling op basis van relais.....	65
Figuur 30: Respons systeem: PID-instelling op basis van relais: instellingen zonder overshoot.....	66
Figuur 31: Model in Simulink: Lineaire tank.....	66
Figuur 32: Respons systeem lineaire tank: MPC geval 1.....	69
Figuur 33: Respons systeem lineaire tank: MPC geval 2.....	70
Figuur 34: Respons systeem lineaire tank: MPC geval 3.....	70
Figuur 35: Respons systeem lineaire tank: MPC geval 4.....	71
Figuur 36: Respons niet-lineaire tank open lus met constructie voor instelregels.....	73
Figuur 37: Respons lineaire tank met PID: wijziging ingestelde waarde, enkel P-actie ingesteld.....	73
Figuur 38: Respons lineaire tank met PID: wijziging ingestelde waarde: PI-actie.....	74
Figuur 39: Regelactie bij de niet-lineaire tank door relais.....	75
Figuur 40: Instellingen PID op basis van relais.....	75
Figuur 41: Respons systeem: PID-instellingen op basis van relais.....	76
Figuur 42: Respons systeem: PI-instellingen op basis van relais.....	76
Figuur 43: Respons systeem niet-lineaire tank: MPC geval 1.....	78

Figuur 44: Respons systeem niet-lineaire tank: MPC geval 2.....	79
Figuur 45: Respons systeem niet-lineaire tank: MPC geval 3.....	80
Figuur 46: Respons systeem niet-lineaire tank: MPC geval 4.....	80
Figuur 47: Model in Simulink: CSTR .....	84
Figuur 48: Respons CSTR 330K open lus met constructie voor instelregels .....	86
Figuur 49: Respons CSTR 330K: PID gesloten lus: stap in gewenste waarde reactortemperatuur .....	87
Figuur 50: Respons koelwater CSTR 330K: PID gesloten lus: stap in gewenste waarde reactortemperatuur .....	87
Figuur 51: Respons CSTR 330K: PID gesloten lus: stap in voedingsdebiet .....	88
Figuur 52: Respons koelwater CSTR 330K: PID gesloten lus: stap in voedingsdebiet .....	88
Figuur 53: Respons CSTR 330K: PID gesloten lus: stap in voedingstemperatuur reagens.....	89
Figuur 54: Respons koelwater CSTR 330K: PID gesloten lus: stap in voedingstemperatuur .....	89
Figuur 55: Respons CSTR 350K: PID gesloten lus: instellingen voor CSTR 330K.....	90
Figuur 56: Respons CSTR 350K: PID gesloten lus instellingen.....	91
Figuur 57: Respons CSTR 350K: PID met instellingen na <i>trial and error</i> .....	92
Figuur 58: Respons CSTR 350K: PID met instellingen na <i>trial and error</i> : koelwaterrespons .....	92
Figuur 59: Respons reactortemperatuur CSTR 350K op stap voedingsdebiet .....	93
Figuur 60: Respons koelwaterdebiet CSTR 350K op stap voedingsdebiet.....	93
Figuur 61: Respons reactortemperatuur CSTR 350K op stap voedingstemperatuur .....	94
Figuur 62: Respons reactortemperatuur CSTR 350K op stap voedingstemperatuur .....	94
Figuur 63: Temperatuurrespons met MPC van de CSTR zonder tijdsvertraging van de temperatuursensor .....	96
Figuur 64: Koelwaterrespons met MPC van CSTR zonder tijdsvertraging van de temperatuursensor .....	96
Figuur 65: Temperatuurrespons MPC 330K: stap in ingestelde waarde: 330 naar 335K.....	97
Figuur 66: Koelwaterrespons MPC 330K: stap in ingestelde waarde: 330 naar 335K.....	98
Figuur 67: Temperatuurrespons MPC 330K: stap in voedingsdebiet	Figuur 68: Koelwaterrespons MPC 330K: stap in voedingsdebiet .....
Figuur 68: Koelwaterrespons MPC 330K: stap in voedingsdebiet .....	
Figuur 69: Temperatuurrespons MPC 330K: stap in voedingstemperatuur .....	100
Figuur 70: Koelwaterrespons MPC 330K: stap in voedingstemperatuur.....	100
Figuur 71: Temperatuurrespons MPC 350K: stap in ingestelde waarde van 350 naar 355K....	101
Figuur 72: Koelwaterrespons MPC 350K: stap in ingestelde waarde van 350 naar 355K.....	102
Figuur 73: Temperatuurrespons MPC 350K: stap in voedingsdebiet.....	102
Figuur 74: Koelwaterrespons MPC 350K: stap in voedingsdebiet.....	103
Figuur 75: Temperatuurrespons MPC 350K: stap in voedingstemperatuur .....	103
Figuur 76: Koelwaterrespons MPC 350K: stap in voedingstemperatuur.....	104
Figuur 77: Temperatuurrespons MPC 330K met 2 variabelen: stap op temperatuur en concentratie.....	105
Figuur 78: Koelwaterrespons MPC 330K met 2 variabelen: stap op temperatuur en concentratie .....	105
Figuur 79: Concentratierespons MPC 330K met 2 variabelen: stap op temperatuur en concentratie.....	106
Figuur 80: Voedingsdebietrespons MPC 330K met 2 variabelen: stap op temperatuur en concentratie.....	106

## Abstract

Een geschikte procesregeling is onmisbaar voor de veilige en efficiënte werking van industriële chemische installaties. Klassiek wordt PID-regeling (proportioneel, integrerend en differentiërend) gebruikt. Deze is populair, goed gekend en eenvoudig, maar heeft tekortkomingen. Er is maar 1 gemanipuleerde variabele en het kan niet om met beperkingen op (proces-)variabelen. Geavanceerde modelgebaseerde regeltechniek (Model Predictieve Controle of MPC) kan een oplossing zijn voor tekortkomingen van PID-regeling. MPC werkt met een procesmodel en houdt rekening met beperkingen in het proces en kan met meerdere gemanipuleerde variabelen tegelijk overweg.

In deze masterproef zijn PID- en MPC- regelschema's uitgewerkt om op praktische wijze in een educatieve context te gebruiken. De toepasbaarheid van beide regelmethoden werd onderzocht aan de hand van verschillende toepassingen uit de chemische industrie, i.e., tanks met in- en uitstroom en een CSTR-reactor. De complexiteit van de beschouwde processen neemt gradueel toe. Van een lineair proces met één te regelen procesvariabele en één regelingang naar een niet-lineair proces met meerdere procesvariabelen en regelingangen. De 3 beschouwde gevallen zijn uitgewerkt in het softwarepakket MATLAB en Simulink waarbij de code, die voor de betreffende simulaties nodig is, overzichtelijk wordt opgeleverd. De voor- en nadelen van PID en MPC komen naar voren en worden besproken aan de hand van resultaten van simulaties. Uit de resultaten blijkt dat MPC een stap vooruit is ten opzichte van PID.



## Abstract in English

An appropriate process control is crucial for safe and efficient production with industrial chemical installations. The PID-controller (proportional, integrative and derivative) is a commonly used control algorithm. It is very popular, well known and simple but has imperfections. It can manipulate only 1 variable and it can't take limitations of certain process variables into account. A solution for these shortcomings can be advanced model-based control (Model Predictive Control, MPC). MPC works with a model of the process it controls and can take limitations into account and can work with multiple manipulated variables.

In this masterthesis, PID and MPC are used in control schemes for practical usage in an educational context. The suitability for both control methods are investigated by different applications in chemical industry, i.e., tanks with in- and outflow and a CSTR-reactor. The complexity of these processes increase gradually. From a linear process with one manipulated variable and one controlled variable to a non-linear process with multiple controlled and manipulated variables. The 3 considered cases are elaborated in MATLAB and Simulink, the code for the considered simulations is delivered in a clarifying manner. The pros and cons of PID and MPC come forward and are discussed by the results of the simulations. From these results it appears that MPC is definitely a step forward in comparison with PID.



# 1 Inleiding

## 1.1 Situering

De chemische procesindustrie streeft ernaar om producten op een optimale manier op industriële schaal te produceren, waarbij een goede afweging gemaakt moet worden tussen verschillende factoren zoals kostprijs, energie-efficiëntie en omzetting. Een productieproces moet veilig en efficiënt verlopen ondanks uitwendige verstoringen en interne fluctuaties. Deze uitwendige verstoringen bestaan bijvoorbeeld uit veranderingen in bepaalde gebruiksstromen zoals stoom en koelwater. Interne fluctuaties zijn het gevolg van veranderingen aan instellingen van het proces om dit optimaler te laten verlopen. (William, L. L., 2007)

Meestal wordt tijdens het ontwerp van een proces gekeken naar een chemisch proces als zijnde een destillatiekolom, of een CSTR-reactor. In de praktijk is dit niet mogelijk. De destillatiekolom of CSTR-reactor vormt immers een deel van een groter geheel. Binnen zo'n groot geheel is het een utopie dat een bepaalde stroom constant blijft. Doordat stromen van voeding, stoom e.d. fluctueren, is het belangrijk te weten hoe een onderdeel van een groter chemisch proces reageert op deze fluctuaties. Dit is de dynamiek van dat onderdeel van een proces. (William, L. L., 2007)

Om deze verstoringen op te vangen is een gepaste procesregeling noodzakelijk. Regeltechniek wordt gebruikt om een proces binnen bepaalde grenzen te regelen naar gewenste waarden voor bijvoorbeeld temperatuur, conversie,... . Om deze regelsystemen te testen, de juiste instellingen van parameters te achterhalen en te bepalen of een proces stabiel kan zijn, doet men beroep op modellering. Het is belangrijk om vooraf te weten of het beschouwde proces uiteindelijk wel uit te voeren is en onder de gegeven condities niet stilvalt of een reactie-ontsporing optreedt. Een reactie-ontsporing is een exotherme reactie waarbij door onvoldoende koeling de temperatuur verder stijgt. De stijgende temperatuur zorgt ervoor dat de reactie nog sneller gaat en er nog meer warmte vrijkomt waardoor de temperatuur verder stijgt. Hierdoor kan er bijvoorbeeld faling van de apparatuur optreden. (Logist, F., 2013)

Door gebruik te maken van computermodellen wordt op een veilige en goedkope manier nagegaan welke invloed bepaalde ingrepen op een proces hebben. Voor het opstellen van deze modellen gebruikt men o.a. energie-, massa-, en componentenbalansen die het proces wiskundig beschrijven. De constanten die nodig zijn in de modellen moeten gekend zijn, zoals bijvoorbeeld de activatie-energie van een reactie. Na de modellering van een proces is het de bedoeling om een volledige regelkring te simuleren.

Regeltechniek en modellering gaan in deze masterproef hand in hand. Door een volledige regelkring op te bouwen kan de dynamiek van het complete systeem achterhaald worden. Dit systeem is het geheel van het chemische proces, samen met sensor, regelaar en regelorgaan. De dynamiek is hierbij belangrijk. Een chemisch proces staat immers nooit op zich, maar heeft een systeem rond zich hangen van o.a. sensoren, signaalverwerking en één of soms zelfs meerdere regelars. De dynamiek bepaalt of het systeem zich als geheel stabiel gedraagt, maar ook hoe het zich gedraagt. Zo kan afhankelijk van de dynamiek een keuze gemaakt worden voor

regelorganen, sensoren en instelling van de regelaar. Uit verder onderzoek wordt bepaald hoe de regelaar ingesteld dient te worden om variaties in het proces op te vangen. (Logist, F., 2013)

## 1.2 Probleemstelling

Uit bovenstaande is duidelijk dat regeltechniek in de chemische industrie belangrijk is. Ook is modellering en simulatie niet meer weg te denken dankzij de steeds krachtigere computers. Deze 3 facetten, nl. modellering, simulatie en regeltechniek, worden zelden samen gezien. Modellering en simulatie wordt bijvoorbeeld toegepast door chemische ingenieurs terwijl de regeltechniek vaak toegepast wordt door automatisatie-ingenieurs. Deze verschillende types van ingenieur hebben elk hun eigen disciplinaire achtergrond en kunnen zich beiden vinden in dit concept.

De bedoeling van deze masterproef is het dichter bij elkaar brengen van modellering, simulatie en regeltechniek zodat het ook voor studenten, die opgeleid worden tot zowel chemische als automatiseringsingenieurs, als geheel beter te begrijpen is. Omdat deze 3 facetten vaak apart worden gezien is de materie als leerstof vrij abstract en daar wordt getracht verbetering in te brengen. (Aström, K. J., 2001)

De beschouwde materie is vaak abstract voor studenten en daardoor moeilijk te doorgronden. Het resultaat van deze thesis zal concreet te gebruiken zijn in onderzoeksgebaseerd onderwijs. Door een duidelijke opbouw van een proces tot gesloten lus controlesysteem, en opbouw van een eenvoudig proces (lineair en 1 variabele) tot een meer geavanceerd proces (niet-lineair en multivariabel) is het de bedoeling dat concepten zoals root locus, instellen van PID-regelaars en meer geavanceerde technieken als MPC op een intuïtieve en gebruiksvriendelijke manier aangeleerd worden.

## 1.3 Doelstellingen

De bedoeling van dit onderzoek is te komen tot enkele volledig uitgewerkte gevallen, van model tot bespreking van de vergelijking tussen PID (proportionele integrerende en differentiërende) regelaar en MPC (modelgebaseerde predictieve regelaar) omtrent modellering en ontwerp van regelsystemen. Deze gevalstudies zijn een tank met lineaire klepkarakteristiek, een tank met niet-lineaire klepkarakteristiek en een wandgekoelde CSTR-reactor. De beschreven tanks zijn hierbij voorzien van een regelbaar toevoerdebiet en een uitstroomdebiet dat afhankelijk is van de vloeistofhoogte (lineaire klepkarakteristiek: uitstroom is recht evenredig met de vloeistofhoogte, niet-lineaire klepkarakteristiek: uitstroom is recht evenredig met de vierkantswortel van de vloeistofhoogte). Hierbij komen verschillende technieken aan bod, gaande van klassieke regeltechniek tot geavanceerde modelgebaseerde predictieve regeling. De eerste 2 gevalstudies dienen om een basis te vormen voor wat betreft modellering en simulatie met de computer. De derde gevalstudie omvat een model van een reële CSTR met wandkoeling waarin een enkelvoudige, irreversibele exotherme reactie plaatsvindt.

Men heeft voor deze concrete gevallen gekozen omdat ze een logische opbouw vormen doordat de modellen steeds complexer worden. Daarnaast wordt veelvuldig gebruik gemaakt van een tank om principes uit de regeltechniek uit te leggen, maar door gebruik van een CSTR met meerdere variabelen (concentratie, temperaturen van reactor en mantel) wordt een



uitgebreider voorbeeld gebruikt zodat dieper op de materie wordt ingegaan en een reëel en attractief voorbeeld gebruikt wordt.

Voor deze modellen wordt een gepast regelsysteem ontworpen zodat het proces dat gesimuleerd wordt binnen de gewenste werkingsvoorwaarden blijft. Ook wordt voor de besproken modellen gekeken of ze intrinsiek (on)stabiel zijn en welke impact het regelsysteem heeft op de (on)stabiliteit. Bij deze uitgewerkte gevallen gaat het om modellen en voorbeelden die gemaakt worden in MATLAB en Simulink. Het is de bedoeling dat de uitgewerkte bestanden, zowel deze met de opbouw van het model als het volledige procesmodel in Simulink, duidelijk becommentarieerd zijn. Op die manier kunnen de resultaten van deze masterproef later aangewend worden voor onderzoek gebaseerd onderwijs in een vak zoals regeltechniek of procescontrole.

Ook wordt gekeken naar de invloed van het veranderen van bepaalde regelparameters, zoals de instellingen van een PID-regelaar. Het onderzoek gaat verder dan de parameters op zich. Zo wordt gekeken naar modelgebaseerde predictieve regeling dat vandaag reeds in de industrie gebruikt wordt. Dit type regeling laat toe om het proces zo optimaal mogelijk te bedienen. De resultaten van deze toepassingen dienen uiteraard ook te worden opgenomen in de masterproef. Binnen MATLAB is er een MPC-regelaar uitgewerkt zonder de voor handen zijnde toolbox van MATLAB zelf zodat de werking van een MPC-regelaar voor een student beter te doorgronden is.



## 2 Literatuurstudie

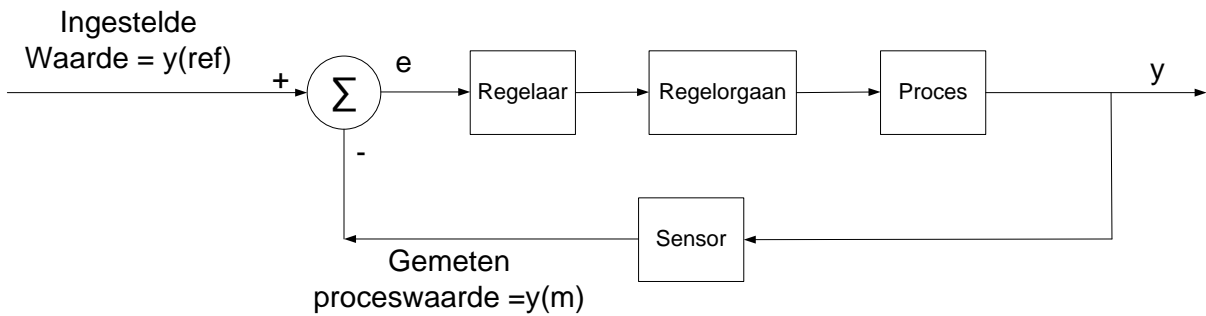
### 2.1 Regeltechniek

Regeltechniek omvat methoden die instaan voor het automatiseren van processen. Dit is nodig omdat grote (chemische) processen enkel mogelijk zijn indien ze geautomatiseerd (kunnen) worden. Hierbij wordt gebruik gemaakt van de controletheorie. Dankzij de controletheorie is het mogelijk om allerlei processen (fysisch, chemisch, mechanisch...) die wiskundig te beschrijven zijn, op een bepaalde manier te controleren en automatiseren.

In de chemische industrie bestaat een proces uit meerdere kleinere eenheden en eenheidsbewerkingen. Om met zo een uitgebreid proces een product te maken op een veilige manier, binnen de specificaties van de klant en liefst zo economisch mogelijk, is er regeltechniek nodig. In de regeltechniek kijkt men naar de eenheidsbewerkingen in het proces en gaat men na op welke manier men deze eenheid kan automatiseren. Een proces wil men immers zo optimaal mogelijk bedrijven. Dit betekent dat hier bij het ontwerp al rekening gehouden moet worden, maar zeker zo belangrijk is een degelijke controlestrategie die het proces stabiel houdt.

Een proceseenheid wordt verder een *proces* genoemd. Dit kan een warmtewisselaar zijn, of een CSTR e.d. Een proces omvat vaak meerdere *variabelen*, zoals debieten, temperaturen en drukken. Hier dient een onderscheid gemaakt te worden tussen variabelen die niet opgenomen worden in de automatisatie en variabelen die wel opgenomen worden in de automatisatie. De keuze van welke variabele wel wordt opgenomen in de automatisatie hangt af van de noodzaak. De variabelen die wel worden opgenomen in de automatisatie kunnen weer opgedeeld worden in de *gecontroleerde variabele(n)* en de *gemanipuleerde variabele(n)*. De gecontroleerde variabele is de variabele die gemeten wordt, de gemanipuleerde variabele is de variabele die door de regelaar wordt aangepast om de gecontroleerde variabele te regelen.

Om een proces te automatiseren zijn nog meer onderdelen nodig dan het proces alleen. Zo zijn er *sensoren* nodig om te weten in welke toestand het proces zich bevindt. Verder is een *regelorgaan* nodig, dat een bepaalde variabele aanpast of regelt. Het is de bedoeling dat de variabele die door het regelorgaan wordt aangepast impact heeft op de door de sensor gemeten variabele. Het *regel- of controleorgaan* wordt aangepast door een *regelaar*. Een regelaar krijgt een signaal binnen. Bij een *gesloten systeem*, zoals hieronder wordt weergegeven in Figuur 1: Schematisch overzicht van de elementen van een regelkring, is dit een zogeheten *fout*. Dit is de afwijking van een ingestelde waarde van de gemeten variabele, ten opzichte van de gemeten waarde van de variabele. Aan de hand van deze fout en het soort regelaar, zal de regelaar een bepaalde instelling doorsturen naar het regelorgaan. Op deze manier is een proces te automatiseren, waardoor het proces kan blijven werken zonder dat continu tussenkomst nodig is van de mens. Bij een gesloten lus zorgt de aanwezigheid van een sensor ervoor dat er een fout kan doorgegeven worden aan de regelaar.



Figuur 1: Schematisch overzicht van de elementen van een regelkring

Naast een gesloten lus bestaat er ook een *open lus systeem*. Bij een open lus vindt er geen terugkoppeling plaats van een gemeten waarde van het systeem naar de regelaar. Een voorbeeld hiervan is een timer die een licht aan en uit schakelt.

Het geheel van proces, sensor, ... gedraagt zich op een bepaalde dynamische manier. Een temperatuursensor zal bijvoorbeeld afwijken van de werkelijke temperatuur doordat er een temperatuurverschil nodig is om warmte toe te voeren of af te voeren en de sensor zelf in temperatuur te laten veranderen. De dynamiek van de aan- en afvoer van warmte aan de temperatuursensor is afhankelijk van het gebruikte materiaal en de warmteoverdrachtscoëfficiënt. Een regelklep is afhankelijk van de klepkarakteristiek.

Hoe een proces reageert op bepaalde veranderingen, kan gesimuleerd worden met de computer. De moeilijkheid hierbij ligt in het wiskundig beschrijven van ieder onderdeel. Het wiskundig beschrijven van een proces noemt men *modelleren*. Dit gebeurt met behulp van o.a. massa- en energiebalansen. Er dient wel op gelet te worden dat de simulaties die op deze manier gebeuren maar zo nauwkeurig zijn als het model zelf. Als het model te sterk afwijkt van het werkelijke proces, zal de simulatie ook niet meer bruikbaar zijn. De dynamiek van een systeem is afhankelijk van alle onderdelen die in een gesloten lus voorkomen. Dit gedrag is dus niet enkel afhankelijk van het proces, maar ook van de sensor of regelklep. Door de eigenschappen van al de onderdelen te modelleren is het mogelijk om een volledig proces te simuleren.

Met behulp van simulatie kan op een goedkope en snelle manier bepaald worden hoe het systeem (de uitgangen) reageert op afwijkingen en veranderingen bij de ingang. Dat maakt het mogelijk om de instellingen voor bijvoorbeeld een PID-regelaar te achterhalen in de simulatie en deze later toe te passen op de PID-regelaar van het echte proces. Zo zal het instellen van een PID-regelaar iets minder kritisch worden voor de chemische installatie, zoals hieronder duidelijk wordt.

Stabiliteit is in de regeltechniek een belangrijk gegeven en dient steeds nagestreefd te worden. Als een systeem niet stabiel is onder bepaalde condities kan dit destructieve gevolgen hebben. Een algemene definitie van stabiliteit is dat een systeem stabiel is als dit systeem naar een constant blijvende waarde geregeld kan worden. Hiermee wordt bedoeld dat eventuele fluctuaties vanzelf gedempt worden. Door onderzoek is gebleken dat wanneer er voor een controlesysteem een transferfunctie op te stellen is, de nulpunten van de noemer in het linker halfvlak moeten liggen van een reëel-imaginair assenstelsel. Wanneer dit het geval is, zal het systeem stabiel zijn en naar een stabiele waarde gaan na verloop van tijd.

In de regeltechniek bestaan er verschillende soorten regelaars. De PID-regelaar, een klassieker in de industrie, wordt veelvuldig gebruikt. Daarnaast is er door tekortkomingen van de PID-regelaar een opmars bezig van geavanceerde procesregeling. Modelgebaseerde regeling is hier een voorbeeld van. Beide algoritmen worden uit de doeken gedaan, samen met voor- en nadelen en eigenschappen.

## 2.2 Klassieke regeling: PID-regelaar

### 2.2.1 Wat is een PID-regelaar?

Een veel gebruikt algoritme voor controleorganen is de *PID-regelaar*. PID-regelaar staat voor Proportionele Integrerende en Differentiërende regelaar. Voordelen van de PID-regelaar zijn het feit dat het een zeer goed gekend algoritme is dat reeds lange tijd gebruikt wordt en voor een grote veelzijdigheid aan processen toepasbaar is. Tekortkomingen uiten zich in het feit dat het niet altijd de beste oplossing is om een proces te controleren omdat een PID-regelaar te weinig informatie van het proces zelf gebruikt. De informatie van het proces wordt vaak grafisch bepaald waardoor fouten optreden. Een PID-regelaar kan ook niet overweg met het regelen van meerdere variabelen en met het opleggen van beperkingen.

Omdat niet ieder proces hetzelfde reageert op een regelactie, moet een PID-regelaar ingesteld worden voor het proces dat het regelt. Door in de simulatie een model te gebruiken, is het mogelijk om voor dat model een PID-regelaar in te stellen aan de hand van klassieke instelregels. De methoden die gebruikt kunnen worden zijn o.a. Ziegler-Nichols en Tyreus-Luyben. Deze 2 methoden worden toegepast in deze MP-thesis. (Skogestad, Probably the best simple PID tuning rules in the world, 2001) (MathWorks)

### 2.2.2 Eigenschappen van een PID-regelaar

Het PID-regelalgoritme werkt in een gesloten lus systeem en neemt de plaats van regelaar in in figuur 1. De sensor meet een bepaalde variabele in het proces. Dit is bijvoorbeeld een druk of temperatuur. Het som-blokje neemt het verschil tussen de ingestelde waarde en de gemeten waarde van de gemeten variabele. Door dit verschil, de fout "e" tussen de gewenste ingestelde en gemeten waarde, naar een regelaar te sturen wordt deze fout in de regelaar gebruikt om een bepaalde regelactie te genereren. In de regelaar wordt er volgens een regelalgoritme bepaald welke regelactie er dient te gebeuren. Bij een PID-regelaar zitten 3 regelacties in het algoritme, die in de naam zijn weergegeven. Een proportionele, een integrerende en een differentiërende actie zitten vervat in een PID-regelaar. Verder worden de eigenschappen van deze acties uit de doeken gedaan. In 2-1 staat  $u(s)$  voor de uitgang van de PID-regelaar.  $e$  is de fout die in de PID-regelaar binnenkomt ( $y_{sp} - y_m$ ). Verder zijn  $K_p$ ,  $\tau_i$  en  $\tau_d$  factoren die bepaald worden afhankelijk van het proces dat de PID-regelaar moet regelen.

$$u(s) = e * \left( K_p * \left( 1 + \frac{1}{\tau_i} + \tau_d \right) \right)$$

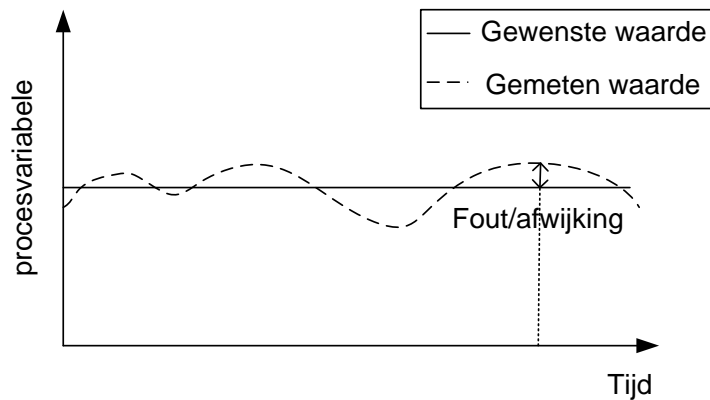
2-1

Deze vorm is de zogenaamde ideale vorm. Er bestaat ook een parallelle vorm van de PID-regelaar die er als volgt uitziet:

$$u(s) = e * \left( K_p + \frac{1}{\tau_i} + \tau_d \right)$$

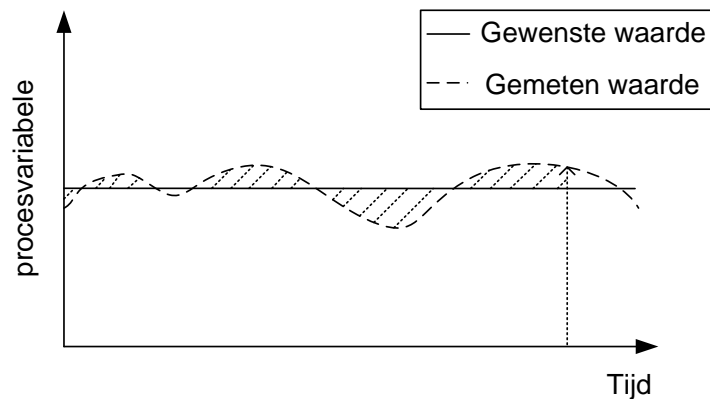
2-2

Welke vorm gebruikt wordt is belangrijk om weten bij het instellen van de PID-regelaar en kan aangepast worden in het PID-blok bij Simulink. Om de concepten achter de 3 acties die in de PID-regelaar vervat zitten worden deze stap voor stap overlopen. De P staat voor een *proportionele actie*. Dit is een versterkingsfactor die de fout die in het PID-regelaarblokje binnenkomt vermenigvuldigt met de hiervoor ingestelde waarde. Dit is het deel van de regelaar die naar het heden “kijkt”.



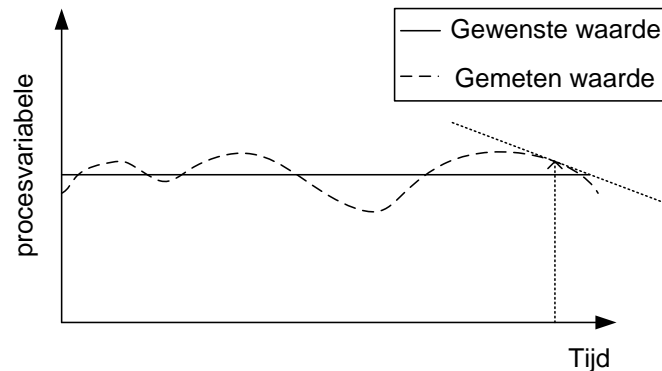
Figuur 2: Uitleg PID: P-actie

De 2<sup>e</sup> regelactie is de I-actie, de *integrerende actie*. Hier wordt de integraal genomen van de fout. Dit komt overeen met het oppervlak tussen de curve van de ingestelde waarde en de gemeten waarde. De integrerende actie wordt vergeleken met een blik op het verleden. Deze blik op het verleden kan oneindig teruggaan, of de regelaar wordt voorzien van een antiresetwindup. Deze functie zit in een PID-regelaar en kan indien nodig ingesteld worden. Het dient om, als de PID-regelaar in verzadiging gaat, de regelactie niet oneindig te laten veranderen en de integrator te ontlasten. Daardoor kan, zodra de PID niet meer in verzadiging is, de PID onmiddellijk regelen, zonder dat de negatieve fout eerst de “opgeslagen” positieve fout teniet moet doen. (Aström K. J., 2012)



Figuur 3: Uitleg PID: I-actie

De laatste regelactie van een PID-regelaar is de D-actie, of de *differentiërende actie*. Deze neemt ogenblikkelijk de afgeleide van de curve van de fout.



Figuur 4: Uitleg PID: D-actie

Afhankelijk van de noodzaak wordt in de PID-regelaar een bepaalde actie op 0 gezet worden waardoor bijvoorbeeld een P-, PI- of PD-regelaar ingesteld worden.

### 2.2.3 Instellen van een PID-regelaar.

#### Gesloten lus

Bij gesloten lus instelling maakt men gebruik van een gesloten systeem om de instelprocedure van de PID-regelaar uit te voeren. Voordeel hiervan is dat de dynamiek van alle onderdelen in het proces worden meegenomen in het bepalen van de PID-parameters.

Men gaat hierbij als volgt te werk: de PID-regelaar wordt ingesteld als P-regelaar. Deze P-actie of versterkingsfactor wordt in stapjes verhoogd tot een gelijkmatige oscillatie optreedt die niet uitdooft of sterker wordt. Aan de hand van deze versterkingsfactor  $K_U$ , en de periode  $P_U$  van de oscillaties zijn er regels om de instellingen van een PID-regelaar te bepalen. Het voordeel van deze manier van werken is dat door het feit dat de versterkingsfactor bepaald wordt met een gesloten lus, de dynamiek van het gehele systeem in rekening wordt gebracht. Dit betekent dat niet alleen de dynamiek van het proces wordt meegenomen, maar ook de dynamiek van de sensoren en signaalverwerking. Een nadeel aan deze manier van werken is dat de *ultieme versterkingsfactor* bepaald moet worden. Een ander bijkomend nadeel is het feit dat een proces dat beschouwd wordt voor de instelling van de regelaar meestal in de industrie niet los staat. Daardoor zullen de oscillaties die ontstaan bij het zoeken naar de ultieme versterkingsfactor te voelen zijn in het verdere proces stroomafwaarts. Dit kan zorgen voor product dat niet meer voldoet aan de specificaties van de klant. Het kan zelfs leiden tot het uitvallen van de installatie. De oscillatie die optreedt kan ook zo groot worden dat er bepaalde werkingsgrenzen overschreden worden waardoor de instellingen zelfs niet achterhaald kunnen worden. (Ruel, 2010)

Methoden die van gesloten lus instelling gebruik maken zijn de methoden van Ziegler-Nichols en Tyreus-Luyben (Luyben, 2007). De regels van Ziegler-Nichols geven voor veel systemen in de procesindustrie een te agressieve regelactie waardoor vaak *overshoot* en oscillaties optreden, hetgeen ongewenst is. Overshoot is het fenomeen dat optreedt wanneer de gecontroleerde procesvariabele tijdens het regelen naar een nieuwe gewenste waarde kortstondig voorbij deze

gewenste waarde schiet alvorens de nieuwe gewenste waarde stabiel te bereiken. Tyreus en Luyben hebben daarom voor PI-regelaars een andere instelling aanbevolen, nl.  $K_c=0.313*K_u$  en  $\tau_i=2.2*P_u$ . (Skogestad, Probably the best simple PID tuning rules in the world, 2001)

Tabel 1: PID-instelparameters op basis van gesloten lus

	$K_p$	$\tau_i$	$\tau_d$
P-regelaar	$0.5*K_u$		
PI-regelaar	$0.45*K_u$	$P_u/1.2$	
PID-regelaar	$0.48*K_u$	$P_u/1.6$	$P_u/10$

### Relais tuning

Naast de geslotenlusmethoden zoals Ziegler Nichols en Tyreus Luyben is het mogelijk om *relais-tuning* toe te passen. Ook op basis daarvan is het mogelijk om een PID-regelaar in te stellen. Bij relais-tuning gaat men als volgt te werk: men vervangt de regelaar door een relais die afhankelijk van de input die hij krijgt een hoge of lage output geeft (bijvoorbeeld wanneer de fout groter is dan 0: 100%, wanneer de fout kleiner is dan 0: 0% regelactie door het regelorgaan). De helft van het verschil tussen de hoge en de lage uitgangswaarde is een bepaalde amplitude B. Het systeem reageert hierop met een oscillatie met een bepaalde amplitude A. Verder zal er ook een periode waargenomen kunnen worden,  $P_U$ , waarmee het systeem tussen een hoge en lage waarde oscilleert. Aan de hand van A en B wordt  $K_U$  berekend zoals in 2-3. De aanbevolen instellingen worden weergegeven in Tabel 2.

$$K_U = \frac{4 * B}{\pi * A}$$

2-3

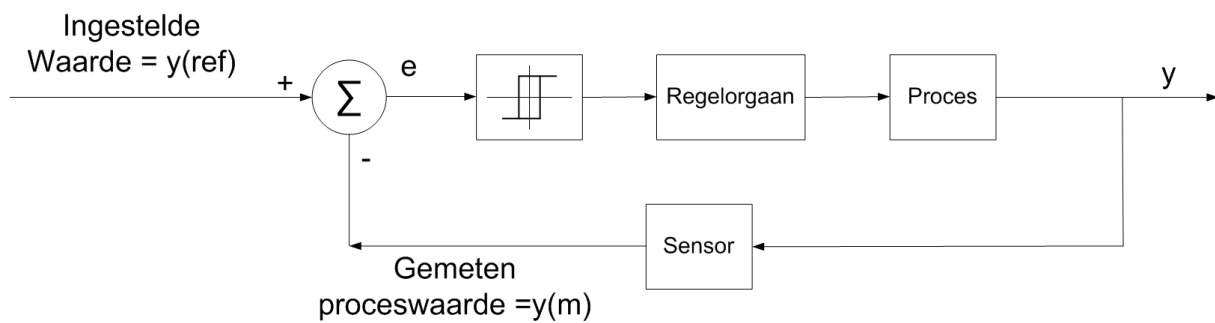
Tabel 2: PID-instelparameters op basis van relais

	$K_p$	$\tau_i$	$\tau_d$
Origineel	$0.6*K_U$	$P_U/2$	$P_U/8$
Matige overshoot	$0.33*K_U$	$P_U/2$	$P_U/3$
Geen overshoot	$0.2*K_U$	$P_U/2$	$P_U/3$

(Wilson, 2005)

Het schema van zo'n systeem wordt in Figuur 5 weergegeven.





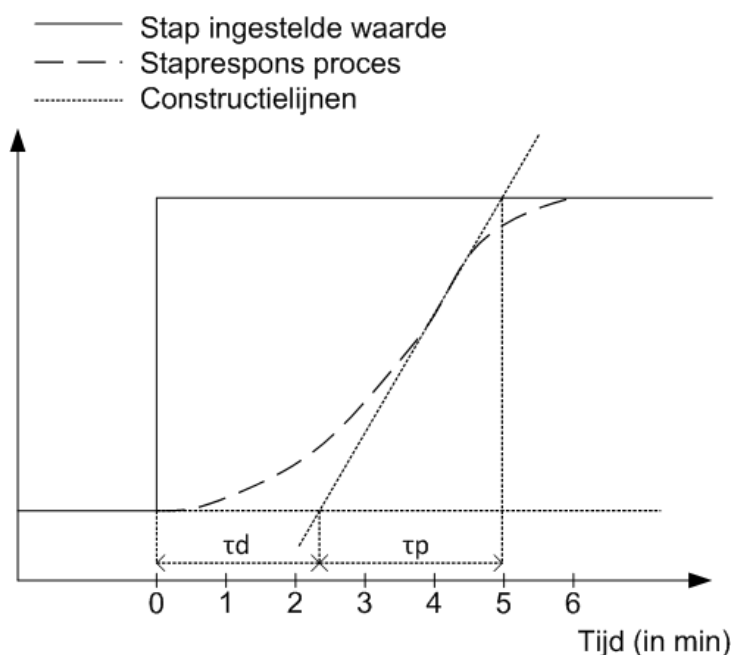
Figuur 5: Schema van een regelkring met relais

Een groot voordeel van relais-tuning is het feit dat het eenvoudig kan toegepast worden doordat de mogelijkheid soms in de PID-regelaar ingebakken zit. Verder treedt er maar een kleine procesverstoring op en heeft men deze met de keuze van de amplitude ook in de hand. (Wilson, 2005)

### Open lus

Het bepalen van deze parameters heeft uiteindelijk tot doel om een gesloten lus systeem te verkrijgen met een PID-regelaar die ingesteld is op basis van de parameters die bekomen worden met deze methode. De openlusmethode wordt meestal als eerste toegepast omwille van zijn eenvoud en omdat er maar een matige procesverstoring optreedt. Bij het bepalen van instellingen van de PID-regelaar in een open lus vertrekt men van het aanleggen van een stap in de gecontroleerde variabele en meet men de respons van het systeem op. Uit de grafiek daaruit volgt, de zogenaamde *reactiecurve*, worden waarden gehaald die kenmerkend zijn voor het systeem. Het gaat hierbij over de *dodetijd* en de *processnelheid*.

Met behulp van deze waarden worden de instellingen berekend die nodig zijn voor een PID-regelaar. De constructie ziet er als volgt uit:



Figuur 6: Open lus respons: constructie om dodetijd en processnelheid te achterhalen, naar (Liptak, 2006)

De berekeningen die hiermee gedaan worden zijn de volgende:

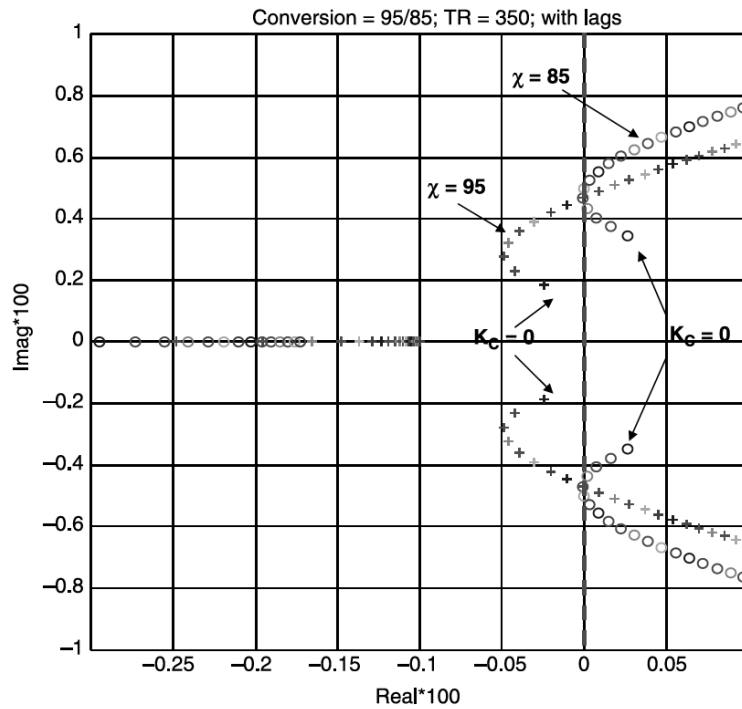
Als  $\tau_d$  kleiner is dan 1, wordt deze gelijk gesteld aan 1. Hierboven wordt  $\tau_d$  2,3 minuten. De waarde van  $\tau_p$  wordt afgelezen op de grafiek (tijdsverschil) waarna deze in seconden wordt gezet indien nodig. Voor bovenstaand voorbeeld wordt dit  $5-2,3=2,7$  minuten of 162 seconden. De waarde van  $K_p$  wordt berekend door de proceswaarde te delen door de output. In bovenstaand voorbeeld is dit 1. Deze waarden worden gebruikt in Tabel 3 om de waarden te berekenen die nodig zijn voor de instelling van de PID-regelaar. De besproken factoren  $\tau_d$ ,  $\tau_p$  en  $K_p$  beschrijven het proces zodat de PID "begrijpt" met welk proces het te maken krijgt. De gegevens kunnen gebruikt worden in Tabel 3, gebaseerd op Ziegler-Nichols: (Liptak, 2006)

Tabel 3: Instelparameters op basis van een open lus systeem

Regelaar	$K_p$	$\tau_i$	$\tau_d$
P	$0,30 \frac{\tau_p}{K_p \tau_d}$		
PD	$0,60 \frac{\tau_p}{K_p \tau_d}$		$0,50 \tau_d$
PI	$0,35 \frac{\tau_p}{K_p \tau_d}$	$1,20 \tau_p$	
PID	$0,60 \frac{\tau_p}{K_p \tau_d}$	$\tau_p$	$0,50 \tau_d$

#### 2.2.4 Root Locus

Verder is het belangrijk om het totale systeem van proces, sensor en regelaar te testen op stabiliteit. Hiermee wordt de gesloten lus-stabiliteit bedoeld. Wanneer het totale systeem niet stabiel is zal het gaan oscilleren en niet te controleren zijn. Het bepalen van de stabiliteit wordt bijvoorbeeld uitgevoerd met de methode van de root locus, waarbij de polen gezocht worden van de transferfunctie van de gehele gesloten lus. Wanneer deze polen een positief reëel deel hebben, zal het proces onstabiel zijn. Algemeen wordt genomen dat er op een grafische manier gekeken wordt hoe, door een parameter in het gesloten lus-systeem te veranderen, de nulpunten van de transferfunctie veranderen. De parameter die hier wordt toegepast is het veranderen van de versterkingsfactor in de P-regelaar, in onderstaande figuur  $K_U$  genoemd. Men laat een versterkingsfactor variëren van 0 tot een bepaalde waarde en kijkt hoe de polen en nulpunten van de transferfunctie zich gedragen. De versterkingsfactor vanaf waar er een positief reëel deel optreedt komt overeen met de  $K_U$  die gevonden wordt bij de instellingsregels van Ziegler-Nichols en Tyreus-Luyben. (Luyben, 2007)



Figuur 7: Root locus plot van een CSTR met reactortemperatuur 330K beschouwd voor 2 conversies (Luyben, 2007)

PID-regelaars zijn vrij eenvoudig en gebruiken weinig informatie van het proces zelf. Mede door dit nadeel en door de steeds krachtigere computers, is modelgebaseerde controle aan een opmars bezig. Deze controlestrategie heeft zijn oorsprong in de industrie omdat de klassieke regelstrategieën niet volstonden om bepaalde processen te regelen. (Maciejowski, 2000)

### 2.3 Modelgebaseerde regeling: MPC-regelaar

De naam *modelgebaseerde controle* geeft aan wat deze regelaar inhoudt. Het regelorgaan bevat een model van het proces dat het regelt. Daardoor is de regelaar in staat te voorspellen hoe het proces zal reageren op een controleactie en hoe een bepaalde nieuwe gewenste waarde zo snel en optimaal mogelijk bereikt kan worden. Dit heeft tot gevolg dat een nieuwe gewenste waarde sneller en beter te bereiken is dan met een PID-regelaar. Het stabiel bedrijven van een proces kan stabielier gemaakt worden dan met een PID-regelaar, omdat de voorspelling die gedaan wordt aan de hand van het model veel krachtiger is dan de ogenblikkelijke bijsturing die een PID-regelaar doet.

Modelgebaseerde controle staat voor een brede waaier aan mogelijkheden en algoritmen om een anticiperende regelaar te maken. MPC is daar 1 van. Het algoritme dat algemeen voorkomt bij modelgebaseerde controle wordt algemeen uitgelegd en gevisualiseerd aan de hand van een eenvoudig voorbeeld.

Vaak wordt in de praktijk als eerste laag in de regelhiërarchie een PID-regelaar toegepast die continu instaat voor de regeling van het proces. Daarboven wordt dan een modelgebaseerde regeling geplaatst die periodiek naar de huidige stand van het proces kijkt en rekt wat het proces in de toekomst zal doen of moet gaan doen (Aström K. &, 2001). Aan de hand van deze voorspelling doet de regelaar dan een regelactie. Modelgebaseerde controle is ook direct te gebruiken wanneer de rekenkracht voor handen is om de periode tussen 2 regelacties zo kort

mogelijk te maken zodat de *discrete* regeling een *continue* regeling benadert. (Maciejowski, 2000)

### 2.3.1 Wat is een MPC-regelaar?

De MPC-regelaar die bij deze masterproef gebruikt wordt heeft een lineair model nodig om te kunnen functioneren. Het voordeel hiervan is dat het algoritme eenvoudig blijft vanwege de lineariteit en het daaruit volgende superpositiebeginsel. In de praktijk kan het zijn dat het lineaire model niet voldoet en dan zal overgegaan moeten worden naar een niet-lineaire MPC. De vorm van model dat gebruikt wordt in de MPC is een discreet toestandsruimtemodel. Een discreet model omdat rekentijd vereist is tussen iedere regelstap. Een toestandsruimtemodel bestaat uit matrices en de matrixvorm laat eenvoudige uitbreiding van het model toe. Een toestandsruimtemodel ziet er in het algemeen uit als in 2-4, waarbij  $x$  de toestanden zijn,  $u$  de ingangsvARIABLEN,  $y$  de uitgangsvARIABLEN en  $A$ ,  $B$ ,  $C$  en  $D$  constanten:

$$\begin{aligned}\frac{dx}{dt} &= Ax + Bu \\ y &= Cx + Du\end{aligned}$$

2-4

En de discrete vorm van het toestandsruimtemodel ziet er als volgt uit:

$$\begin{aligned}x_{k+1} &= A_d x_k + B_d u_k \\ y_{k+1} &= C_d x_{k+1} + D_d u_{k+1}\end{aligned}$$

2-5

De term discreet duidt erop dat de MPC maar op bepaalde tijdstippen meet, waartussen een rekenactie en regelactie moet gebeuren. In bovenstaand discreet model 2-5 staat  $k$  voor de huidige tijd waarop de MPC zich bevindt en  $k+1$  staat voor de eerste stap die de MPC vooruit kijkt. Tijdens de rekenactie berekent de MPC op basis van de afwijking tussen gemeten en gewenste waarde een regeltraject voor een aantal tijdsstappen in de toekomst. Dit aantal tijdsstappen in de toekomst ligt vast door de voorspellingshorizon  $H_p$ . Na deze berekening wordt de regelactie van de eerste voorspelde stap overgenomen om als regelactie aan het proces aan te leggen. De tijd tussen 2 metingen wordt de staalnametijd of *sample time*  $t_s$  genoemd.  $H_p$  en  $t_s$  zijn 2 belangrijke factoren bij MPC.  $t_s$  is best vrij kort omdat de MPC anders niet goed kan reageren op verstoringen. De staalnametijd mag ook niet te kort zijn omdat de regelactie anders sneller wordt aangelegd dan het regelorgaan kan regelen. Veel hiervan hangt af van wat voor systeem wordt geregeld. Een traag systeem met een lange dodetijd reageert immers heel anders dan een snel systeem met geen of korte dodetijd.

De tijdsstappen worden in het algemeen aangegeven als  $k$  voor de tijd dat er nu gekeken wordt,  $k+1$  voor een tijdsstap vooruit,  $k+2$  voor 2 stappen vooruit enz. De notatie  $k-1$  etc. wordt dan weer gebruikt voor de tijdsstappen in het verleden aan te duiden. Voor iedere tijdsstap kan er dan een nieuwe berekening uitgevoerd worden voor de toestand. Dit ziet er als volgt uit:

- 1<sup>e</sup> tijdsstap:

$$\begin{aligned}x_{k+1} &= A_d x_k + B_d u_k \\ y_{k+1} &= C_d x_{k+1} + D_d u_{k+1} = C_d (A_d x_k + B_d u_k) + D_d u_{k+1}\end{aligned}$$

2-6

- 2<sup>e</sup> tijdsstap:

$$\begin{aligned}x_{k+2} &= A_d x_{k+1} + B_d u_{k+1} = A_d (A_d x_k + B_d u_k) + B_d u_{k+1} \\y_{k+2} &= C_d (A_d (A_d x_k + B_d u_k) + B_d u_{k+1}) + D_d u_{k+2}\end{aligned}$$

2-7

Voor iedere tijdsstap gaat dit verder tot aan het einde van de voorspellingshorizon waar de tijd  $k+n$  is. Algemeen worden  $x_{k+n}$  en  $y_{k+n}$  dan:

$$x_{k+n} = A_d^n x_k + A_d^{n-1} B_d u_k + A_d^{n-2} B_d u_{k+1} + \dots + A_d B_d u_{k+n-2} + B_d u_{k+n-1}$$

$$y_{k+n} = C_d A_d^n x_k + C_d (A_d^{n-1} B_d u_k + A_d^{n-2} B_d u_{k+1} + \dots + A_d B_d u_{k+n-2} + B_d u_{k+n-1}) + D_d u_{k+n}$$

2-8

Voor deze toestand op tijd  $k+n$  te kunnen berekenen moet de toestand op tijdstip  $k$  gekend zijn. Om deze formules in een regeling te kunnen gebruiken moet optimalisatie gebeuren waarbij  $x$  gemeten wordt en aan de hand daarvan een  $u$  berekend. Algemeen komt deze optimalisatie neer op het uitvoeren van volgende berekening:

$$J = \sum_{i=1}^{H_P} (y_i - y_{i,ref}) Q (y_i - y_{i,ref}) + (\Delta u_i - \Delta u_{i,ref}) R (\Delta u_i - \Delta u_{i,ref})$$

2-9

$J$  wordt de kostfunctie genoemd. De factoren  $Q$  en  $R$  zijn matrices die als instellingen in de MPC ingevoerd worden. Het komt er in feite op neer dat  $Q$  een compensatie geeft wanneer meerdere variabelen gemeten worden die van eenheid en grootteorde verschillen.  $R$  kan dan weer ingesteld worden als een vorm van afstraffing voor het aanleggen van teveel grote regelacties achter elkaar. De kostfunctie maakt nog steeds gebruik van factoren uit het model:

$$\begin{aligned}x_{k+1} &= A_d x_k + B_d u_k \\y_{k+1} &= C_d x_{k+1} + D_d u_{k+1}\end{aligned}$$

2-10

En aan beperkingen die opgelegd kunnen worden zoals:

$$x_{min} \leq x_k \leq x_{max}$$

$$u_{min} \leq u_k \leq u_{max}$$

$$\Delta u_{min} \leq \Delta u_k \leq \Delta u_{max} \text{ met } \Delta u_k = u_k - u_{k-1}$$

2-11

Het probleem dat zich eigenlijk stelt is het feit dat er een optimale controleactie bepaald moet worden om de ingestelde waarde van  $y$  zo snel mogelijk te behalen. Hierbij dient een functie geminimaliseerd te worden. Deze functie is de fout tussen de gewenste  $y$  aan de ene kant en de gemeten  $y$  en voorspelde  $y$  aan de andere kant en is hetgeen in de kostfunctie  $J$  beschreven wordt (Wang, 2009). Deze probleemstelling kan omschreven worden naar een kwadratisch optimaliseringsprobleem. Binnen MATLAB bestaat een functie om deze op te lossen, nl. quadprog. Deze functie lost een probleem op van de volgende vorm, een kwadratisch programma:

$$\min\left(\frac{1}{2}y'.H.y + f'.y\right)$$

2-12

Bij het oplossen van deze functie kan rekening worden gehouden met beperkingen zoals:

$$G_{eq}.y = g_{eq}$$

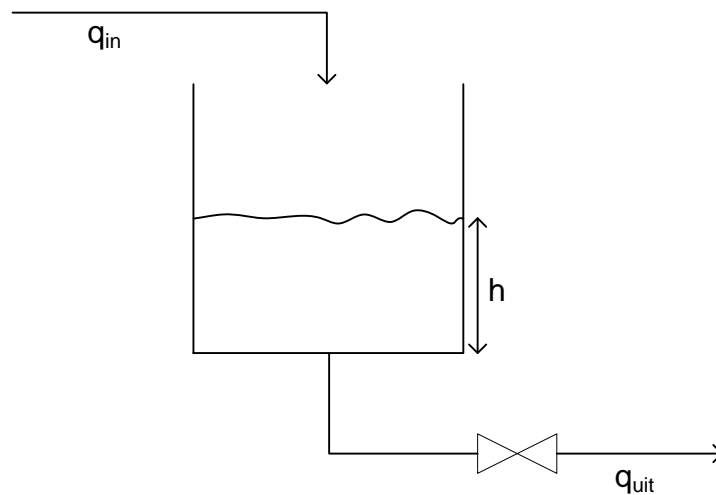
$$G_{eq}.y \leq g_{eq}$$

$$y_{min} \leq y \leq y_{max}$$

2-13

In deze formule staat H voor de hessiaan-matrix die staat voor de kromming van de functie bij het verloop naar de volgende tijdstappen.  $f'$  staat voor de gradient van de functie, deze geeft de richting aan van de steilste helling. (Maciejowski, 2000)

Het gemakkelijkste om MPC uit te leggen is door het te visualiseren aan de hand van opbouwende grafieken en een concreet voorbeeld. Als voorbeeld wordt een tank beschouwd met een constante doorsnede waar een toevoer en een afvoer aan is gemaakt. Het afvoerdebiet is afhankelijk van de vloeistofhoogte, het toevoerdebiet is een gemanipuleerde variabele en de hoogte van het vloeistofniveau is de gecontroleerde variabele. De debieten waarmee gewerkt wordt zijn volumetrische debieten aangezien de densiteit constant wordt beschouwd. Figuur 8 toont een schematische voorstelling van dit proces.



Figuur 8: Schematische voorstelling van een tank

De hoogte van de tank is afhankelijk van het toegevoerde en afgevoerde debiet volgens massabalans 2-14:

$$\frac{dh}{dt} = -\frac{h}{AR} + \frac{q_{in}}{A}$$

2-14

Hierbij zijn  $A$  en  $R$  constanten, namelijk respectievelijk het oppervlak van de tank (dwarsdoorsnede) en de weerstand van de klep in de uitstroom. In dit voorbeeld wordt voor  $A$   $0,5\text{m}^2$  aangenomen en  $R$  wordt verondersteld  $2\text{s/m}^2$  te zijn. Onder stabiele condities is:

$$\frac{dh}{dt} = 0 = -\frac{h}{AR} + \frac{q_{in}}{A}$$

2-15

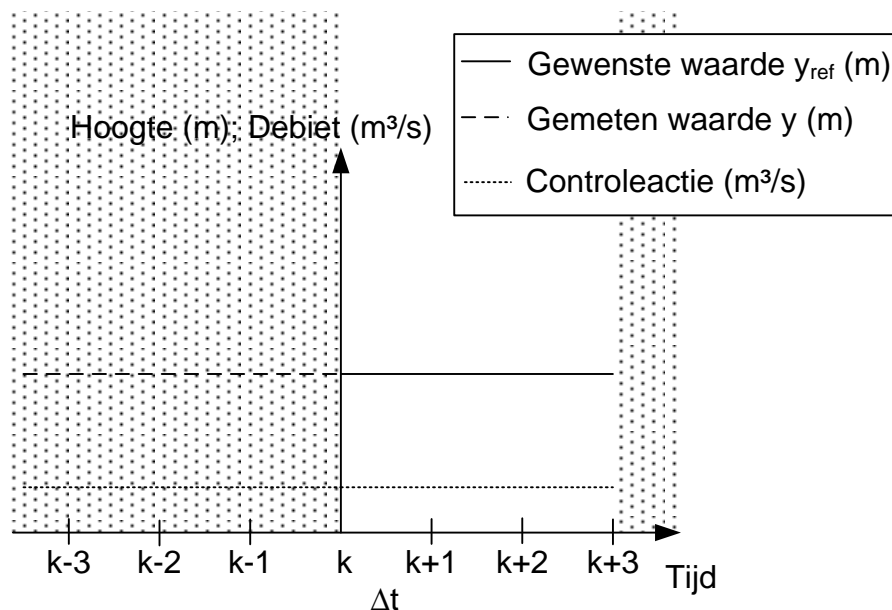
Wat leidt tot:

$$h = q_{in}R$$

2-16

Wanneer vergelijking 2-16 wordt opgelost voor een stabiele vloeistofhoogte van 2 meter betekent dit dat  $q_{in}$   $1\text{m}^3/\text{s}$  moet zijn. Ter controle worden al deze gegevens in de lineaire differentiaalvergelijking ingevuld en dit geeft 0.

In Figuur 9 wordt de toestand weergegeven op het moment  $k$  van het proces dat hierboven werd beschreven met een gemeten waarde van 2m vloeistofhoogte. De gemeten en gewenste waarden zijn de vloeistofhoogte  $h$ , de controleactie is de waarde van het debiet  $q_{in}$  die reeds berekend werd.



Figuur 9: MPC: uitleg voorbeeld: stabiel proces

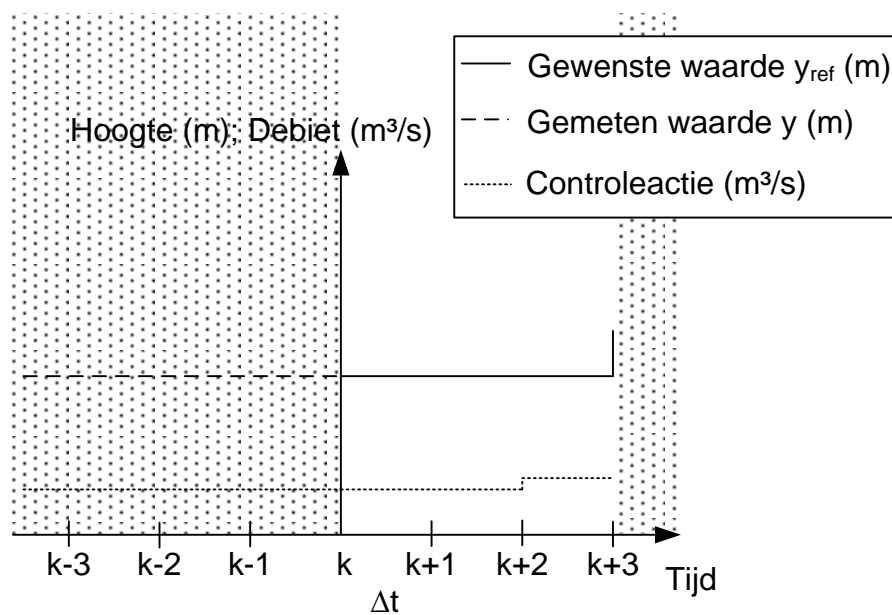
Deze toestand op moment  $k$  is de input die de regelaar krijgt. Wanneer in de toekomst de gewenste waarde hetzelfde blijft, blijft ook de regelactie constant zoals hierboven wordt weergegeven.

De sterkte in MPC ligt in 3 zaken:

- Met meerdere variabelen werken;

- Mogelijk om een proces te regelen aan de hand van variaties in de toekomst zullen gebeuren: stabielere bedrijven van een proces wordt mogelijk waardoor er dichterbij de werkinggrenzen geregeld kan worden;
- Werken met beperkingen in bepaalde variabelen. Zo is het bijvoorbeeld mogelijk om beperkingen aan te leggen in maximale stap die een debiet kan afwijken ten opzichte van de waarde in de vorige stap (een regelklep heeft een bepaalde traagheid waardoor niet onmiddellijk een bepaalde debietverandering kan aangelegd worden), of een debiet dat niet negatief kan zijn.

Er wordt zoals te zien is in Figuur 10 op het beschouwde proces een stap aangelegd van 2 meter vloeistofhoogte naar 3 meter. Op dit moment bevindt zich deze stap nog op  $k+3$ . Dit is nog net binnen de voorspellingshorizon die aangegeven wordt tussen de grijze velden, waardoor er ook een eerste verandering in regelactie zal optreden. Dit betekent dat voordat de werkelijke verandering in gewenste waarde wordt aangelegd aan het proces, reeds een verandering wordt voorgesteld door de MPC. Iedere stap die nu gemaakt wordt vooruit, blijft  $k$  het moment waarop gekeken wordt, een stap vooruit betekent dat de volgende  $k$  overeenkomt met  $k+1$  uit de voorgaande figuur.

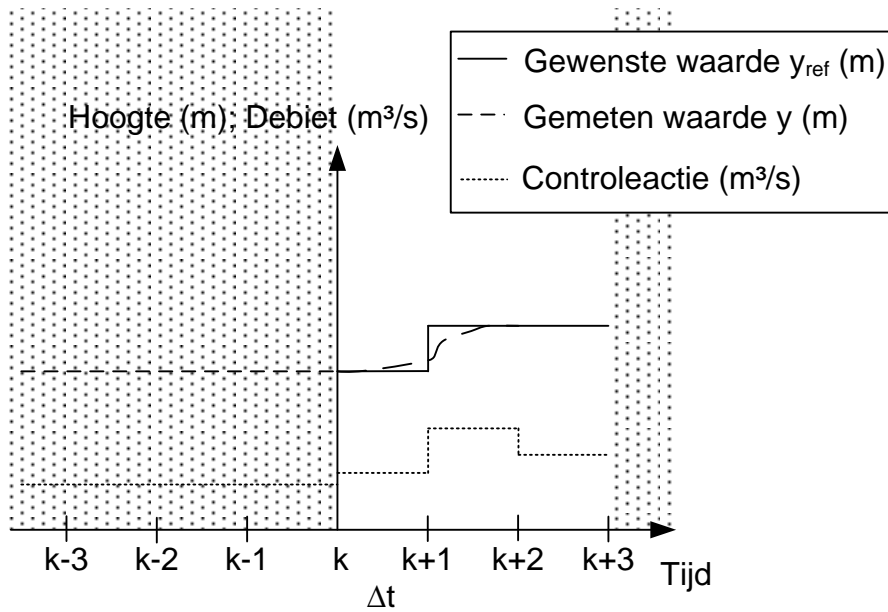


Figuur 10: MPC: uitleg voorbeeld: aanleggen van stap op  $k+3t$

Dat deze stap iedere tijdsstap  $\Delta t$  opschuift is duidelijk. Iedere tijdsstap zal de stap in gewenste waarde zich dichterbij  $k$  bevinden waardoor er ook steeds een verandering in voorspelling van controleactie zal plaatsvinden op iedere tijdsstap. Wanneer de stap zich op tijd  $k+1$  bevindt zal de MPC een eerste verandering in regelactie uitvoeren. De voorspellingshorizon van 3 betekent ook dat de MPC een voorspelling zal doen als regelactie voor de volgende 3 stappen met behulp van het kwadratisch programma dat reeds besproken is.

Doordat er een voorspelling wordt gedaan en de eerste regelactie reeds gebeurt voordat de eigenlijke stap op tijd  $k$  komt, zal het proces veel sneller een nieuwe stabiele toestand bereiken. Op onderstaande figuur is te zien dat wanneer de stap in gewenste waarde zich op  $k+1$  bevindt, de eerste regelactie reeds wordt uitgevoerd.





Figuur 11: MPC: uitleg voorbeeld: eerste regelactie aan proces

Onderstaande figuur geeft wederom de toekomst en het verleden weer, alleen bevindt de stap in gewenste waarde zich nu in de tegenwoordige tijd, op tijd  $k$ . Het is duidelijk dat hier ook de grootste stap in regelactie plaatsvindt. Dit hoeft niet noodzakelijk zo te zijn. Ook wordt zichtbaar dat na deze regelactie, de volgende regelactie reeds een actie is naar een nieuwe stabiele toestand. Deze toestand kunnen we weer berekenen aan de hand van de lineaire differentiaalvergelijking die het proces beschrijft:

$$\frac{dh}{dt} = -\frac{h}{AR} + \frac{q_{in}}{A}$$

2-17

Wanneer de gekende waarden weer worden ingevuld in de volgende afgeleide formule:

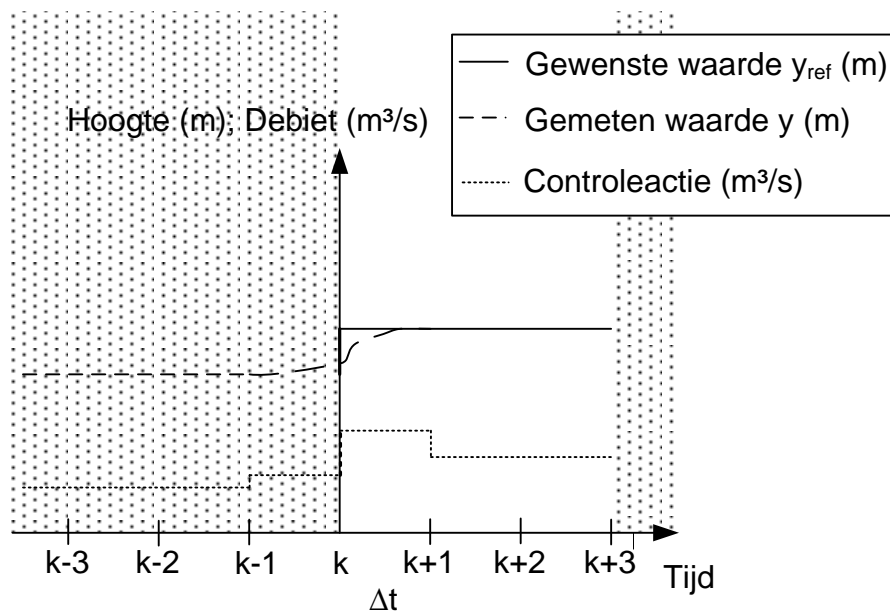
$$\frac{dh}{dt} = 0$$

$$\frac{h}{AR} = \frac{q_{in}}{A}$$

$$q_{in} = \frac{h}{R}$$

2-18

Dan wordt voor  $q_{in}$  een waarde bekomen van  $1,5\text{m}^3/\text{s}$ . In Figuur 12 is bij de gemeten waarde zichtbaar dat deze na de eerste stap in de regelactie een vloeiende respons vertoont in de gemeten waarde, naar de gewenste waarde toe. Het gedeelte van de gemeten waarde dat zich in het niet-gekleurde stuk bevindt is het door de MPC voorspelde pad van de gemeten waarde.



Figuur 12: MPC: uitleg voorbeeld: stap nieuwe ingestelde waarde

### 2.3.2 Eigenschappen van een MPC-regelaar

Een modelgebaseerde regelaar kan overweg met beperkingen en werkingsgrenzen, alsook met systemen waarin meerdere variabelen afhankelijk zijn van elkaar. Wanneer modelgebaseerde controllers gebruikt worden, is het eenvoudiger om in het toestands-ruimedomein te werken om het beschouwde proces te modelleren. (Maciejowski, 2000) Dit komt doordat processen met meerdere variabelen in het toestands-ruimedomein makkelijker toe te passen zijn. Een nadeel is wel dat modelgebaseerde regelingen soms moeilijk in te stellen zijn.

Het gebruikte model van een proces kan op verschillende manieren tot stand komen. In sommige gevallen is het mogelijk om voor de gemeten uitgangen de desbetreffende differentiaalvergelijkingen te bepalen. Met behulp van software zijn dan aan de hand van een aangelegde ingang de uitgangen te berekenen. Wanneer dit niet mogelijk is kan aan de hand van tests een *black box model* van het werkelijke proces bepaald worden. Dit kan op 2 manieren uitgevoerd worden. Het systeem kan geïdentificeerd worden aan de hand van curve-fitting-methodes. De curve waarop men de fitting uitvoert is dan het gevolg van een op het proces aangelegd signaal, zoals bijvoorbeeld een stap-verandering of sinusgolf op een bepaalde input. Deze benadering is een zogenaamde *black box*-benadering, waarbij de dynamiek van het proces niet in detail gekend is, maar enkel een output linkt aan een input. Men dient zich er dan wel van te vergewissen dat een benaderend model gebruikt wordt om een dynamisch proces te regelen. Dit maakt dat wanneer het proces meer afwijkt van het punt waarrond het model werd opgesteld, er ook een grotere afwijking zal optreden tussen model en proces. (Logist, 2013)

Naast lineaire modellen behoren niet-lineaire modellen ook tot de mogelijkheden. Dit gebeurt op basis van de balansen. Voor industriële chemische processen kunnen deze echter zeer gecompliceerd worden. Daardoor zijn ze vrij duur om te ontwikkelen. Dit wordt echter steeds vaker gedaan, door bijvoorbeeld de ontwikkeling op een originele manier te bekijken. Door de modellen op te stellen voor training van de operator (e.g. opstellen van een simulator) of voor veiligheidscertificaten, zijn ze achteraf ook te gebruiken in regelaars. Wanneer deze niet-lineaire modellen gebruikt worden in een regelaar dienen deze eerst gelineariseerd te worden rondom

een werkingspunt. Wanneer het proces teveel gaat afwijken van dit werkingspunt zal het model ook meer afwijken van het werkelijke proces. Dit is hetzelfde bij de *black box benadering*.

Een modelgebaseerde regelaar kan ook overweg met grenzen en voorwaarden. Het is bijvoorbeeld mogelijk om een lineair model te gebruiken wanneer de afwijking van de gemeten waarde ten opzichte van de gewenste waarde niet te groot is. Dit betekent dat de regelaar snel kan reageren voor wat betreft rekenkracht, maar niet erg hard moet ingrijpen om het proces de gewenste waarde opnieuw te laten bereiken. Wanneer de afwijking te groot wordt en de werkingsgrenzen benaderd worden, kan de regelaar op een niet-lineaire manier reageren en op een harde manier bijregelen. Wanneer de werkingsgrenzen toch bereikt worden zal de regelaar niet meer verder ingrijpen maar neemt een hoger niveau het over, zoals een veiligheids-PLC of een operator.

Wanneer de regelaar wiskundig benaderd wordt zal opvallen dat deze tijdsinvariant is. Dat betekent dat er enkel gekeken wordt naar de ingang van de regelaar en aan de hand daarvan wordt een bepaalde uitgang berekend om als regelactie te dienen.

Uit al het voorgaande is een groot verschil gebleken tussen regeling met PID en MPC. Een aantal basisbegrippen uit de regeltechniek werden aangehaald en indien nodig werd ook dieper ingegaan op bepaalde begrippen. Diverse methoden om een PID-regelaar in te stellen werden overlopen in functie van het gebruik in deze masterproef. Deze kennis zal gebruikt worden bij het uitvoeren van de simulaties van de gevalsstudies die worden uitgevoerd met de software, beiden (gevalsstudeis en software) worden beschreven in volgend hoofdstuk.



### 3 Materialen en methode

In dit hoofdstuk wordt besproken welke software gebruikt wordt voor de simulaties, welke speciale functies zijn toegepast en wordt het modelleren eerst kort algemeen besproken waarna de modellen voor de gevalstudies in de masterproef uitgebreid aan bod komen. Het modelleren komt eerst kort algemeen aan bod omdat het vrij abstracte materie is maar toch belangrijk om het concept bij de gevalstudies te begrijpen.

#### 3.1 Software

Gedurende deze masterproef wordt intensief gebruik gemaakt van de software MATLAB en Simulink voor het simuleren van diverse modellen. Deze software is afkomstig van MathWorks.

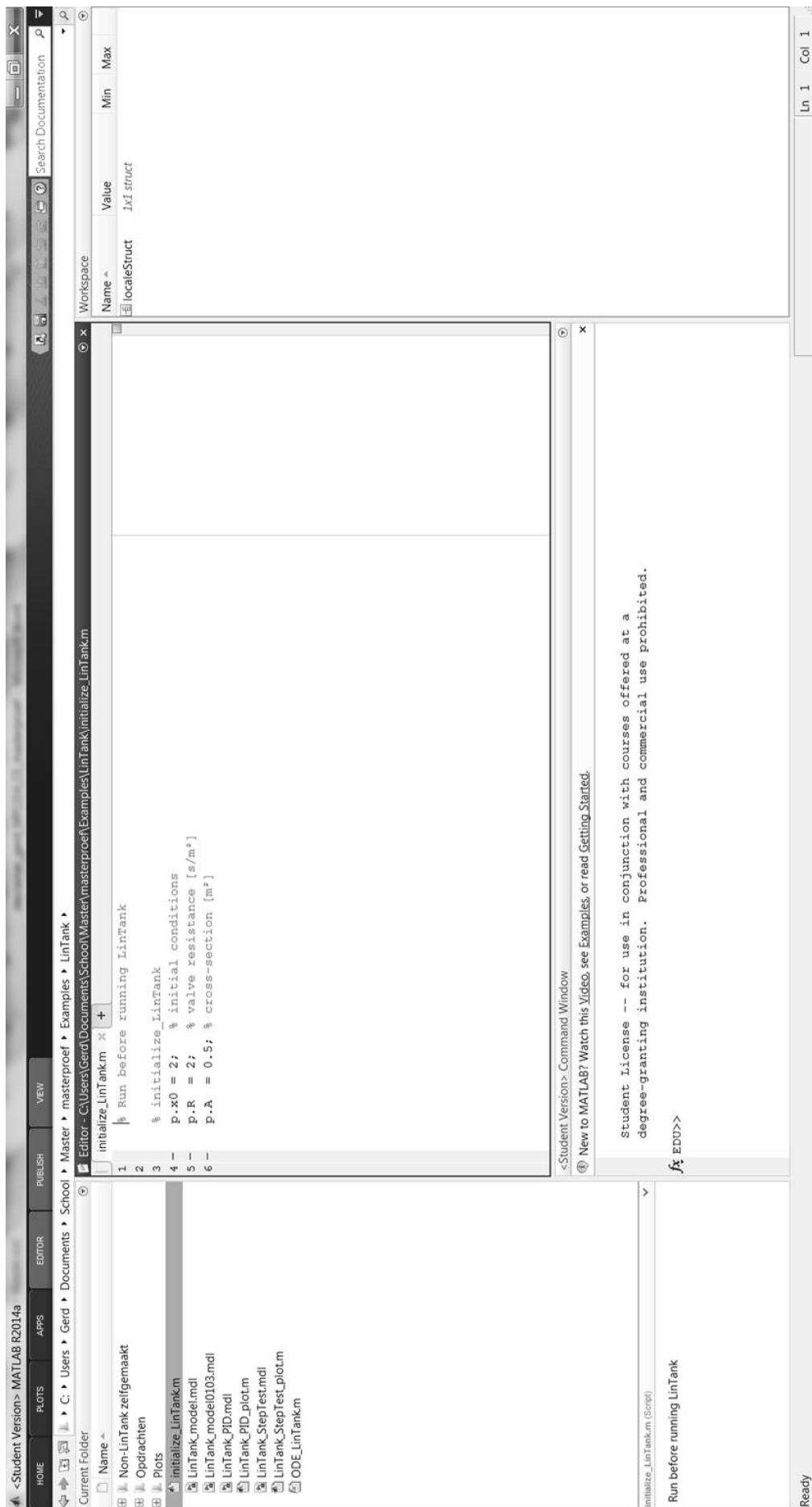
##### 3.1.1 MATLAB

MATLAB is een computerprogramma waarmee, gebruik makend van de eigen gelijknamige programmeertaal, een zeer breed scala aan wetenschappelijke berekeningen kan uitgevoerd worden. De gebruikte versie is MATLAB 2014 64-bit student versie.

Volgende bespreking geldt voor Figuur 13. Afhankelijk van persoonlijke instellingen zal het uitzicht anders zijn. De cursieve benamingen zijn leesbaar in de figuur. In het deelvenster aan de linkerkant wordt de huidige map inclusief inhoud weergegeven, de zogeheten *current folder*. Zo wordt het mogelijk om snel andere bestanden die vaak gelinkt zijn met elkaar te openen. Onder dit grotere deelvenster bevindt zich links onderaan een venster waarin commentaar wordt weergegeven betreffende het bestand dat geselecteerd is in het deelvenster *current folder*. Het deelvenster aan de rechterkant geeft de gebruikte variabelen weer. Zo is het mogelijk om in 1 oogopslag te zien of een bepaalde variabele reeds gebruikt werd en dus vrij is om nog te gebruiken of niet.

De grootste 2 deelvensters bevinden zich in het midden. De bovenste is de *editor*. Hierin kunnen bestanden geopend worden en aangepast. De *editor* wordt o.a. gebruikt om bestanden te maken die gebruikt worden in simulaties, zoals het definiëren van begintoestanden, gebruikte vergelijkingen. Het onderste venster is de *command window*. Hier kan snel en kort getest worden of een bepaalde formule of uitdrukking die gebruikt zou worden in de editor ook effectief werkt. Ook kan dit deelvenster gebruikt worden voor het openen van bestanden en het snel en eenvoudig plotten van variabelen die in de *workspace* worden weergegeven.

MATLAB heeft verschillende functies die in het programma zitten. Zo is het bijvoorbeeld mogelijk om met behulp van de functie `ODE45()` een differentiaalvergelijking numeriek op te lossen met behulp van de methode van Runge-Kutta.



Figuur 13: Overzicht MATLAB 2014

Afhankelijk van welke zogeheten *toolboxes* geïnstalleerd zijn, zijn bepaalde functies wel of niet beschikbaar. Deze *toolboxes* geven MATLAB extra functionaliteit afhankelijk van het domein waarin MATLAB gebruikt wordt. Zo kan het bijvoorbeeld interessant zijn om de modelpredictieve toolbox te installeren indien hier veel mee gewerkt wordt. Het nadeel hiervan is dat deze toolboxes zeer duur zijn voor bedrijven en scholen. De modelgebaseerde regeling die in deze thesis wordt toegepast is volledig in MATLAB-code uitgewerkt en daardoor is er geen specifieke *toolbox* nodig en waardoor de kostprijs gedrukt wordt. Bijkomend voordeel is dat deze manier van werken meer inzicht verschaft over de werking van MPC.

Naast MATLAB wordt Simulink gebruikt voor het opbouwen van een visueel aantrekkelijker model van regelkringen.

### 3.1.2 Simulink

Simulink is een programma dat volledig geïntegreerd is in MATLAB en dat door middel van een omgeving met blokdiagramma's toelaat om simulaties op te bouwen en uit te voeren in silico. Dit is niet alleen nuttig in de proceschemie, maar omdat wiskunde een universele taal is, is deze software multidisciplinair en wordt deze veelvuldig gebruikt in tal van vakgebieden.

Hieronder is Figuur 14 toegevoegd van een model in Simulink dat gebruikt werd tijdens deze masterproef. Dit model wordt nu niet uit de doeken gedaan, wel wordt gekeken naar veelgebruikte functies die nuttig zijn tijdens de masterproef. Een van de belangrijkste functies is de *Simulink library browser*. Door middel van dit venster is het mogelijk om met behulp van slepen nieuwe blokken in het hoofdvenster te brengen, waar deze gebruikt kunnen worden om een bepaald model op te bouwen. Wanneer deze blokken in het werkveld zijn aangebracht kunnen deze naar wens worden ingesteld worden, afhankelijk van de functie van elk blokje. Zo zijn er blokken die een constante *output* genereren, of blokken die een grafiek genereren zodat de *input* grafisch weergegeven wordt. Als alle blokken correct zijn ingesteld kunnen ze met behulp van signaallijnen worden verbonden. Daardoor wordt communicatie mogelijk en wordt het model als alles goed is een werkend geheel. Na het opbouwen van het model start de simulatie door op de *playknop* te drukken. De simulatie wordt gedurende een opgegeven tijd uitgevoerd. Deze tijd kan ingevuld worden in het veld waar in Figuur 14 *n\_simulated* staat.

Hierboven is reeds vermeld dat Simulink werkt met een grafische interface en blokschema's om een model op te bouwen. Dit zorgt voor een grafische en zeer intuïtieve interface maar is moeilijk te gebruiken voor het opbouwen van modellen met differentiaalvergelijkingen. De gewoonte binnen chemische ingenieurswetenschappen is om deze te gebruiken in de vorm van een vergelijking. Als een differentiaalvergelijking in Simulink wordt gebruikt moet deze van vergelijking omgebouwd worden naar een vorm met integratieblokken. Een mogelijkheid die de combinatie MATLAB en Simulink biedt voor dit probleem is het gebruik van de S-functie. Nadat het opstellen van modellen en het nut ervan voor simulaties wordt besproken, wordt dieper ingegaan op de S-functie, waar deze modellen in gebruikt kunnen worden.

#### S-functie

De S-functie is een blok in Simulink waarmee het mogelijk wordt om andere code in Simulink in te brengen. Daardoor verhogen de mogelijkheden van Simulink aanzienlijk. De kracht van de S-

functie bestaat erin dat code in zowel bijvoorbeeld MATLAB, C als C++ kan gebeuren en deelneemt aan de simulatie.

#### ODE

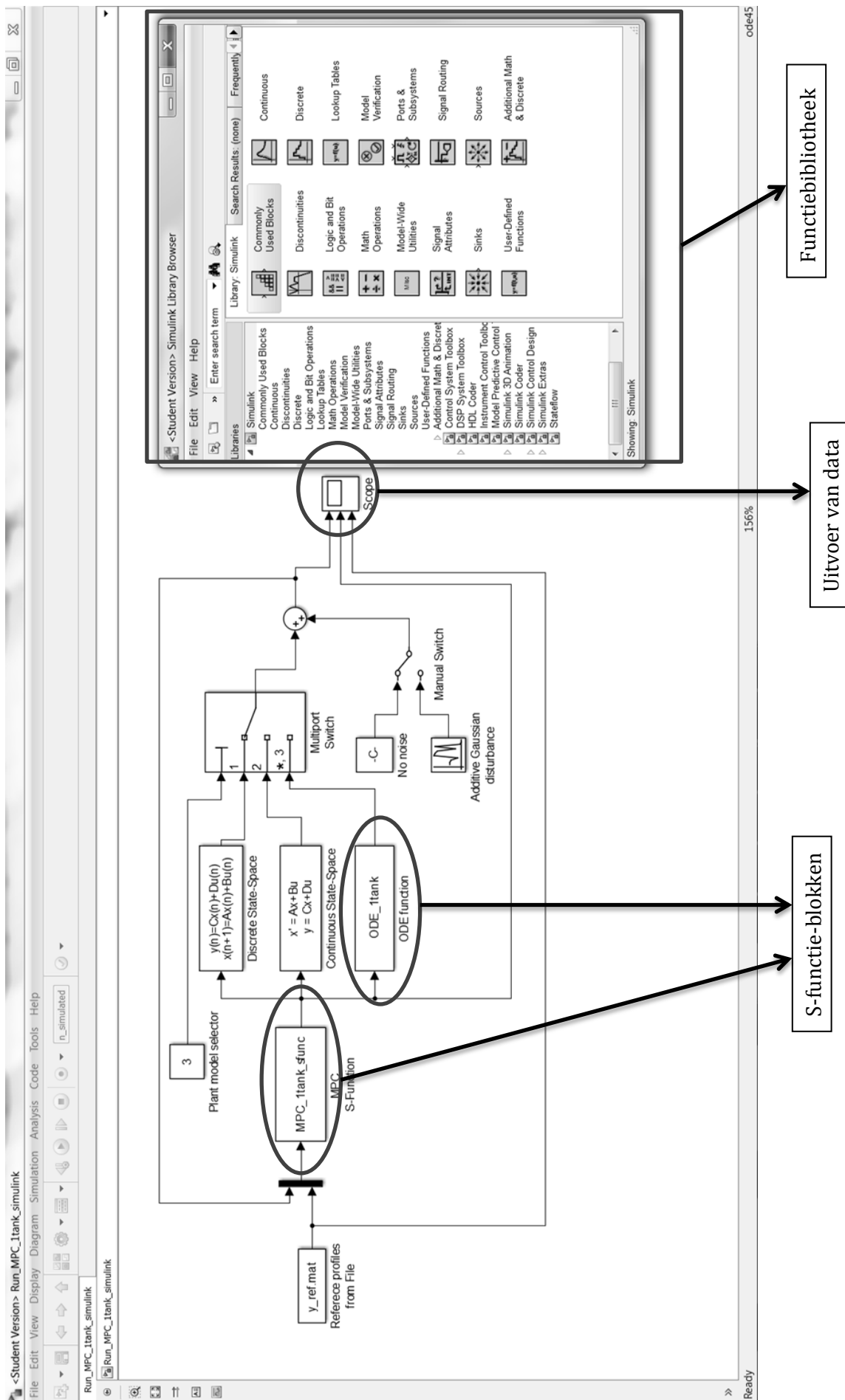
Wanneer de code voor een S-functie wordt opgebouwd dient deze aan bepaalde voorwaarden te voldoen, die beschreven worden in zowel de handleiding van MATLAB (MathWorks) als in uitleg die bijvoorbeeld op de website van MATLAB (van Schijndel, 2014) te vinden is. Een MATLAB-bestand kan bijvoorbeeld een beschrijving bevatten van een (chemisch) proces op basis van differentiaalvergelijkingen. (van Schijndel, 2014). Wanneer zo'n bestand met code voor een S-functie 1 maal is opgebouwd is deze wel snel en eenvoudig aan te passen voor andere processen met andere differentiaalvergelijkingen.

#### MPC

Naast het zorgen van input voor Simulink in de vorm van een model kan de S-functie ook ingewikkeldere code bevatten. Tijdens deze thesis wordt gebruik gemaakt van een MPC-regelaar die in MATLAB was opgebouwd, bijna af was en tijdens de thesis werd dat afgemaakt. Het is daardoor mogelijk om simulaties betreffende MPC uit te voeren zonder de modelpredictieve toolbox die MATLAB aanbiedt maar duur is. Bijkomend voordeel is het feit dat de werking van de MPC, door de uitgeschreven code in MATLAB, te doorgronden is. Studenten kunnen hiermee in de toekomst zelf simulaties uitvoeren.

Bij het gebruik van MPC is het mogelijk om in de regelaar te werken met niet-lineaire modellen die numeriek opgelost worden, maar dit vereist nog vaak teveel rekenkracht waardoor men de gelineariseerde variant gebruikt in de modelgebaseerde regeling. Het toestandsruimte model is hiervoor een veel gebruikte vorm omdat deze een duidelijke vorm heeft om het concept van MPC te leren begrijpen en omdat het toestandsruimte model vrij eenvoudig aan te passen is voor andere processen die in de MPC ingevoerd worden.





Figuur 14: Overzicht Simulink

## 3.2 Opstellen van procesmodellen

Om chemische processen te simuleren zijn wiskundige modellen nodig. Deze modellen worden opgedeeld in 2 grote groepen, namelijk op basis van lineariteit. Aan de ene kant bestaan er lineaire processen, die beschreven worden door lineaire modellen. Aan de andere kant bestaan er niet-lineaire processen die beschreven worden door niet-lineaire modellen. Wanneer een proces niet-lineair is en zo ook toegepast wordt in de software resulteert dit in een niet-gelineariseerd model. Wanneer een proces niet-lineair is maar voor de vereenvoudiging gelineariseerd wordt, dan spreken we van een gelineariseerd model. Dit lineariseren gebeurt rond een bepaalde werkingstoestand. Wanneer voor eenzelfde proces deze verder van de ontwerptoestand evolueert, zal de afwijking tussen het gelineariseerde en het niet-gelineariseerde model groter worden.

De modellen die hieronder beschreven worden zijn allemaal in functie van het gebruik in deze masterproef. Daarom worden ze eerst kort theoretisch aangehaald om daarna in de gevalstudies uitgewerkt te worden.

### 3.2.1 Lineaire modellen in continue tijd

Een lineair continu model is een model dat de variabelen beschouwt op infinitesimaal korte tijdstappen. Daardoor is het mogelijk de variabele(n) te berekenen op eender welk tijdstip. Het model dat hierbij wordt opgebouwd kan van de 0<sup>e</sup> tot de n<sup>e</sup> orde zijn. Dit betekent dat er geen (0<sup>e</sup> orde) tot een afgeleide van de n<sup>e</sup> orde in de vergelijking zit. De weergave gebeurt typisch als toestandsruimtemodel. De kracht van het toestandsruimtemodel is het feit dat alles in matrices wordt gegoten en zo duidelijkheid schept over de opbouw.

#### 3.2.1.1 0<sup>e</sup>-ordeproces

Een proces van de 0<sup>e</sup> orde geeft een 0<sup>e</sup>-ordemodel. Dit kan bijvoorbeeld een vergelijking zijn van de vorm  $ax=bu$ . Dit is dan eigenlijk niet meer dan een versterkingsfactor van  $\frac{a}{b}x = u$ .

#### 3.2.1.2 1<sup>e</sup>-ordeproces

Bij een 1<sup>e</sup>-ordeproces is de hoogste afgeleide een afgeleide van de 1<sup>e</sup> graad. Een eenvoudige vergelijking zou dan  $a\frac{dx}{dt} + b \cdot x = c \cdot u$ . Een eenvoudig voorbeeld hiervan is het vullen van een vat zonder uitstroom, constante b is daarbij 0. Maar ook het vullen van een vat met een uitstroom die recht evenredig is met de hoogte van het vloeistofniveau is een 1<sup>e</sup>-ordeproces waarbij constante b geen 0 is.

#### 3.2.1.3 2<sup>e</sup>-ordeproces

Een 2<sup>e</sup>-ordeproces heeft in de beschrijvende vergelijking een hoogste orde van 2 voor de afgeleide. Algemeen is dit een vergelijking van de vorm  $a\frac{d^2x}{dt^2} + b\frac{dx}{dt} + c \cdot x = e$ .

### 3.2.1.4 Toestandsruimtemodel

Een lineair model kan gegoten worden in de vorm van een toestandsruimtemodel. Deze vorm maakt het werken met systemen met meerdere variabelen eenvoudiger en sneller. De algemene vorm van het toestandsruimtemodel is:

$$\begin{aligned}\frac{dx}{dt} &= Ax + Bu \\ y &= Cx + Du\end{aligned}$$

3-1

Hierbij zijn A, B, C en D constanten, x zijn de toestandsvariabelen, u zijn de input- of gemanipuleerde variabelen en y zijn de uitgangsvaariabelen. Deze factoren worden weergegeven in de vorm van matrices. In deze masterproef werd het toestandsruimtemodel specifiek gebruikt voor de input van het procesmodel in de MPC. Daarvoor moet het model discreet zijn.

### 3.2.2 Lineaire modellen in discrete tijd

Wanneer het opgebouwde model gebruikt dient te worden om een MPC op te bouwen, is het nodig om het model te discretiseren. Dit wil zeggen dat de MPC, net zoals alle digitale apparatuur, niet continu aan meten doet, maar slechts op bepaalde tijdstippen een meting doet. Deze metingen hebben een vast tijdsverschil, de staalname tijd, sampletijd of  $t_s$  tussen ieder tijdstip waarop de staalname of meting gebeurt. Om een lineair continu model te gebruiken dient dit gediscrètiseerd te worden. Hierbij gebruikt men een principe dat men in de numerieke wiskunde gebruikt, het benaderen van de functie door een rechte te trekken door 2 punten met  $t_s$  ertussen.

De discretisatie gebeurt door volgende benadering uit de numerieke wiskunde:

$$\frac{dx}{dt} \approx \frac{x(k+1) - x(k)}{t_s}$$

3-2

Via discretisatie wordt dan het toestandsruimtemodel afgeleid dat er als volgt uitziet:

$$\begin{aligned}x_{k+1} &= A_d x_k + B_d u_k \\ y_{k+1} &= C_d x_{k+1} + D_d u_{k+1}\end{aligned}$$

3-3

Het komt erop neer dat voor de eerste vergelijking de differentiaalvergelijking geschreven wordt als  $\frac{dx}{dt} = \dots$ . Door de rechterterm met  $t_s$  te vermenigvuldigen en  $x(k)$  af te trekken bekomt men de nieuwe matrices  $A_d$ ,  $B_d$ ,  $C_d$  en  $D_d$ . Hier staat de d in de index voor het feit dat het gaat om de gediscrètiseerde vorm van de matrices. De x-, u- en y-matrix houden hun inhoud maar in plaats van in functie van t zijn ze nu in functie van de tijdsstap.

Het verschil tussen continu en discreet kan geïllustreerd worden met het verschil tussen 2 technologieën, namelijk analoog enerzijds en digitaal anderzijds. Analoge elektronica kan continu veranderen, bijvoorbeeld een analoge multimeter die de weerstand van een potentiometer meet die rond wordt gedraaid. Hierbij zal de aanwijzingsnaald geleidelijk veranderen en niet in stapjes. Wanneer hetzelfde wordt gemeten met een digitale multimeter,

zal de snelheid van het veranderen van de meetwaarde op het display afhankelijk zijn van o.a. de sampletijd van het apparaat. De sampletijd is het tijdsinterval tussen 2 weerstandswaarden die gemeten kunnen worden. Het display kan ook maar een bepaald bereik van meetwaarden weergeven, waar bij analogo de wijzer tussen 2 waarden in kan staan.

De tijd van het interval tussen 2 discrete waarden is een belangrijk gegeven in simulatie maar ook bij metingen. Hoe korter de tijd, hoe meer waarden, hoe meer computerkracht er vereist is om de simulatie uit te voeren of om gegevens te verwerken. Net zoals bij numerieke wiskunde. Als daar de stapgrootte kleiner wordt, wordt de benadering beter maar zal er ook meer rekenkracht vereist zijn.

### 3.2.3 Niet-lineaire modellen in continue tijd

Het is mogelijk dat in een bepaald proces niet-lineair is. Dit geeft aanleiding tot een niet-lineair model dat bestaat uit 1 of meerdere vergelijkingen die niet-lineair gedrag vertonen. Het kan bijvoorbeeld zijn dat een variabele onder een vierkantswortel zit of in een exponent. De algemene vorm kan als volgt geschreven worden, waarin x en u 2 verschillende variabelen zijn:

$$\begin{aligned}\frac{dx}{dt} &= f(x, u) \\ y &= g(x, u)\end{aligned}$$

3-4

Het nadeel van zulke niet-lineaire modellen is het feit dat in de regeltechniek voornamelijk tools bestaan en gebruikt worden voor het werken met lineaire modellen. Daarom moeten zulke niet-lineaire modellen gelineariseerd worden. Dit kan bijvoorbeeld door het schrijven van zo'n model in een Taylor-expansie. De niet-lineaire vergelijking wordt daarbij gelineariseerd rond een bepaald werkpunt. Algemeen ziet een Taylor-expansie er als volgt uit, waarbij na de 2<sup>e</sup> term wordt afgebroken (de expansie gaat oneindig verder met iedere term een bijkomende orde van afgeleide):

$$f(x, u) \approx f(x_0, u_0) + \frac{\partial f}{\partial x}(x_0, u_0)(x - x_0) + \frac{\partial f}{\partial u}(x_0, u_0)(u - u_0)$$

3-5

Nadat het niet-lineaire model gelineariseerd is met behulp van een Taylor-expansie ontstaat er een lineair model. Hierbij dient wel opgepast te worden dat dit lineaire model slechts in beperkte gevallen te gebruiken is. Het model wordt immers gelineariseerd rond een bepaalde stabiele waarde  $(x_0, u_0)$  die gekozen kan worden. Indien er te veel afgeweken wordt van deze stabiele waarde is het mogelijk dat het gelineariseerde model grote afwijkingen vertoont met het werkelijke proces waar het model op gebaseerd is.

Binnen de regeltechniek is het ook interessant om het beschouwde gelineariseerde model te gebruiken in afwijkingsvariabelen. Dit komt doordat een chemisch proces vaak geregeld dient te worden rond een stabiele waarde. Deze waarde kan ingebracht worden in het model, waarrond gelineariseerd wordt. Door daarna afwijkingsvariabelen te gebruiken kan gemakkelijk gekeken worden hoe het proces afwijkt van zijn stabiele waarde en hoe het proces zich daarbij gedraagt. Daarbij komt nog dat het werken in afwijkingsvariabelen veelvuldig gebruikt wordt in de regeltechniek net omdat algemeen genomen de linearisatie gebeurt rond een werkpunt.

Door te werken met afwijkingsvariabelen wordt er gekeken hoe het systeem zich gedraagt bij afwijkingen ten opzichte van het werkingpunt, de stabiele waarde waarrond de linearisatie is uitgevoerd. In het toestandsruimtemodel ziet dit er als volgt uit:

$$\begin{aligned}\frac{d\Delta x}{dt} &= A\Delta x + B\Delta u \\ \Delta y &= C\Delta x + D\Delta u\end{aligned}$$

3-6

In de gebruikte software is het mogelijk om niet-lineaire modellen rechtstreeks te gebruiken. Wanneer deze niet-lineaire vergelijkingen opgelost worden dan gebeurt dit met numerieke wiskunde. Met numerieke wiskunde wordt de vergelijking opgelost op een discrete manier. Tussen iedere berekening zit er een bepaalde stapgrootte. Na iedere stap wordt de uitkomst van de vergelijking berekend en gebruikt voor verdere verwerking. Methoden voor het numeriek oplossen van niet-lineaire vergelijkingen zijn de Newton-, secant-, en Runge-Kutta-Methode. De grootte van de tijdsstap is wel zeer klein, afhankelijk van de rekenkracht, en geeft daarom grafiek met een continue vorm. Dit is te zien in de simulaties, waarbij de procesrespons een continue lijn vertoont terwijl dit toch de uitgang is van een S-functieblok met ODE's in waarmee het proces beschreven wordt.

### 3.2.4 Opstellen van modellen voor procesregeling

Bij het opstellen van modellen voor procesregeling gaat men systematisch te werk. Indien mogelijk wordt op basis van balansen een model opgesteld. Deze balansen kunnen massa- en energiebalansen zijn. Als deze balansen niet-lineair zijn gaat men over tot een linearisatie op basis van Taylor. De modellen die zo bekomen kunnen worden zijn het toestandsruimtemodel, zowel in continue als in discrete tijd. Hier wordt er gewerkt naar een discrete tijd omdat het toestandsruimtemodel gebruikt wordt in de MPC. Indien nodig kan er ook een transferfunctie opgesteld worden, dit is de 3<sup>e</sup> vorm om een model te beschrijven.

De volgende 3 kopjes geven de algemene opbouw weer van het opstellen van modellen gebruikt bij de 3 gevallen, daarna worden de 3 gevallen concreet bekeken en uitgewerkt.

#### 3.2.4.1 *Op basis van balansen*

Bij het gebruik van balansen wordt een proces omschreven met behulp van materiaal- en energiebalansen. Door deze te gebruiken zal het gebruikte model de werkelijkheid vrij goed benaderen. Het nadeel is wel dat voor dit soort modellen een grotere rekenkracht vereist is omdat de differentiaalvergelijkingen numeriek opgelost moeten worden. Ook kan het zijn dat er in werkelijkheid afwijkingen optreden ten opzichte van gesimuleerde gegevens. Dit komt doordat het mogelijk is dat er bij het opstellen van de balansen bepaalde factoren zoals dichtheid als constante kunnen worden beschouwd, terwijl ze dat in werkelijkheid niet zijn.

Na het opstellen van deze balansen kan het zijn dat de vergelijkingen niet-lineair zijn. Om deze toch te kunnen gebruiken is linearisatie nodig.

### 3.2.4.2 Op basis van toestandsruimtemodel

Het opbouwen van het toestandsruimtemodel gebeurt in wezen op dezelfde manier als de opbouw van de transferfunctie. De kracht van het toestandsruimtemodel is het feit dat er eenvoudig en snel gewerkt kan worden met systemen met meerdere variabelen, waar dat bij werken met de transferfunctie moeilijker is. Het lineariseren en discretiseren worden hier niet meer herhaald, daarvoor wordt verwezen naar pagina 41.

Hierbij staat  $x$  voor de mogelijke toestanden van het systeem,  $u$  voor de variabelen in het systeem die aangepast worden en  $y$  voor de variabelen die uit het model berekend worden.  $A$ ,  $B$ ,  $C$  en  $D$  zijn factoren die bepalen hoe de verschillende veranderlijken  $x$ ,  $u$  en  $y$  zich gedragen ten opzichte van elkaar. Alle hiervoor beschreven factoren staan in matrixvorm.

Het toestandsruimtemodel is typisch een continu model. In deze thesis wordt het toestandsruimtemodel enkel gebruikt binnen de MPC. De MPC heeft een bepaalde staalnametijd, dit is het tijdsinterval tussen de eerste keer ( $k$ ) dat gemeten en gerekend wordt door de MPC en de volgende keer ( $k+1$ ) dat er gemeten en gerekend wordt door de MPC. Een belangrijke voorwaarde voor de staalnametijd is dat deze langer is dan de rekentijd nodig voor het algoritme dat wordt uitgevoerd binnen de MPC. In de MATLAB-code van de MPC staat deze staalnametijd omschreven als  $T_s$  en daarom wordt dit symbool verder gebruikt in de afleiding.

### 3.2.4.3 Op basis van transferfunctie

Wanneer men de transferfunctie gebruikt om een proces te modelleren, zal het gebruikte model eerst gelineariseerd moeten worden. Er wordt dan vertrokken van de materiaal- en energiebalansen bij een bepaalde stabiele ontwerptoestand. Wanneer het model dan gelineariseerd is wordt er de Laplacetransformatie op toegepast waardoor een transferfunctie bekomen wordt. Zo een model maakt gebruik van afwijkingsvariabelen ten opzichte van de ontwerptoestand, in plaats van absolute variabelen.

## 3.3 Gevalstudies

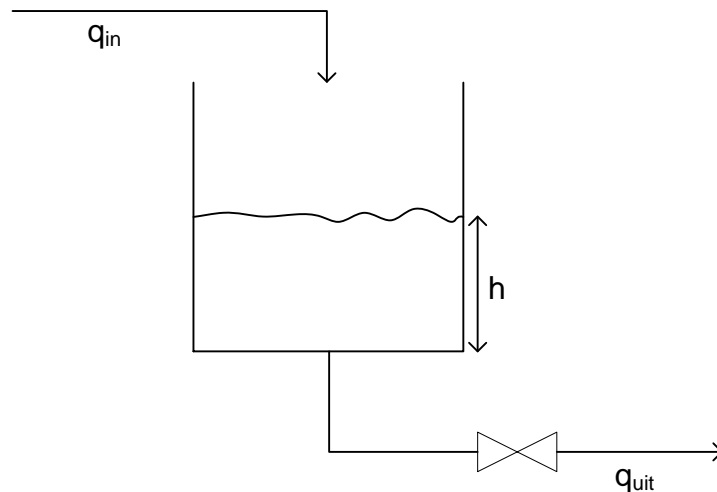
De gevalstudies die worden beschreven worden gebruikt in Simulink om simulaties uit te voeren. De input van de beschreven vergelijkingen gebeurt met behulp van de  $S$ -functie die vorige paragraaf is behandeld. Daarnaast worden de modellen van de niet-lineaire processen gelineariseerd omdat de gelineariseerde vorm in de vorm van het toestandsruimtemodel gebruikt wordt als model voor de MPC. Aan de hand van dit model gaat de MPC voorspellen hoe het proces, dat in de simulatie het originele, niet-lineaire model bevat, zich zal gedragen.

### 3.3.1 Lineaire tank

Het model dat lineaire tank wordt genoemd is een model van een eenvoudig proces met een tank met constante dwarsdoorsnede, een bepaalde vloeistofhoogte, een uitstromend debiet dat recht evenredig is met de vloeistofhoogte en een instromend debiet dat geregeld kan worden. De debieten zijn volumetrisch, dichtheden worden constant geacht. De gebruikte symbolen voor de beschreven parameters zijn de volgende:

- $h$  = hoogte van het vloeistofniveau in de tank (m);

- $A$  = dwarsdoorsnede van de tank ( $m^2$ );
- $q_{in}$  = volumetrisch debiet naar de tank ( $m^3/s$ );
- $R$  = weerstand van de regelklep ( $s/m^2$ );
- $q_{uit}$  = volumetrisch debiet uit de tank ( $m^3/s$ ).



Figuur 15: Schematische weergave van de tank

Onderstaand model beschrijft het proces in de afbeelding.  $A$  en  $R$  zijn constanten met respectievelijke waarden 0,5 en 2. De simulatie wordt uitgevoerd voor een verandering van ingestelde waarde van 3m naar 5m vloeistofhoogte in de tank.

$$\frac{dh}{dt} = \frac{q_{in}}{A} - \frac{q_{uit}}{A}; q_{uit} = \frac{h}{R}$$

$$\frac{dh}{dt} = -\frac{h}{AR} + \frac{q_{in}}{A}$$

3-7

Het proces is lineair, waardoor het model niet gelineariseerd hoeft te worden. Het toestandsruimtemodel kan rechtstreeks opgebouwd worden. Voor de algemene vorm wordt verwezen naar pagina 41. De symbolen die gebruikt worden, worden ingevuld voor deze casus en geven het volgende:

$$x = [q_{uit}]; u = [q_{in}]; y = [h]$$

3-8

$$A = \begin{bmatrix} -1 \\ AR \end{bmatrix}; B = \begin{bmatrix} 1 \\ A \end{bmatrix}; C = [1]; D = [0]$$

3-9

Dit model wordt gebruikt als MPC-model. Het model dat in de MPC gebruikt wordt dient gediscretiseerd te worden omdat het model daarin gebruik maakt van de tijdsstap  $t_s$ . Discretisatie van het vorige toestandsruimtemodel levert:

$$x = [q_{uit}]; u = [q_{in}]; y = [h]$$

3-10

$$A_d = \left[1 - \frac{t_s}{AR}\right]; B_d = \left[\frac{t_s}{A}\right]; C_d = [1]; D_d = [0]$$

3-11

### 3.3.2 Niet-lineaire tank

Het proces van de niet-lineaire tank is identiek met het model van de lineaire tank met het verschil dat het debiet aan de uitstroomopening niet recht evenredig is met de hoogte van de tank maar met de vierkantswortel van de hoogte van de tank. Dit betekent dat het model niet-lineair is wat naar exact oplossen van de vergelijking moeilijkheden geeft. De massabalans is het enige verschil met het vorige proces, daarom dat voor de uitleg van de variabelen, constanten en figuur verwezen wordt naar de gevalstudie van de lineaire tank.

$$\frac{dh}{dt} = \frac{-q_{uit}}{A} + \frac{q_{in}}{A} = -\frac{\sqrt{h}}{AR} + \frac{q_{in}}{A}$$

3-12

Het oplossen van bovenstaande vergelijking kan enkel op een numerieke manier. Dit gebeurt tijdens de simulatie, waarbij het opvalt dat de benodigde tijd langer is dan de simulaties met de lineaire tank, hetgeen te wijten is aan de numerieke verwerking van het model.

De linearisatie vindt plaats rond de algemene factoren  $h_0$ ,  $u_0$  en  $y_0$  met  $u_0$  de ingang die aanleiding geeft tot een constante waarde  $h_0$ . Deze waarden met index 0 geven de stabiele waarde aan waarrond gelineariseerd werd en waarrond het gelineariseerde model geldig is. Linearisatie volgens Taylor-reeks geeft volgende vergelijking:

$$\frac{\partial \left(\frac{dh}{dt}\right)}{\partial h} = \frac{\partial}{\partial h} \left(-\frac{\sqrt{h}}{AR} + \frac{q_{in}}{A}\right) + \frac{\partial}{\partial q_{in}} \left(-\frac{\sqrt{h}}{AR} + \frac{q_{in}}{A}\right) = -\frac{1}{2AR\sqrt{h_0}} \cdot \Delta h + \frac{1}{A} \cdot \Delta q_{in}$$

3-13

Deze vergelijking staat rechtstreeks in afwijkingsvariabelen ten opzichte van de waarde die als stabiele toestandswaarde genomen wordt,  $h_0$ . Na deze linearisatie is het mogelijk om deze vorm om te vormen naar het toestandsruimtemodel. Dit toestandsruimtemodel is opgebouwd rond een bepaalde stabiele toestand en wordt geschreven in afwijkingsvariabelen.

$$\begin{aligned} \frac{d\Delta h}{dt} &= A\Delta h + B\Delta q_{in} \\ \Delta h &= C\Delta h + D\Delta q \end{aligned}$$

3-14

Voor bovenstaande vorm staan de factoren A, B, C en D voor het volgende:

$$A = \left[\frac{1}{2AR\sqrt{h_0}}\right]; B = \left[\frac{1}{A}\right]; C = [1]; D = [0]$$

3-15



Dit model wordt uiteindelijk in de MPC gebruikt, terwijl de originele niet-lineaire differentiaalvergelijking in het model wordt gebruikt. Daarom moet dit toestandsruimtemodel nog gediscretiseerd worden. De tijdstap die daarbij gehanteerd wordt krijgt de benoeming  $T_s$ .

$$\frac{d\Delta h}{dt} \approx \frac{\Delta h(k+1) - \Delta h(k)}{t_s} = -\frac{1}{2AR\sqrt{h_0}} \cdot \Delta h(k) + \frac{1}{A} \cdot \Delta q_{in}(k)$$

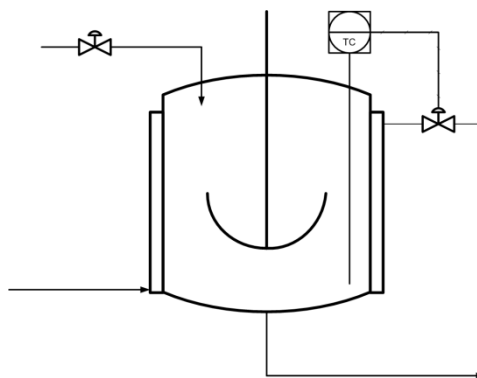
3-16

$$A_d = \left[ 1 - \frac{t_s}{2AR\sqrt{h_0}} \right]; B_d = \left[ \frac{t_s}{A} \right]; C_d = [1]; D_d = [0]$$

3-17

### 3.3.3 CSTR

De beschouwde CSTR, of *Continuously Stirred Tank Reactor*, is een reactor waarin reagens A exotherm reageert tot product B. Bij een CSTR wordt de veronderstelling gemaakt dat de inhoud perfect gemengd is waardoor geen concentratie-of temperatuurgradiënten aanwezig zijn in de reactor. De koeling gebeurt door koelwater doorheen een mantel rond de reactor te sturen. Schematisch wordt deze CSTR weergegeven in Figuur 16. Er wordt verondersteld dat de niveauregeling perfect het niveau houdt waarbij het volume van de reactor afhangt van de conversie.



Figuur 16: Schema van de CSTR

Het model dat wordt opgebouwd omtrent de CSTR gaat uit van enkele algemene eigenschappen van een CSTR. Eén daarvan is het feit dat wanneer het reagens in de reactor komt, deze onmiddellijk volledig vermengd wordt met het reactiemengsel. Een 2<sup>e</sup> eigenschap is het feit dat, door de intense, perfect veronderstelde, menging de temperatuur op alle punten in de reactor identiek is. Verder wordt een vereenvoudiging gemaakt door ten eerste een gangbare waarde voor  $U$ , warmteoverdrachtcoëfficiënt te gebruiken. Ten 2<sup>e</sup> worden bepaalde fysische constanten die afhankelijk zijn van temperatuur toch constant beschouwd (zoals de dichtheid en de soortelijke warmtecapaciteit) en ten derde wordt het temperatuurverschil tussen reactor en mantel lineair beschouwd.

De opbouw van het model gebeurt op basis van de volgende 3 balansen:

- de materiaalbalans van reagens A;

$$\frac{dC_A}{dt} = \left(\frac{F}{V_R}\right)(C_{A0} - C_A) - C_A k_0 e^{\left(\frac{-E}{RT_R}\right)}$$

3-18

- de energiebalans van de reactor;

$$\frac{dT_R}{dt} = \frac{F}{V_R}(T_0 - T_R) - \frac{\lambda C_A k_0 e^{\left(\frac{-E}{RT_R}\right)}}{\rho c_p} - \frac{UA_j(T_R - T_J)}{V_R \rho c_p}$$

3-19

- de energiebalans van de koelmantel.

$$\frac{dT_J}{dt} = \frac{F_J}{V_J}(T_{cin} - T_J) + \frac{UA_j(T_R - T_J)}{V_J \rho_j c_{pj}}$$

3-20

De gebruikte constanten en variabelen hieronder worden weergegeven. De variabelen die berekend worden zijn voor de CSTR-reactor bekeken bij een instelling van  $T_R = 330\text{K}$  en conversie van A,  $X_A = 95\%$ .

- $C_A$  = concentratie van A in de reactor  $((1-\text{conversie}) \cdot C_{A0})$ ;
- $F_{in}$  = voedingsdebiet van de reactor ( $\text{m}^3/\text{s}$ );
- $V_R$  = reactorvolume ( $434\text{m}^3$ );
- $C_{A0}$  = concentratie van A in het voedingsdebiet ( $8,01 \text{ kmol}/\text{m}^3$ );
- $k_0$  = pre-exponentiële factor van de reactiesnelheid  $R_A$  ( $20,75 \cdot 10^6 \text{ s}^{-1}$ );
- $e$  = grondtal van het natuurlijk logaritme;
- $E$  = activatie-energie ( $69,71 \cdot 10^6 \text{ J}/\text{kmol}$ );
- $R$  = universele gasconstante ( $8314 \text{ J}/\text{mol.K}$ );
- $T_R$  = reactortemperatuur (K);
- $T_0$  = temperatuur van de voeding van reagens A voor de reactor ( $294\text{K}$ );
- $\lambda$  = reactiewarmte ( $-69,71 \cdot 10^6 \text{ J}/\text{kmol}$ );
- $\rho$  = dichtheid van de voedingsstroom, wordt verondersteld constant te zijn ( $801 \text{ kg}/\text{m}^3$ );
- $c_p$  = warmtecapaciteit van reagens A, wordt verondersteld constant te zijn ( $3137 \text{ J}/\text{kg.K}$ );
- $U$  = warmteoverdrachtscoëfficiënt tussen de reactorinhoud en het koelwater ( $851 \text{ W}/\text{K.m}^2$ );

- $A_j$  = oppervlakte van de buitenkant van de reactor, waar het koelwater tegen stroomt ( $m^2 = \pi \cdot D \cdot H = 266m^2$ );
- $D$  = diameter van de reactor (6,5m);
- $H$  = reactorhoogte op basis van aspectratio  $H/D=2$  (13m);
- $T_j$  = temperatuur van het koelwater bij verlaten van de mantel (K);
- $F_j$  = volumedebiet van het koelwater ( $m^3/s$ );
- $V_j$  = inhoud van de mantel ( $A_j \cdot \text{wanddikte} = 266 \cdot 0,1 = 26,6m^3$ );
- $T_{cin}$  = ingangstemperatuur van het koelwater (294K);
- $\rho_j$  = dichtheid van het koelwater, wordt verondersteld constant te zijn ( $1000kg/m^3$ );
- $c_{pj}$  = warmtecapaciteit van het koelwater, wordt verondersteld constant te zijn (4183 J/kg.K).

Linearisatie volgens taylorreeks geeft volgende vergelijkingen:

$$\frac{dC_A}{dt}$$

$$\frac{\partial \left( \frac{dC_A}{dt} \right)}{\partial C_A} = \frac{\partial}{\partial C_A} \left( \left( \frac{F_R}{V_r} \right) (C_{A0} - C_A) - C_A k_0 e^{\left( -\frac{E}{RT_R} \right)} \right) = -\frac{F_R}{V_R} - \bar{k}$$

$$\frac{\partial \left( \frac{dC_A}{dt} \right)}{\partial T_R} = \frac{\partial}{\partial T_R} \left( \left( \frac{F_R}{V_r} \right) (C_{A0} - C_A) - C_A k_0 e^{\left( -\frac{E}{RT_R} \right)} \right) = 0 - 0 - \frac{C_A E \bar{k}}{RT_R^2} = -\frac{C_A E \bar{k}}{RT_R^2}$$

$$\frac{\partial \left( \frac{dC_A}{dt} \right)}{\partial T_j} = \frac{\partial}{\partial T_j} \left( \left( \frac{F_R}{V_r} \right) (C_{A0} - C_A) - C_A k_0 e^{\left( -\frac{E}{RT_R} \right)} \right) = 0$$

$$\frac{\partial \left( \frac{dC_A}{dt} \right)}{\partial F_R} = \frac{\partial}{\partial F_R} \left( \left( \frac{F_R}{V_r} \right) (C_{A0} - C_A) - C_A k_0 e^{\left( -\frac{E}{RT_R} \right)} \right) = \frac{C_{A0} - C_A}{V_R}$$

$$\frac{\partial \left( \frac{dC_A}{dt} \right)}{\partial F_j} = \frac{\partial}{\partial F_j} \left( \left( \frac{F_R}{V_r} \right) (C_{A0} - C_A) - C_A k_0 e^{\left( -\frac{E}{RT_R} \right)} \right) = 0$$

3-21

$$\frac{dT_R}{dt}$$

$$\frac{\partial \left( \frac{dT_R}{dt} \right)}{\partial C_A} = \frac{\partial}{\partial C_A} \left( \left( \frac{F_R}{V_R} \right) (T_0 - T_R) - \frac{\lambda C_A k_0 e^{\left( \frac{-E}{RT_R} \right)}}{\rho c_p} - \frac{UA_j (T_R - T_J)}{V_R \rho c_p} \right) = -\frac{\lambda \bar{k}}{\rho c_p}$$

$$\begin{aligned} \frac{\partial \left( \frac{dT_R}{dt} \right)}{\partial T_R} &= \frac{\partial}{\partial T_R} \left( \left( \frac{F_R}{V_R} \right) (T_0 - T_R) - \frac{\lambda C_A k_0 e^{\left( \frac{-E}{RT_R} \right)}}{\rho c_p} - \frac{UA_j (T_R - T_J)}{V_R \rho c_p} \right) \\ &= 0 - \frac{F_R}{V_R} - \frac{E \lambda C_A k_0 e^{\left( \frac{-E}{RT_R} \right)}}{\rho c_p R T_R^2} - \frac{UA}{V_R \rho c_p} - 0 = -\frac{F_R}{V_R} - \frac{E \lambda C_A k_0 e^{\left( \frac{-E}{RT_R} \right)}}{\rho c_p R T_R^2} - \frac{UA}{V_R \rho c_p} \end{aligned}$$

$$\begin{aligned} \frac{\partial \left( \frac{dT_R}{dt} \right)}{\partial T_J} &= \frac{\partial}{\partial T_J} \left( \left( \frac{F_R}{V_R} \right) (T_0 - T_R) - \frac{\lambda C_A k_0 e^{\left( \frac{-E}{RT_R} \right)}}{\rho c_p} - \frac{UA_j (T_R - T_J)}{V_R \rho c_p} \right) = 0 - 0 - 0 - 0 + \frac{UA}{V_R \rho c_p} \\ &= \frac{UA}{V_R \rho c_p} \end{aligned}$$

$$\frac{\partial \left( \frac{dT_R}{dt} \right)}{\partial F_R} = \frac{\partial}{\partial F_R} \left( \left( \frac{F_R}{V_R} \right) (T_0 - T_R) - \frac{\lambda C_A k_0 e^{\left( \frac{-E}{RT_R} \right)}}{\rho c_p} - \frac{UA_j (T_R - T_J)}{V_R \rho c_p} \right) = \frac{T_0 - T_R}{V_R}$$

$$\frac{\partial \left( \frac{dT_R}{dt} \right)}{\partial F_J} = \frac{\partial}{\partial F_J} \left( \left( \frac{F_R}{V_R} \right) (T_0 - T_R) - \frac{\lambda C_A k_0 e^{\left( \frac{-E}{RT_R} \right)}}{\rho c_p} - \frac{UA_j (T_R - T_J)}{V_R \rho c_p} \right) = 0$$

3-22

$$\frac{dT_J}{dt}$$

$$\frac{\partial \left( \frac{dT_J}{dt} \right)}{\partial C_A} = \frac{\partial}{\partial C_A} \left( \frac{F_J}{V_J} (T_{cin} - T_J) + \frac{UA_j (T_R - T_J)}{V_J \rho_j c_{pj}} \right) = 0$$

$$\frac{\partial \left( \frac{dT_J}{dt} \right)}{\partial T_R} = \frac{\partial}{\partial T_R} \left( \frac{F_J}{V_J} (T_{cin} - T_J) + \frac{UA_j (T_R - T_J)}{V_J \rho_j c_{pj}} \right) = 0 - 0 + 0 + \frac{UA_j}{V_J \rho_j c_{pj}} = \frac{UA_j}{V_J \rho_j c_{pj}}$$

$$\frac{\partial \left( \frac{dT_J}{dt} \right)}{\partial T_J} = \frac{\partial}{\partial T_J} \left( \frac{F_J}{V_J} (T_{cin} - T_J) + \frac{UA_j (T_R - T_J)}{V_J \rho_j c_{pj}} \right) = 0 - \frac{F_J}{V_J} + 0 - \frac{UA}{V_J \rho_j c_{pj}} = -\frac{F_J}{V_J} - \frac{UA}{V_J \rho_j c_{pj}}$$

$$\frac{\partial \left( \frac{dT_J}{dt} \right)}{\partial F_R} = \frac{\partial}{\partial F_R} \left( \frac{F_J}{V_J} (T_{cin} - T_J) + \frac{UA_j (T_R - T_J)}{V_J \rho_j c_{pj}} \right) = 0$$

$$\frac{\partial \left( \frac{dT_J}{dt} \right)}{\partial F_J} = \frac{\partial}{\partial F_J} \left( \frac{F_J}{V_J} (T_{cin} - T_J) + \frac{UA_J (T_R - T_J)}{V_J \rho_J c_{pJ}} \right) = \frac{T_{cin} - T_J}{V_J}$$

3-23

Achtereenvolgens is dit verkort voor de 3 differentiaalvergelijkingen:

$$\frac{d\Delta C_A}{dt} = \left( -\frac{F_R}{V_R} - \bar{k} \right) \Delta C_A + \left( -\frac{C_A E \bar{k}}{RT_R^2} \right) \Delta T_R + 0 \Delta T_J + \left( \frac{C_{A0} - C_A}{V_R} \right) \Delta F_R + 0 \Delta F_J$$

3-24

$$\frac{d\Delta T_R}{dt} = \left( -\frac{\lambda \bar{k}}{\rho c_p} \right) \Delta C_A + \left( -\frac{F_R}{V_R} - \frac{E \lambda C_A k_0 e^{\left( -\frac{E}{RT_R} \right)}}{\rho c_p RT_R^2} - \frac{UA}{V_R \rho c_p} \right) \Delta T_R + \left( \frac{UA}{V_R \rho c_p} \right) \Delta T_J + \left( \frac{T_0 - T_R}{V_R} \right) \Delta F_R + 0 \Delta F_J$$

3-25

$$\frac{d\Delta T_J}{dt} = 0 \Delta C_A + \left( \frac{UA_J}{V_J \rho_J c_{pJ}} \right) \Delta T_R + \left( -\frac{F_J}{V_J} - \frac{UA}{V_J \rho_J c_{pJ}} \right) \Delta T_J + 0 \Delta F_R + \left( \frac{T_{cin} - T_J}{V_J} \right) \Delta F_J$$

3-26

Dit kan verkort geschreven worden als volgt, waarbij erop gelet moet worden dat deze toestand bekeken wordt vanuit een stabiele waarde  $C_{Ass}$ ,  $T_{Rss}$ ,  $T_{Jss}$ ,  $F_{Jss}$  en  $F_{INss}$ , met ss van *steady state* of *stabiele waarde*:

$$\frac{d\Delta C_A}{dt} = a_{11} \Delta C_A + a_{12} \Delta T_R + a_{13} \Delta T_J + b_{11} \Delta F_R + b_{12} \Delta F_J$$

3-27

$$\frac{d\Delta T_R}{dt} = a_{21} \Delta C_A + a_{22} \Delta T_R + a_{23} \Delta T_J + b_{21} \Delta F_R + b_{22} \Delta F_J$$

3-28

$$\frac{d\Delta T_J}{dt} = a_{31} \Delta C_A + a_{32} \Delta T_R + a_{33} \Delta T_J + b_{31} \Delta F_R + b_{32} \Delta F_J$$

3-29

Matrices A en B zijn opgebouwd uit volgende elementen:

$$a_{11} = -\frac{F_R}{V_R} - \bar{k}; a_{12} = -\frac{C_A E \bar{k}}{RT_R^2}; a_{13} = 0; b_{11} = \frac{C_{A0} - C_A}{V_R}; b_{12} = 0$$

3-30

$$a_{21} = -\frac{\lambda \bar{k}}{\rho c_p}; a_{22} = -\frac{F_R}{V_R} - \frac{E \lambda C_A k_0 e^{\left( -\frac{E}{RT_R} \right)}}{\rho c_p RT_R^2} - \frac{UA}{V_R \rho c_p}; a_{23} = \frac{UA}{V_R \rho c_p}; b_{21} = \frac{T_0 - T_R}{V_R}; b_{22} = 0$$

3-31

$$a_{31} = 0; a_{32} = \frac{UA_J}{V_J \rho_J c_{pJ}}; a_{33} = -\frac{F_J}{V_J} - \frac{UA}{V_J \rho_J c_{pJ}}; b_{31} = 0; b_{32} = \frac{T_{cin} - T_J}{V_J}$$

3-32

De matrices van x, y en u zien er dan als volgt uit:

$$x = \begin{bmatrix} C_A \\ T_R \\ T_J \end{bmatrix}; u = [F_J]; y = \begin{bmatrix} C_A \\ T_R \\ T_J \end{bmatrix}$$

3-33

De matrices A, B, C en D bevatten volgende elementen, waarbij opvalt dat de B-matrix maar 1 kolom heeft, omdat er maar 1 gemanipuleerde variabele in dit model gebruikt wordt, namelijk het koelwaterdebiet. Het is ook mogelijk om het model op te stellen waarbij het voedingsdebiet én het koelwaterdebiet gemanipuleerde variabelen zijn. Er komt dan een extra factor in de u-matrix en een extra rij in de B-matrix.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}; B = \begin{bmatrix} b_{12} \\ b_{22} \\ b_{32} \end{bmatrix}$$

$$C = [0 \ 1 \ 0]; D = [0]$$

3-34

#### Uitgebreid gelineariseerd toestandsruimtemodel met tijdsvertraging en discretisatie.

Bij de simulaties met de MPC als regelaar moet er ook een toestandsruimtemodel uitgewerkt worden. In dit model moet ook de tijdsvertraging ingebouwd worden als gevolg van de temperatuursensor. Bij de simulaties is gekeken wat het gevolg is van het werken met 2 tijdsvertragingen bij de temperatuursensor, zonder dat het model van de MPC deze tijdsvertragingen bevat. (p.80-81). De MPC bleek problemen te hebben met het controleren van de CSTR wanneer de tijdsvertraging van de temperatuursensor niet in rekening gebracht. Daarom wordt het model uitgebreid en daarna gediscretiseerd, omdat dit het concrete model is dat gebruikt wordt in de MPC.

Naast de 3 differentiaalvergelijkingen van de concentratie, reactortemperatuur en koelwatertemperatuur komen er nu 2 differentiaalvergelijkingen bij van de eerste orde. De factor  $\tau_m$  is de tijdsconstante en bedraagt in deze casus voor beide tijdsvertragingen 60s:

$$\frac{\tau_m dT_m}{dt} = T_{in} - T_m$$

3-35

In afwijkingsvariabelen wordt dit:

$$\frac{\tau_m d\Delta T_m}{dt} = \Delta T_{in} - \Delta T_m$$

3-36

Omdat er 2 tijdsvertragingen zijn, zijn er ook 2 vergelijkingen, 1 voor iedere eerste orde tijdsvertraging binnen de temperatuursensor. Voor de eerste tijdsvertraging is  $T_{in}$  de reactortemperatuur, voor de 2<sup>e</sup> tijdsvertraging is  $T_{in}$   $T_m$  van de 1<sup>e</sup> tijdsvertraging:

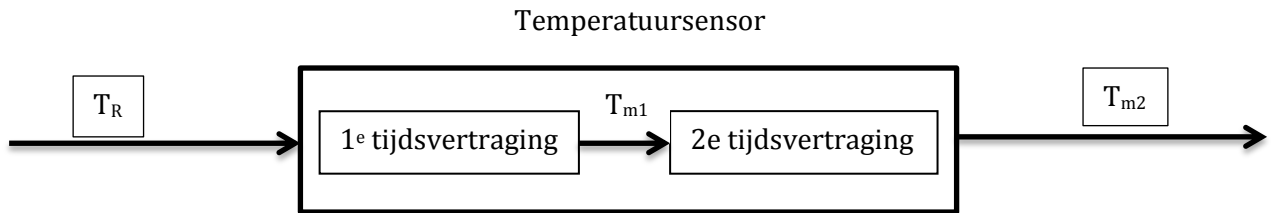
$$\text{eerste tijdsvertraging: } \frac{\tau_m dT_{m1}}{dt} = T_R - T_{m1}$$

3-37

$$\text{tweede tijdsvertraging: } \frac{\tau_m dT_{m2}}{dt} = T_{m1} - T_{m2}$$

3-38

Om dit even te verduidelijken:



Figuur 17: Schematische weergave van de temperatuursensor

Uiteindelijk geeft dit volgend toestandsruimtemodel op basis van het vorige toestandsruimtemodel. Daarbij is ook een discretisatiestap uitgevoerd wegens het feit dat dit model in de MPC gebruikt zal worden:

$$x = \begin{bmatrix} \Delta C_A \\ \Delta T_R \\ \Delta T_J \\ \Delta T_{m1} \\ \Delta T_{m2} \end{bmatrix}; u = [F_J]; y = [\Delta T_{m2}]$$

3-39

De matrices A, B, C en D geven zonder discretisatie het volgende resultaat:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 \\ 0 & \frac{1}{\tau_m} & 0 & -\frac{1}{\tau_m} & 0 \\ 0 & 0 & 0 & \frac{1}{\tau_m} & -\frac{1}{\tau_m} \end{bmatrix}; B = \begin{bmatrix} b_{12} \\ b_{22} \\ b_{23} \\ 0 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}; D = [0]$$

3-40

Wanneer discretisatie wordt uitgevoerd geeft dit:

$$A = \begin{bmatrix} 1 + a_{11}t_s & a_{12}t_s & a_{13}t_s & 0 & 0 \\ a_{21}t_s & 1 + a_{22}t_s & a_{23}t_s & 0 & 0 \\ a_{31}t_s & a_{32}t_s & 1 + a_{33}t_s & 0 & 0 \\ 0 & \frac{t_s}{\tau_m} & 0 & 1 - \frac{t_s}{\tau_m} & 0 \\ 0 & 0 & 0 & \frac{t_s}{\tau_m} & 1 - \frac{t_s}{\tau_m} \end{bmatrix}; B = \begin{bmatrix} b_{12}t_s \\ b_{22}t_s \\ b_{23}t_s \\ 0 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}; D = [0]$$

3-41

Laplace

De laplacetransformatie van de gelineariseerde differentiaalvergelijkingen is nodig voor het achterhalen van de stabiliteit van het systeem met behulp van de methode van root locus en geeft volgende vergelijkingen:

$$sC_A(s) = a_{11}C_A(s) + a_{12}T_R(s) + a_{13}T_J(s) + b_{11}F_R(s) + b_{12}F_J(s)$$

3-42

$$sT_R(s) = a_{21}C_A(s) + a_{22}T_R(s) + a_{23}T_J(s) + b_{21}F_R(s) + b_{22}F_J(s)$$

3-43

$$sT_J(s) = a_{31}C_A(s) + a_{32}T_R(s) + a_{33}T_J(s) + b_{31}F_R(s) + b_{32}F_J(s)$$

3-44

Voor het zoeken naar de root locus om de stabiliteit van het proces te bepalen is de transferfunctie nodig van dit proces met als input de variabele  $\Delta F_J$  en als output  $\Delta T_R$ . Om vereenvoudigd te werken wordt gebruik gemaakt van de afkortingen die hierboven staan weergegeven en worden de factoren in volgende formule ook weer verkort weergegeven.

$$H(s) = \frac{Y(s)}{U(s)} = \frac{T_R(s)}{F_J(s)} = \frac{c_1s + c_0}{(s^3 + c_4s^2 + c_3s + c_2)}$$

3-45

$$c_0 = -a_{11}a_{23}b_{32}; c_1 = a_{23}b_{23}; c_2 = a_{12}a_{21}a_{33} - a_{11}a_{22}a_{33} + a_{11}a_{23}a_{32}$$

$$c_3 = a_{11}a_{22} + a_{11}a_{33} + a_{22}a_{33} - a_{12}a_{22} - a_{23}a_{32}; c_4 = -a_{11} - a_{22} - a_{33}$$

3-46

Het proces wordt voorzien van 2 tijdsvertragingen als gevolg van de temperatuursmeting. Deze 2 tijdsvertragingen zijn beide van de eerste orde. Hierin is  $\tau_m$  de tijdsconstante. Bij de simulaties wordt deze tijdsvertraging ingesteld als 60s.

$$\frac{\tau_m dT_m}{dt} = T_{in} - T_m$$

3-47

De laplacetransformatie hiervan resulteert in volgende vergelijking:

$$(T_m \tau_m s + T_m \tau_m) = T_{in}$$

3-48

De transferfunctie voor de meting wordt dan:



$$H(s) = \frac{\text{output}}{\text{input}} = \frac{Y(s)}{U(s)} = \frac{T_m(s)}{T_{in}(s)} = \frac{1}{\tau_m s + 1}$$

3-49

Wanneer deze transferfunctie mee vervat wordt in de transferfunctie van de CSTR wordt de volgende vergelijking bekomen:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{T_R(s)}{F_f(s)} = \frac{c_1 s + c_0}{(s^3 + c_4 s^2 + c_3 s + c_2)(\tau_m s + 1)^2}$$

3-50

Met het model van de CSTR in bovenstaande vorm, als zijnde transferfunctie, kan er een root locus plot gemaakt.

### Conclusie

In het voorgaande hoofdstuk werd een inleiding gegeven in de gebruikte software. De link naar de modellering werd aangehaald in de uitleg over de S-functie binnen Simulink. Om de modellen van de gevalstudies niet te abstract voor de dag te laten komen werd de theorie aangehaald rond de manieren waarop modellen opgesteld kunnen worden. Daarna zijn de 3 gevalstudies uitgewerkt die tijdens deze masterproef gebruikt werden tijdens simulaties om het verschil tussen PID en MPC aan te geven. De volgende 2 hoofdstukken beschrijven deze resultaten, eerst met betrekking tot de 2 modellen betreffende de tank en dan over het model van de CSTR. Daarbij wordt nagegaan hoe de systemen reageren op externe veranderingen en of het gebruik van PID en MPC hier een grote invloed op heeft.



## 4 Resultaten: Regeling van het niveau in een opslagvat

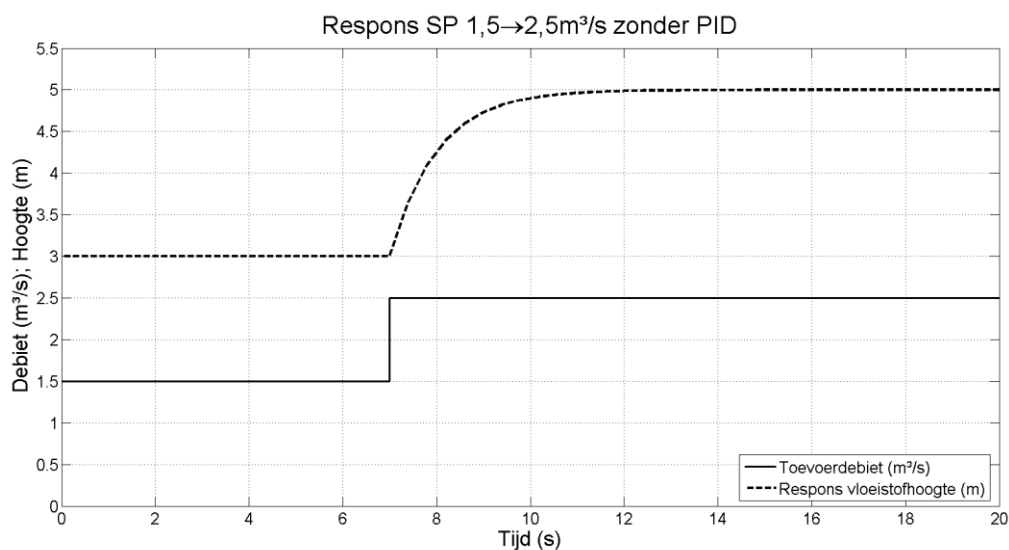
In het voorgaande hoofdstuk zijn de modellen afgeleid voor de 2 opslagvaten die in volgende simulaties gebruikt werden. Door deze simulaties uit te voeren voor vergelijkbare situaties wordt het verschil aangegeven tussen een lineair en niet-lineair model van een tank. Het verschil tussen regelen met PID en MPC wordt ook nagegaan, waarbij gekeken wordt naar de reactie van het systeem op het aanleggen van een verandering van ingestelde waarde van 3 naar 5m hoogte in te tank.

Er wordt ook aandacht geschonken aan de realistische tint van de simulaties door o.a. een minimaal en maximaal ingangsdebiet in te stellen in de PID-regelaar, zie verder.

### 4.1 Lineaire tank

#### 4.1.1 Geen regeling, open lus

Eerst wordt er nagegaan hoe het systeem reageert op wijzigingen in het toevoerdebiet. Het toevoerdebiet wijzigt na 10s simulatie van  $1,5\text{m}^3/\text{u}$  naar  $2,5\text{m}^3/\text{u}$ . De totale simulatie duurt 20s. Daarbij verandert de vloeistofhoogte van 3m naar 5m. Figuur 18 geeft het resultaat weer van deze simulatie.

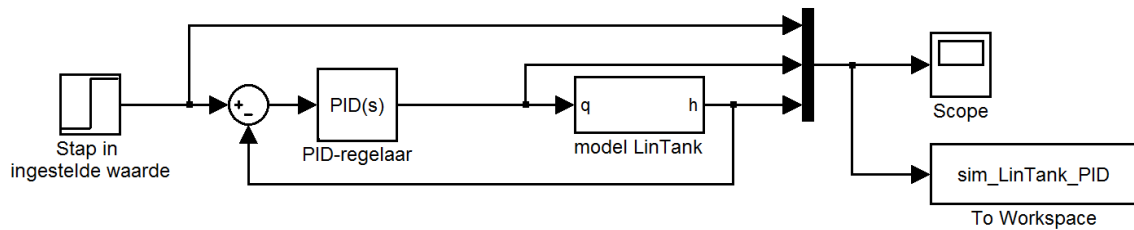


Figuur 18: Respons open lus: geen regelaar

#### 4.1.2 Klassieke regeling: PID

Vooraleer PID en MPC met elkaar vergeleken worden moet de PID-regelaar eerst ingesteld worden. Hiervoor worden open lus, gesloten lus en relais-methode aangehaald om de regelaar in te stellen. Daarna wordt nagegaan welke instellingen tot een bevredigende regeling van het systeem komen. Het resultaat hiervan wordt gebruikt om de PID-regeling te vergelijken met MPC. Het blokschema dat in Simulink gebruikt is voor de simulaties is in Figuur 19 te zien. Het eerste blok geeft de stap van ingestelde waarde weer, deze heeft een instelling van 3 naar 5m op 7s simulatietijd. De totale simulatie bedraagt 20s. Het blokje model LinTank bevat een ODE-beschrijving van het model. De Scope laat toe om de signalen die binnen komen in een grafische

weergave van de simulatie te gieten. Het blokje *To Workspace* slaat alle gegevens op die naar de scope gevoerd worden, zodat de gegevens later te verwerken zijn in een grafiek.

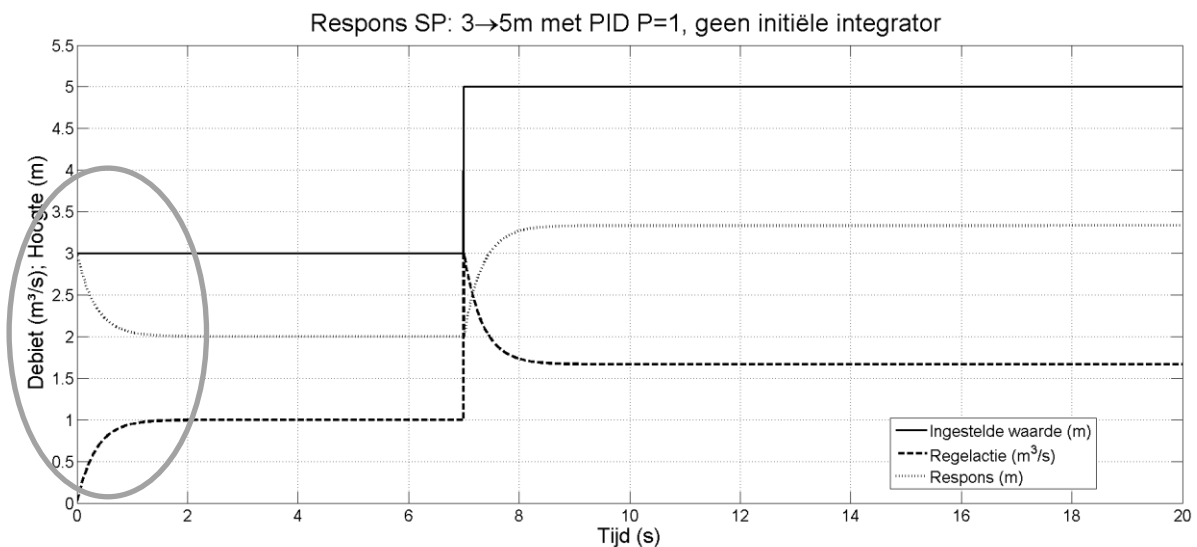


Figuur 19: Schema simulink: simulatie PID

Wanneer nu een PID-regelalgoritme aan dit systeem wordt toegevoegd, kan er van een regeling gesproken worden. Hiervoor was het een open lus, maar de lus wordt nu gesloten. De instellingen die in de PID-regelaar worden ingesteld zijn o.a. saturatiegrenzen (ondergrens:  $0\text{m}^3/\text{s}$  en bovengrens:  $5\text{m}^3/\text{s}$ , een vloeistofstroom zal immers altijd in een bepaalde richting stromen in een proces en zal ook niet boven een bepaalde maximale waarde uitkomen) en een antiresetwindup (gezien de korte simulatietijd is dit niet verplicht). Het type PID-regelaar dat is ingesteld is een zogenaamde *parallele* PID. Verder was het nodig om de PID bij de integrator een initiële waarde te geven. Anders weet de PID-regelaar niet dat er bij de initiële hoogte van de tank (2m) een debiet van  $1\text{m}^3/\text{u}$  hoort en zal er al een regelactie gebeuren voor een stabiele waarde. Deze 2m hoogte is het vertrekpunt van de simulatie.

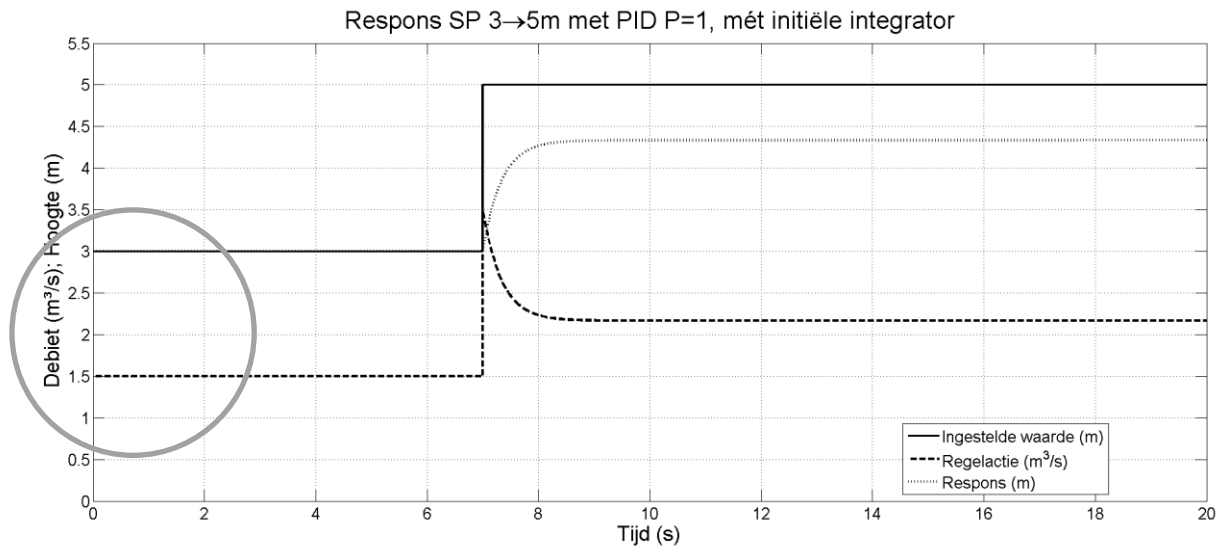
### Initiële condities

Hieronder wordt weergegeven wat er gebeurt als de initiële integrator fout is ingesteld.  $K_p$ , de proportionele actie van de PID-regelaar is de enige die is ingesteld. De waarde van  $K_p$  is 1. Wat onmiddellijk opvalt is de statische fout na de stap in ingestelde waarde op 7s. Deze statische fout, tussen de ingestelde waarde en de respons, is te wijten aan het feit dat de initiële integrator ingesteld staat op de initiële hoogte. Er zit verder geen integrator in de PID waardoor deze afwijking niet wordt weggewerkt.



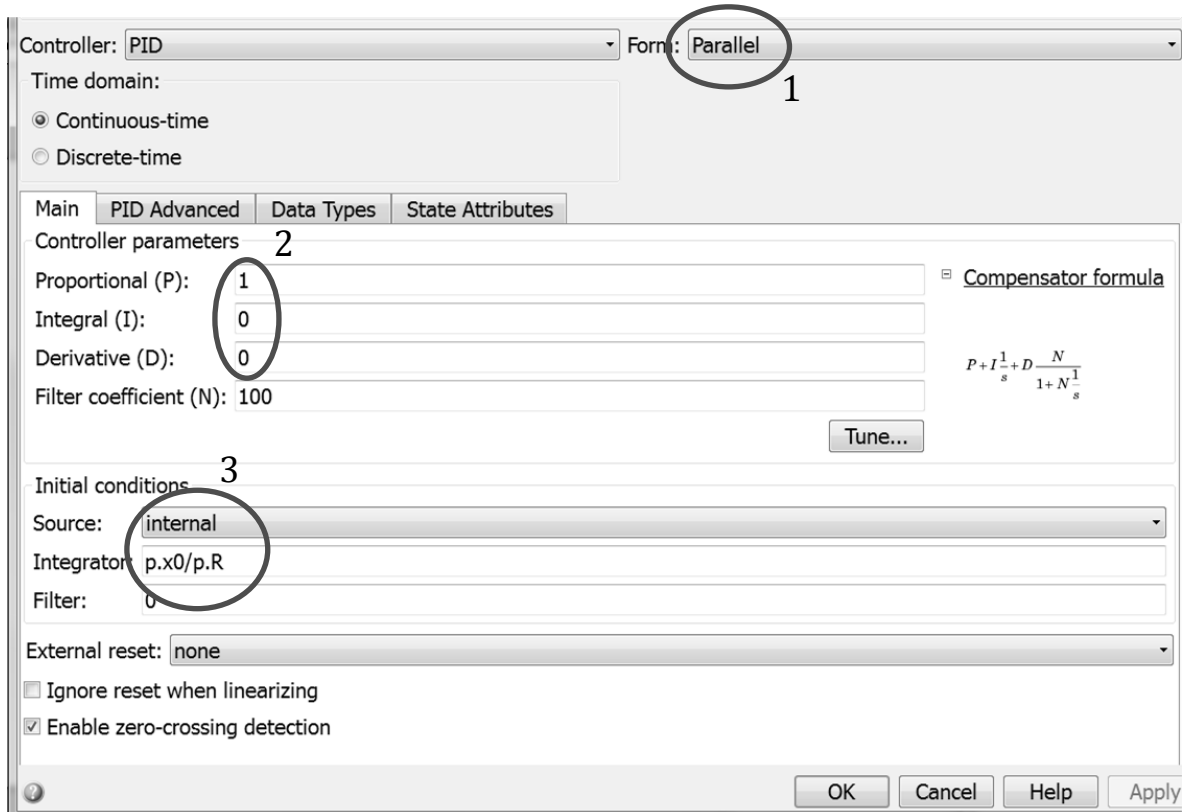
Figuur 20: Respons met PID zonder initiële integrator

Wanneer deze simulatie wordt uitgevoerd met een correct ingestelde initiële integrator ( $h_0/R$ ) dan wordt, voor dezelfde instellingen voor de PID, Figuur 21 gegenereerd. De enige factor die in de PID-regelaar is ingesteld is de versterkingsfactor die op 1 is ingesteld. Eén van de tekortkomingen van het enkel instellen van de P-factor wordt hier duidelijk. Er treedt namelijk een statische fout op. Het systeem zal niet naar een hoogte van 3m kunnen naderen.



Figuur 21: Respons met PID met initiële integrator

Figuur 22 geeft aan welke instellingen in de PID bij Simulink belangrijk zijn. Ballon 1 geeft aan welk type PID-regelaar er is ingesteld. Dit is essentieel omdat er 2 vormen van PID-regelaar zijn. De methoden om een PID-regelaar in te stellen geven de parameters voor 1 van de 2 vormen en moeten omgevormd worden als ze in een ander type gebruikt worden. Ballon 2 geeft de instellingen van de 3 parameters van de PID-regelaar. aan de rechterkant, onder de tekst *compensator formula* toont op welke manier deze parameters gebruikt worden. Bij het veranderen van de vorm (parallel of ideaal) verandert deze formule ook. Ballon 3 ten laatste geeft het veld aan waar de initiële waarde van de PID-regelaar ingesteld kan worden. Dit is een manuele reset.



Figuur 22: Instellingen van de PID in Simulink

Door de manuele reset in te stellen op de initiële ingestelde waarde (hoogte van de tank) kan de PID-regelaar een initiële regelactie genereren, zonder statische afwijking. Doordat deze berekening enkel geldt voor de initiële ingestelde waarde zal er, nadat de ingestelde waarde wijzigt van 3m naar 5m, een statische afwijking optreden die ongewenst is. De initiële integrator geldt immers enkel voor de initiële ingestelde waarde en niet voor de ingestelde waarde na de stap.

Naast de instelling van de initiële integrator worden nog 2 parameters ingesteld in het PID-blok, namelijk een minimaal en maximaal debiet waarbinnen de PID de waarde kan veranderen. Dit wordt zo gedaan om de simulaties een realistische tint te geven, een debiet is in de praktijk immers ook niet oneindig groot en kan nooit in een tegengestelde richting geregeld worden.

### Open lus tuning

Om de PID-regelaar volledig in te kunnen stellen wordt gebruik gemaakt van de responscurve waarbij het voedingsdebiet een stap maakt van  $0\text{m}^3/\text{u}$  tot  $0,5\text{m}^3/\text{u}$ . Op deze responscurve wordt er een raaklijn getekend aan het steilste stuk van de respons van het proces. Het punt waar deze raaklijn de lijn van de nieuwe stabiele waarde van het proces snijdt, geeft een snijpunt met een x-en y-waarde. De x-waarde, de tijd, is de waarde die nodig is om de verdere berekeningen uit te voeren.

De gevolgde methode kan in de literatuurstudie gevonden worden onder het luik klassieke regeling: open lus afstelling. De instellingen op basis van open lus moet normaal gezien een dodeltijd hebben. Dit proces is een 1<sup>e</sup>-ordeproces zonder dodeltijd. Daarom is deze dodeltijd 0 en is er een variatie nodig waarbij de dodeltijd gelijkgesteld wordt aan 1 in plaats van 0. De gevonden waarden voor  $\tau_d$  en  $\tau_p$  zijn 1 (indien  $\tau_d$  0 is, stelt men  $\tau_d$  gelijk aan 1) en  $11,08 \cdot 10 = 1,08$ .

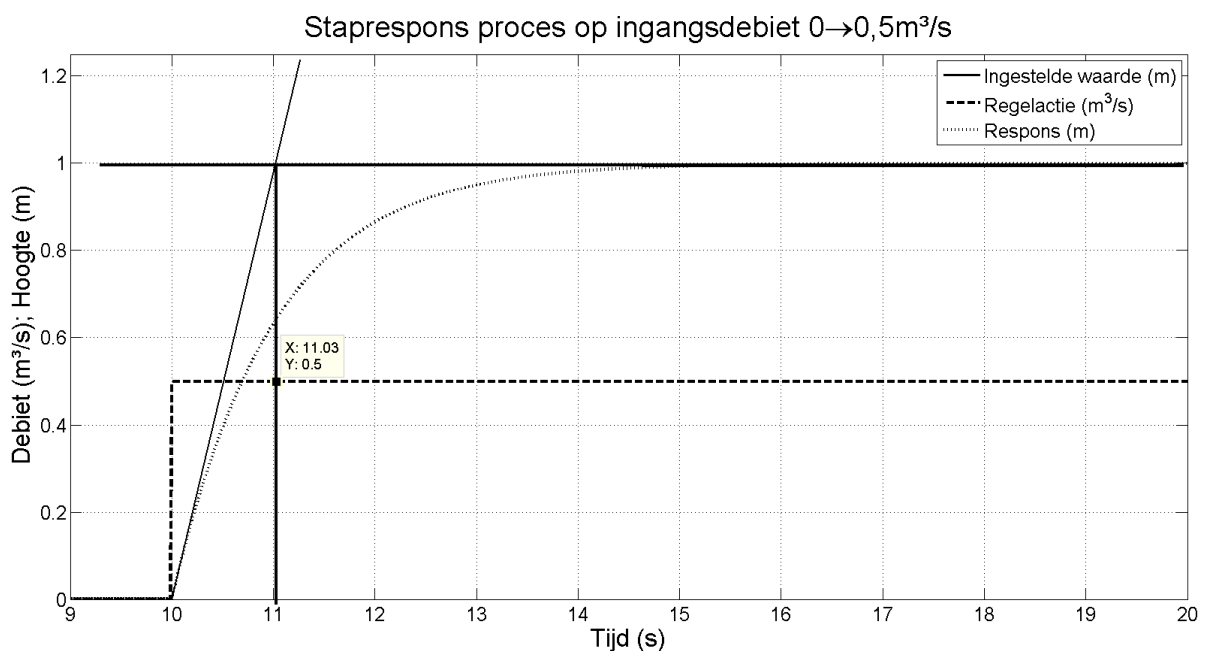
$K_p$ , de versterkingsfactor van het proces, is de resulterende proceswaarde/de aangelegde variatie in inputvariabele (hier het voedingsdebiet).  $K_p$  is dus  $1/0,5=2$ .

De gebruikte omrekeningsfactoren worden weergegeven in onderstaande figuur voor diverse mogelijke instellingen van een PID-regelaar: (figuur en werkwijze afkomstig uit labohandleiding oefenzittingen KULeuven)

Omgerekend geeft dit de volgende bekomen waarden voor deze casus:

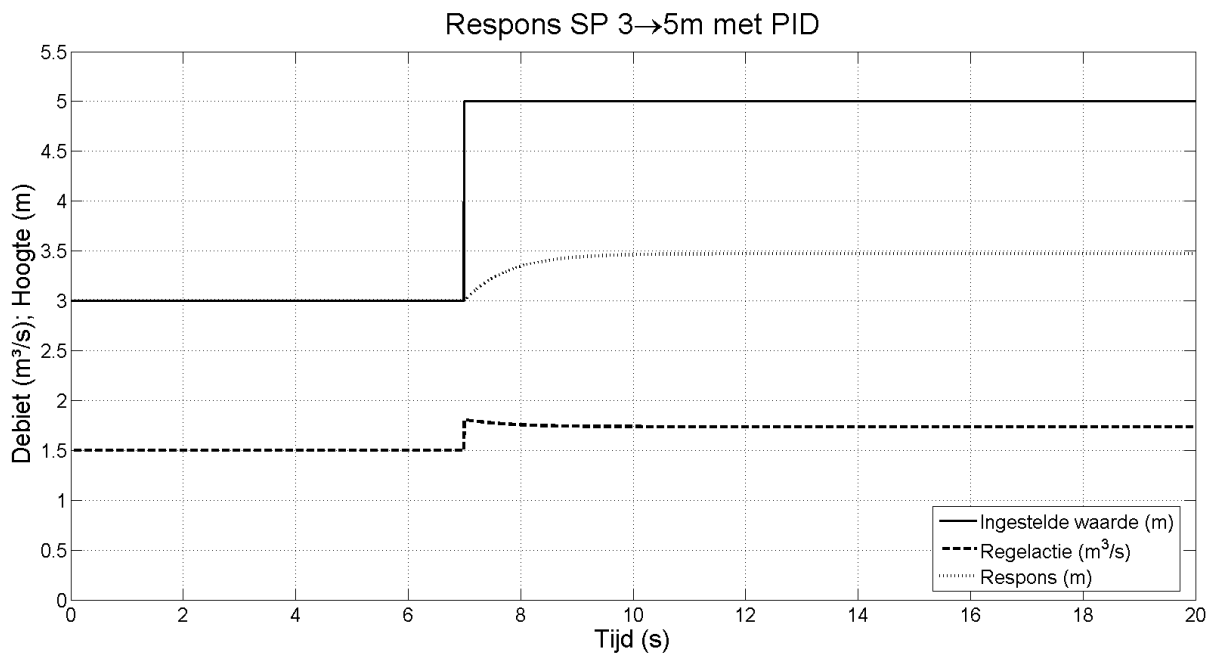
Tabel 4: PID-instellingen lineaire tank op basis van open lus

Regelaar	$K_p$	$\tau_i$	$\tau_d$
P	0,1545		
PD	0,309		0,5
PI	0,18025	1,236	
PID	0,309	1,03	0,5



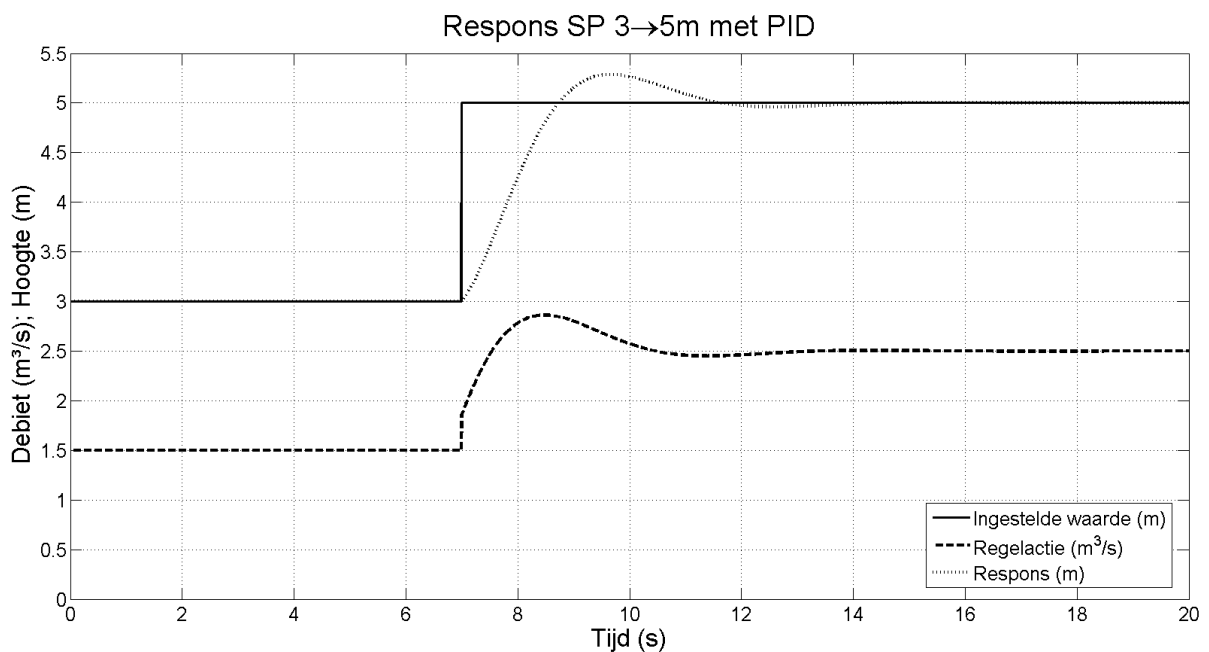
Figuur 23: Respons lineaire tank open lus met constructie voor instelregels

Met de instellingen uit de voorgaande tabel worden een P- en een PI-regelaar ingesteld voor dit proces. Een D-actie wordt overbodig geacht gezien de onmiddellijke respons van het proces. Bij het proces waarbij er in de PID-regelaar enkel een P-actie wordt ingesteld (zie Tabel 4) wordt Figuur 24 verkregen.



Figuur 24: Respons lineaire tank: enkel P-actie op basis van open lus

De volgende instelling gebeurt op basis van de rij van PI in Tabel 4. Enkel de P- en de I-actie worden ingesteld. Figuur 25 geeft het resultaat weer van deze simulatie. Het grote verschil wordt hier onmiddellijk duidelijk tussen de P-regelaar en de PI-regelaar. Bij de P-regelaar blijft er een grote statische fout tussen de gewenste waarde en de waarde die het proces effectief zal bereiken. Wanneer de regelaar een P-én I-factor krijgt ingesteld zal het proces geen statische fout vertonen en de ingestelde waarde na een kleine *overshoot* bereiken.



Figuur 25: Respons lineaire tank: PI-actie op basis van open lus



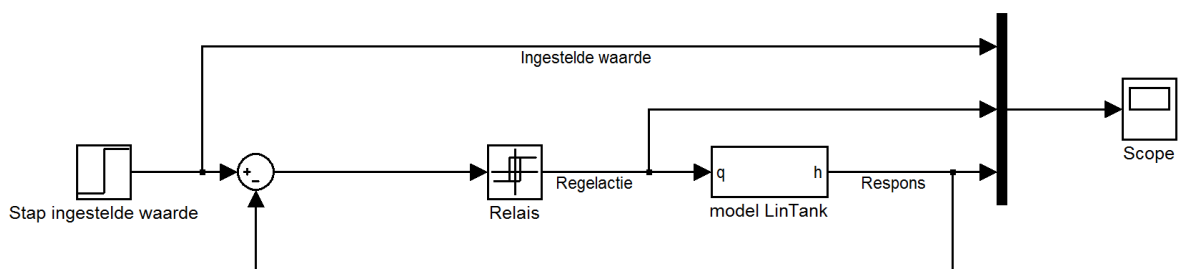
## Gesloten lus tuning

### Ziegler-Nichols gesloten lus tuning met ultieme versterkingsfactor

Via de methode die in de literatuurstudie uit de doeken wordt gedaan worden de instellingen van de PID-regelaar gezocht. Hierbij dient een ultieme versterkingsfactor gezocht te worden. Bij dit proces is het niet mogelijk om de ultieme versterkingsfactor te achterhalen omdat het proces een proces van de 1<sup>e</sup> orde is.

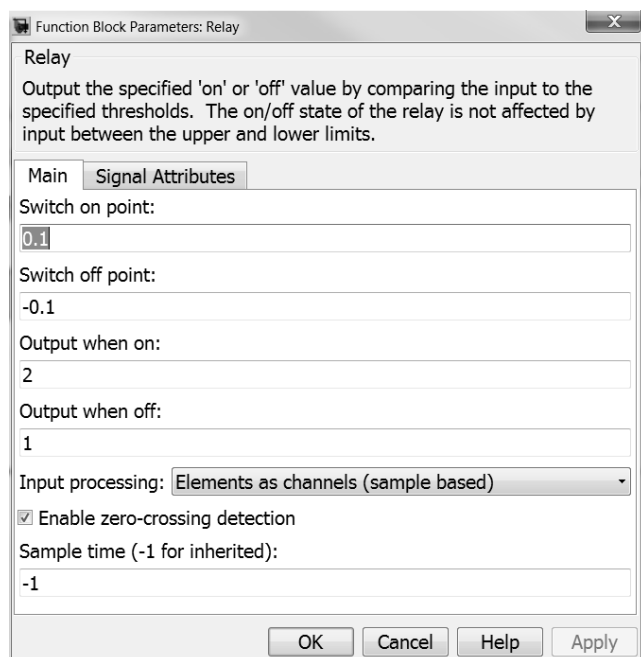
### Relais tuning

Relais-tuning kan worden toegepast in elk systeem, ook bij dit systeem. De gebruikte methode wordt beschreven in de literatuurstudie. De opbouw van het gebruikte model in Simulink ziet er als volgt uit:



Figuur 26: Blokschema relaistuning in Simulink

Bij dit systeem wordt de relais ingesteld zoals in Figuur 27, waarbij *switch on point* en *switch off point* staan voor de fout ten opzichte van de ingestelde waarde. *Output when on* en *output when off* geven aan wat de regelactie bedraagt wanneer de fout positief of negatief is. De gedachte hierachter is dat bij een negatieve afwijking er een hoger debiet naar de tank moet gestuurd worden en vice versa.

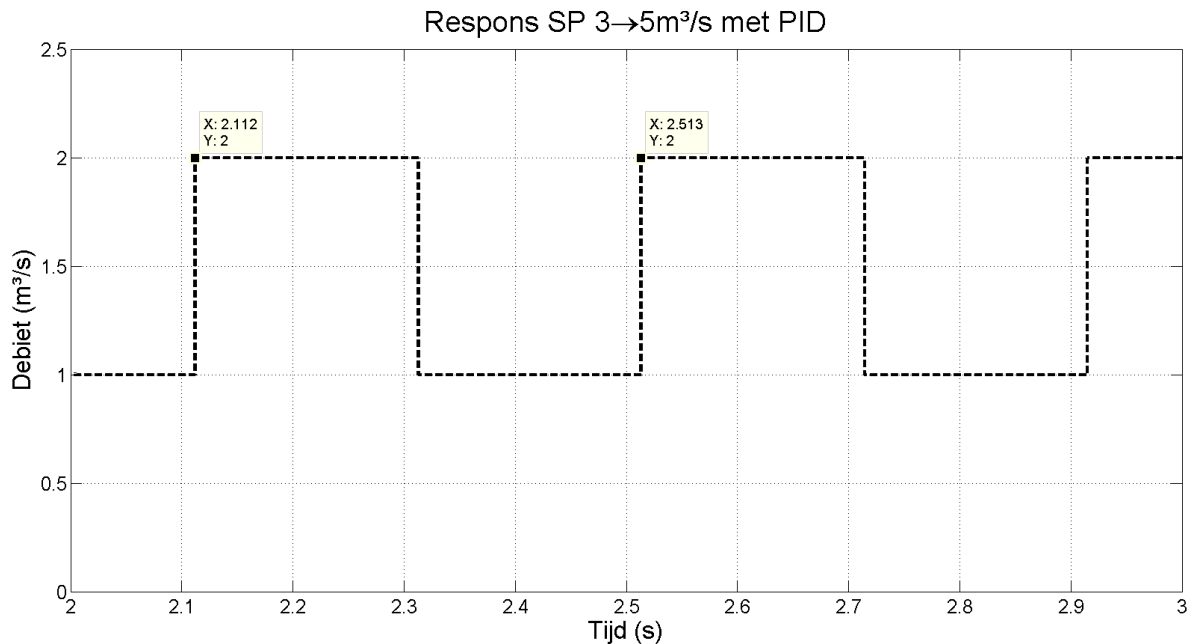


Figuur 27: Instellingen Relais

Wanneer de simulatie wordt uitgevoerd met deze instellingen worden volgende factoren bekomen:

- $A = 0,2$  (verschil tussen *switch on point* en *switch off point*);
- $B = 0,5$  (verschil tussen hoog en laag debiet gedeeld door 2);
- $P_U = 2,513 - 2,112 = 0,401\text{s}$  (tijd tussen hoge en lage waarde).

Dit geeft voor  $K_U = \frac{4B}{\pi A} = \frac{2}{0,628} = 3,14$



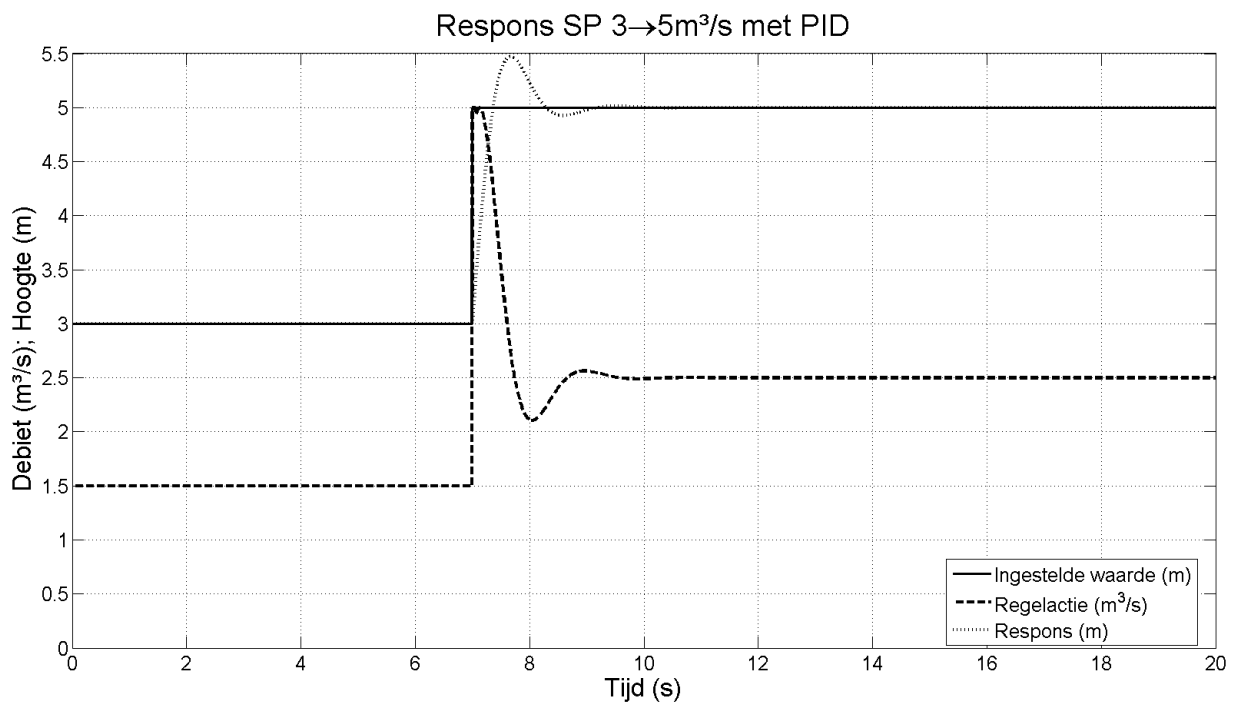
Figuur 28: Regelactie bij de lineaire tank door relais

Op basis van de tabel in de literatuurstudie geeft dit voor de instellingen van een PID-regelaar volgens de 1<sup>e</sup> rij volgend resultaat:

Tabel 5: Instellingen PID op basis van relais

	$K_P$	$\tau_i$	$\tau_d$
Instelling	1,884	0,2005	0,050125

De reactie van het proces op de PID-regelaar ingesteld met bovenstaande gegevens geeft volgende figuur:

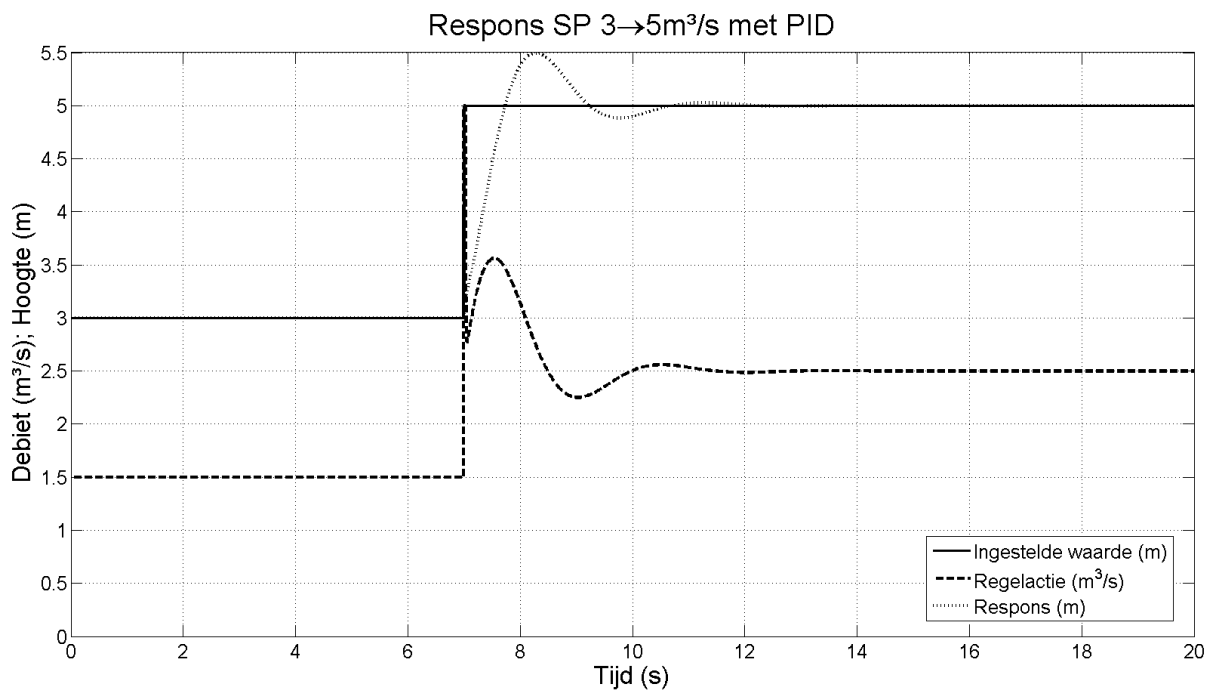


Figuur 29: Respons systeem: PID-instelling op basis van relais

Het is duidelijk dat er een *overshoot* optreedt en het proces trager naar een nieuwe stabiele waarde nadert op de nieuwe ingestelde waarde. Daarom wordt de simulatie herhaald met de instellingen die te vinden zijn in de tabel op de rij “geen overshoot”:

	$K_P$	$\tau_i$	$\tau_d$
Instelling	0,628	0,2005	0,134

Dit geeft volgend resultaat, waarbij opvalt dat er nog steeds overshoot aanwezig is en het systeem zelfs trager een nieuwe stabiele waarde haalt:

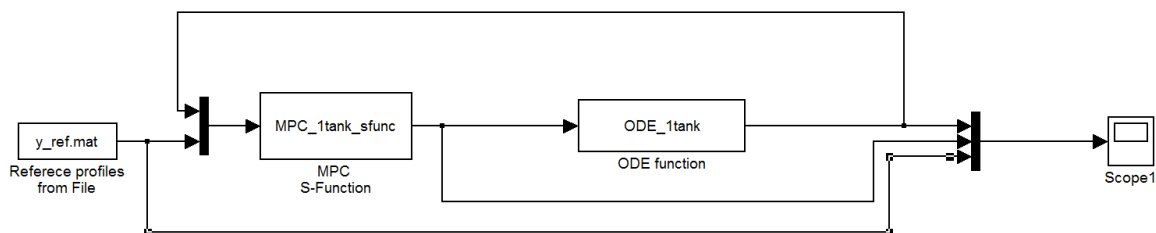


Figuur 30: Respons systeem: PID-instelling op basis van relais: instellingen zonder overshoot

Het beste resultaat met de instellingen van de PID worden behaald met de instellingen op basis van de relaisinstelregels. Hierbij wordt snel en met een matige *overshoot* een nieuwe gewenste waarde behaald.

#### 4.1.3 Geavanceerde procescontrole: MPC

De resultaten van de PID-regelaar zijn nu gekend maar moeten nog vergeleken worden met het regelen van het proces met een MPC. Eerst wordt aangehaald wat de mogelijkheden zijn van de MPC voor wat betreft de oplegging van beperkingen en de relevantie ervan. Het gebruikte model is identiek als het model toegepast bij de controle met behulp van PID. Ook de aangelegde stap van de ingestelde waarde (hoogte van de tank) is hetzelfde als bij de simulatie met PID, zodat de 2 regelmethodes goed te vergelijken zijn. In het model zitten 2 S-functies. 1 voor de MPC, de linkse S-functie, en 1 S-functie voor het beschrijven van het proces. De inhoud van deze functies kunnen in bijlage gevonden worden. De inhoud van de S-functie van het proces is te vinden in de bijlagen onder 8.2. De inhoud van de S-functie van de MPC is te vinden in de bijlagen onder 8.3. In de MPC wordt het gediscretiseerde toestandsruimtemodel gebruikt, in het proces wordt de massabalans gebruikt in de vorm van een ODE. Dit gediscretiseerde toestandsruimtemodel is te vinden in de bijlagen onder 8.4.



Figuur 31: Model in Simulink: Lineaire tank

Bij het toepassen van MPC is het mogelijk om bepaalde parameters in te stellen. Deze worden even in een lijst weergegeven zodat deze duidelijk zijn:

- Sampletijd  $t_s$ : de tijd die verloopt tussen iedere meting. Deze tijd moet lang genoeg zijn om de berekening en regelactie die op de berekening volgt uit te voeren
- Beperkingen: afhankelijk van het systeem kan het nodig zijn om beperkingen op te leggen voor zowel de gecontroleerde als de gemanipuleerde variabelen. Dit kan bijvoorbeeld een maximaal verschil zijn in debiet, temperatuur, druk of andere parameters. Naast het verschil kan ook op de absolute waarde van debiet, temperatuur of druk een minimum of maximum opgelegd worden. Ook is het mogelijk om minimale en maximale waarden op te leggen voor de veranderingssnelheid van de gemanipuleerde variabele. Dit kan van pas komen omdat in een reëel proces een variabele ook niet oneindig (snel) aan te passen is.
- Voorspellingshorizon  $H_p$ : aantal stappen die de regelaar vooruit kijkt en op basis van het model het verloop van het proces voorspelt. De tijd die hiermee overeenkomt kan berekend worden uit volgende formule: tijd = aantal stappen  $\cdot$  sampletijd.
- Controlehorizon  $H_c$ : aantal stappen die de regelaar vooruit kijkt om de controleacties te berekenen.
- De matrix  $R$ : wordt meestal gezien als instelbare parameter om een bevredigende dynamische reactie te krijgen in het systeem.  $R$  is een matrix die in de kwadratische vergelijking die geoptimaliseerd wordt in rekening wordt gebracht om de te berekenen controleparameter af te straffen op de verandering die de stap ervoor werd aangelegd op deze controleparameter.
- De matrix  $Q$ : wordt meestal gezien als instelbare parameter om een bevredigende dynamische reactie te krijgen in het systeem.  $Q$  is een matrix die in de kwadratische vergelijking die geoptimaliseerd wordt in rekening wordt gebracht om gewichten te geven aan de verschillende te volgen referenties. Een temperatuur heeft bijvoorbeeld een andere grootteorde dan concentratie.

Als deze parameters vergeleken worden met de parameters die bij een PID-regelaar ingesteld kunnen worden, is een duidelijk verschil te merken. De PID-regelaar heeft 3 parameters (P-, I-, en D-actie) die bepaald worden om de eigenschappen van het systeem te beschrijven. Dit gebeurt bij een MPC met behulp van het model. De parameters die zonet besproken zijn voor MPC zijn voor een groot deel zelf in te vullen. Hier speelt ervaring een rol, maar ook het begrijpen van het systeem waarmee gewerkt wordt. Een snel systeem heeft bijvoorbeeld beter een korte sampletijd en heeft een niet zo grote voorspellingshorizon nodig. Een traag systeem daarentegen neemt genoeg met een grotere sampletijd en heeft een grotere voorspellingshorizon nodig.

De beperkingen die ingesteld worden gaan over de maximale stapgrootte die in 1 tijdsstap gedaan kan worden door het regelorgaan, over de maximale regelactie die gebeurt over het

voorspellingsinterval en de maximale regelactie die kan gebeuren in het totaal (bijvoorbeeld door het maximale debiet van een pomp) tot beperkingen in de toestanden van het systeem, zoals een minimale en maximale tankhoogte die gerespecteerd dienen te worden. Wanneer een tank een hoogte heeft van 5m zou deze overlopen indien de regelaar een ingestelde waarde van 6m krijgt. Door de beperking in de regelaar in te voeren worden fouten die tijdens de bediening van de regelaar (verandering van instelling) mogelijk gebeuren uitgesloten.

Hieronder worden de resultaten vergeleken. In de tabel worden eerst de parameters weergegeven, waarna de grafiek volgt met het resultaat. Tabel 6 bevat de algemene parameters die voor alle simulaties identiek zijn. Iedere situatie die wordt beschreven en gesimuleerd wordt in een overzicht weergegeven in Tabel 7 . Er is hier geen geval gesimuleerd waarbij geen beperkingen worden opgelegd. Deze wordt wel uitgevoerd met het model van de niet-lineaire tank.

Tabel 6: Instellingen MPC simulaties lineaire tank

$t_s$	0,1
Hp	6
Hc	4
R	0
Q	1

Tabel 7 geeft het overzicht weer van de beschreven gevallen. U geeft beperkingen voor de gemanipuleerde variabele, het voedingsdebiet in  $m^3/s$ .  $x$  geeft beperkingen op de toestand, hetgeen hier de vloeistofhoogte in de tank is. De eerste beperking is de maximale sprong die de regelactie iedere tijdsstap mag maken. De 2<sup>e</sup> beperking is de maximale stijging of daling die de regelactie mag doen tijdens de volledige voorspelling. De voorlaatste kolom geeft een beperking op de absolute waarden van het debiet en de laatste kolom geeft een beperking op de absolute waarden van de hoogte van de tank. Deze opbouw van beperkingen heeft tot doel een logische gestructureerde opbouw te visualiseren van de mogelijkheden van de beperkingen.

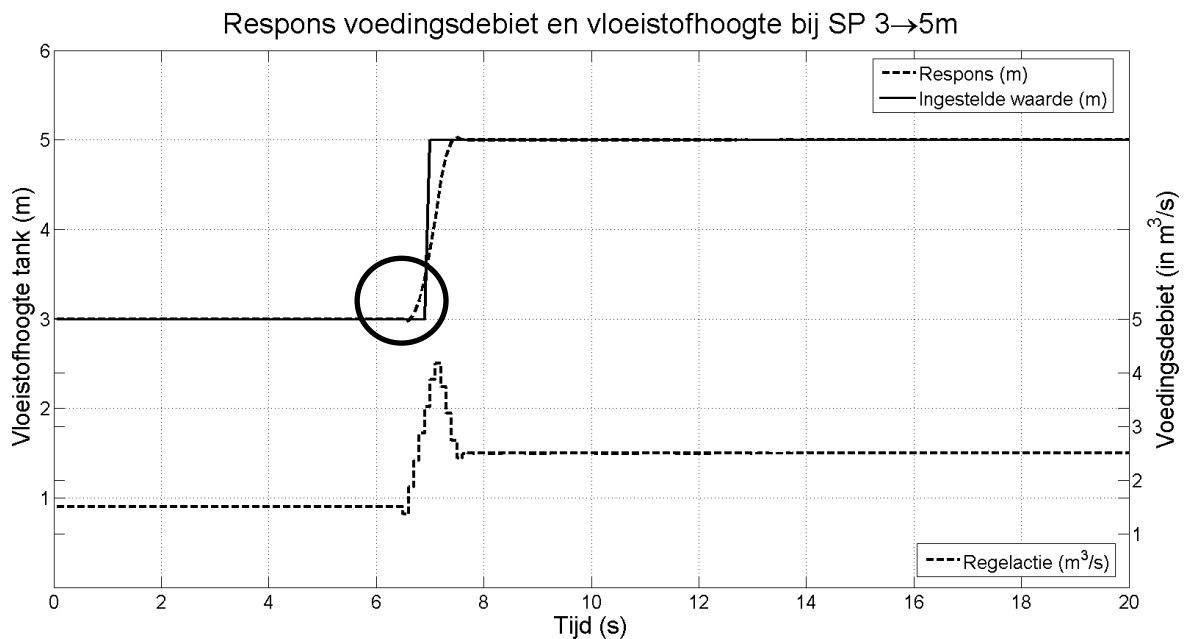
Tabel 7: Overzicht gevallen MPC lineaire tank

	$\frac{\Delta u}{t_s}$	$\frac{\Delta u_{max}}{Hp \cdot Ts}$	$u_{max}$ en $u_{min}$	$x_{max}$ en $x_{min}$
Geval 1	0,5	/	/	/
Geval 2	0,5	1,5	/	/
Geval 3	0,5	/	3,3 en 0	/
Geval 4	0,5	/	3,3 en 0	4,7 en 0

De eerste situatie heeft zoals reeds vermeld enkel een beperking op de stapgrootte die iedere staalnametijd maximaal aan de regelactie opgelegd kan worden. Een regelorgaan zoals een klep

is immers niet oneindig snel en een pomp kan ook niet altijd onmiddellijk het gewenste debiet afleveren. Verder worden in Tabel 6 de andere instellingen weergegeven die tijdens de simulaties met de lineaire tank gelden. Bij de situaties met andere beperkingen blijven deze parameters constant.

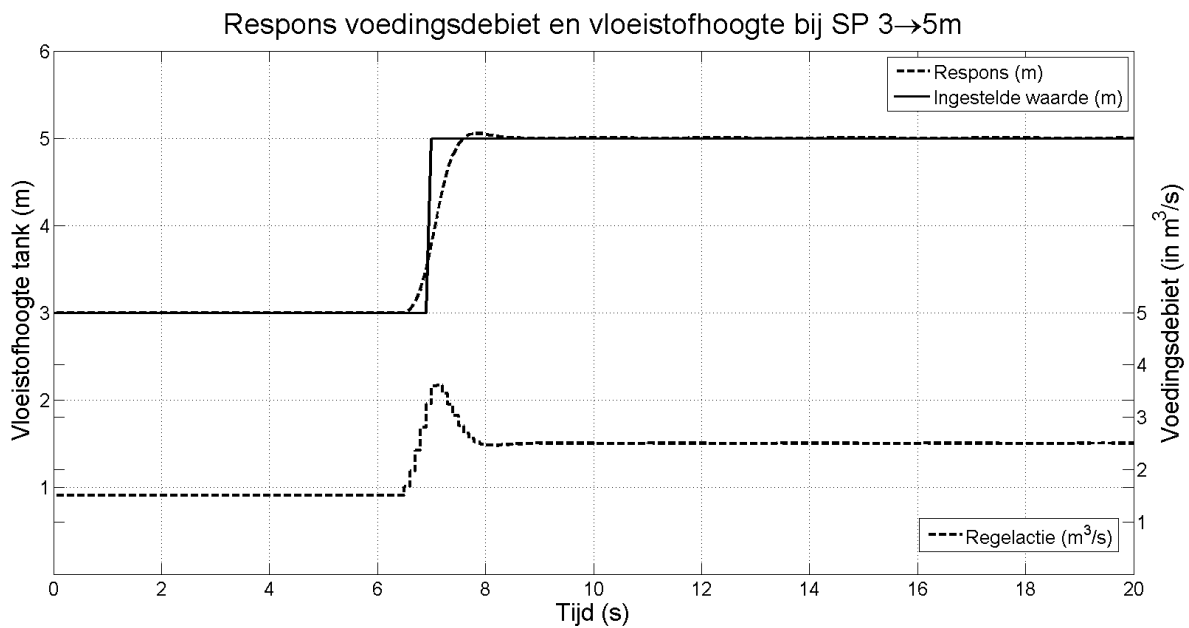
Figuur 32 geeft het resultaat weer van de simulatie met deze instellingen. Wat hier opvalt is dat de regelaar reeds voor de stap van de ingestelde waarde een regulatie begint uit te voeren en dat het proces ook al voor deze stap naar de nieuwe ingestelde waarde wordt geregeld. De stap die iedere  $t_s$  wordt aangelegd is steeds maximaal  $0,5\text{m}^3/\text{s}$  zoals uit de grafiek is af te leiden. Ook treedt er zo goed als geen overshoot op.



Figuur 32: Respons systeem lineaire tank: MPC geval 1

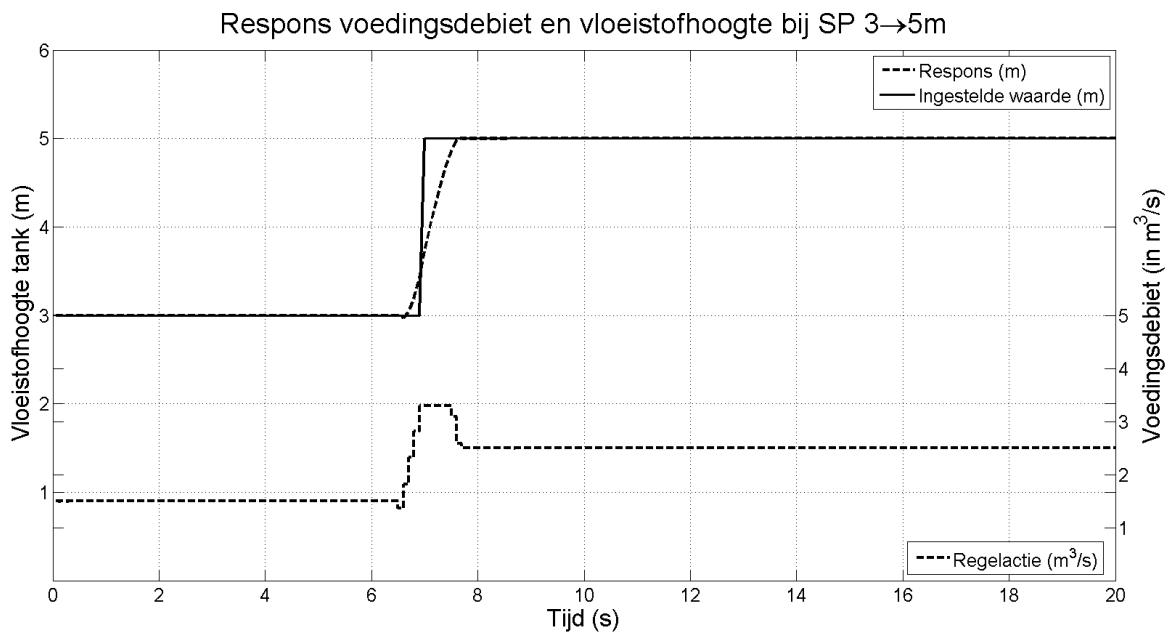
Bij de 2<sup>e</sup> situatie wordt een beperking opgelegd aan de maximale grootte die iedere volledige voorspelling kan vervatten. Het model maakt immers een voorspelling over de toekomst, opgelegd door de controlehorizon. Een controlehorizon van 4 stappen en een tijdsstap van 0,1s maakt dat de stap van de regelactie 0,4s voorloopt op de tijd dat de stap effectief wordt aangelegd. Dit is zichtbaar op alle grafieken. Het is nuttig om de maximale stapgrootte tijdens deze 0,4s als beperking op te leggen.

Dit geeft volgend resultaat. Als gefocust wordt op de tijd tussen 6 en 8,5 seconden bij deze situatie en de vorige situatie wordt vastgesteld dat het systeem er in deze situatie met deze beperkingen het systeem iets trager de ingestelde waarde haalt. De verklaring ligt het feit dat voor de eerste situatie in 0,4 seconden een stapverschil aangelegd kon worden van  $2\text{m}^3/\text{s}$ , waar dit bij de 2<sup>e</sup> situatie slechts  $1,5\text{m}^3/\text{s}$  is.



Figuur 33: Respons systeem lineaire tank: MPC geval 2

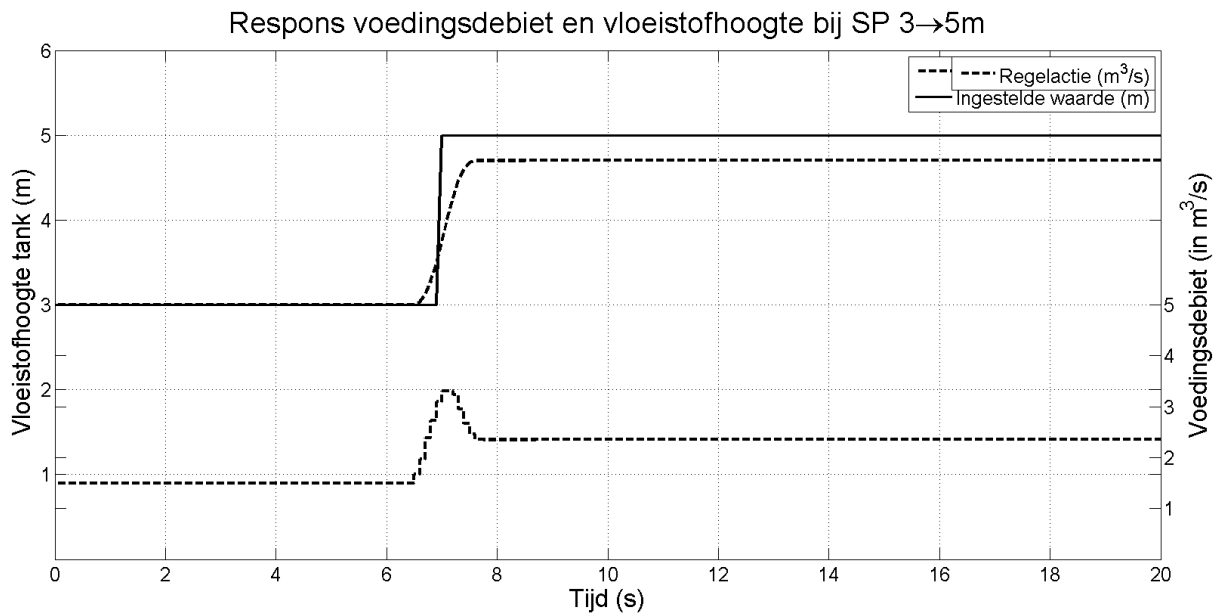
De voorlaatste beperking die wordt opgelegd zijn uitersten in het debiet. Deze worden opgelegd samen met de maximale regelactie die iedere staalnametijd mogelijk is. In de grafiek is duidelijk te zien dat de lijn van de regelactie een saturatie bereikt op  $3,3\text{m}^3/\text{s}$ . De beperking die in vorige situatie werd toegevoegd wordt verwijderd omdat deze praktisch gezien geen grote meerwaarde biedt bij het vergelijken van de situaties.



Figuur 34: Respons systeem lineaire tank: MPC geval 3

De laatste beperking die wordt opgelegd is de maximale hoogte van de tank. Deze is ingesteld op 4,7m, bewust lager dan de nieuwe ingestelde waarde. De MPC houdt zich aan de opgelegde beperking en houdt het systeem op een waarde die toegelaten wordt en het dichtst aanleunt bij de gewenste waarde.

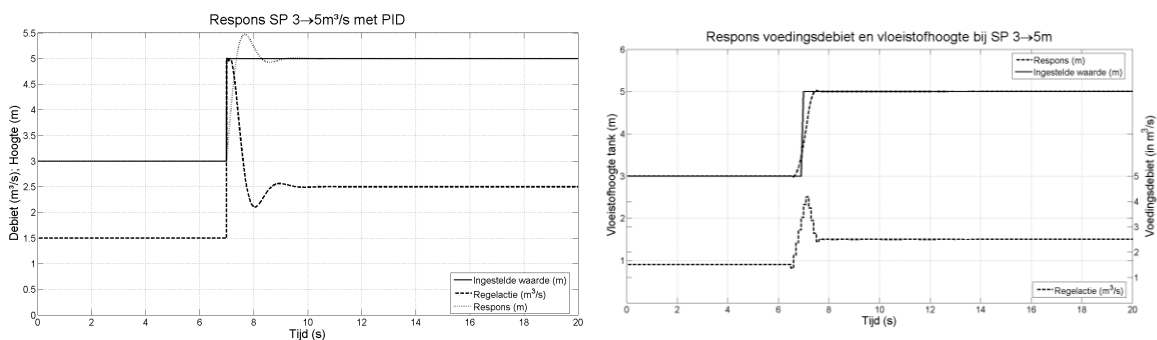




Figuur 35: Respons systeem lineaire tank: MPC geval 4

### Conclusie

Om een concrete vergelijking te maken worden de beste en meest representatieve casussen langs elkaar gelegd. Voor de PID is dit de instelling op basis van de relaisinstelregels, voor de MPC is dit de casus met de maximale stapgrootte in de regelactie. De MPC krijgt daardoor een vorm van traagheid die in een reëel systeem ook zou zitten in de vorm van traagheid van het regelorgaan. Als de MPC geen beperking had dan zou de tank onmiddellijk in 0,1s een nieuwe ingestelde hoogte halen, dit wordt kort aangehaald bij de simulaties van de niet-lineaire tank. Deze traagheid wordt hier gebruikt om de realiteit meer te benaderen. Beide figuren worden kort herhaald om ze langs elkaar te kunnen vergelijken.



Met PID behaalt het systeem na 2 seconden de nieuwe gewenste waarde terwijl hier bij MPC 1 seconde voor nodig is. Waar de PID vloeiend de gemanipuleerde stroom regelt, gebeurt dit bij de MPC in werkelijkheid wel voor zover de regelklep een bepaalde traagheid bezit. De regelaar zal echter discrete stappen als regelactie geven, wat te zien is in de grafiek van de regelactie bij de MPC. Het grote voordeel dat hierbij komt kijken is dat voor eenvoudige en goedgekende systemen geen instelling van de PID-regelaar nodig is als MPC gebruikt zou worden.

Daarnaast is het systeem dat met PID geregeld wordt onderhevig aan een aanzienlijke *overshoot*, een fenomeen dat bij de MPC onbestaande is.

## 4.2 Niet-lineaire tank

De casus van de lineaire tank bevatte een model en proces dat lineair was. De volgende stap die uitgevoerd wordt is een niet-lineariteit in het model invoegen, zie de gevalstudies. Het opzet van de simulaties is identiek aan het proces met de lineaire tank, net zoals bepaalde gebruikte figuren in Simulink. Het enige dat hiertussen verandert is het feit dat het gebruikte model nu niet meer van een lineaire tank is maar van een niet-lineaire tank. Het model dat in de MPC gebruikt wordt is dus ook dat van een lineaire tank.

### 4.2.1 Klassieke regeling: PID

#### Algemene instellingen

Bij de simulaties van de niet-lineaire tank wordt geen gebruik gemaakt van de functie *antiresetwindup*. Gezien de korte simulatietijd is dit overbodig. Ook wordt er een initiële waarde toegekend aan de output van het PID-blok. De uitleg daaromtrent is te vinden onder de resultaten van de lineaire tank, bij de instellingen van PID. De waarde van het initiële stabiele ingangsdebiet wordt ad hoc berekend uit  $\sqrt{\frac{h}{R^2}}$ .

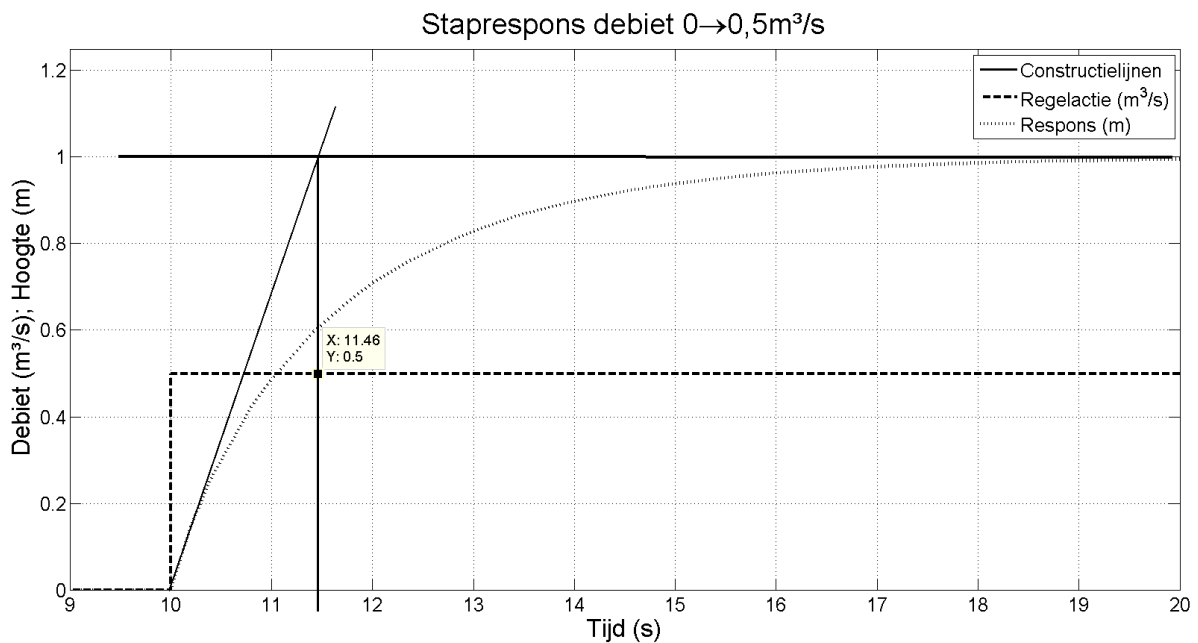
#### Open lus instelling

De gevolgde methode kan in de literatuurstudie gevonden worden onder het luik klassieke regeling: open lus instelling. De gevonden waarden voor  $\tau_d$  en  $\tau_p$  zijn 1 (indien  $\tau_d$  0 is, stelt men  $\tau_d$  gelijk aan 1) en  $11,08-10=1,08$ .  $K_p$ , de versterkingsfactor van het proces, is de resulterende proceswaarde/de aangelegde variatie in inputvariabele (hier het voedingsdebiet).  $K_p$  is dus  $1/0,5=2$ .

Omgerekend geeft dit voor de bekomen waarden voor deze casus:

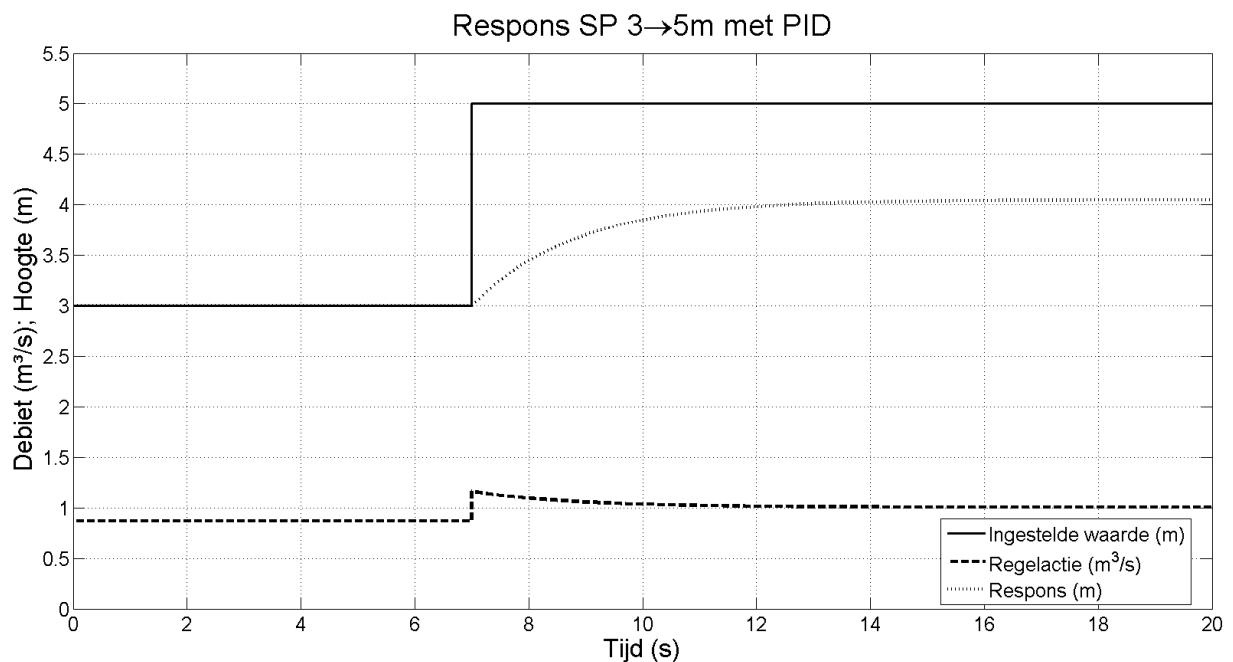
Tabel 8: PID-instellingen niet-lineaire tank op basis van open lus

Regelaar	$K_p$	$\tau_i$	$\tau_d$
P	0,147541		
PD	0,295082		0,5
PI	0,172131	1,752	
PID	0,295082	1,46	0,5



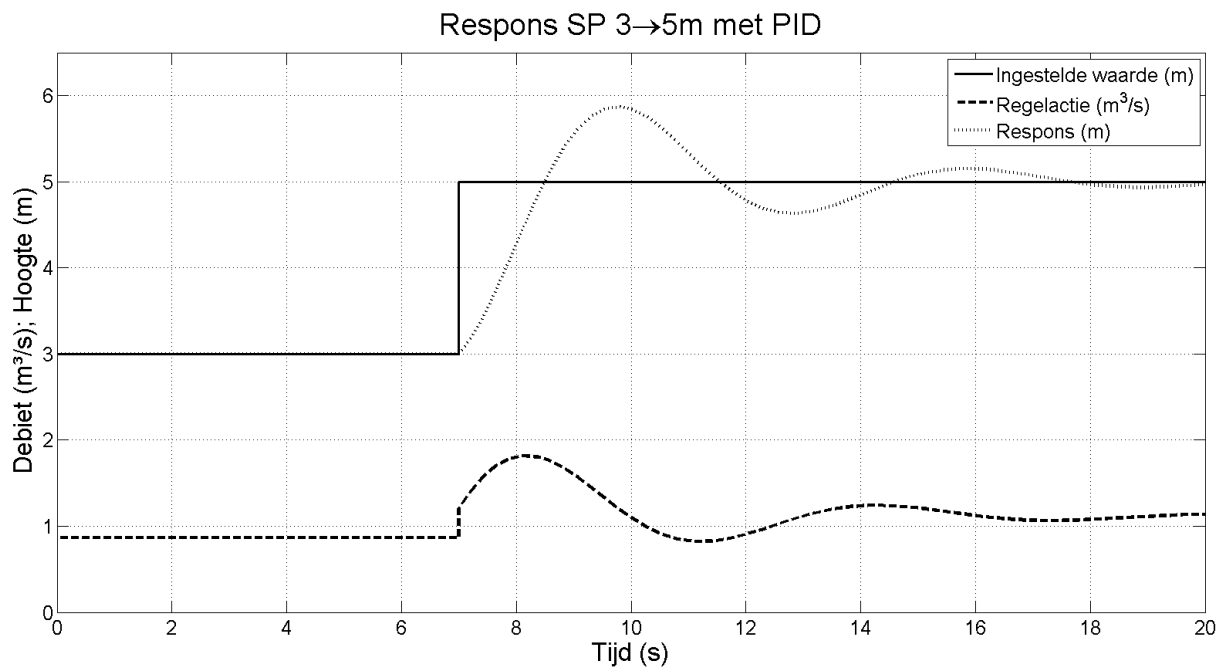
Figuur 36: Respons niet-lineaire tank open lus met constructie voor instelregels

Met de instellingen uit de voorgaande tabel worden een P- en een PI-regelaar ingesteld voor dit proces. Een D-actie wordt overbodig geacht gezien de onmiddellijke respons van het proces. Bij het proces waarbij de PID-regelaar enkel een P-actie krijgt ingesteld



Figuur 37: Respons lineaire tank met PID: wijziging ingestelde waarde, enkel P-actie ingesteld

De instellingen van de PI-regelaar wordt hieronder weergegeven. Hierbij valt op dat de  $\tau_i$  in de regelaar van Simulink in wordt gegeven als  $1/\tau_i$ . De grafiek van deze simulatie met behulp van een PI-regelaar wordt weergegeven in Figuur 38. Het grote verschil wordt hier onmiddellijk duidelijk tussen de P-regelaar en de PI-regelaar. Bij de P-regelaar blijft er een statische fout tussen de gewenste waarde en de waarde die het proces effectief zal halen. De PI-regelaar daarentegen zal geen statische fout vertonen en met lichte overshoot zal het proces de ingestelde gewenste waarde benaderen.



Figuur 38: Respons lineaire tank met PID: wijziging ingestelde waarde: PI-actie

### Gesloten lus tuning

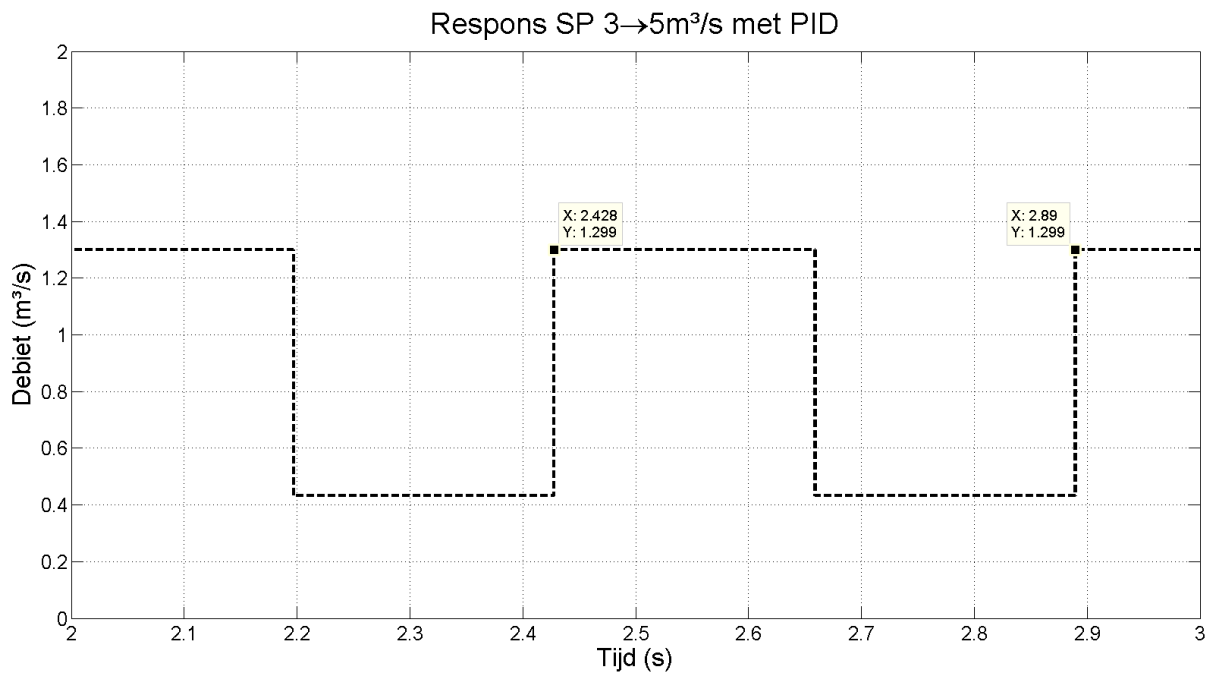
#### Ziegler-Nichols gesloten lus tuning met ultieme versterkingsfactor

Via de methode die in de literatuurstudie uit de doeken wordt gedaan worden de instellingen van de PID-regelaar gezocht. Hierbij dient een ultieme versterkingsfactor gezocht te worden. Bij dit proces is het niet mogelijk om zo een ultieme versterkingsfactor te achterhalen omdat het proces een proces van de 1<sup>e</sup> orde is.

#### Relais tuning

Ook bij relais tuning wordt de methode gebruikt beschreven in de literatuurstudie. De factoren die in de relais worden ingegeven zijn voor het *switch on- en switch off-point* 0,1 en -0,1. De *output when on* is 1,5 keer de stabiele waarde op een vloeistofhoogte van  $3\text{m} \left(\frac{\sqrt{3}}{2}\right)$ . De *output when off* is 0,5 keer de stabiele waarde op een vloeistofhoogte van  $3\text{m} \left(\frac{\sqrt{3}}{2}\right)$ .

Deze instellingen geven de regelactie te zien in Figuur 39.



Figuur 39: Regelactie bij de niet-lineaire tank door relais

- $A = 0,2$  (verschil tussen *switch on point* en *switch off point*)
- $B = 0,866$  (verschil tussen hoog en laag debiet gedeeld door 2)
- $P_u = 2,89 - 2,428 = 0,463s$  (tijd tussen hoge en lage waarde)

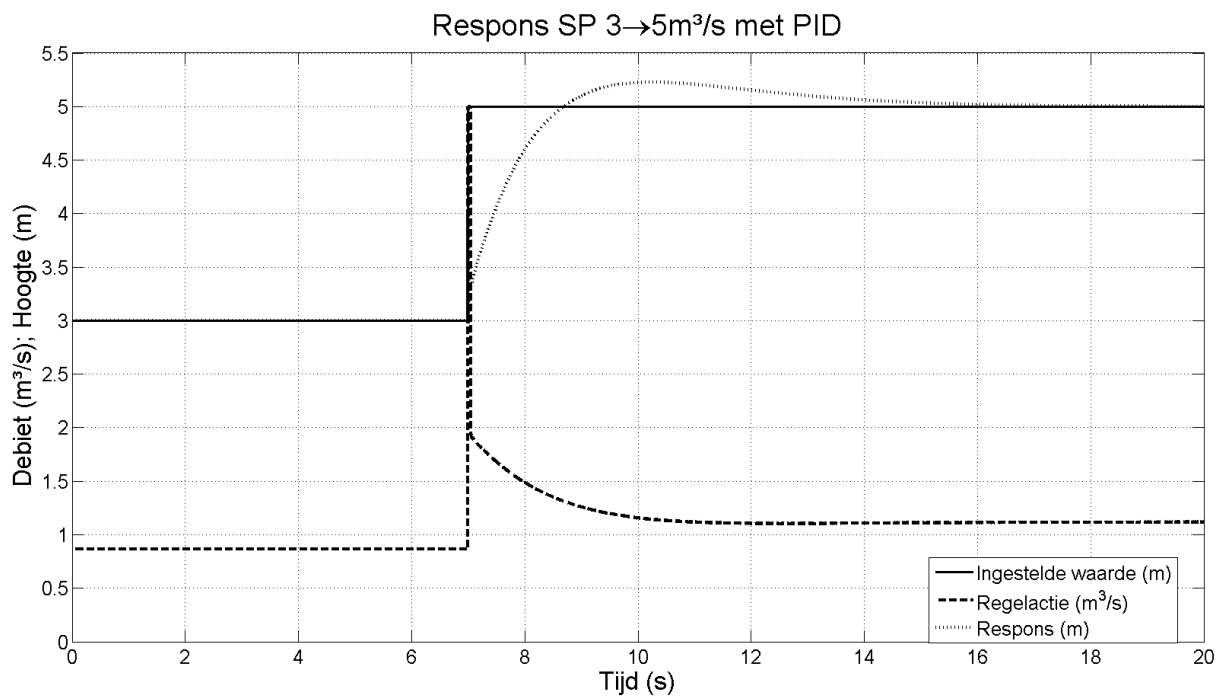
Dit geeft voor  $K_u = \frac{4B}{\pi A} = \frac{3,464}{0,628} = 5,5159$

Dit geeft op basis van de tabel uit de literatuurstudie volgende gegevens als gebruik gemaakt wordt van de rij "origineel":

	$K_p$	$\tau_i$	$\tau_d$
Instelling	3,30954	2,75795	0,6895

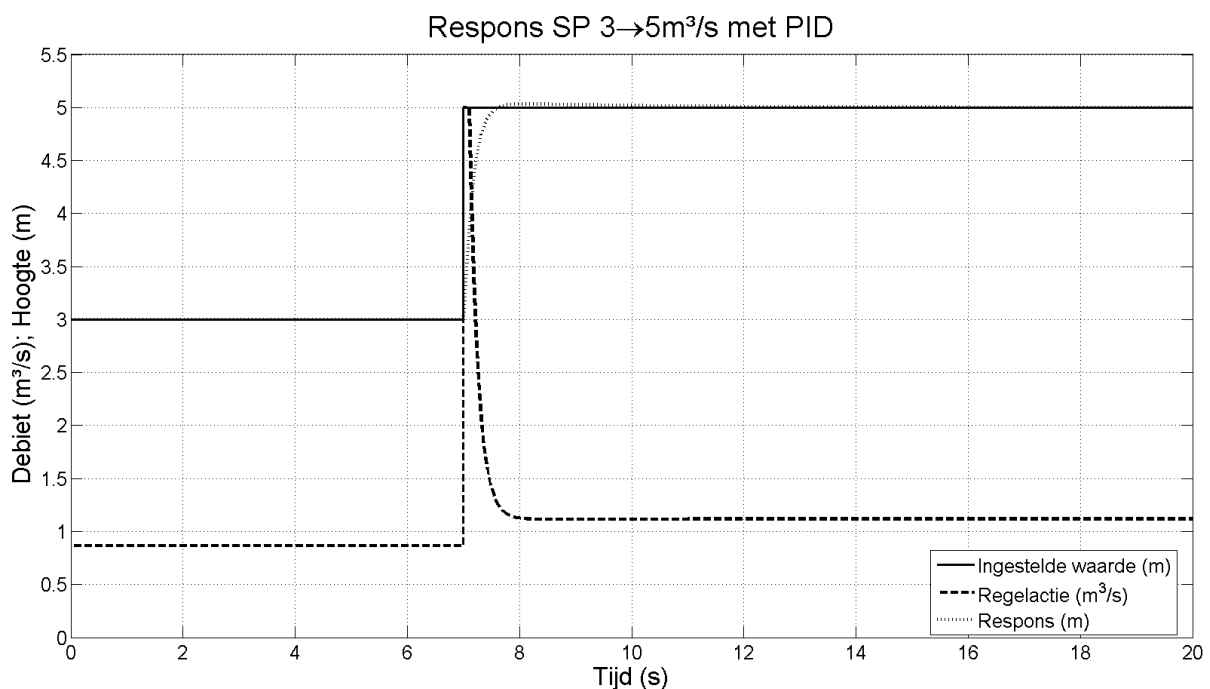
Figuur 40: Instellingen PID op basis van relais

De reactie van het proces op de PID-regelaar ingesteld met bovenstaande gegevens geeft Figuur 41. Het is duidelijk dat er een lange overshoot optreedt en het proces heel traag naar een nieuwe stabiele waarde nadert op de nieuwe ingestelde waarde:



Figuur 41: Respons systeem: PID-instellingen op basis van relais

Door de D-actie te verwijderen en de P-en I-actie te laten staan wordt gekeken of de trage regelactie niet sneller gemaakt kan worden. Deze nieuwe instelling geeft Figuur 42.



Figuur 42: Respons systeem:PI-instellingen op basis van relais

Het is duidelijk dat met behulp van deze instellingen het proces zeer snel een nieuwe ingestelde waarde bereikt. De beperking die weliswaar nog werd opgelegd in de PID-regelaar is het feit dat de minimale output  $0\text{m}^3/\text{s}$  is en de maximale output  $5\text{m}^3/\text{s}$ . Dit wordt gedaan om de simulatie een realistische tint te geven, in een echt proces kan een parameter ook niet oneindig worden aangepast.

## 4.2.2 Geavanceerde procescontrole: MPC

Nu de resultaten van het niet-lineaire proces geregeld met een PID gekend zijn wordt net zoals bij het lineaire proces nu een MPC gebruikt als regelaar. Daarna kunnen de resultaten met PID en MPC vergeleken worden met elkaar. De gevolgde werkwijze is vergelijkbaar als bij het lineaire proces. De screenshots betreffende de opbouw van het model e.d. zijn vergelijkbaar met het lineaire proces. In de blokken worden wel andere modellen gebruikt, naar het niet-lineaire proces.

### 4.2.2.1 Methode

Het gebruikte model is een niet-lineair model. Om dit proces in een MPC te gebruiken, dient dit eerst gelineariseerd te worden. Het is mogelijk om een niet-lineair model in een MPC te gebruiken, maar dit zou te ver gaan en nog meer rekenkracht vereisen. De linearisatie die wordt toegepast, gebeurt altijd in de buurt van een bepaalde werkingswaarde die als stabiele waarde voor dat proces gezien wordt. Er wordt gesproken van een model dat werkt met afwijkingsvariabelen in plaats van absolute variabelen. Hierbij wordt de verandering van de variabelen bekeken ten opzichte van de stabiele werkingswaarde, in plaats van de verandering van de absolute waarde van de variabelen.

### 4.2.2.2 Resultaten

De werkwijze die hier gevolgd wordt is identiek aan de werkwijze bij de simulaties van de lineaire tank. Toch wordt er hier weergegeven hoe het systeem reageert wanneer er geen beperkingen worden opgelegd, om een beeld te kunnen vormen. Hetgeen hierbij opvalt is het feit dat binnen 1 staalnametijd van 0,1s de nieuwe ingestelde waarde wordt bereikt. Hiervoor is wel een zeer kortstondig piekdebiet nodig van meer dan  $20\text{m}^3/\text{s}$ . Een debiet zal zelden of nooit in de praktijk zo snel zo een grote verandering aankunnen, daarom is het toevoegen van beperkingen niet alleen nodig om de verschillen aan te duiden, maar ook om de simulaties realistisch te houden. Net zoals het toevoegen van werkingsgrenzen bij de PID-regelaar.

Onderstaande tabel geeft de ingestelde parameters weer. Alle parameters onder de beperkingen blijven constant voor alle simulaties met het model van de niet-lineaire tank.

Tabel 9: Instellingen MPC simulaties niet-lineaire tank

$t_s$	0,1
$H_p$	6
$H_c$	4
R	0
Q	1

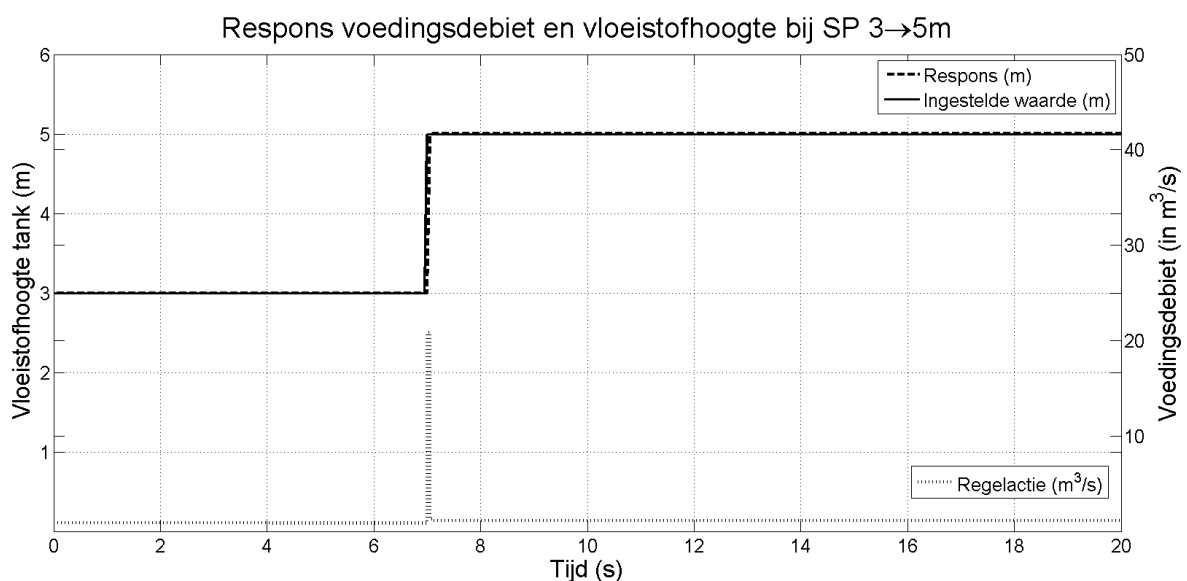
Tabel 9 geeft het overzicht weer. U geeft beperkingen voor de gemanipuleerde variabele, het voedingsdebiet in  $\text{m}^3/\text{s}$ .  $x$  geeft beperkingen op de toestand, hetgeen hier de vloeistofhoogte in de tank is. De eerste beperking is de maximale sprong die de regelactie iedere tijdsstap mag maken. De 2<sup>e</sup> beperking is de maximale stijging of daling die de regelactie mag doen tijdens de

volledige voorspelling. De voorlaatste kolom geeft een beperking op de absolute waarden van het debiet en de laatste kolom geeft een beperking op de absolute waarden van de hoogte van de tank.

Tabel 10: Overzicht gevallen MPC niet-lineaire tank

	$\frac{\Delta u}{t_s}$	$\frac{\Delta u_{max}}{Hp.Ts}$	$u_{max}$ en $u_{min}$	$x_{max}$ en $x_{min}$
Geval 1	/	/	/	/
Geval 2	0,5	/	/	/
Geval 3	0,5	/	3,3 en 0	/
Geval 4	0,5	/	3,3 en 0	4,7 en 0

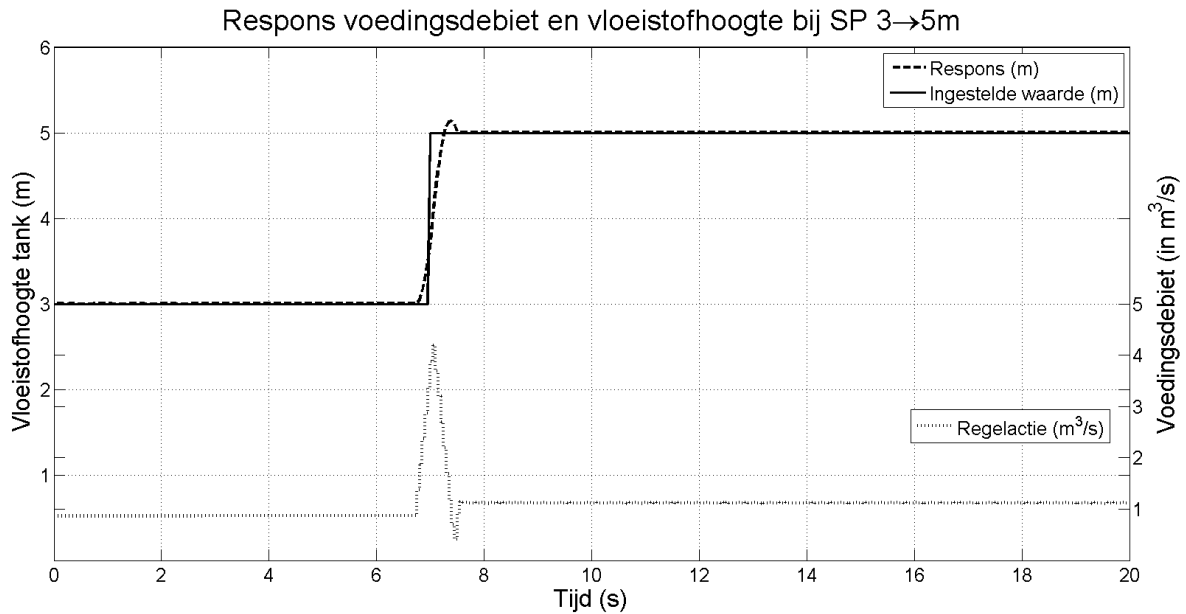
De eerste simulatie bevat geen beperkingen. Dit wordt gedaan om aan te geven wat de MPC doet in het geval er geen rekening wordt gehouden met de realiteit. Het is duidelijk in Figuur 43 dat het debiet zeer kortstondig naar een maximum van meer dan 20m<sup>3</sup>/s wordt gebracht. Daarna wordt het gebracht op een stabiele waarde waarbij de hoogte van 5m wordt aangehouden.



Figuur 43: Respons systeem niet-lineaire tank: MPC geval 1

De 2<sup>e</sup> beperking is weer de maximale verandering in debiet die iedere tijdsstap kan gebeuren. Het systeem reageert door snel, binnen 1s, van de oude naar de nieuwe ingestelde waarde te gaan. Bij de simulatie met PID op basis van de instellingen op basis van relais, wordt de stap overbrugd in 1,5s. Omdat hier een maximaal debiet van 5m<sup>3</sup>/s als regelactie plaatsvindt en bij deze situatie met MPC het maximum debiet deze waarde niet maximaal wordt gehaald, is het toch mogelijk om beide situaties te vergelijken.

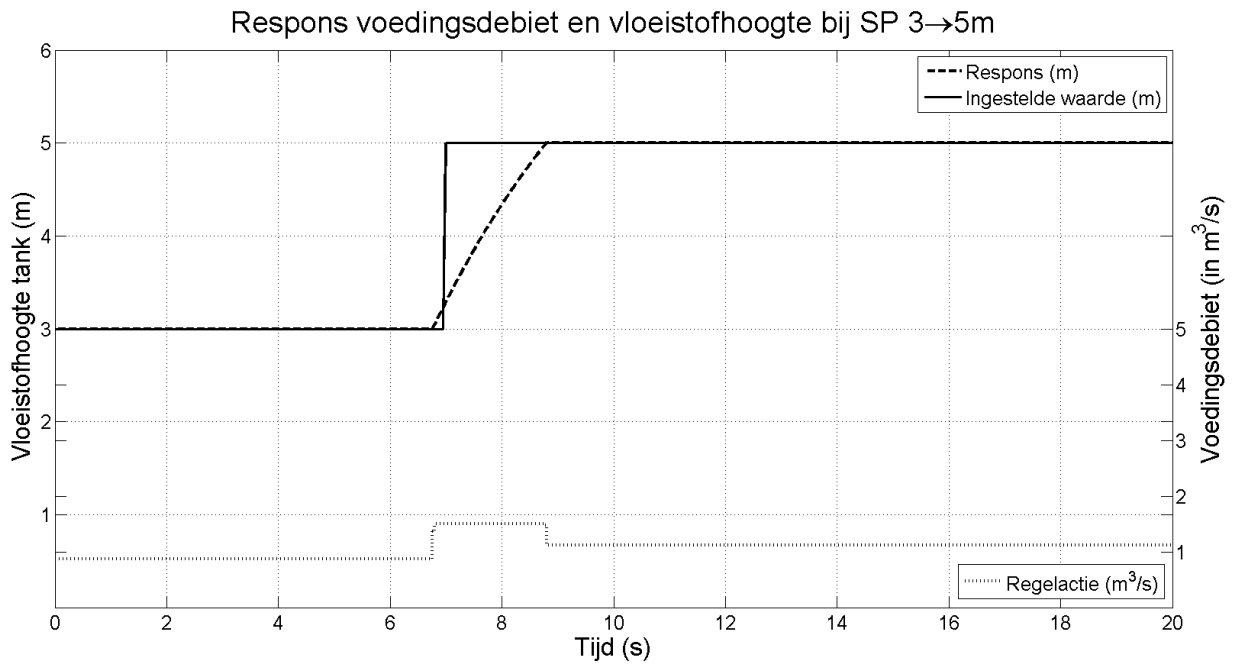




Figuur 44: Respons systeem niet-lineaire tank: MPC geval 2

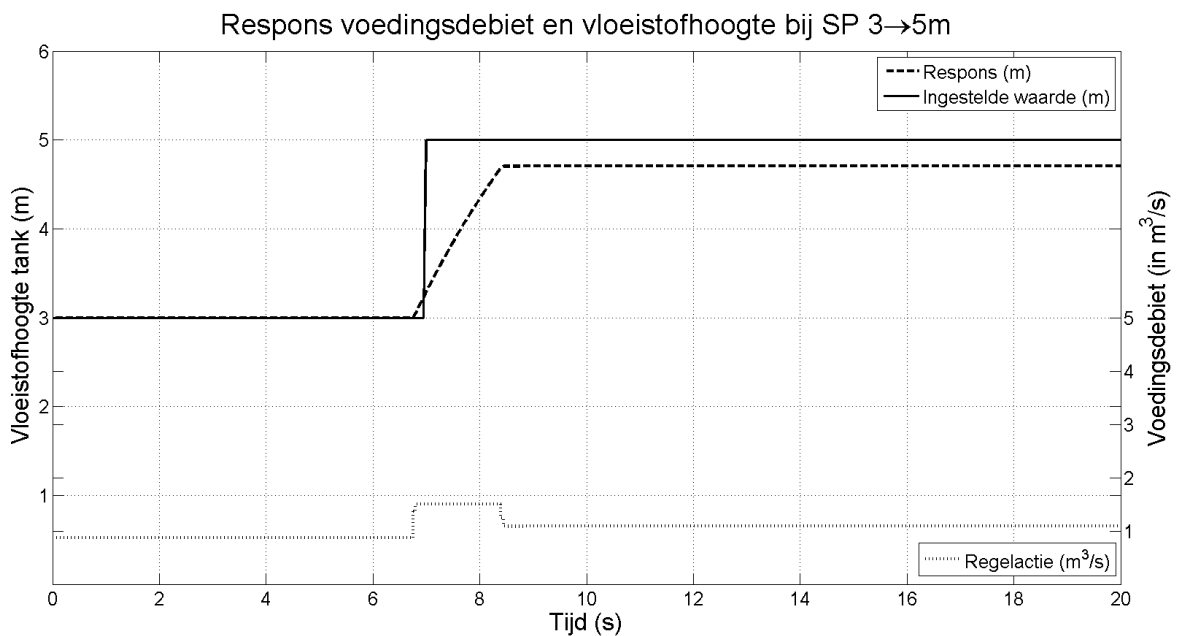
De situatie waarbij de controleactie beperkt werd door een minimale en maximale waarde over de voorspelde tijd wordt niet uitgevoerd wegens weinig relevant. Bij de lineaire tank werd deze uitgevoerd om de mogelijkheden aan te geven.

De volgende situaties bevatten stapsgewijs meer beperkingen, waardoor ze niet meer zomaar met de situaties met PID-regelaar te vergelijken zijn. Wat telkens opvalt is de manier waarop de ingestelde waarde gehaald wordt in de volgende situatie. Regelactie en proces lopen duidelijk iets voor op de ingestelde waarde. Het maximale mogelijke debiet wordt in deze simulatie snel behaald en aangehouden tot de gewenste waarde bijna bereikt is door het proces.



Figuur 45: Respons systeem niet-lineaire tank: MPC geval 3

De laatste situatie die gesimuleerd wordt is met, bovenop de vorige beperkingen, een beperking op de maximale hoogte. Hier gebeurt hetzelfde als bij de simulatie met het model van de lineaire tank. Het biedt ook hetzelfde voordeel.



Figuur 46: Respons systeem niet-lineaire tank: MPC geval 4

## Conclusies

### PID-MPC

Wanneer voor het geval van de niet-lineaire tank de 2 meest representatieve situaties langs elkaar gelegd worden, moet dit gebeuren met Figuur 42 en Figuur 44. De beperkingen die hierbij opgelegd worden komen het meeste met elkaar overeen. Als deze grafieken vergeleken worden met elkaar valt op dat met de PID het systeem vrij snel reageert, even snel als de MPC. Hier hangt veel af van de instelling van de PID zelf. Het enige voordeel dat de MPC in dit geval heeft ten opzichte van de PID-regelaar is het feit dat de MPC al regelt voordat de stap effectief is aangelegd.

### Lineaire tank – niet-lineaire tank

Uiteraard is er een verschil tussen de lineaire tank en de niet-lineaire tank omdat het proces verschilt. Toch is het nodig om even stil te staan tussen het verschil in respons van beide systemen met de PID-regelaar. Bij het lineaire proces in Figuur 29 vertoont deze overshoot en is het proces merkbaar trager dan bij de niet-lineaire tank in Figuur 42. Dit geeft op een praktische wijze weer hoe afhankelijk een PID-regelaar is van de gebruikte instellingen en afwijkingen indien een grafische methode werd gebruikt zoals bij openlusinstellingen het geval is (raaklijn).

Voor wat betreft de vergelijking van het lineaire proces in Figuur 32 en het niet-lineaire proces in Figuur 44 geregeld met MPC is het opvallendste dat bij de niet-lineaire tank een kleine overshoot aanwezig is. Dit is meer dan waarschijnlijk te wijten aan het feit dat dit model gelineariseerd is en dus een benadering is van het werkelijke proces. De regelactie ziet er ook anders uit tussen de 2 processen omdat de respons van het systeem op de regelactie anders is.



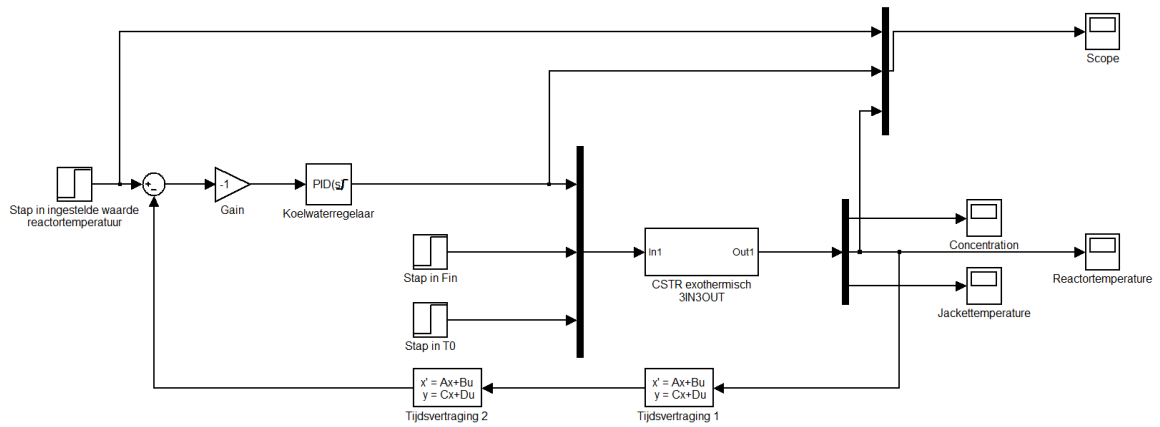
## 5 Resultaten: Regeling van een wandgekoelde chemische reactor

De casus van de CSTR wordt uitgewerkt voor 2 situaties. De eerste situatie is deze waarbij de CSTR een initiële reactortemperatuur heeft van 330K en een conversie van 95%. Een 2<sup>e</sup> situatie wordt uitgewerkt voor een CSTR met een initiële reactortemperatuur van 350K en 85% conversie. Beide situaties worden gebruikt om de simulaties uit te voeren met een PID-regelaar en een MPC. Op deze manier kan een vergelijking gemaakt worden tussen beide regelmethoden. Bijkomend wordt gekeken naar de regeling van het CSTR-model met behulp van MPC waarbij 2 variabelen geregeld worden. Hierbij wordt de concentratie en reactortemperatuur zo stabiel mogelijk gehouden.

Het gebruikte model, zie Figuur 47, bevat 2 nieuwe zaken die nog niet zijn uitgelegd. Links van het blok van het model ("CSTR exothermisch 3IN3OUT") is er een zogeheten mux-blok die van 3 signalen 1 signaal maakt. Dit is nodig omdat het model hier gebruik maakt van 3 ingangen om de simulatie uit te voeren. Het koelwaterdebiet, die gemanipuleerd wordt door de regelaar, en  $F_{in}$  en  $T_0$ . Dat is nodig om externe veranderingen aan te brengen in het model waarna gekeken kan worden hoe het proces en vooral, het proces met regelaar, hierop reageert. Het andere wat hier nieuw is, zijn de tijdsvertragingen. De reactortemperatuur die door het proces gegenereerd wordt, moet in werkelijkheid ook opgemeten worden. Afhankelijk van de sensor zit hier een bepaalde tijdsvertraging op, o.a. afhankelijk van het gebruikte materiaal. De tijdsvertragingen moeten daarom op het signaal van de reactor naar de regelaar aangebracht worden op de manier die hieronder wordt weergegeven.

### 5.1 Klassieke regeling: PID

De 2 casussen die besproken worden moeten allebei bepaalde PID-instellingen krijgen die niet gelijk zijn aan elkaar. De 2 casussen verschillen veel van elkaar en de regelaar is niet zomaar over te nemen voor een nieuw proces. Het verschil is vooral te wijten aan het feit dat de conversie van de 2 systemen anders is waardoor het reactorvolume veel verschilt. Het geval van 95% heeft een reactorvolume van ongeveer 422m<sup>3</sup> terwijl dit bij het geval met 85% conversie slechts 33m<sup>3</sup> is. De bufferende werking is daardoor bij het geval met 95% conversie veel groter en zal daardoor trager reageren. Door de kleine reactorinhoud is het geval met 85% conversie sneller en onstabiel. Dit is een gegeven waar in de industrie, zeker naar veiligheid toe, rekening mee wordt gehouden. De manier van werken is vergelijkbaar als in het vorige hoofdstuk. Eerst openlusinstellingen, daarna gesloten lus. De relais-instellingen worden hier niet toegepast omdat het werk anders te uitgebreid zou worden.



Figuur 47: Model in Simulink: CSTR

### 5.1.1.1 Methode

De algemene instellingen van de PID zijn de volgende:

- Initiële integrator = p.Fjss
- Regelbereik = 0 tot 4.p.Fjss. Dit wil zeggen dat het ontwerpdebiet van het koelwater  $1/4^e$  is van het maximale koelwaterdebiet
- Voor de PID-regelaar wordt het signaal van de fout omgekeerd door te vermenigvuldigen met -1. Dit wordt gedaan omwille van het feit dat wanneer de reactortemperatuur moet stijgen, het koelwaterdebiet zal moeten dalen. Dit komt overeen met indirecte regelaars die in de industrie gebruikt wordt.

De besproken casussen voor PID zijn de volgende:

Tabel 11: Totaaloverzicht casussen CSTR: PID

Eigenschappen CSTR	Ingestelde waarde temperatuur	Veranderde variabele voor stap	Veranderde variabele na stap	Korte uitleg
330K, 95%		$F_j$	$1,25F_j$	openlusrespons
	330	330	335	Setpoint tracking
	330	$F_{in}$	$1,2F_{in}$	Disturbance rejection
	330	$T_0$	$T_0+5$	Disturbance rejection
350, 85%	350	350	355	Vergelijking instellingen PID 330K
	350	350	355	Openlusinstellingen
	350	350	355	<i>Trial and error</i> instelling, setpoint tracking
	350	$F_{in}$	$1,2F_{in}$	Disturbance rejection
	350	$T_0$	$T_0+5$	Disturbance rejection

### 5.1.1.2 Resultaten

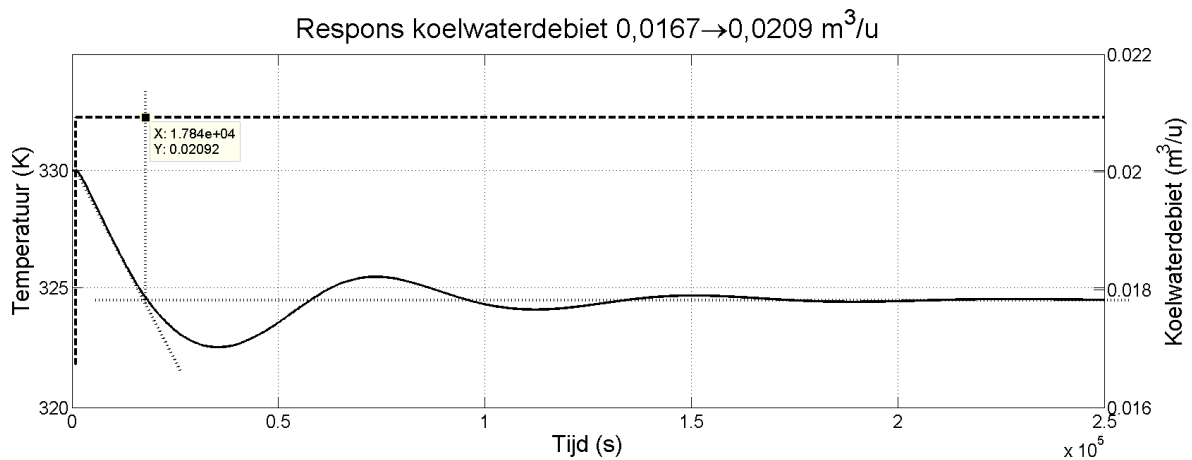
De gevolgde methode wordt in de literatuurstudie gevonden onder het luik klassieke regeling: open lus afstelling. De gevonden waarden voor  $\tau_d$  en  $\tau_p$  zijn  $1349-1000=349$  en  $17840-1349=16491$ .  $K_p$ , de versterkingsfactor van het proces, is de resulterende proceswaarde/de aangelegde variatie in inputvariabele (hier het voedingsdebiet).  $K_p$  is dus  $PV/OP=(330-324,8)/(0,0209-0,0167)=5,2/0,0042=1238,1$ .

Omgerekend geeft dit voor de bekomen waarden voor deze casus:

Tabel 12: PID-instellingen CSTR op basis van open lus

Regelaar	$K_p$	$\tau_i$	$\tau_d$
P	0,01145		
PD	0,022899		2,908333
PI	0,013358	329,82	
PID	0,022899	274,85	2,908333

Eén van de nadelen van de instelling op basis van de open loop respons curve is het feit dat deze niet uitvoerbaar is voor een gedempte oscillatie. Wanneer de instellingen worden toegepast en de PID ingesteld wordt als PID-regelaar zoals in de tabel treden er oneindige oscillaties op. Simulaties op basis van deze instellingen zijn verder nutteloos.



Figuur 48: Respons CSTR 330K open lus met constructie voor instelregels

Reactortemperatuur = 330K en conversie = 95%

De instellingen van de PI-regelaar wordt hieronder weergegeven. De gebruikte methode is de geslotenlusmethode uit de literatuurstudie. De regelaar wordt ingesteld als PID-regelaar van het ideale type, dit wil zeggen dat de transferfunctie van de vorm  $P(1+I+D)$  is. Daarbij zijn I en D respectievelijk gelijk aan  $1/\tau_i$  en  $\tau_d$ .

Tabel 13: PID-instellingen CSTR 330K op basis van gesloten lus

	$K_p$	$\tau_i$	$\tau_d$
Instelling	0,072	725	181,25

In ondergaande situaties wordt nagegaan hoe het proces met deze PID-instellingen omgaat met veranderingen die worden aangelegd. Zo wordt nagegaan hoe het proces reageert op een verandering van ingestelde waarde van de reactortemperatuur, maar ook een verandering in voedingsdebiet en een verandering in temperatuur van de voeding. Bij deze laatste 2 zal de ingestelde waarde van de reactortemperatuur stabiel blijven op 330K. De aangelegde veranderingen worden steeds na 1000s simuleren ingevoerd en worden hieronder schematisch weergegeven:

Tabel 14: Schematisch overzicht van de casussen die worden uitgevoerd voor wat betreft externe verstoringen

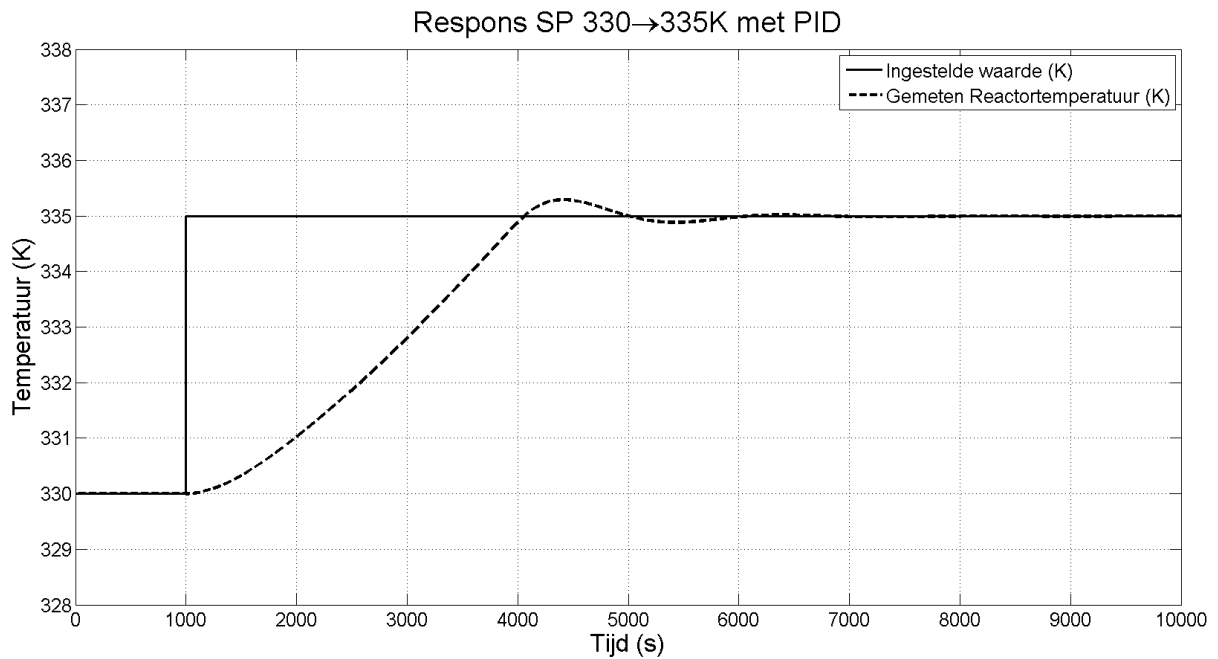
	Initieel	Na 1000s
Reactortemperatuur $T_R$	330	335
Voedingsdebiet $F_{in}$	0.0044	0.0053



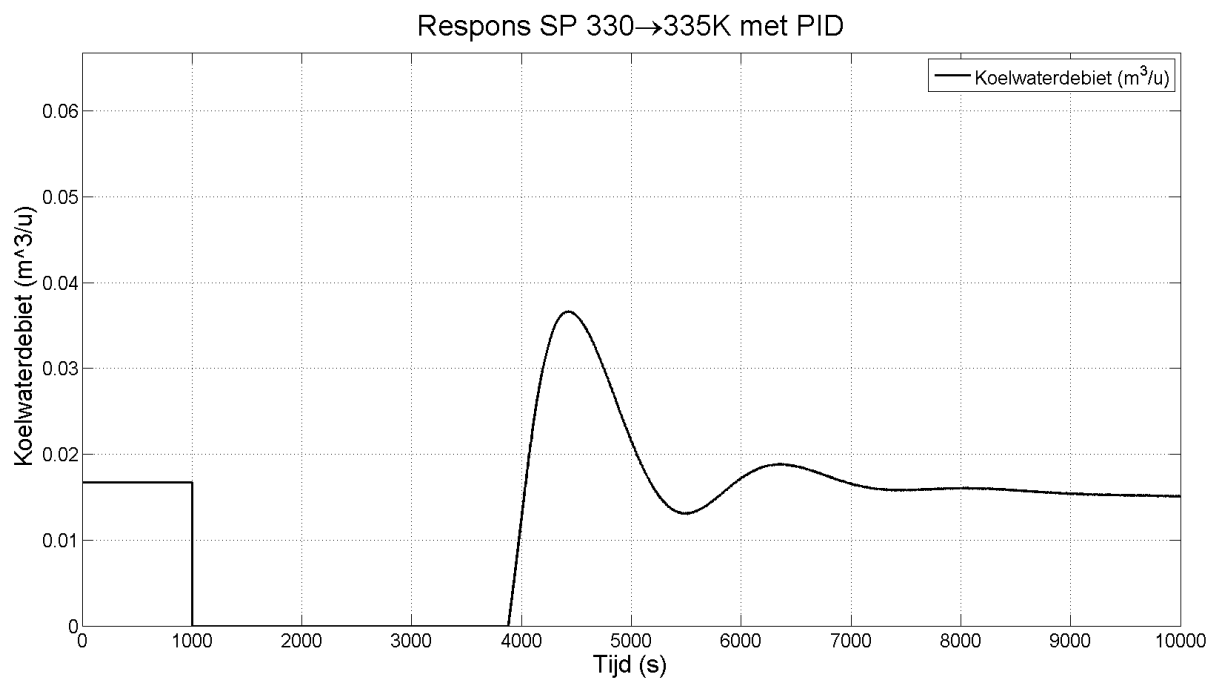
Voedingstemperatuur $T_0$	294	299
---------------------------	-----	-----

Van deze situaties wordt een grafiek gegeven van de manier waarop de reactortemperatuur verandert en de manier waarop het koelwaterdebiet verandert. Een overzicht van de situaties

Wanneer het proces een verandering in ingestelde waarde van de reactortemperatuur ondergaat van 330K naar 335K wordt onderstaande grafiek bekomen:



Figuur 49: Respons CSTR 330K: PID gesloten lus: stap in gewenste waarde reactortemperatuur

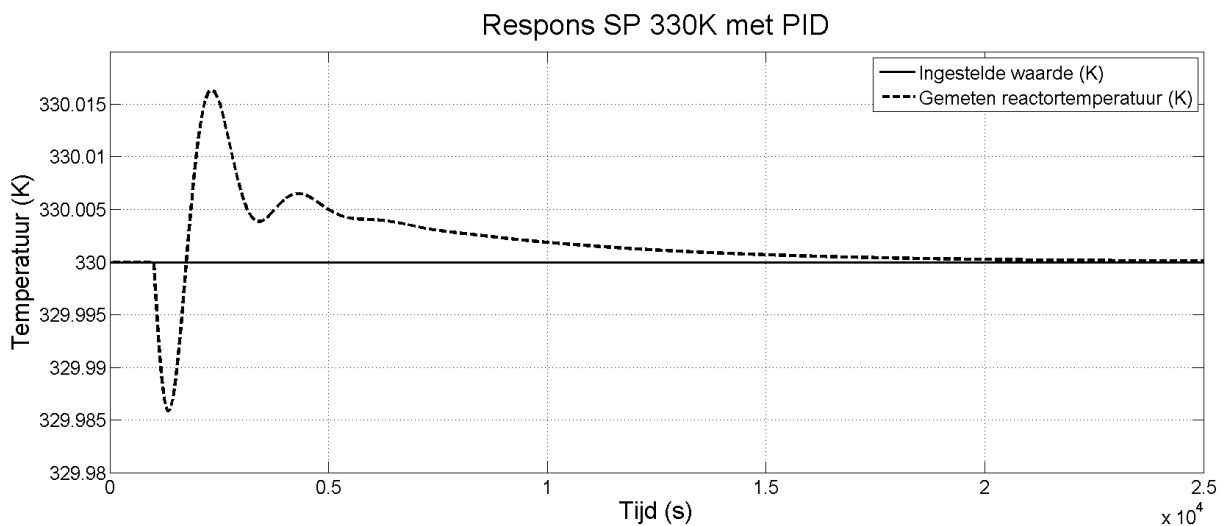


Figuur 50: Respons koelwater CSTR 330K: PID gesloten lus: stap in gewenste waarde reactortemperatuur

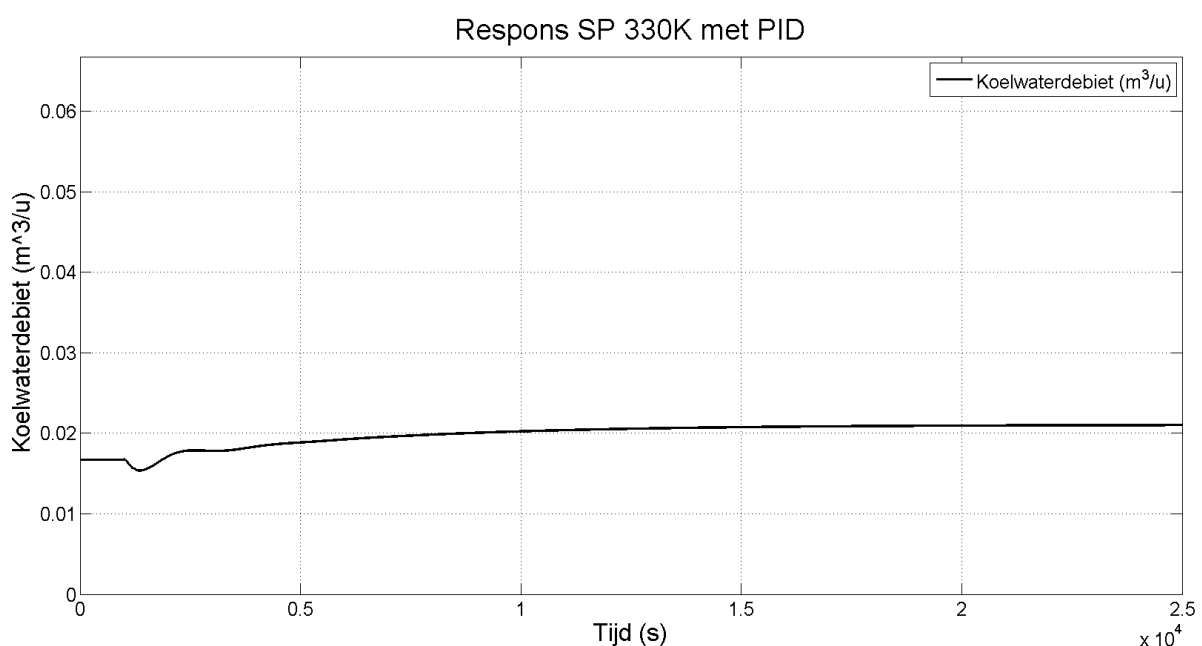
Het koelwater dat geregeld wordt is weergegeven in Figuur 50. Hierbij is de minimale waarde op de y-as ingesteld op 0 en de maximale waarde op 4 keer de stabiele waarde van het proces. Dit is het bereik waarbinnen het debiet geregeld kan worden. Er wordt hier verondersteld dat het designdebiet van het proces  $1/4^e$  is van het maximale koelwaterdebiet dat voor de mantel ter beschikking is.

Verder wordt nagegaan hoe het proces met deze PID-instellingen reageert op externe veranderingen. De volgende situatie is een verandering in voedingsdebiet van de reactor. Deze zal veranderen van de ontwerpwaarde van  $4,4 \cdot 10^{-3}$  naar 1,2 keer dit debiet, of een verhoging van 20%, naar  $5,3 \cdot 10^{-3}$ .

Bij de volgende gesimuleerde situatie wordt de ingestelde waarde van de reactortemperatuur op 330K gehouden en wordt er een stapverandering aangelegd zoals net besproken. In onderstaande figuur wordt het verloop weergegeven van de reactortemperatuur ten opzichte van de ingestelde waarde.

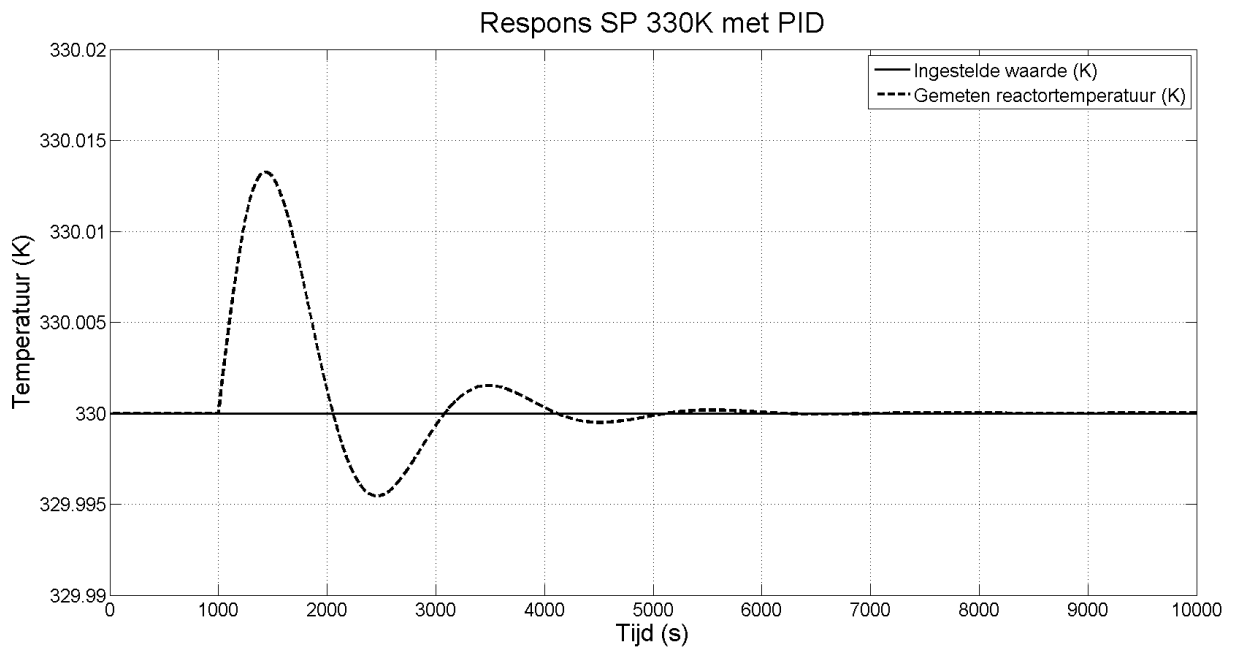


Figuur 51: Respons CSTR 330K: PID gesloten lus: stap in voedingsdebiet

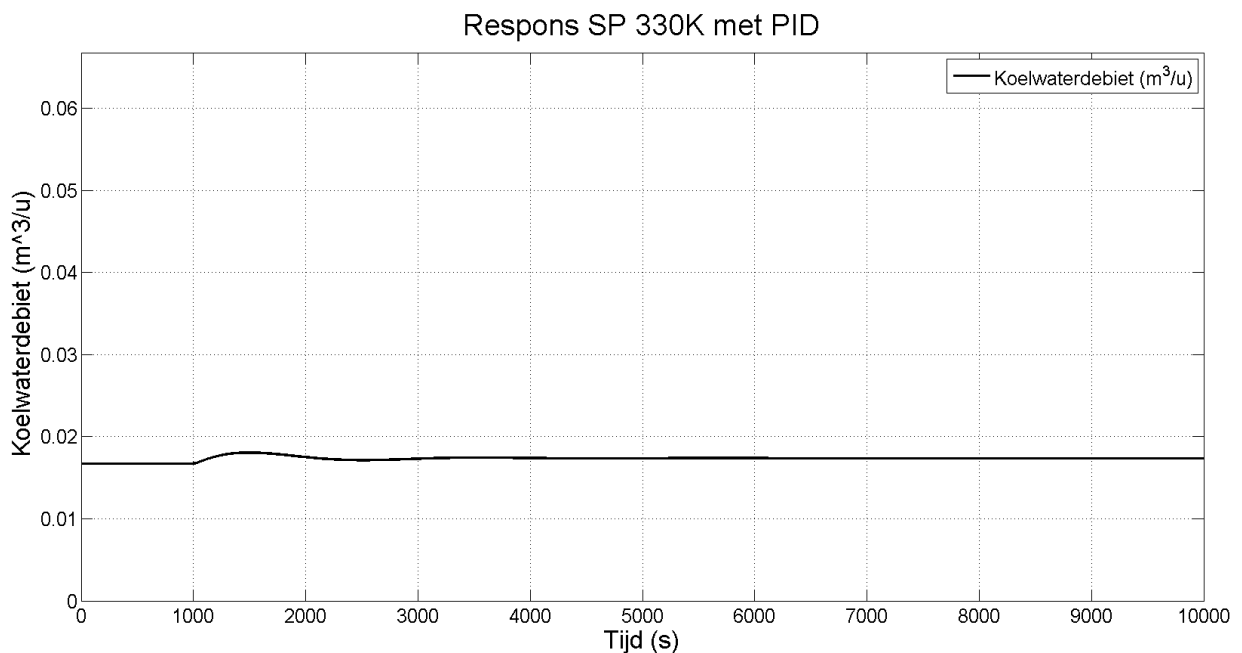


Figuur 52: Respons koelwater CSTR 330K: PID gesloten lus: stap in voedingsdebiet

In de laatste simulatie wordt er een stap aangelegd in de voedingstemperatuur van 294K naar 299K, dit is een stap van 5K. De temperatuurvariatie die hierbij gesimuleerd wordt is in werkelijkheid niet te meten, daarom wordt hier gesteld dat door de bufferende werking van het systeem er geen temperatuurvariatie zal ontstaan. Uit de simulaties blijkt wel dat de PID-regelaar het systeem stabiliseert na deze externe variaties.



Figuur 53: Respons CSTR 330K: PID gesloten lus: stap in voedingstemperatuur reagens



Figuur 54: Respons koelwater CSTR 330K: PID gesloten lus: stap in voedingstemperatuur

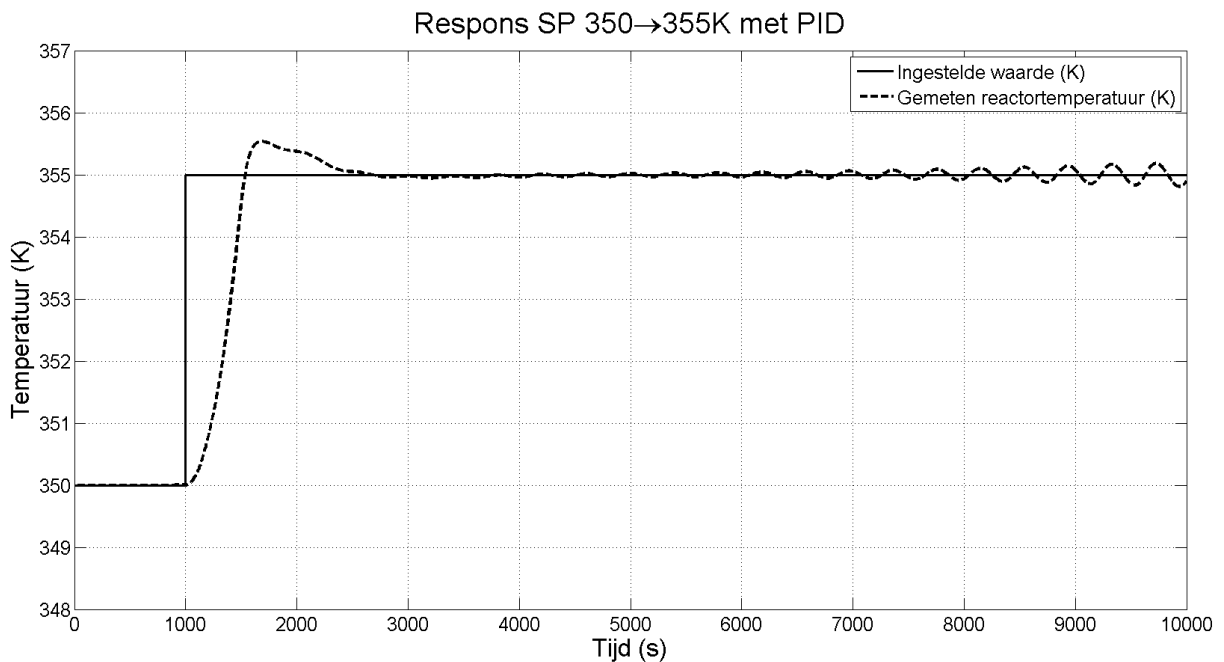
## Conclusie

Wanneer gekeken wordt naar de eerste simulatie waarbij de ingestelde waarde van de reactortemperatuur verandert van 330 naar 335K, dan wordt duidelijk dat het systeem traag reageert. Na 7000s of bijna 2 uren wordt er pas een nieuwe stabiele waarde bereikt. Dit is in de

praktijk niet gewenst en hiermee dient bij het regelen rekening te worden gehouden. De andere aangelegde variaties resulteren in een nog tragere stabilisatie wanneer het voedingsdebiet wordt aangepast en een snellere stabilisatie wanneer de temperatuur van de voeding wordt aangepast.

Reactortemperatuur = 350K en conversie = 85%

Wanneer de reactor wordt ingesteld op een reactortemperatuur van 350K en 85% conversie, wordt er eerst gekeken wat het effect is van de instellingen van de PID-regelaar die golden voor de reactor met instellingen van 330K en 95%. Bij deze simulaties dient in acht genomen te worden dat de nieuwe ontwerpwaarden anders zijn. Zo is het koelwaterdebiet waarop de reactor ontworpen is nu  $19,4 \cdot 10^{-3}$  in plaats van  $16,7 \cdot 10^{-3}$ .



Figuur 55: Respons CSTR 350K: PID gesloten lus: instellingen voor CSTR 330K

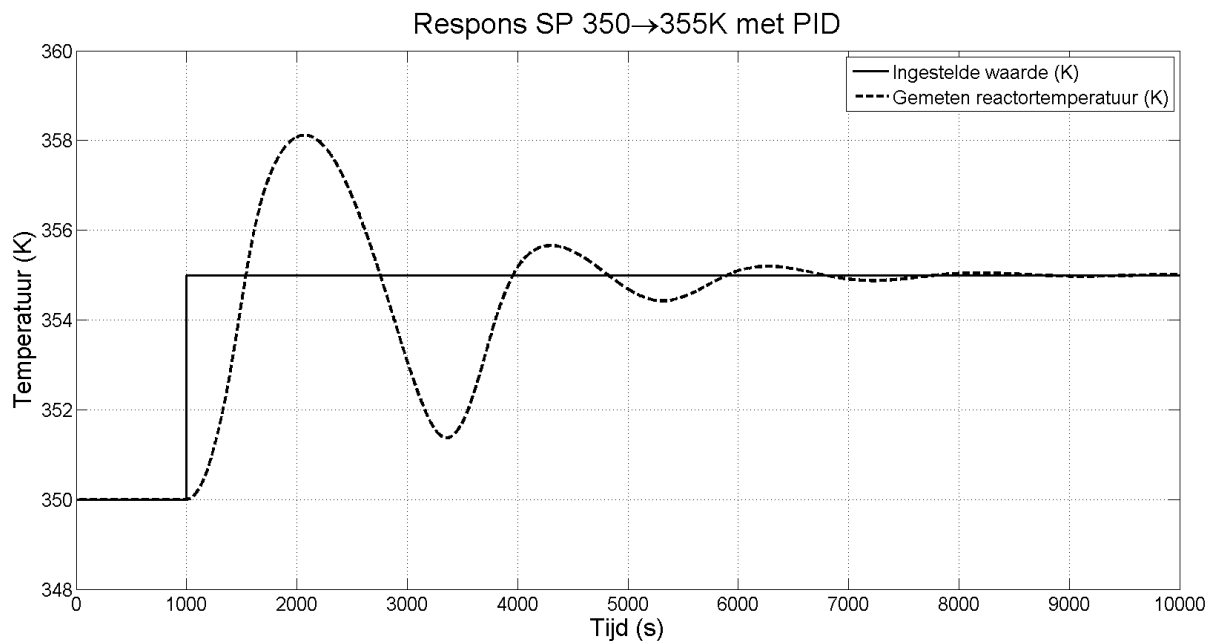
Hierbij valt op dat de oscillatie niet weg wordt gewerkt door de regelaar en na een periode van kleine oscillaties, deze steeds groter worden. Dit wil zeggen dat het systeem niet stabiel is en nieuwe instellingen gezocht zullen moeten worden.

Op basis van het resultaat van de casus met reactortemperatuur = 330K wordt hier geen openlusmethode toegepast. De geslotenlusmethode met relais wordt onmiddellijk gebruikt en geeft volgende instellingen. De  $K_U$  die gevonden wordt is 0,017 en  $P_U=1260s$ . De PID-regelaar wordt ingesteld met de volgende waarden:

Tabel 15: PID-instellingen CSTR op basis van gesloten lus

Regelaar	$K_p$	$\tau_i$	$\tau_d$
P	0,0085		
PI	0,00765	1050	
<b>PID</b>	<b>0,0102</b>	<b>630</b>	<b>157,5</b>

Wanneer de simulatie wordt uitgevoerd voor een verandering in ingestelde waarde van 350K naar 355K wordt volgende grafiek bekomen:



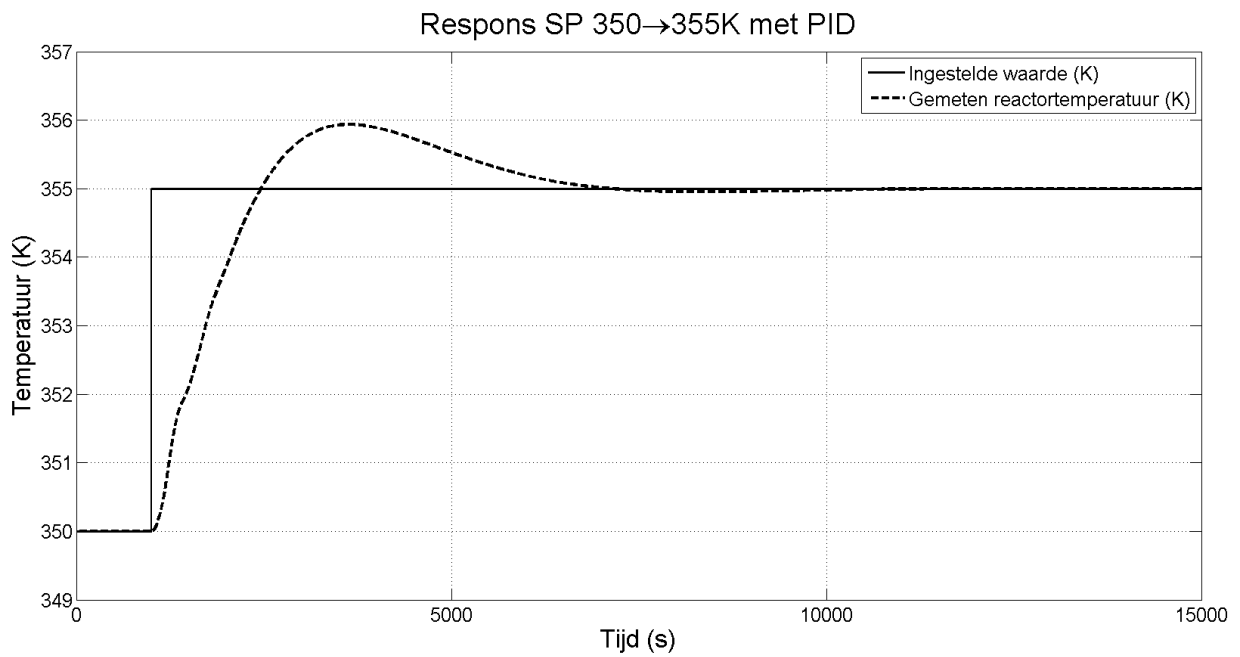
Figuur 56: Respons CSTR 350K: PID gesloten lus instellingen

Het dient gezegd te worden dat de respons van het systeem zeker niet bevredigend is, gezien de stapgrootte maar 5K bedraagt en de overshoot 3K. De undershoot bedraagt zelfs meer dan 3K. Dit is meer dan 60% van de aangelegde stapgrootte van de ingestelde waarde en zeker ongewenst. Via *trial and error* worden nieuwe instellingen gezocht en gevonden:

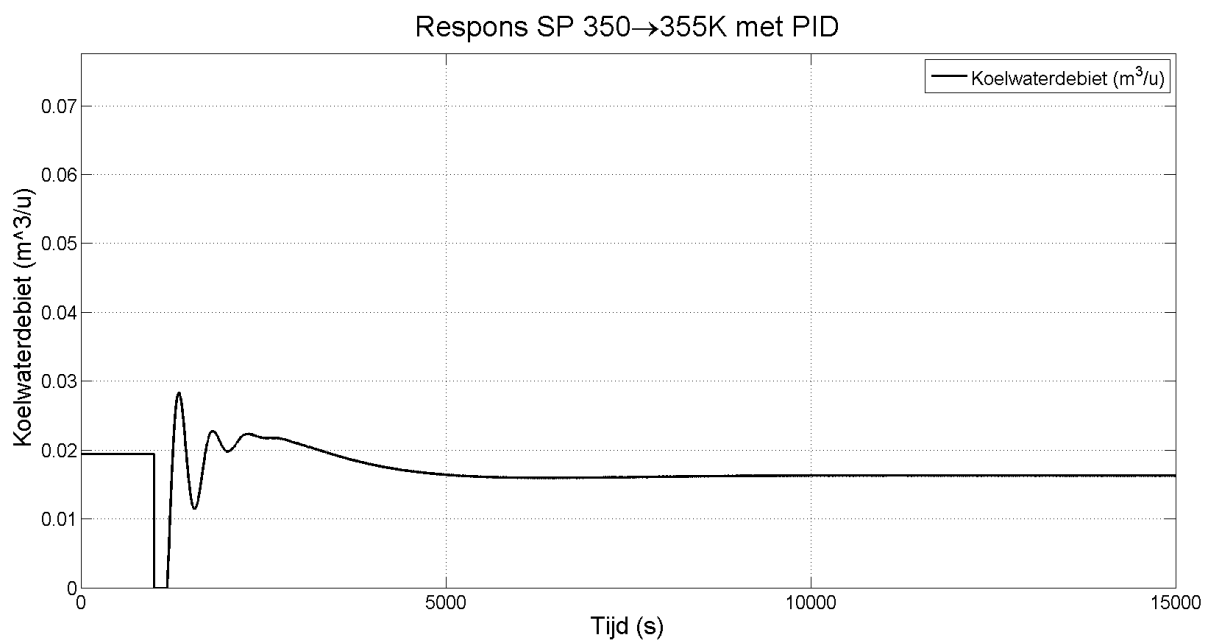
Tabel 16: PID-instellingen CSTR na trial and error

	$K_p$	$\tau_i$	$\tau_d$
Instelling	0,009	2000	900

Het proces reageert hier ongeveer even traag, maar wel met een veel matigere *overshoot*. Er zou verder gezocht kunnen worden naar instellingen die een nog betere respons geven maar deze instellingen zijn bevredigend genoeg om voor de volgende simulaties te gebruiken.

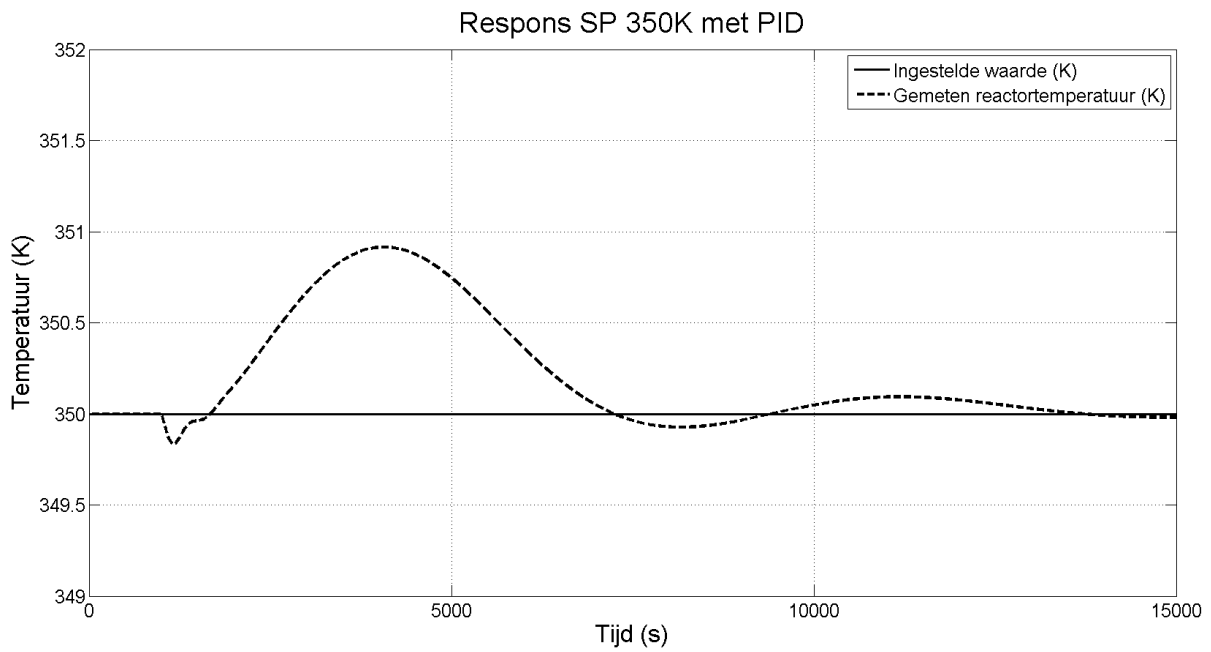


Figuur 57: Respons CSTR 350K: PID met instellingen na *trial and error*

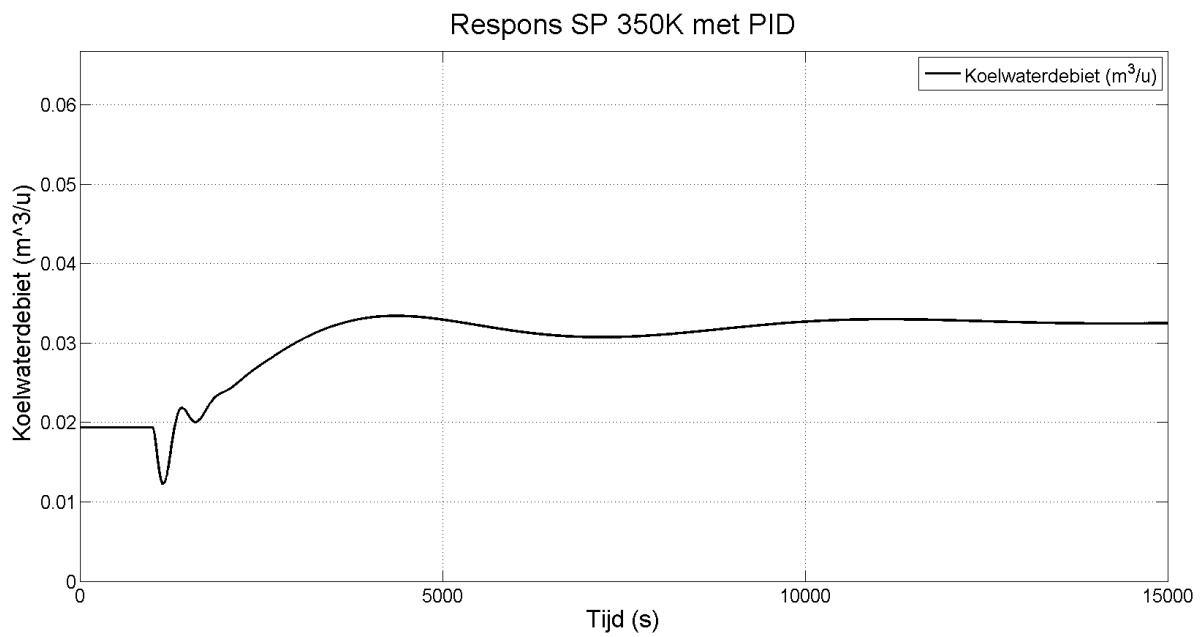


Figuur 58: Respons CSTR 350K: PID met instellingen na *trial and error*: koelwaterrespons

Met deze instellingen worden vervolgens, net zoals bij de casus van 330K en 95% conversie, gekeken wat de impact is van een stapverandering in voedingsdebiet en voedingstemperatuur. De eerste grafiek geeft de respons weer van de reactortemperatuur bij het veranderen van het voedingsdebiet van  $4,4 \cdot 10^{-3}$  naar  $5,3 \cdot 10^{-3}$ .

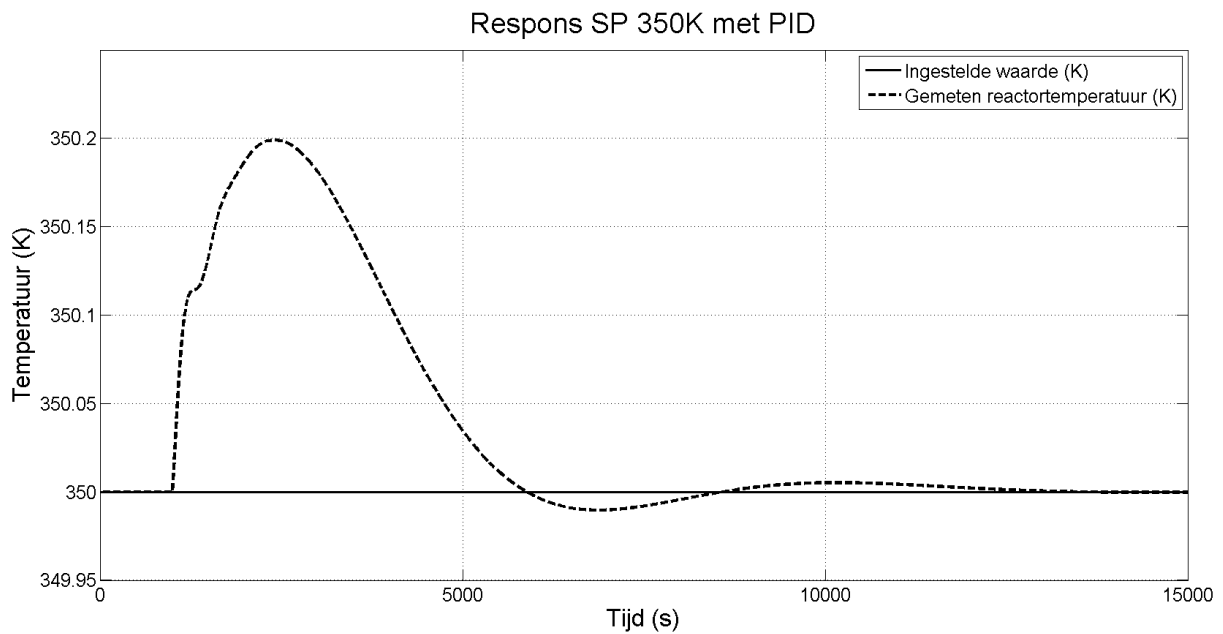


Figuur 59: Respons reactortemperatuur CSTR 350K op stap voedingsdebiet

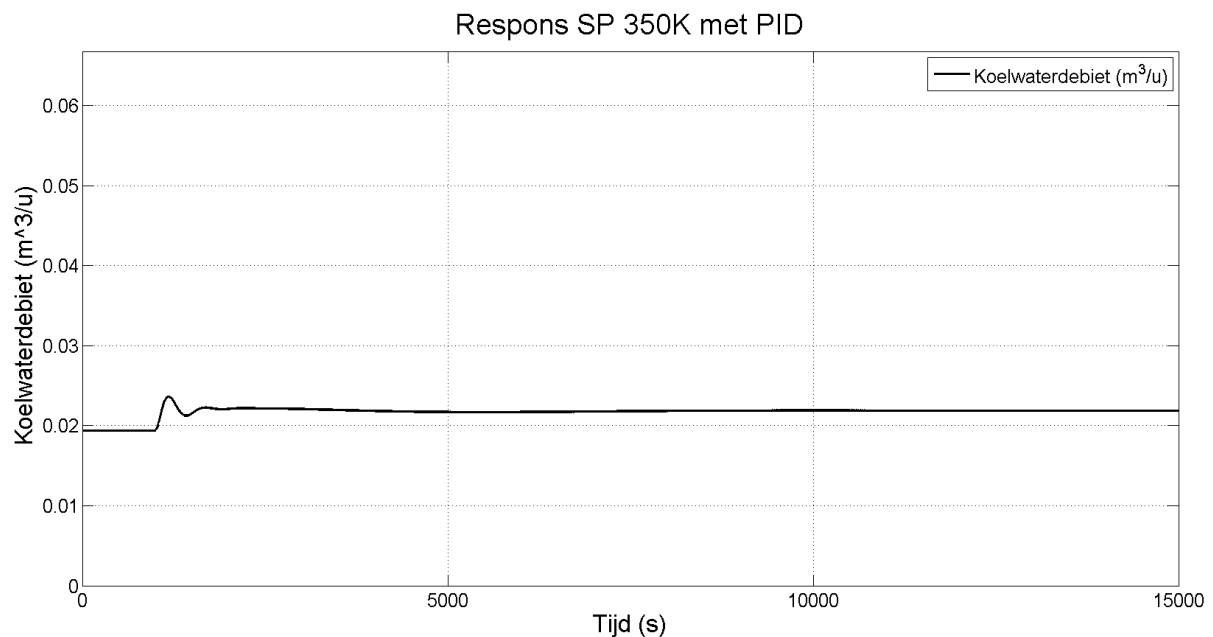


Figuur 60: Respons koelwaterdebiet CSTR 350K op stap voedingsdebiet

Wanneer de voedingstemperatuur veranderd wordt van 294 naar 299K geeft dit volgend verloop in reactortemperatuur:



Figuur 61: Respons reactortemperatuur CSTR 350K op stap voedingstemperatuur



Figuur 62: Respons reactortemperatuur CSTR 350K op stap voedingstemperatuur

Wat bij deze laatste situatie opvalt is het feit dat de reactortemperatuur niet veel afwijkt als gevolg van de bufferende werking van de reactorinhoud, maar ook dat ondanks het feit dat de reactortemperatuur vrij lang afwijkt, de regelactie relatief snel vrij stabiel blijft. De reactortemperatuur vertoont tot 5000s nog een afwijking, terwijl de afwijking in de regelactie vooral groot is net na de stap en de eerste paar honderd seconden erna.

## 5.2 Geavanceerde procescontrole: MPC

In onderstaand gedeelte wordt uitgebreid over het gebruik van MPC voor de controle van het CSTR-model. De gecontroleerde variabele is hierbij nog steeds het koelwaterdebiet, net zoals wanneer de PID gebruikt werd om het proces te controleren. De gemeten variabele is de temperatuur van het reactiemengsel in de reactor.



### 5.2.1.1 Methode

Bij het simuleren met de MPC als regelaar wordt geen enkele parameter aangepast tussen de simulaties. Bepaalde waarden zoals  $u_{\max}$  worden ad hoc opgeroepen uit het geheugen.  $U_{\max}$  is namelijk het maximale koelwaterdebiet. Er wordt vanuit gegaan dat tijdens stabiel bedrijf van de CSTR het koelwaterdebiet 25% van het maximale koelwaterdebiet bedraagt. Omdat er 2 situaties worden bekeken, een reactor met een reactortemperatuur van 330K en 95% conversie en een reactor met een reactortemperatuur van 350K en 95% conversie, is het koelwaterdebiet dat nodig is bij stabiel bedrijf verschillend van elkaar. Door de ad-hoc berekening hoeft deze factor niet telkens aangepast te worden.

$t_s$	5
Hp	100
Hc	80
R	0,5
Q	1

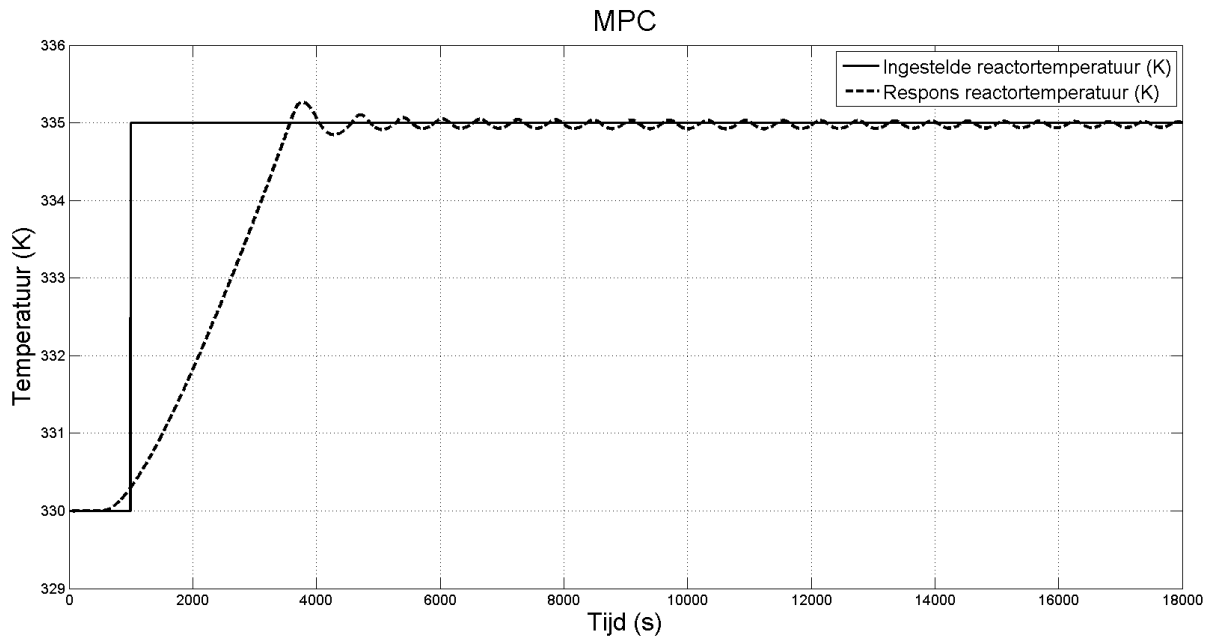
De besproken casussen zijn analoog aan deze met PID. De bespreking van het instellen van de PID-regelaar valt hier wel weg.

Tabel 17: Totaaloverzicht casussen CSTR: PID

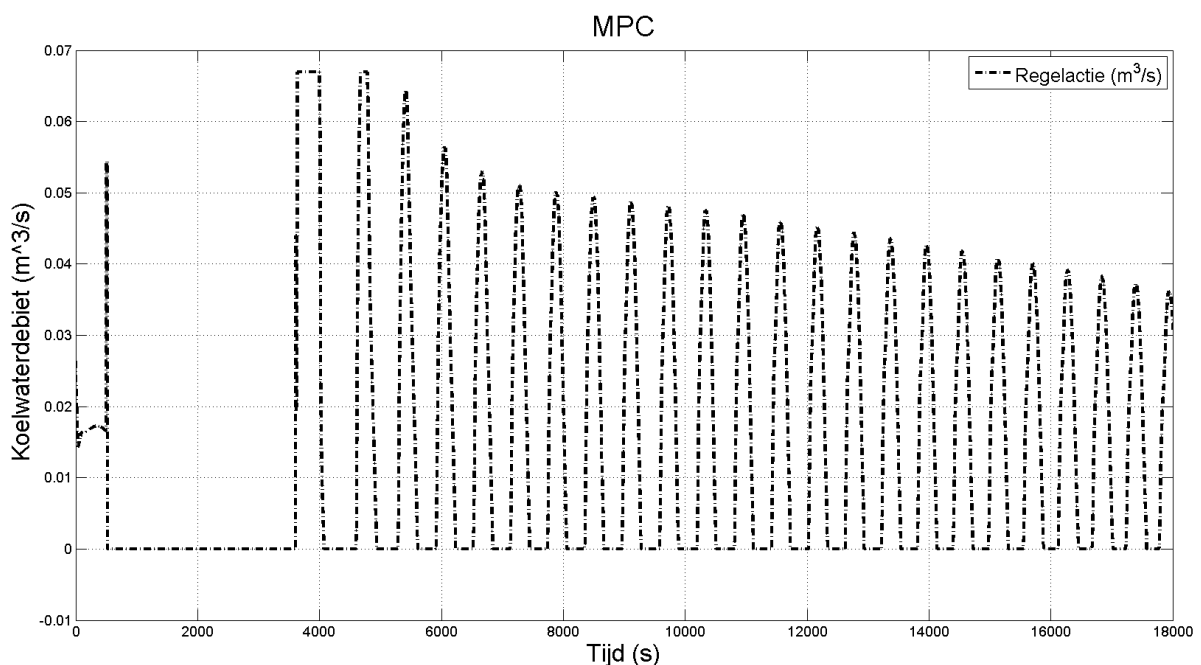
Eigenschappen CSTR	Ingestelde waarde temperatuur	Veranderde variabele vóór stap	Veranderde variabele na stap	Korte uitleg
330K, 95%	330	330	335	Proces met tijdsvertraging, model zonder tijdsvertraging
330K, 95%	330	330	335	Setpoint tracking
	330	$F_{in}$	$1,2F_{in}$	Disturbance rejection
	330	$T_0$	$T_0+5$	Disturbance rejection
350, 85%	350	350	355	setpoint tracking
	350	$F_{in}$	$1,2F_{in}$	Disturbance rejection
	350	$T_0$	$T_0+5$	Disturbance rejection

### 5.2.1.2 Resultaten

Net zoals bij de simulaties van de CSTR met PID worden ook hier 2 tijdsvertragingen toegevoegd. Het model dat eerst in de MPC gebruikt wordt is een model zonder deze 2 tijdsvertragingen in vervat, terwijl het proces wel 2 tijdsvertragingen heeft. Dit wordt gedaan om na te gaan wat er gebeurt wanneer het model wel het proces omschrijft, maar stappen die er in werkelijkheid ook tussen zitten, vergeten worden.



Figuur 63: Temperatuurrepons met MPC van de CSTR zonder tijdsvertraging van de temperatuursensor



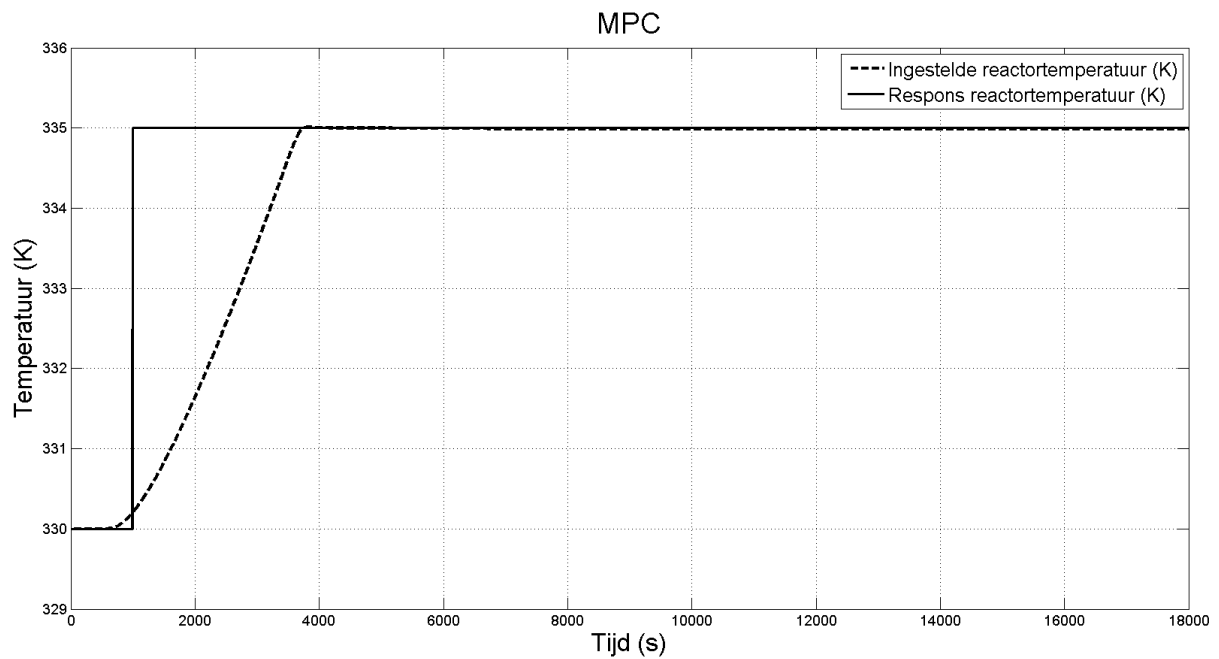
Figuur 64: Koelwaterrespons met MPC van CSTR zonder tijdsvertraging van de temperatuursensor

Het is zeer duidelijk dat de MPC het proces niet snel genoeg stabiel krijgt, er blijven oscillaties aanwezig. Voor de temperatuur is dit geen grote ramp, de oscillaties zijn niet zo heel groot, maar wanneer gekeken wordt naar Figuur 64, wordt duidelijk hoe onrustig het systeem is.

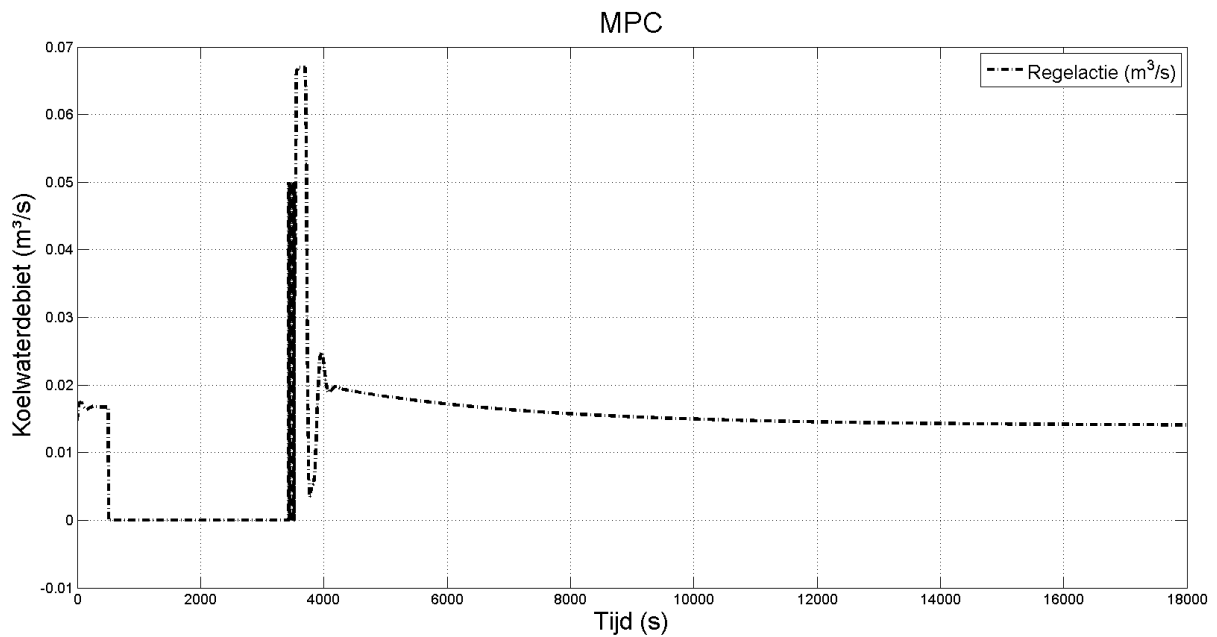
De regelactie die met de oscillaties in de temperatuur overeenkomt gaat van 0% naar 100% en terug. Daarom wordt in het model dat in de MPC gebruikt wordt de tijdsvertraging van de metingen toegevoegd. Deze aanpassing wordt toegepast voor alle simulaties.

Reactortemperatuur = 330K en conversie = 95%

Het eerste geval dat bekeken wordt is een verandering van de ingestelde waarde van de reactortemperatuur. Deze krijgt een stap op 1000 seconden van 330K naar 335K en resulteert in Figuur 65 waarbij opvalt dat ook hier weer het proces iets voor de ingestelde waarde reageert. De nieuwe ingestelde waarde wordt na minder dan 3000s of 50 minuten gehaald. De reactortemperatuur blijft daarna stabiel, terwijl de temperatuur van de koelmantel en het koelwater wel nog varieert nadat de nieuwe ingestelde waarde bereikt wordt.



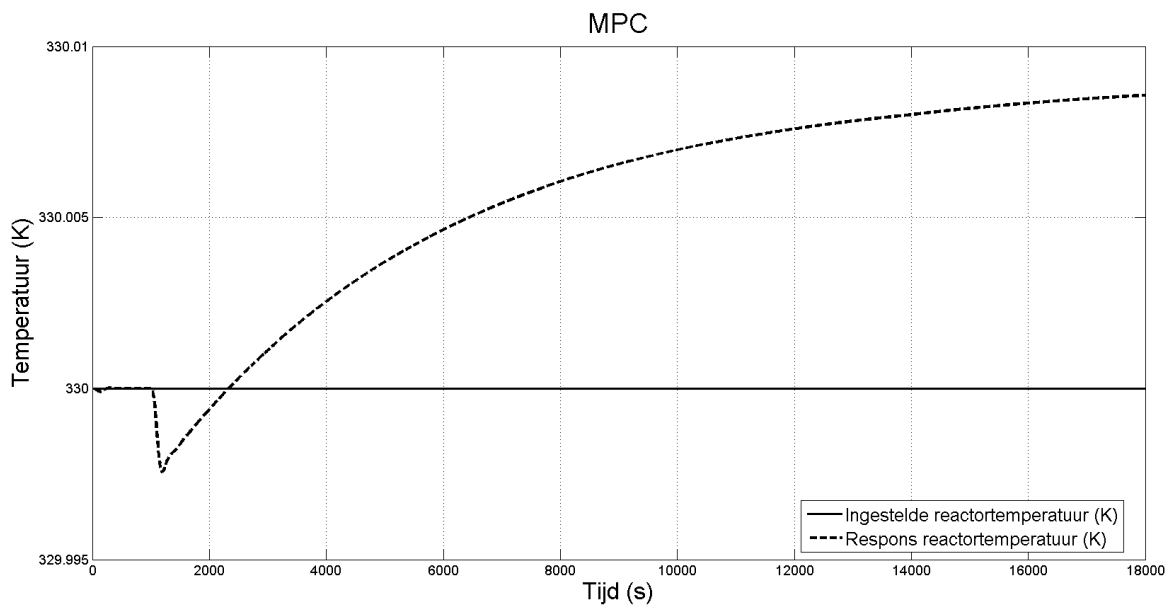
Figuur 65: Temperatuurrepons MPC 330K: stap in ingestelde waarde: 330 naar 335K



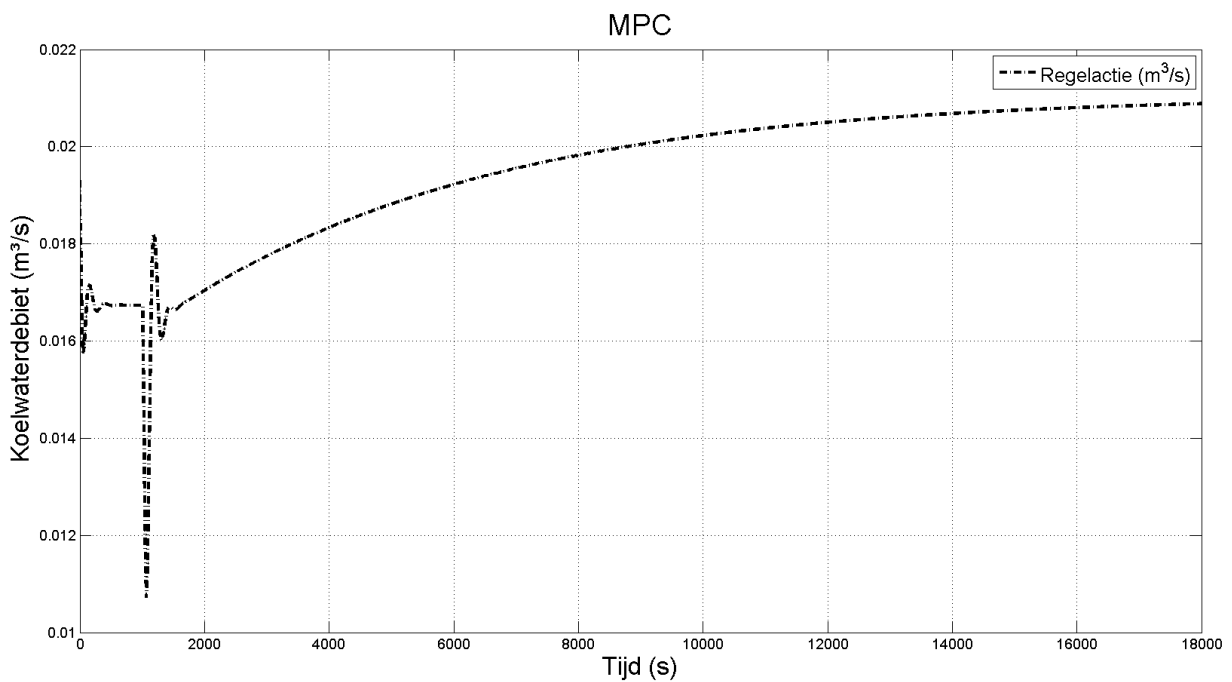
Figuur 66: Koelwaterrespons MPC 330K: stap in ingestelde waarde: 330 naar 335K

Voor het 2<sup>e</sup> geval wordt er een verandering van het voedingsdebiet aangelegd van  $4,377 \cdot 10^{-3} \text{ m}^3/\text{s}$  naar  $5,25 \cdot 10^{-3} \text{ m}^3/\text{s}$  op 1000s. Dit wordt gedaan om na te gaan hoe goed de MPC een externe variatie kan opvangen. Aangezien er in de simulatie maar een variatie te zien is tot bijna  $1/100^{\text{e}}$ , kan besloten worden dat de variatie hierbij zeer klein is en de regelaar snel en correct reageert. Dit wordt weergegeven in Figuur 67. In de legende staat respons koelwatertemperatuur maar deze valt weg omdat er gefocust is op de reactortemperatuur.

Wat niet te zien is, omdat de simulatietijd te kort is, is het feit of de reactortemperatuur zich stabiliseert op een waarde die afwijkt van de gewenste waarde of dat de reactortemperatuur nog nadert naar de gewenste waarde. Hier kan geen conclusie over worden getrokken omdat in Figuur 68 de regelactie ook nog steeds niet stabiel is. De stabilisatie van de reactortemperatuur op een temperatuur die afwijkt van de gewenste waarde is te zien in het volgende geval.



Figuur 67: Temperatuurrepons MPC 330K: stap in voedingsdebiet

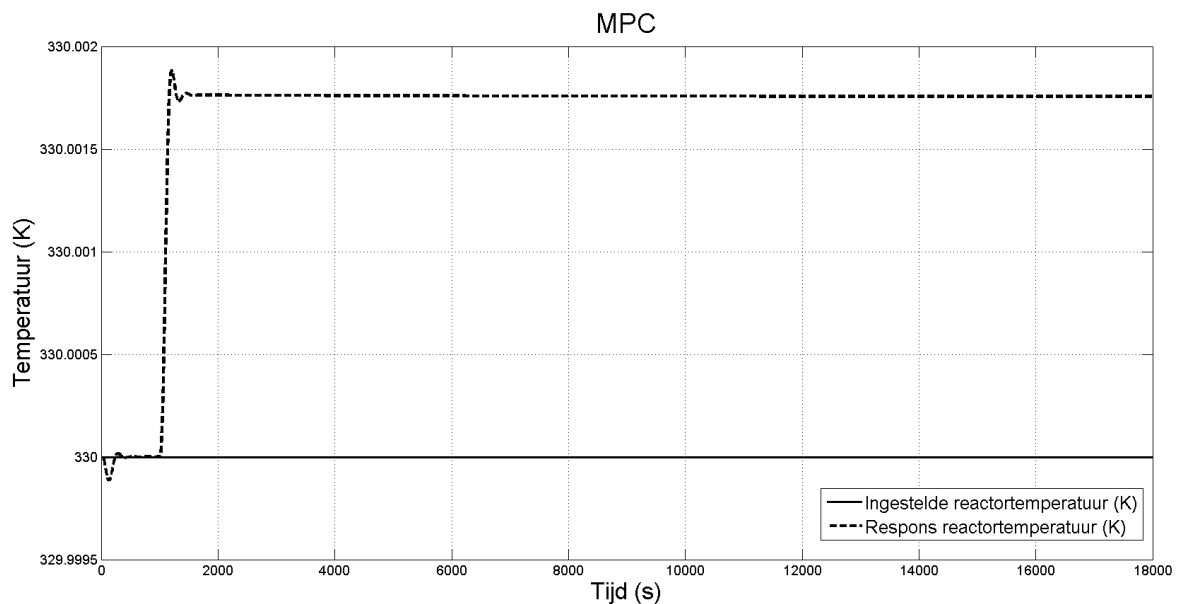


Figuur 68: Koelwaterrespons MPC 330K: stap in voedingsdebiet

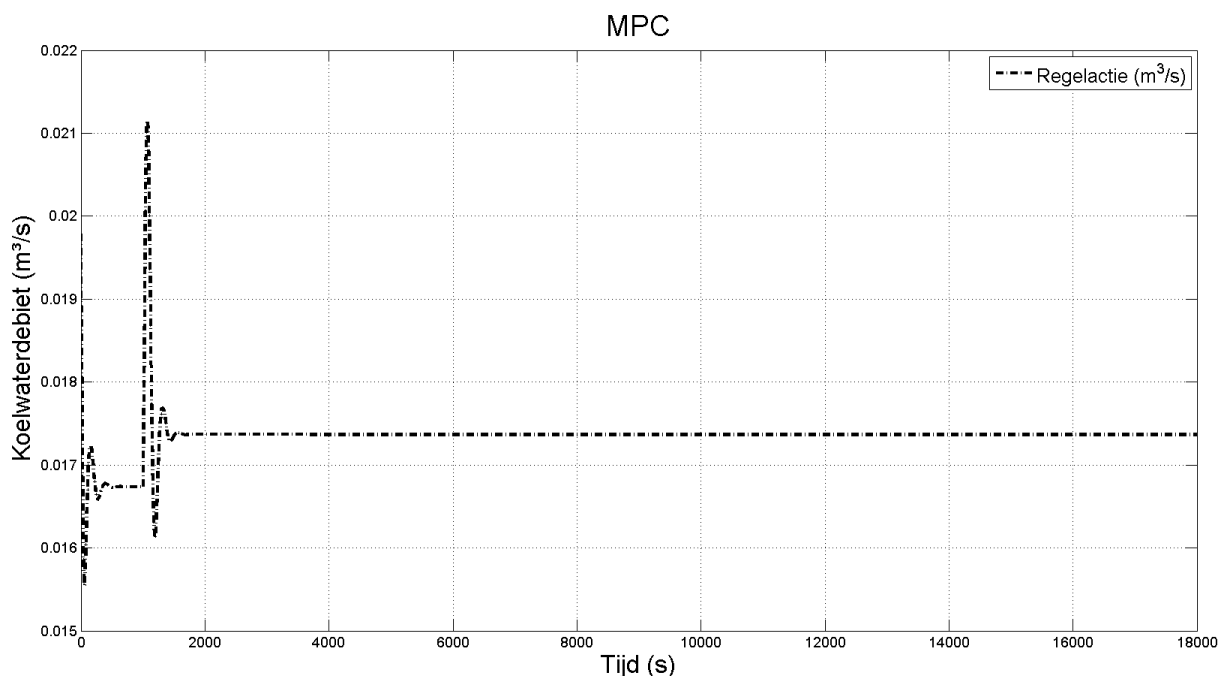
Het koelwaterdebiet dat voor deze verandering zorgt wordt in Figuur 68 weergegeven. De regelaar regelt snel met een vrij grote stap wanneer het voedingsdebiet net verandert en vangt daarom de verstoring zo snel op.

Het 3<sup>e</sup> geval dat bekeken wordt is de verandering van de voedingstemperatuur van het reagens van 294 naar 299K op 1000s. Hetgeen hier het meeste opvalt is het feit dat er een soort van statische afwijking optreedt. De respons van de reactortemperatuur blijft stabiel op een waarde, die niet overeenkomt met de gewenste waarde. Dit kan te wijten zijn aan een numerieke fout binnen de simulatie. Ook hier valt in de grafiek de respons van de koelwatertemperatuur weg wegens de focus op de reactortemperatuur.

Bij Figuur 70 valt op dat het regelen van het koelwaterdebiet resulteert in een stabiel debiet reeds na 600s nadat de stap in de voedingstemperatuur wordt aangelegd. Daaruit is, samen met het verloop in Figuur 69 zeker te concluderen dat de blijvende afwijking in reactortemperatuur te wijten is aan een numerieke fout die inherent is aan het simuleren. Daarbij dient gelet te worden op het feit dat de fout slechts een variatie heeft van nog geen  $2/1000^e$  wat zeer klein is en in de praktijk niet meetbaar is.



Figuur 69: Temperatuurrespons MPC 330K: stap in voedingstemperatuur



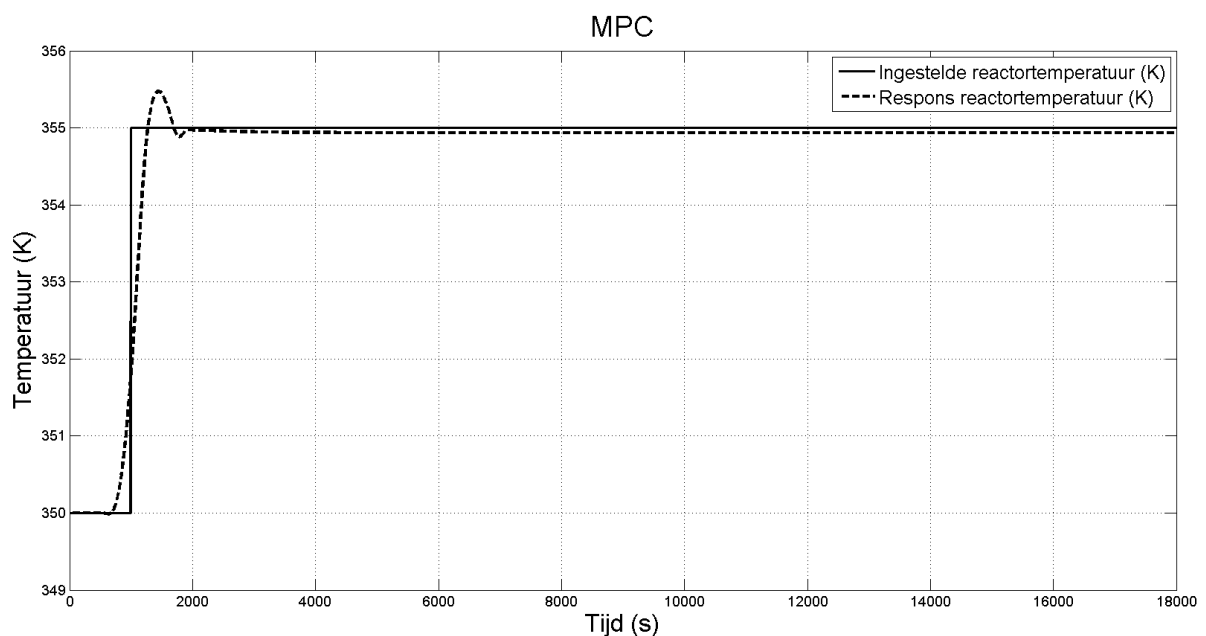
Figuur 70: Koelwaterrespons MPC 330K: stap in voedingstemperatuur

Reactortemperatuur = 350K en conversie = 85%

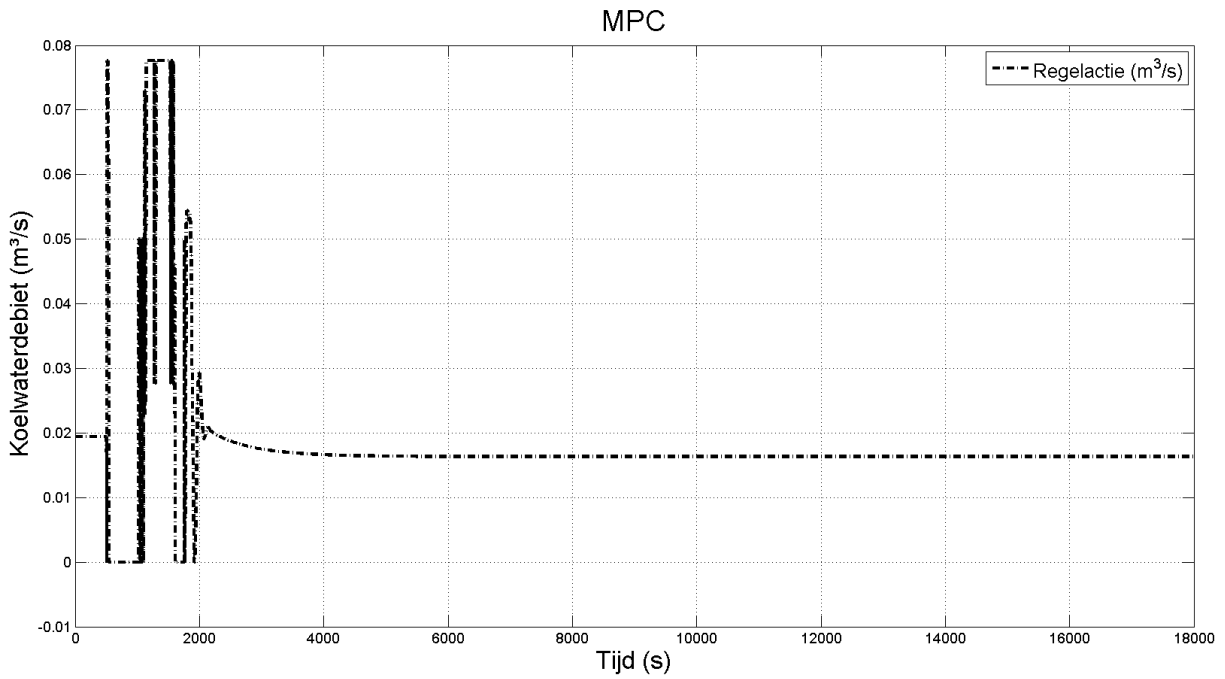
De volgende gevallen die bekeken worden gelden voor dezelfde CSTR maar met nieuwe instellingen. De reactortemperatuur wordt 350K en de conversie wordt 85%. Dit houdt in dat

bepaalde constanten veranderen zoals het reactorvolume, dit kan gevonden worden onder de kop Resultaten: . De gevallen die hieronder bekeken worden hebben dezelfde opbouw als met de CSTR met de vorig instellingen (reactortemperatuur = 330K en conversie = 95%)

Het eerste geval dat hier beschouwd wordt is de verandering van de ingestelde waarde van de reactortemperatuur: stap op 1000 seconden van 350K naar 355K. het proces doet er nu iets langer over om de nieuwe gewenste waarde te bereiken in vergelijking met de andere instellingen voor de CSTR. Ook wordt er een kleine overshoot vastgesteld als gevolg van de thermische inertie van het systeem. Als de voorspellingshorizon vergroot zou worden kan de MPC deze overshoot wegwerken. Het koelwaterdebiet in Figuur 72 maakt grote sprongen die eigenlijk niet kunnen, daarvoor dienen in de werkelijkheid zeker beperkingen opgelegd te worden aangezien de sprongen gaan van minimaal naar maximaal koelwaterdebiet. Het is ook mogelijk om de matrix R andere waarden te geven. Deze staat in voor een soort van afstraffing op de grootte van de stap die berekend wordt voor de regelactie. Het debiet wordt wel relatief snel weer stabiel ten opzichte van de volgende gevallen.

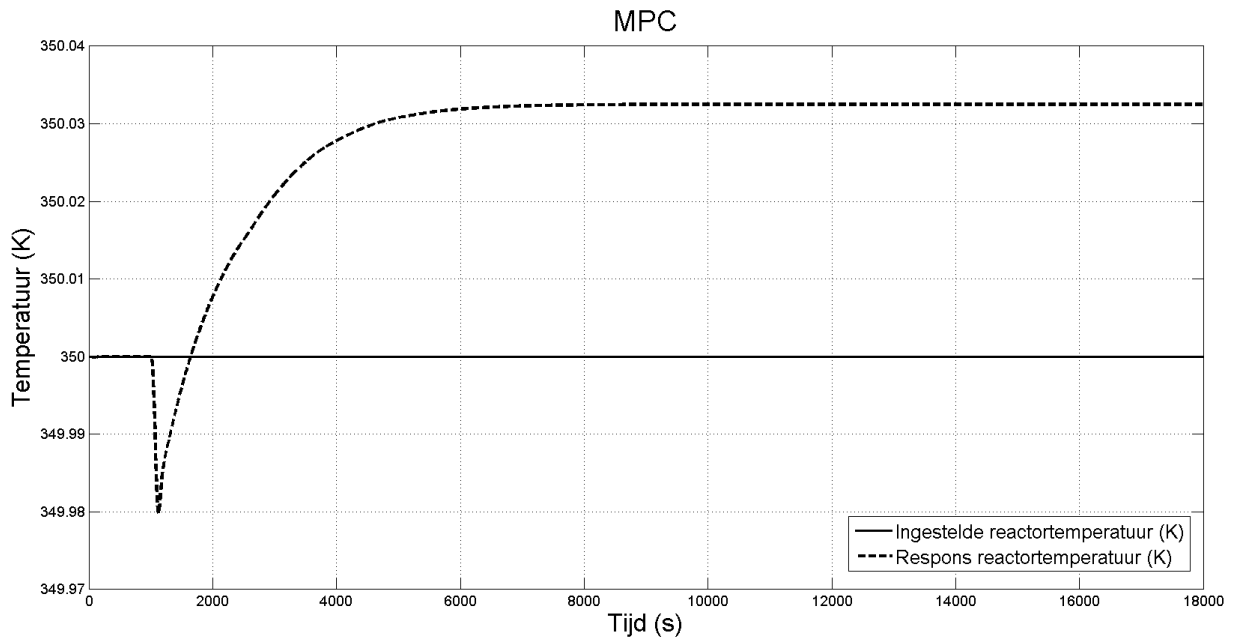


Figuur 71: Temperatuurrepons MPC 350K: stap in ingestelde waarde van 350 naar 355K



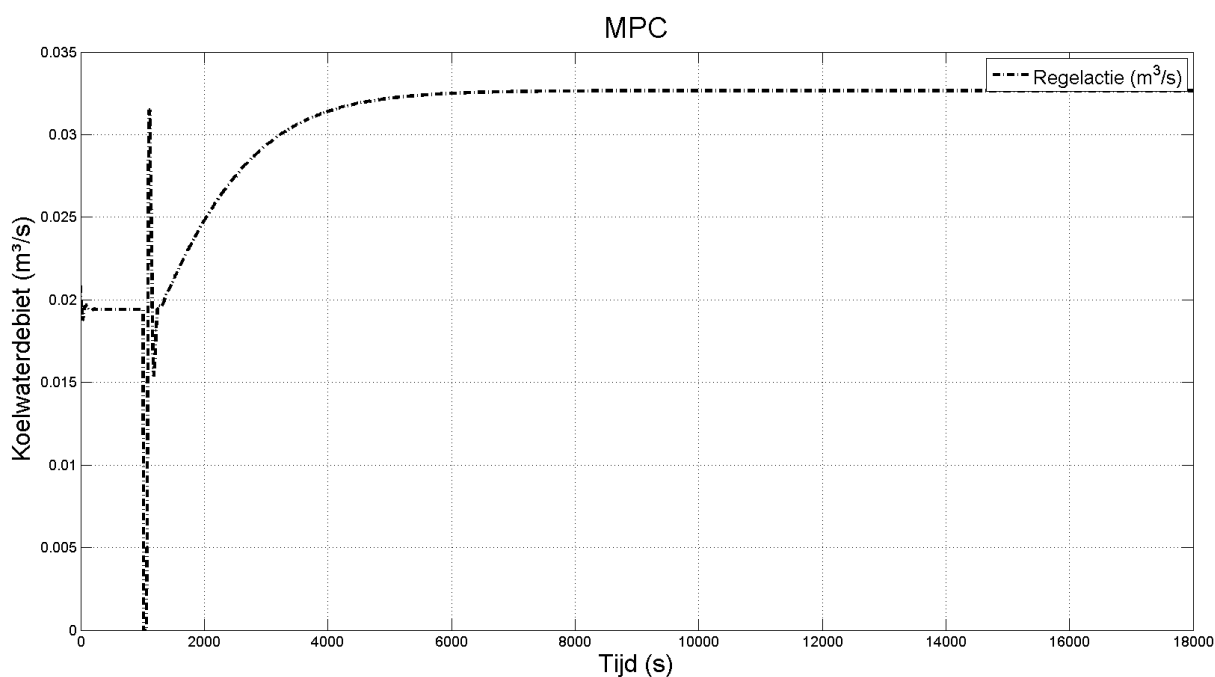
Figuur 72: Koelwaterrespons MPC 350K: stap in ingestelde waarde van 350 naar 355K

Het 2<sup>e</sup> geval wordt een verandering in het voedingsdebiet aangebracht van  $4,377 \cdot 10^{-3} \text{m}^3/\text{s}$  naar  $5,25 \cdot 10^{-3} \text{m}^3/\text{s}$  op 1000s. De respons van de reactortemperatuur, te zien in Figuur 73, is hierbij minimaal, eigenlijk kan besloten worden dat de reactortemperatuur constant blijft omdat de variatie kleiner is dan hetgeen in de praktijk gemeten kan worden. Het koelwaterdebiet in Figuur 74 maakt niet meer zo grote sprongen als het vorige geval, de verandering is hier ook niet zo groot. Toch maakt deze op korte tijd vrij grote sprongen en zullen in de praktijk beperkingen opgelegd moeten worden.



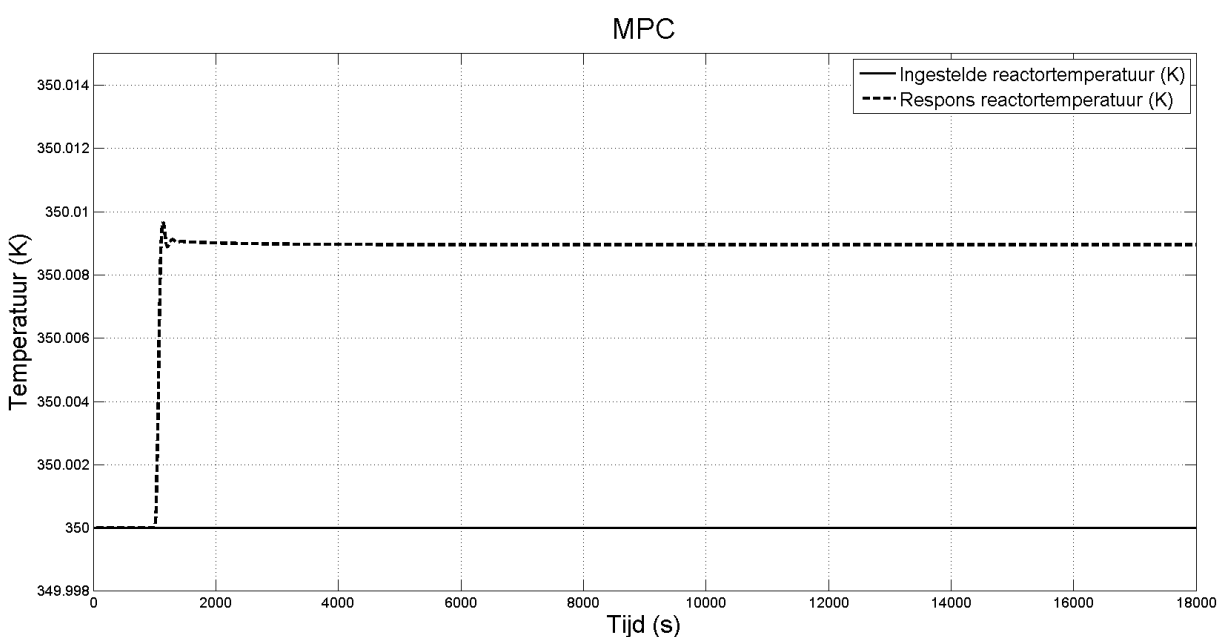
Figuur 73: Temperatuurrepons MPC 350K: stap in voedingsdebiet



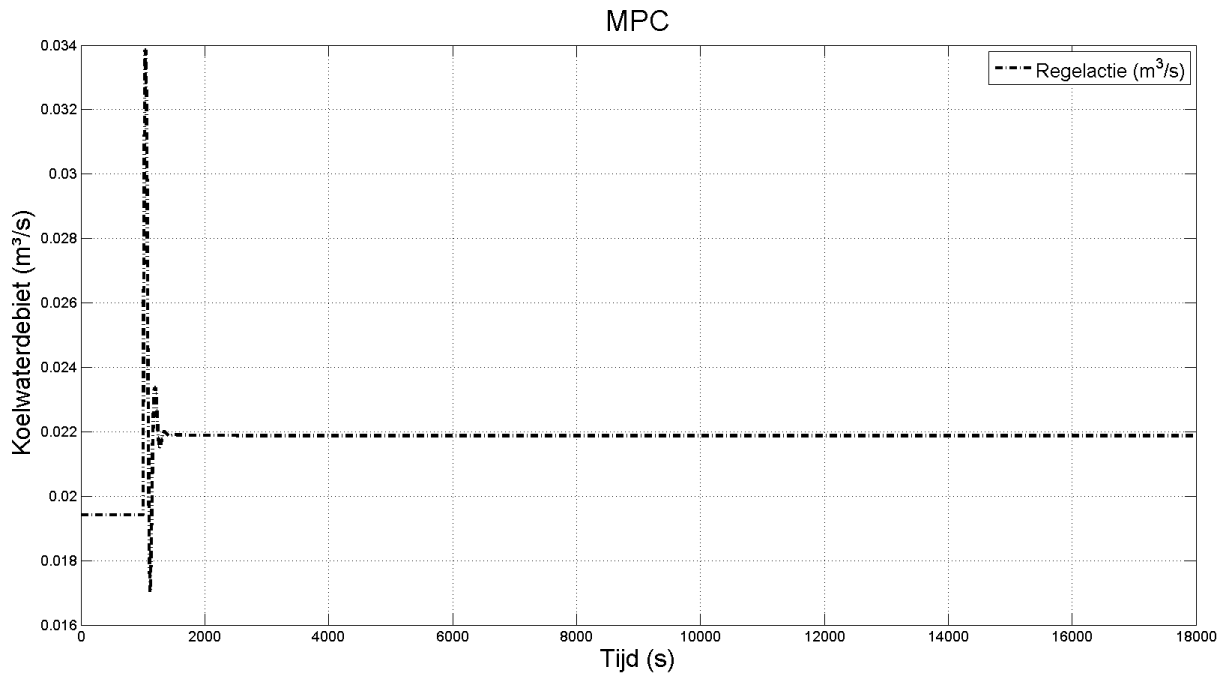


Figuur 74: Koelwaterrespons MPC 350K: stap in voedingsdebiet

Het 3<sup>e</sup> en laatste geval dat bekeken wordt is de verandering van de voedingstemperatuur van 294 naar 299K op 1000s. Voor de reactortemperatuur wijzigt dit in een kleine afwijking in temperatuur. Hier kan dezelfde conclusie getrokken worden als bij het vorige geval, namelijk dat er eigenlijk geen verandering in reactortemperatuur optreedt. De statische afwijking die hieronder wordt weergegeven is waarschijnlijk te wijten aan de numerieke benadering van het proces. Of het ook daadwerkelijk te wijten is aan numerieke afwijking kan eventueel achterhaald worden door het aanpassen van toleranties en iteratiestappen. Wanneer door het aanpassen van zulke factoren de stabiele waarde van de temperatuur een andere waarde benadert kan dit geconcludeerd worden. Dit wordt hier verder niet besproken. Het koelwaterdebiet in Figuur 76 maakt kortstondig enkele sprongen maar gaat zeer snel naar een nieuw stabiel debiet.



Figuur 75: Temperatuurrepons MPC 350K: stap in voedingstemperatuur

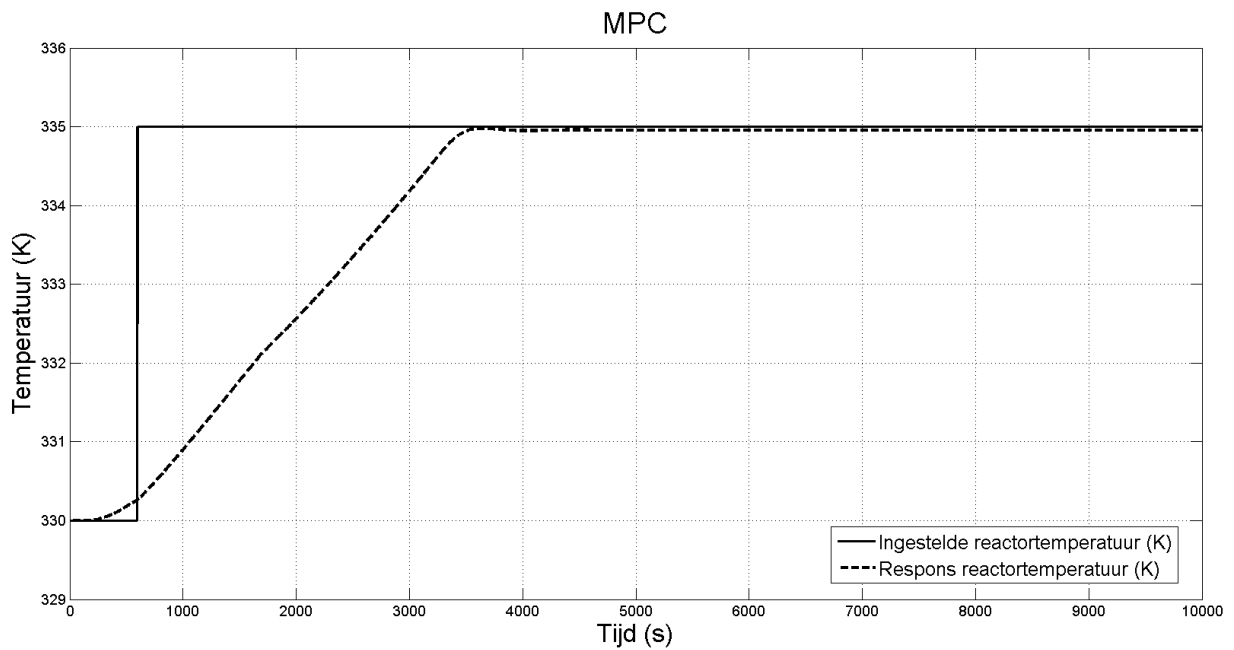


Figuur 76: Koelwaterrespons MPC 350K: stap in voedingstemperatuur

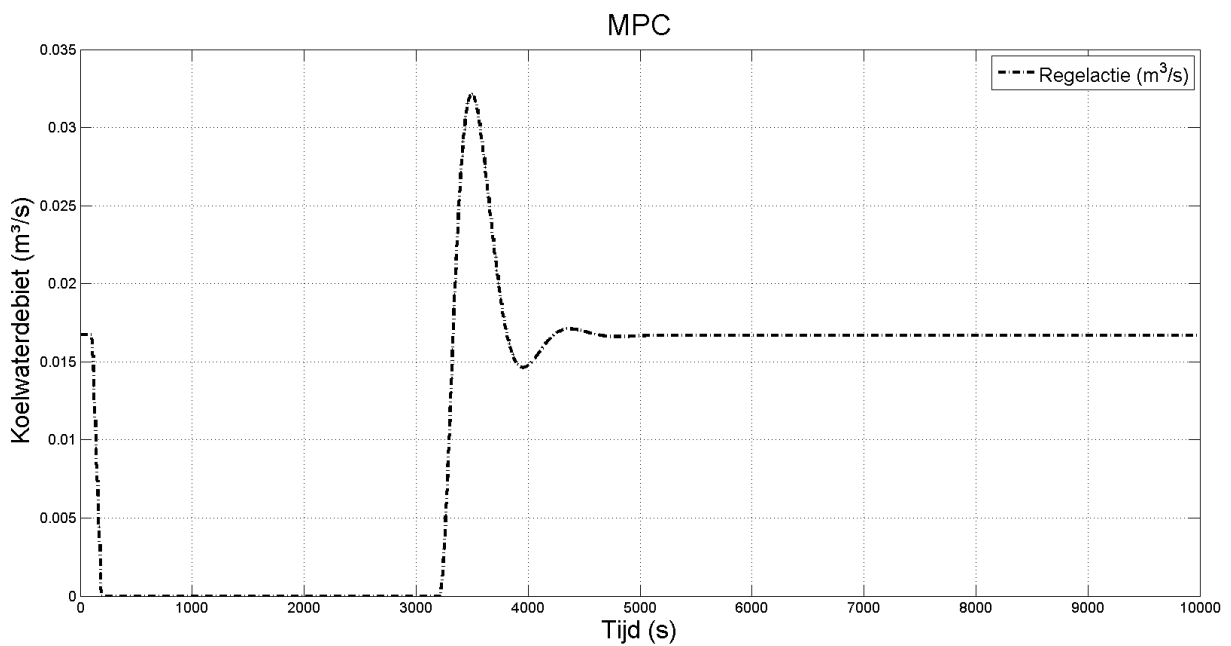
Multivariabel, reactortemperatuur = 330K en conversie = 95%

Bij laatste simulatie die wordt uitgevoerd wordt de reactortemperatuur én de concentratie beiden geregeld. De reactortemperatuur met behulp van het koelwaterdebiet en de concentratie met behulp van het voedingsdebiet. De casus die hiervoor gebruikt wordt is de CSTR met een reactortemperatuur van 330K. Er is geen specifieke reden waarom deze instelling gekozen is. Bij het model is geen rekening gehouden met een tijdsvertraging door een temperatuursensor om het concept te vereenvoudigen. De temperatuur krijgt na 600s een stap van 330K naar 335K, de concentratie krijgt op 1000s een stap van 95% conversie of  $8,01\text{kmol/m}^3 \cdot (1-0,95) = 4,005$  naar 96% conversie of  $8,01\text{kmol/m}^3 \cdot (1-0,96) = 0,3204\text{kmol/m}^3$ . De resultaten van deze simulatie worden hieronder weergegeven.

Figuur 77 geeft de stap en de respons weer van de reactortemperatuur. De respons van de reactortemperatuur blijft afwijken van de ingestelde waarde. Deze afwijking is niet groot, stabiel, maar toch aanwezig. Dit kan te wijten zijn aan een numerieke fout aan de ene kant, maar ook het feit dat het systeem berekend is op een conversie van 95% en niet 96% waardoor de linearisatie het proces niet meer accuraat omschrijft. Figuur 78 geeft de aangelegde regelactie op het koelwater weer. Hierop is te zien dat de reactortemperatuur niet sneller kan stijgen, het koelwaterdebiet is immers minimaal.

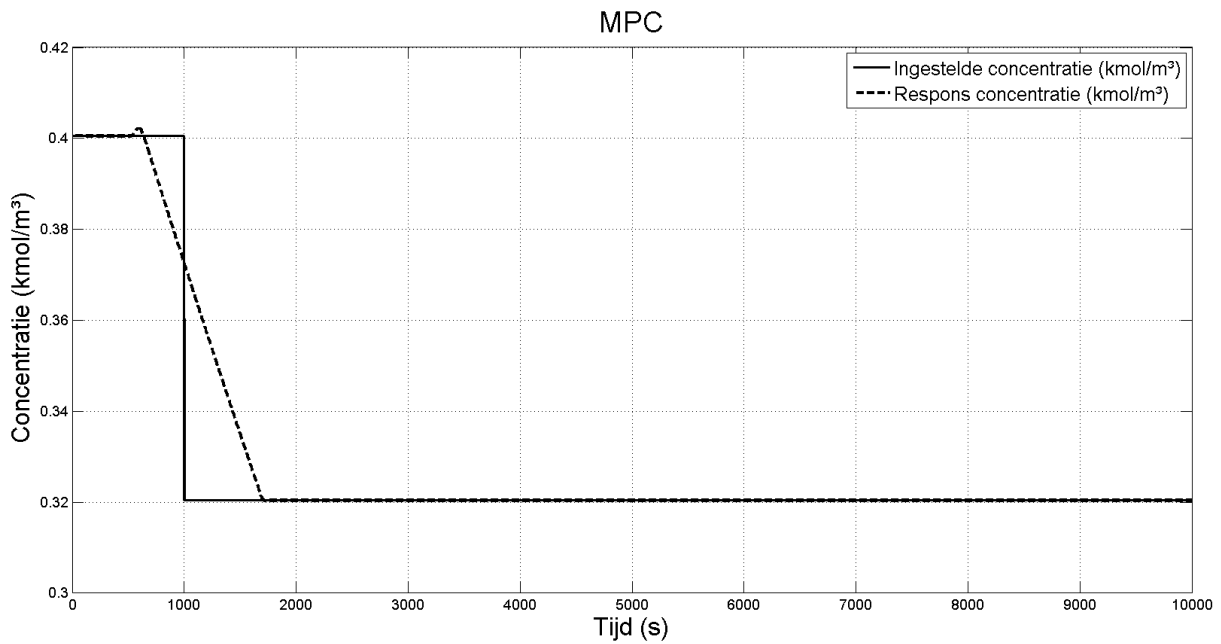


Figuur 77: Temperatuurrepons MPC 330K met 2 variabelen: stap op temperatuur en concentratie

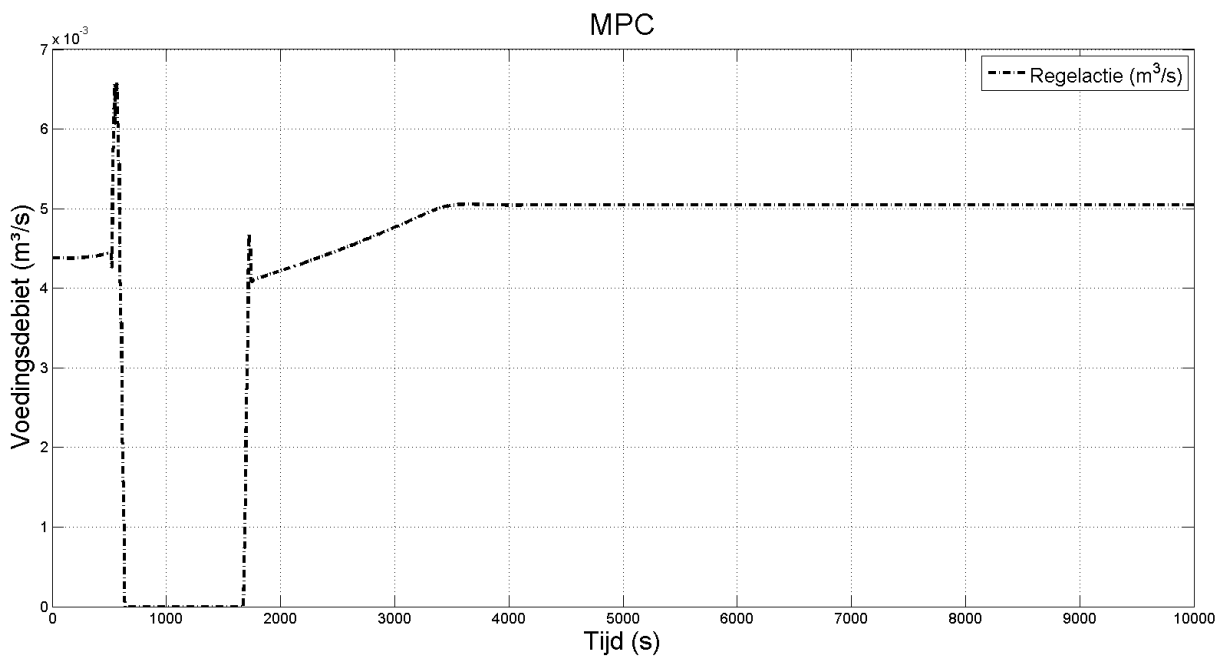


Figuur 78: Koelwaterrespons MPC 330K met 2 variabelen: stap op temperatuur en concentratie

Figuur 79 geeft het resultaat van deze simulatie weer voor wat betreft de concentratie. De stap van de ingestelde waarde wordt snel gehaald. Het hiermee corresponderende voedingsdebiet wordt weergegeven in Figuur 80. Ook hier wordt duidelijk dat de concentratie niet sneller kan veranderen dan gebeurt in de simulatie, omdat het voedingsdebiet naar 0 valt.



Figuur 79: Concentratierespons MPC 330K met 2 variabelen: stap op temperatuur en concentratie



Figuur 80: Voedingsdebietrespons MPC 330K met 2 variabelen: stap op temperatuur en concentratie

Het is duidelijk dat de MPC het proces naar behoren regelt. Zowel de concentratie als de reactortemperatuur worden simultaan geregeld naar hun nieuwe ingestelde waarde. Hoewel de stap pas plaatsvindt op 1000s vindt de eerste regelactie reeds eerder plaats.

### Conclusie

Met deze simulatie wordt de kracht van MPC werkelijk duidelijk. Beide parameters, zowel de concentratie als de reactortemperatuur, worden zo snel als voor het systeem mogelijk is,

veranderd (aangezien beide debieten 0 worden en lager dan 0 gaat niet). Daarna regelt de MPC het systeem ook zo dat de stabiele waarde behouden blijft.

### 5.2.1.3 Conclusie CSTR

Gezien de grote tijdsschalen waarmee gewerkt wordt in de grafieken zijn de stapjes in de regelactie niet meer duidelijk, maar deze zijn er nog wel wanneer wordt gefocust op de grafiek van de regelactie.

#### Geval 330K - 350K

Figuur 65 geeft de respons weer voor een verandering van ingestelde waarde van CSTR bij 330K. Figuur 71 geeft dezelfde respons weer voor een CSTR bij 350K. Het verschil dat hierbij opvalt is de snelheid waarmee de nieuwe ingestelde waarde bereikt wordt. De tijd die nodig is om deze waarde te bereiken is voor de CSTR op 330K ongeveer 3000s waar dit bij de CSTR bij 350K maar 1000s zijn. Dit komt o.a. doordat de reactor die ontworpen is voor 350K een kleinere reactorinhoud en oppervlak. Het specifiek oppervlak is groter waardoor de warmtewisseling beter gebeurt dan bij de CSTR bij 330K.

#### PID -MPC

Voor het verschil te bespreken tussen PID en MPC wordt de CSTR bekeken bij 330K. Tussen Figuur 49 en Figuur 65 moet wel opgelet worden dat de x-as een andere schaal heeft. Met een PID-regelaar haalt het systeem met de besproken instellingen na 5000s de nieuwe ingestelde waarde en is dan stabiel. Met de MPC doet het systeem er maar 3000s over, wat bijna een halvering is. Dit is een grote verbetering voor een industrieel proces, waarbij veel afhangt van de instelling van de PID. Met MPC is het systeem niet alleen sneller maar ook optimaler omdat er geen overshoot optreedt. Dit wordt toegeschreven aan het voorspellende karakter van de MPC. Dit is ook te zien in het feit dat de reactortemperatuur wanneer geregeld door MPC eerder stijgt dan bij de PID.

#### Multivariabel

Uit de simulatie met de CSTR waarbij 2 variabelen geregeld worden kan de conclusie overgenomen worden. Beide variabelen worden correct geregeld en bereiken binnen een minimale tijd hun nieuwe ingestelde waarde, waarbij de regelacties geen al te grote ongewenste variaties vertonen. Dit is helemaal in lijn van wat wordt verwacht met MPC.



## 6 Conclusie en blik vooruit

Tijdens deze masterproef werden 3 gevallen die gradueel stegen in complexiteit overlopen waarbij de instellingen voor de PID op diverse manieren werd uitgevoerd en een MPC voor werd ontworpen in MATLAB en Simulink. Daarbij lag de focus aan de ene kant op de verschillen tussen PID en MPC en het opdoen van ervaring voor het instellen van PID-regelaars. Langs de andere kant was er ook een focus op het uitwerken van een MPC in MATLAB en Simulink zonder gebruik te maken van de modelpredictieve *toolbox*. Daardoor is het achterliggend algoritme beter te achterhalen. De simulaties werden allemaal uitgewerkt met het oog op educatie, waarbij de opgebouwde modellen in Simulink later te gebruiken zijn in onderzoek gebaseerd onderwijs. De materie is zowel interessant voor chemische ingenieurs die regeltechniek aanleren, omdat een van de gevallen een concreet chemisch proces, als voor ingenieurs automatisatie die te maken krijgen met een uitgebreider reëel proces.

Door het simuleren kan snel en eenvoudig de reactie van een proces getest worden. Hierbij moet altijd in het achterhoofd gehouden worden dat simulaties niet perfect zijn en sterk afhangen van het model. Daarnaast kunnen numerieke fouten altijd optreden waardoor de simulatie wel nog altijd bruikbaar is om een beeld te schetsen van een proces en hoe het reageert.

De conclusies die getrokken kunnen worden voor wat betreft het verschil tussen PID en MPC zijn dezelfde als reeds op voorhand vermoed werden. De resultaten met de PID hangen sterk af van de instellingen, die dan weer afhangen van de gebruikte instelregels. Het instellen van MPC gebeurt vrij eenvoudig en buiten het invoeren van het model op basis van een discreet toestandsruimtemodel is iedere MPC hetzelfde voor alle behandelde processen. Wanneer voor een reëel proces de afweging gemaakt dient te worden tussen PID en MPC zijn naast de mogelijkheden van de regelaar ook de kostprijs en de performantie factoren die mee een rol spelen bij de beslissing. Gezien de kostprijs en het feit dat er ook processen zijn die goed geregeld kunnen worden met PID, zal PID niet snel weg te denken zijn in de chemische industrie.

MPC heeft verschillende voordelen. Ten eerste kan MPC goed overweg met variaties bij externe factoren die niet rechtstreeks gemeten worden. De temperatuur wordt over het algemeen beter en sneller geregeld dan dat met PID het geval is. In dit opzicht is de MPC superieur aan de PID.

Ten tweede kunnen aan een MPC beperkingen worden opgelegd aan verschillende variabelen, waarmee de MPC bij de berekening van de regelactie rekening mee zal houden. De verschillende soorten beperkingen zijn op een opbouwende manier weergegeven in de resultaten zodat de impact ervan stap voor stap duidelijk wordt. De werkingsgrenzen die in het PID-blok in Simulink worden opgelegd bestaan in werkelijkheid niet. In werkelijkheid regelt een PID een stuursignaal van 4 tot 20mA wat overeenkomt met 0 tot 100% klepstand in het geval dat een corrigerend orgaan een klep is. Een PID kan dus niet overweg met werkingsgrenzen van een proces.

Het derde voordeel van MPC situeert zich in het regelen van *MIMO*-systemen. Dit staat voor *Multiple Inputs, Multiple Outputs*. De manier waarop MPC hiermee omgaat is te zien in de simulatie waarbij zowel reactortemperatuur als concentratie geregeld worden. Dit is een grote stap vooruit omdat, indien dit met PID opgelost zou worden, er 2 PID-regelaars nodig zijn, die beiden ingesteld moeten worden en die geen rekening houden met elkaar. De manier waarop MPC werkt houdt wel rekening met andere variabelen.

Naast deze conclusies die getrokken worden uit de resultaten is het belangrijk om dit onderzoek te kaderen in de toestand voor deze masterproef en hoe het verder kan na de masterproef. Het is de bedoeling dat dankzij deze thesis de leerstof betreffende regeltechniek minder abstract wordt voor de student. Door gebruik te maken van een uitgebreider proces wordt zelfs integratie mogelijk tussen regeltechniek en reactorkunde. Daarnaast wordt voor de simulaties gebruik gemaakt van MATLAB en Simulink, wat een bepaalde kennis in programmeren vereist maar waardoor het algoritme wel duidelijk is. Ook is er geen speciale toolbox nodig voor het uitvoeren van de simulaties, wat de kosten drukt. Dit geldt zeker voor de MPC die niet zoals PID standaard in Simulink zit en volledig in MATLAB-code is uitgewerkt. Dankzij Simulink en de daarbij horende intuïtieve grafische interface is het eenvoudig om zaken in het model aan te passen. Daarnaast is het mogelijk om bepaalde factoren, zoals instellingen van MPC of het model dat door MPC wordt gebruikt, eenvoudig aan te passen omwille van het feit dat deze duidelijk uit de code springen.

In de toekomst is het mogelijk om meerdere gevallen verder uit te werken, vooral voor wat betreft *MIMO* bij MPC lijkt dit interessant. Zo staat er in (Luyben, 2007) een proces beschreven van een CSTR en destillatiekolom dat uitgewerkt zou kunnen worden. Verder werd bij het modelleren gebruik gemaakt van een model in afwijkingsvariabelen, het is ook mogelijk om dit verder uit te werken naar absolute variabelen. De MPC kan ook voorzien worden van een niet-lineair model waardoor de MPC bruikbaar wordt voor simulaties betreffende niet-lineaire modelpredictieve controle.



## 7 Bibliografie

- Aström, K. &. (2001). The Future of PID Control. *Control Engineering Practice*, 13.
- Aström, K. J. (2012, september 28). Feedback Systems: An Introduction for Scientists and Engineers. Princeton, New Jersey, VS: Oxford University Press.
- Di Ruscio, D. (2001). *Model Predictive Control and optimization*. z.u.
- Khalore, A. G. (2012). Relay approach for tuning of PID controller. *International Journal Computer Technology & Applications*, 6.
- Liptak, B. G. (2006). Tuning PID controllers. In B. G. Liptak, *Instrument Engineers' handbook* (p. 2388). Boca Raton, FL: CRC Press.
- Logist, F. (2013). *Cursus: Measurement and Control Technology for the (bio)chemical industry*.
- Luyben, W. L. (2007). *Chemical Reactor Design and Control*. Hoboken, New Jersey: John Wiley & Sons.
- Maciejowski, J. (2000). *Predictive Control with constraints*. Cambridge.
- MathWorks. (sd). MATLAB Help. MathWorks.
- Ogata, K. (2002). *Modern Control Engineering*. New Jersey: Prentice-Hall.
- Okko, H. B. (2001). *Design Methods for Control Systems*. z.u.
- Qin, S. J. (2002). A survey of industrial model predictive control technology. *Control Engineering Practice*, 32.
- Ruel, M. (2010, z.d.). *Closed loop tuning vs. closed loop tuning: tuning all your loops while the process is running is now possible*. Opgehaald van [www.topcontrol.com: http://www.topcontrol.com/fichiers/CLOSED\\_LOOP\\_TUNING\\_VS\\_OPEN\\_LOOP\\_TUNING\\_Tuning\\_all\\_your\\_loops\\_while\\_the\\_process\\_is\\_running.pdf](http://www.topcontrol.com/fichiers/CLOSED_LOOP_TUNING_VS_OPEN_LOOP_TUNING_Tuning_all_your_loops_while_the_process_is_running.pdf)
- Skogestad, S. (2001). Probably the best simple PID tuning rules in the world. *Journal of Process Control*, 27.
- Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, 19.
- van Schijndel, J. (2014, 01 14). *Simulating ODEs with S-functions, step by step*. Opgehaald van [Matlab Central - File exchange: http://www.mathworks.com/matlabcentral/fileexchange/4367-simulating-odes-with-s-functions--step-by-step](http://www.mathworks.com/matlabcentral/fileexchange/4367-simulating-odes-with-s-functions--step-by-step)
- Wang, L. (2009). *Model Predictive Control System Design and Implementation using MATLAB*. London: Springer-Verlag.

Wilson, D. I. (2005). Relay-based PID tuning. *Automation & Control*, 3.

## 8 Bijlagen

In deze bijlagen wordt de MATLAB-code van enkele belangrijke delen bij de simulaties weergegeven. Om de complexiteit van de MATLAB-code weer te geven wordt gebruik gemaakt van het aantal regels dat een MATLAB-bestand telt. Het laagste aantal regels bevindt zich in het bestand waarin de initiële parameters worden ingegeven voor de simulaties van de lineaire tank met de PID (geen invoer van factoren met betrekking tot MPC). Dit bestand bevatte 5 regels. Het bestand met de meeste regels MATLAB-code bevatte 100 regels en maakte deel uit van de MPC.

Omwille van de lengte worden in de bijlagen enkel de code weergegeven van de S-functies en van het model gebruikt door de MPC, enkel voor het geval van de lineaire tank.

### 8.1 Simulink lineaire tank

Zie Figuur 31.

### 8.2 ODE lineaire tank

```
function [sys,x0,str,ts] = ODE_LinTank(t,x,u,flag,p)
% LinTank model description
% The water inflow q [m³/s] into a tank can be manipulated. The water
% outflow is determined by a valve such that it is proportional
% to the water level h [m]. The resistance to water flow is R [s/m²]. The
% tank has a constant cross-section A [m²] and the water has a constant
% density rho [kg/m³].

% inputs u: inflow Fin
% states x: water level in tank h
% output h: water level in tank h

% Matlab/Simulink automatically selects the appropriate function output.
switch flag,
    % Initialization %
    case 0,
        [sys,x0,str,ts] = mdlInitializeSizes(p);
    % Derivatives %
    case 1,
        sys = mdlDerivatives(t,x,u,p);
    % Outputs %
    case 3,
        sys = mdlOutputs(t,x,u,p);
    % Unhandled flags %
    case { 2, 4, 9 },
        sys = [];
    % Unexpected flags %
    otherwise
        error(['Unhandled flag = ',num2str(flag)]);
end

%=====
% mdlInitializeSizes
% Return the sizes, initial conditions and sample times for the S-function.
```

```

%=====
function [sys,x0,str,ts] = mdlInitializeSizes(p)
sizes = simsizes;
sizes.NumContStates = 1;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 1;
sizes.NumInputs = 1;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0 = p.x0;
str = [];
ts = [0];

%=====
% mdlDerivatives
% Return the derivatives for the continuous states.
%=====
function sys = mdlDerivatives(t,x,u,p)
% model parameters
R = p.R; % valve resistance [s/m2]
A = p.A; % cross-section [m2]
% inputs
q = max(u(1),0); % inlet
% states
h = x(1);
% ODEs
hdot = min(-1/(A*R)*h,0) + 1/A*q;
% return derivatives
sys = [hdot];

%=====
% mdlOutputs
% Return the block outputs.
%=====
function sys = mdlOutputs(t,x,u,p)
% states
h = x(1);
% outputs
y = h;
% return outputs
sys = y;

```

### 8.3 S-functie lineaire tank

```

function controller = my_MPC_1tank(model,p)

% Specify controller parameters
controller.H_p = 6; % prediction horizon;
controller.H_c = 4; % control horizon

% Specify reference

```

```

y = load('y_ref.mat');
controller.t_ref_init = y.ans.Time';
controller.ref_init = y.ans.Data';
controller.t_ref = y.ans.Time';
controller.ref = y.ans.Data';

% Specify last control action
controller.uk_1init = [model.xinit/p.R]; % last control applied before initialisation
controller.uk_1 = controller.uk_1init; % last control applied
global uk_1
uk_1 = controller.uk_1;

% controller.Q = input('Give weight matrix Q: ')
controller.Q = [1];

% controller.R = input('Give weight matrix R: ')
controller.R = [1];

% controller.QP solver: ('quadprog' (default) or 'qpOASES')
controller.QP solver.type = 'quadprog'; % 'qpOASES'

% For the constrained case
% *****
% controller.DeltaU_min = input('Give vector of lower bounds on control moves: ')
controller.DeltaU_min = [-0.5];%[-1.5];

% controller.DeltaU_max = input('Give vector of upper bounds on control moves: ')
controller.DeltaU_max = [0.5];%[1.5];

% controller.E = input('Give constraint matrix E: E*Delta u <=e: ')
% controller.e = input('Give constraint vector e: E*Delta u <=e: ')
controller.E = [0];%[1;-1];
controller.e = [0];%[1.5;1.5];

% controller.F = input('Give constraint matrix F: F*u <=f: ')
% controller.f = input('Give constraint vector f: F*u <=f: ')
controller.F = [1;-1];
controller.f = [3.3;-0.8];

% controller.G = input('Give constraint matrix G: G*x <=g: ')
% controller.g = input('Give constraint vector g: G*x <=g: ')
controller.G = [1;-1];
controller.g = [4.7;0];

% *****
% DO NOT CHANGE FROM HERE ON
% *****
% Check consistency of the MPC
check_consistency_my_MPC(model,controller)

% Build the MPC
controller = build_MPC(model,controller);

```

## 8.4 MPC-model lineaire tank

```
function [model]=my_model_1tank(p)
% 1-tank model
Ts = 0.1;          % sample time
model.n_control = 1; % # controls
model.n_states = 1; % # states
model.n_meas = 1; % # measurments
model.t_sample = Ts; % # sample time

A=p.A;
R=p.R;

model.A = [1-(Ts/(A*R))];
model.B = [Ts/A];
model.Cz = [1];
model.D = [0];

model.xinit = [3]; % initial state
model.xk = model.xinit; % current state

% *****
% DO NOT CHANGE FROM HERE ON
% *****
% Check consistency of my model
check_consistency_my_model(model)
```

## Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:

**Simuleren en ontwerpen van klassieke en geavanceerde regelschema's in de chemische industrie**

Richting: **master in de industriële wetenschappen: chemie**

Jaar: **2014**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

**Decramer, Gerd**

Datum: **11/06/2014**