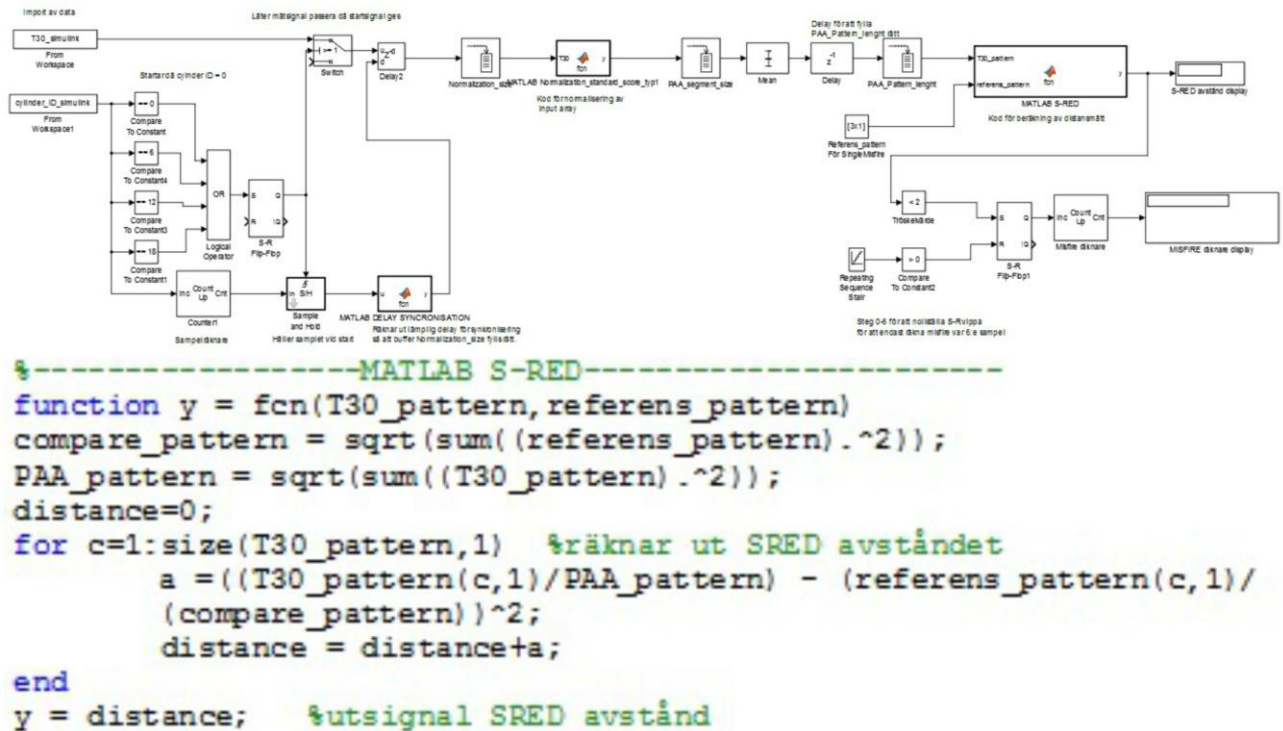




Misfiredetektion baserat på mönsterigenkänning

Misfire detection based on pattern recognition

Examensarbete inom högskoleingenjörsprogrammet Mekatronik

Oscar Olhag Olausson
Rikard Friberg

FÖRORD

Det här examensarbetet är det avslutande momentet på högskoleingenjörsprogrammet Mekatronik (180Hp) vid Chalmers Tekniska Högskola. Arbetet utfördes under 8 veckor på våren 2016 på uppdrag av Volvo Cars. Vi vill tacka vår handledare Niklas Särnberger på Volvo Cars som gjort detta arbete möjligt och för all hjälp han bistått med. Vi vill även tacka vår handledare och examinator Thomas McKelvey.

Oscar Olhag Olausson och Rikard Friberg, Göteborg 2016

SAMMANFATTNING

Den här rapporten beskriver processen för att utvärdera ett befintligt koncept för detektion av uteblivna förbränningar i en fyrcylindrig bensinmotor från Volvo Cars och implementationen av konceptet i Simulink. Att detektera uteblivna förbränningar även kallat misfire är av stor vikt i On Board Diagnostics (OBD II).

Genom komprimering av data från motorns vevaxelgivare är det möjligt att detektera en misfire om datat jämförs med ett referensmönster. Detektion sker med en variant av Euklidiskt distansmått, en metod för att bestämma avståndet mellan två punkter i ett euklidiskt rum.

I arbetet används främst mjukvaran Matlab från Mathworks. Detektionsalgoritmen är implementerad och testad i Simulink från Mathworks.

Resultatet av arbetet är att den framtagna detektionsalgoritmen inte är tillräckligt noggrann för att detektera misfire i alla testfall. Noggrannheten av algoritmen minskar med ökande varvtal hos motorn som en följd av den ökande likheten mellan en misfire och en normal förbränning.

ABSTRACT

This report describes the process to evaluate an existing concept for detection of incomplete combustions in a four cylinder petrol engine from Volvo Cars and implement the concept in Mathworks Simulink. Detection of incomplete combustions, also called misfire is of great importance in On Board Diagnostics (OBD II).

Thru compression of data from the angular velocity of the engine flywheel it is possible to detect a misfire if the data is compared to a reference pattern. The detection uses a variant of the Euclidean distance, a method to determine the distance between two points in an Euclidean space.

The work is mainly performed with the software Matlab from Mathworks. The detection algorithm is implemented and tested in Mathworks Simulink.

The result of the project is that the produced detection algorithm is not sufficiently to detect misfire in all test cases. The accuracy of the detection decreases when the speed of the engine increases as a result of the increasing similarity of a misfire and a normal combustion.

INNEHÅLLSFÖRTECKNING

1.	INLEDNING	1
1.1	Bakgrund	1
1.2	Syfte	1
1.3	Konceptbeskrivning.....	1
1.4	Avgränsningar	2
2.	TEORI/TEKNIK BAKGRUND.....	3
2.1	OBD II.....	3
2.2	Misfire	3
2.3	Vevaxelgivare.....	4
2.4	Fixed Sliding window	4
2.5	Normalisering.....	4
2.6	Root-Mean-Squared Error	6
2.7	Normaliserad Root-Mean-Square Error	6
2.8	Piecewise Aggregate Approximation (PAA)	7
2.9	Symbolic Aggregate Approximation (SAX)	7
2.10	Euklidisk distans.....	9
2.11	Histogram	10
2.12	CSV-fil	11
3.	METOD.....	11
3.1	PAA eller SAX?	11
3.2	Val av normalisering metod och längd.....	12
3.3	Mönsterlängd bestämning	12
3.4	Mätdata.....	13
3.5	Mätsignaler och övningsdata	13
3.6	Metodik för val av referensmönster	14
3.7	Metodik för verifiering av referensmönster.....	15
3.8	Matlab program	15
3.9	Simulink-modell	16
4.	LÖSNINGSALTERNATIV	18
4.1	Val av testfall.....	18
5.	RESULTAT.....	19
5.1	Utvärdering av normaliseringsmetod m.h.a NRMSE.....	19
5.2	Utvärdering av testfall	20
5.3	Implementering i Mathworks Simulink.....	27
6.	Diskussion och slutsats.....	29

7. Rekommendation för framtida arbete.....	31
8. Referenser.....	32

Bilaga 1: Flödesschema och matlabkod program 1.....	
Bilaga 2: Flödesschema och matlabkod program 2.....	
Bilaga 3: Flödesschema och matlabkod program 3.....	
Bilaga 4: Simulink modell.....	
Bilaga 5: Möjlig detektion i följande testfall.....	
Bilaga 6: Noggrannhet i detektion per mätfil	
Bilaga 7: Referensmönster misfire	
Bilaga 8: Referensmönster normal förbränning	

TERMINOLOGI / FÖRKORTNINGAR

Misfire	Ofullständig förbränning
OBD	On-Board Diagnostics
PAA	Piecewise Aggregate Approximation
RED	Relativ Euklidisk Distans
RMSE	Root Mean Square Error
SAX	Symbolic Aggregate Approximation
SRED	Kvadratisk Relativ Euklidisk Distans

1. INLEDNING

Inledningens syfte är att ge klarhet över bakgrunden till arbetet, dess syfte och eventuella avgränsningar.

1.1 Bakgrund

Inom OBD-lagstiftningen (On-Board Diagnostics) förekommer begreppet Misfire (misständning). Definitionen av Misfire är att det på en eller flera cylindrar i en motor med styrd tändning inte uppstår någon förbränning av bränslet. Detta kallas även för utebliven förbränning och måste detekteras enligt lagkrav. Lagkraven är marknadsberoende gällande fordonets emissioner men gemensamt är att en utebliven förbränning måste kunna detekteras med avseende på komponentskydd och emissioner.

Utebliven förbränning kan leda till förhöjd avgastemperatur som i sin tur kan skada och i vissa fall förstöra fordonets katalysator och påverka fordonets emissioner negativt. Förhöjda emissioner påverkar luftkvaliteten och människors hälsa negativt samt kan leda till förurning av mark och sjö samt förhöjda halter växthusgaser i atmosfären. Enligt OBD-lagstiftningen skall alla händelser som påverkar fordonets emissioner negativt loggas med felkoder i fordonets OBD-system. Där lagras alla tillfällen då uteblivna förbränningar detekteras. OBD-systemet skall även kunna detektera på vilken cylinder en ofullständig förbränning sker.

1.2 Syfte

Syftet med projektet är att implementera och utvärdera ett befintligt diagnos koncept för misfire från Volvo Cars i Mathworks Simulink. Målen i projektet är att undersöka om det befintliga konceptet kan detektera misfire samt få ett mätresultat på hur robust konceptet är vid olika körfall.

Frågeställningarna i projektet är:

1. Kan mönsterigenkänningsalgoritmen detektera misfire?
2. Fungerar konceptet för olika last och varvtal på motorn?
3. Hur påverkas mönsterigenkänningen av olika normaliseringsmetoder samt mönsterlängd?

Beroende på utfallet i frågeställning 3 kommer vi att undersöka möjligheten att hitta en metod för bestämning av mönsterlängd

1.3 Konceptbeskrivning

Konceptet från Volvo Cars som skall utvärderas baseras på komprimering av en mätsignal från vevaxelgivaren med hjälp av Piecewise Aggregate Approximation (PAA) samt mönsterigenkänning med metoden Symbolic Aggregate Approximation (SAX).

Mätsignalen från vevaxelgivaren kommer från en av Volvo Cars 4-cylindriga bensin motorer.

1.4 Avgränsningar

Projektet begränsas till att utföras med hjälp av Matlab och Simulink från Mathworks samt MDA 7.1 från Etas som Volvo Cars använder för att analysera signaler. Alla beräkningar kommer ske i Matlab/Simulink.

Adaption av svänghjul på vevaxeln kommer att utelämnas i detta projekt. Inga nya mätserier kommer att utföras då detta ligger utanför tidsramen för projektet.

Detta projekt innefattar en undersökning och utvärdering av Volvo Cars koncept. Därför kommer simulering av konceptet utföras men implementation i fordon av utelämnas.

2. TEORI/TEKNIK BAKGRUND

2.1 OBD II

OBD är ett diagnosystem som integreras både i personbilar och lätta fordonets datorsystem [1]. Den nuvarande versionen av diagnosystemet, OBD II infördes först i alla nya fordon under 1996 i Kalifornien och har därefter blivit en standard för alla nytillverkade personbilar och lätta fordon världen över.

OBD-systemet övervakar fordonets alla komponenter som kan påverka fordonets emissioner negativt.

Om ett fel detekteras av OBD-systemet ska Malfunction Indicator Lamp, MIL tändas. MIL indikatorn är även känd under namnet "Check Engine". Vid fel skall även information loggas av OBD-systemet. Syftet med detta är att underlätta för service- och reparationspersonal vid eventuella fel och därmed hålla utsläppen på en acceptabel nivå under hela fordonets livstid.

2.2 Misfire

Misständning eller även kallat misfire innebär avsaknaden av förbränning i en eller flera av motorns cylindrar p.g.a. bland annat otillräcklig kompression av bränslet eller felande tändning [2]. Uppdragsgivaren definierar misfire som ofullständig förbränning på en eller flera cylindrar i en motor med styrd tändning, denna definition kommer användas i resterande rapport.

Misfire diagnostik är en stor del utav OBD och innebär att förbränningar som kan öka fordonets emissioner över gällande gränsvärden eller som kan vara skadliga för fordonets katalysator utav det totala antalet förbränningar skall observeras. Enligt gällande lagstiftning skall dessa händelser loggas i fordonets OBD-system med information om antalet uteblivna förbränningar och på vilken cylinder det skedde. Syftet med diagnostiken är därmed att minska fordonets emissioner samt skydda fordonets katalysator.

I detta projekt behandlas fyra fall av misfire:

- Single misfire – misständning på en cylinder
- Pair misfire – misständning på två cylindrar med en normal förbränning mellan
- Consecutive misfire – misständning på två cylindrar i följd
- First/last misfire – misständning på första och sista cylindern i tändningsföljden

I tabell 2.1 visas antalet kombinationer av misfire som kan inträffa för en hel tändcykel.

Tabell 2.1: Kombinationen av misfire typer för en hel tändningscykel

Misfire typ	Cylinder 1	Cylinder 3	Cylinder 4	Cylinder 2
Single	X			
Single				x
Single		X		
Single			x	
Consecutive	X	X		
Consecutive		X	x	
Consecutive			x	x
Pair	X		x	
Pair		X		x
First/last	X			x

2.3 Vevaxelgivare

Vevaxelgivaren mäter tiden det tar för vevaxeln att rotera i detta fall 30 grader och beskriver detta med ekvation 2.1 [2]. Ekvationen beskriver antalet sampel per tändcykel hos en fyrtaktsmotor. Där 2×360 grader motsvarar en hel tändcykel och 30 grader motsvarar upplösningen för samplingen.

$$\frac{2 \times 360^\circ}{30^\circ} = 24 \text{ sampel} \quad (2.1)$$

2.4 Fixed Sliding window

Analys av ett dataset med sliding window innebär att ett glidande fönster appliceras på datasetet [3]. Det glidande fönstret är det område av datasetet som betraktas vid ett specifikt tillfälle, exempelvis kan detta användas vid analys av realtidsdata. Glidande fönster kan antingen överlappa varandra, se tabell 2.2 alternativt påbörja där det senaste fönstret slutar, se tabell 2.3.

Tabell 2.2: Visualisering av glidande fönster med fem sampel storlek med överlappning.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Fönster 1	0	1	2	3	4	5	6	7	8	9
Fönster 2	0	1	2	3	4	5	6	7	8	9
Fönster 3	0	1	2	3	4	5	6	7	8	9
Fönster 4	0	1	2	3	4	5	6	7	8	9
Fönster 5	0	1	2	3	4	5	6	7	8	9
Fönster 6	0	1	2	3	4	5	6	7	8	9

Tabell 2.3: Visualisering av glidande fönster med fem sampel storlek utan överlappning.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Fönster 1	0	1	2	3	4	5	6	7	8	9
Fönster 2						5	6	7	8	9

2.5 Normalisering

Normalisering används för att skala data till ett specifikt intervall. Ett vanligt normaliseringsintervall är $[-1,1]$. Normalisering används exempelvis vid jämförelse av signaler med olika amplitud eller förskjutningar.

2.5.1 Normalisering med Standard score

Normalisering med avseende på standardavvikelse är även känt som Standard score och z-score [4]. Det normaliserade värdet kan ses som ett distansmått från medelvärdet för det aktuella datasetet. Om det normaliserade värdet är positivt ligger mätvärdet över medelvärdet för datasetet och är värdet negativt är mätvärdet mindre än medelvärdet för datasetet. Normalisering med avseende på standard avvikelse är endast definierat om datasetet som betraktas är känt.

Normalisering med avseende på standardavvikelse beskrivs enligt ekvation 2.2.

$$z = \frac{x - \mu}{\sigma} \quad (2.2)$$

Där z är det normaliserade värdet, x är det samplade värdet, μ är medelvärdet för datasetet och σ är standardavvikelsen för datasetet.

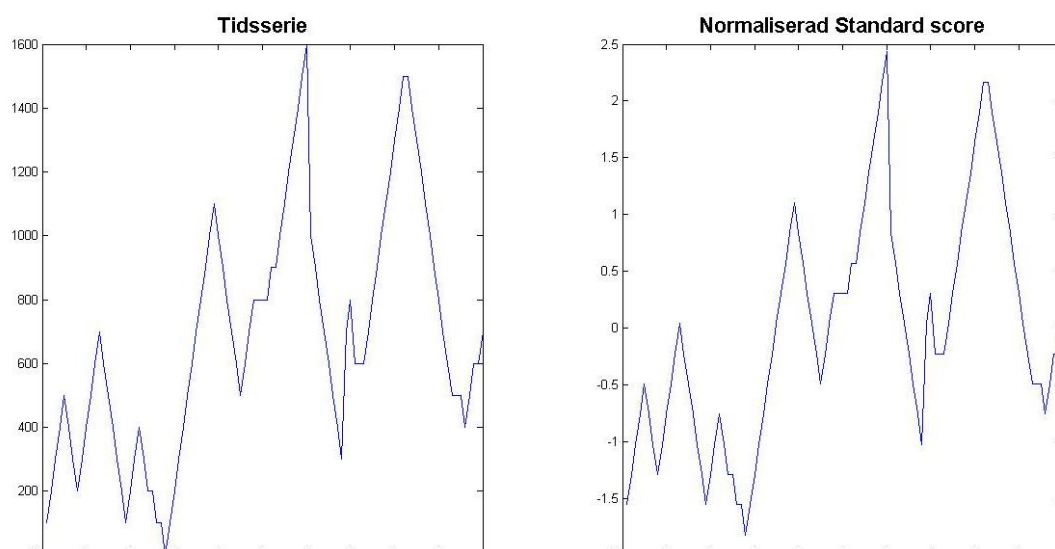
Medelvärdet beräknas utifrån summan av alla samplade värden i datasetet dividerat med antalet samplade värden enligt:

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (2.3)$$

Standard avvikelse beräknas genom att summera differensen i kvadrat mellan varje sampel minus medelvärdet för det aktuella datasetet. Summan divideras sedan med antalet sampel och därefter dras kvadratroten enligt [5].

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (2.4)$$

I figur 2.1 jämförs en tidsserie före och efter normalisering med metoden standard score. Tydligt syns att serien skalas från 0-1600 före normalisering till -2-2,5 efter normaliseringen och samtidigt behåller sitt utseende.



Figur 2.1: Tidsserie jämfört med normaliserad diskret tidsserie med 100 sampel med metoden standard score.

2.5.2 Normalisering med Feature Scaling

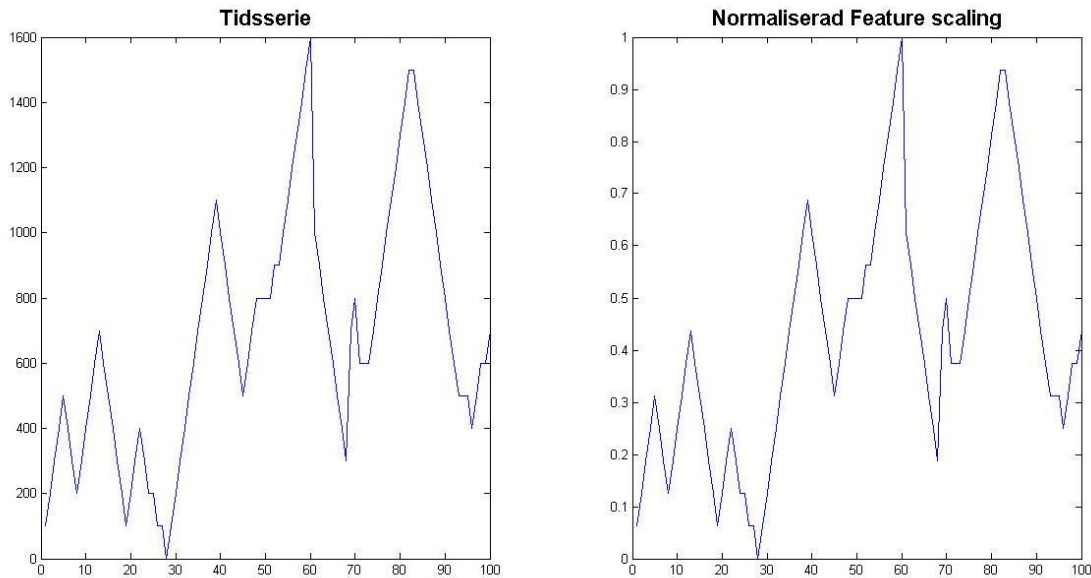
Att normalisera med Feature scaling innebär att datasetet skalas inom intervallet $[0,1]$ eller $[-1,1]$. Där det minsta respektive största samplade värdet utgör gränsvärdena i intervallet.

Normalisering med Feature scaling utförs genom att ta differensen av det samplade värdet och det minsta samplade värdet i datasetet. Följt av division differensen av det största och det minsta samplade värdet [6]. Beräkningarna sker enligt

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (2.5)$$

Där x' är det normaliserade värdet, x är det samplade värdet, $\min(x)$ är det minsta värdet inom datasetet samt $\max(x)$ är det största värdet i datasetet.

I figur 2.2 jämförs en tidsserie före och efter normalisering med Feature scaling i intervallet [0,1]. Likaså som i normalisering med standard score metoden behåller tidsserien sitt utseende efter normaliseringen.



Figur 2.2: Tidsserie jämfört med normaliserad tidsserie med metoden feature scaling

Normalisering med Feature Scaling kan även göras i intervallet [-1,1] enligt:

$$x' = 2 \times \left(\frac{x - \min(x)}{\max(x) - \min(x)} \right) - 1 \quad (2.6)$$

2.6 Root-Mean-Squared Error

Root-Mean-Squared Error, RMSE kan användas för att bestämma noggrannheten mellan olika modeller för analys av tidsserie [7]. RMSE beskrivs enligt:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (x_{1,t} - x_{2,t})^2}{n}} \quad (2.7)$$

Där $x_{1,t}$ och $x_{2,t}$ är tidsserierna som skall jämföras varav $x_{1,t}$ är referens tidsserien.

2.7 Normaliserad Root-Mean-Square Error

Normaliserad RMSE, förkortat NRMSE är ett sätt att jämföra olika modeller eller dataset med olika skalor [7]. Ett vanligt sätt att beräkna NRMSE är genom att dividera RMSE värdet med dess max och minvärdet datasetet antar enligt:

$$NRMSE = \frac{RMSE}{x_{2,t,MAX} - x_{2,t,MIN}} \quad (2.8)$$

Där $x_{2,t,MAX}$ och $x_{2,t,MIN}$ är det största respektive minsta värdet tidsserien $x_{2,t}$ antar.

Ofta visas NRMSE resultatet som en procentsats enligt:

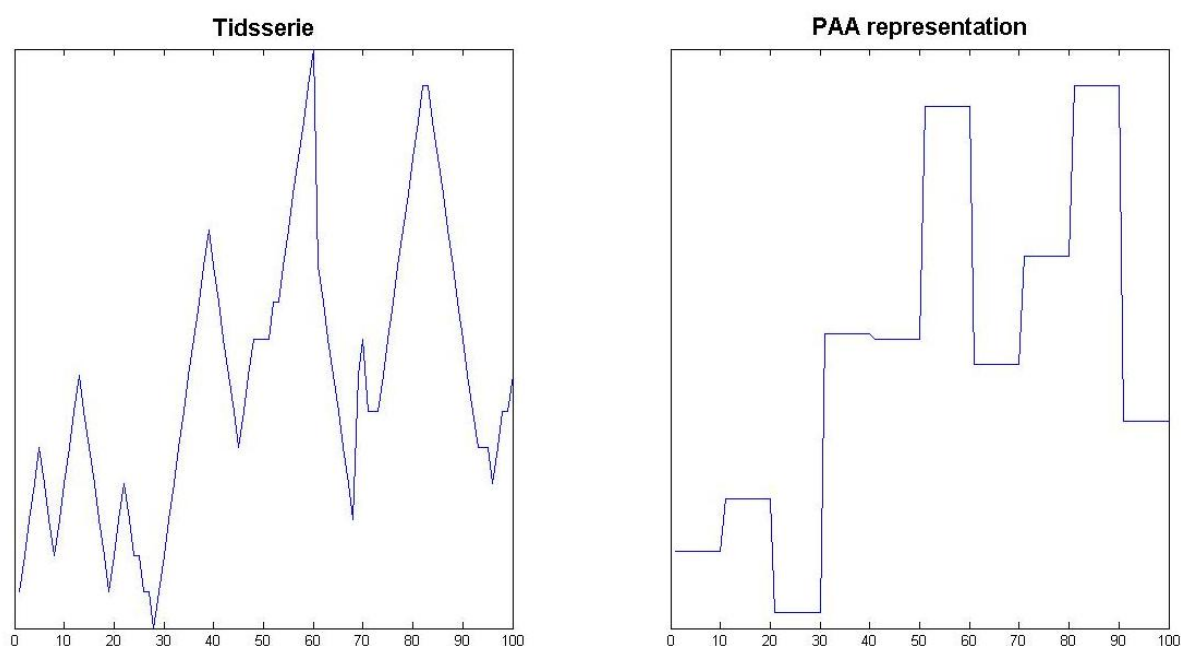
$$NRMSE (\%) = \left(\frac{RMSE}{x_{2,t,MAX} - x_{2,t,MIN}} \right) \times 100 \quad (2.9)$$

Där 0 % motsvarar att tidserierna är identiska och där en ökande procentsats motsvarar en ökande olikhet mellan tidsserierna.

2.8 Piecewise Aggregate Approximation (PAA)

Piecewise Aggregate Approximation (PAA) är en metod för datautvinning där en tidsserie n , delas upp i N lika stora segment [8]. Följt av en beräkning av medelvärdet för varje segment. Resultatet från metoden är en vektor med reducerad data. Eftersom metoden endast innebär enkla beräkningar är den lämplig för stora dataset på grund av dess effektivitet [9].

I figur 2.3 görs en jämförelse mellan en tidsserie med $n=100$ och dess PAA representation där med $N=10$.

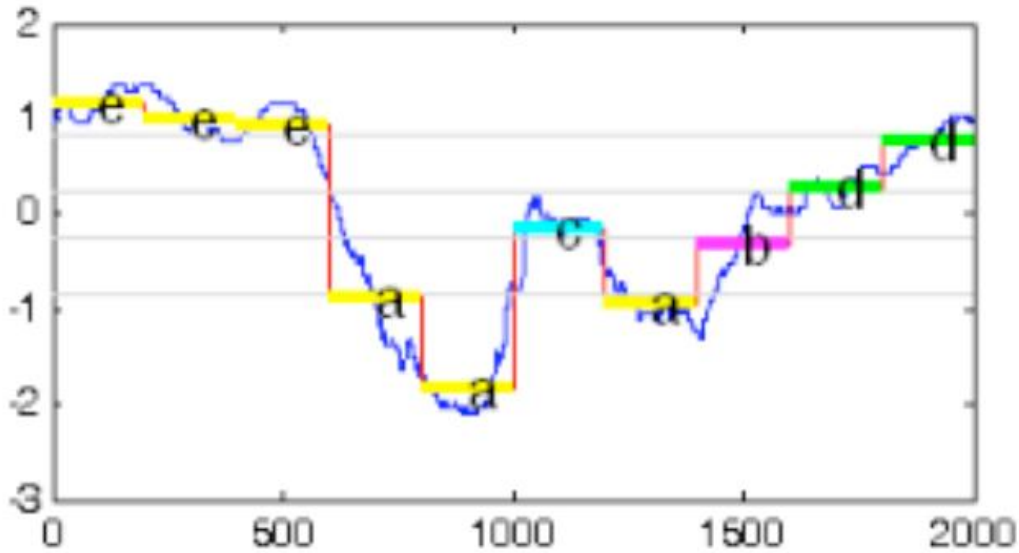


Figur 2.3: Plot av tidsserie ($n=100$) och dess PAA representation med $N=10$

Specialfallen då $N=1$ samt $N=n$ bör uppmärksammas. Då $N=1$ ges ett medelvärde över hela tidsserien n . Då $N=n$ ges blir PAA representationen identisk med tidsserien n , ingen komprimering av data sker [9].

2.9 Symbolic Aggregate Approximation (SAX)

Symbolic Aggregate Approximation, även kallat SAX, är en metod för komprimering av data. Metoden på normalisering av en tidsserie följt av bestämning av en PAA representation [8]. PAA representationen delas därefter upp i förutbestämda nivåer som vanligen presenteras av en bokstav. En visualisering kan ses i figur 2.4 för hur en tidsserie kan representeras med PAA.



Figur 2.4: Normaliserad tidsserie där PAA metoden applicerats. Därefter delats upp i förutbestämda nivåer som representeras av en bokstav enligt SAX metoden. I SAX metoden sker kategoriseringen av nivåerna sker med en hjälp utav en normalfördelningskurva. Bild hämtad från [10].

SAX metoden kan uttryckas som tidsserien C med längd n och representeras av en vektor med w dimensioner. Vektorn uttrycks som $\bar{C} = \bar{c}_1, \dots, \bar{c}_w$ där element i av $\bar{C}$ beräknas enligt:

$$\bar{c}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} C_j \quad (2.7)$$

Där $\bar{C}$ är medelvärdet över mätdatat i det aktuella datasetet

Enligt [8] definieras brytpunkterna för nivåerna enligt nedan:

Definition. Brytpunkter: Brytpunkter är en sorterad lista med nummer $B = \beta_1, \dots, \beta_{a-1}$ så att arean under en $N(0,1)$ gaussisk kurva från β_i till $\beta_{i+1} = 1/a$ där β_0 och β_a är definierade som $-\infty$ respektive ∞ .

Enligt [8] definieras längden av alfabetstecket enligt:

Definition. Ord: En subsekvens av tidsserien C med längden n kan bli representerad som ett ord $\hat{C} = \hat{c}_1, \dots, \hat{c}_w$. Låt α_i vara element i i alfabetet, det vill säga $\alpha_1 = a$ och $\alpha_2 = b$ o.s.v. Då fås mappningen av PAA representationen $\bar{C}$ av subsekvensen till ett ord $\hat{C}$ enligt ekvation 2.8

$$\hat{c}_i = \alpha_i \quad \text{om} \quad \beta_{j-1} \leq \bar{c}_i \leq \beta_j \quad (2.8)$$

Enligt [8] påverkar värdena av α och ord längden w endast effektiviteten på SAX algoritmen och inte det slutgiltiga resultatet. Valet av ord längden w beror på tidsseriens utseende och komplexitet. Generellt menar [8] att en jämn och långsamt förändrade tidsserie gynnas av ett lägre w värde medans en mer komplex tidsserie bör ha ett högre w värde. Val av SAX alfabetlängd a bör vara tre till fyra tecken långt.

2.10 Euklidisk distans

Euklidisk distans används för att beräkna ett linjärt avstånd mellan två punkter i ett Euklidiskt rum [11]. Metoden bygger på Pythagoras sats men kan tillämpas i en till n dimensioner jämfört med Pythagoras sats där endast två dimensioner används.

Euklidisk distans i n dimensioner där p_i och q_i är punkterna och avståndet mellan punkterna är $d(p, q)$ beräknas enligt ekvation 2.8.

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (2.8)$$

2.10.1 Kvadratisk Euklidisk distans

Kvadratisk Euklidisk distansmätt är en variant av det ursprungliga Euklidiska distansmättet där uttrycket kvadreras enligt:

$$d(p, q) = \sum_{i=1}^n (p_i - q_i)^2 \quad (2.9)$$

Detta används då större vikt vill läggas vid de punkter som ligger långt ifrån varandra [11].

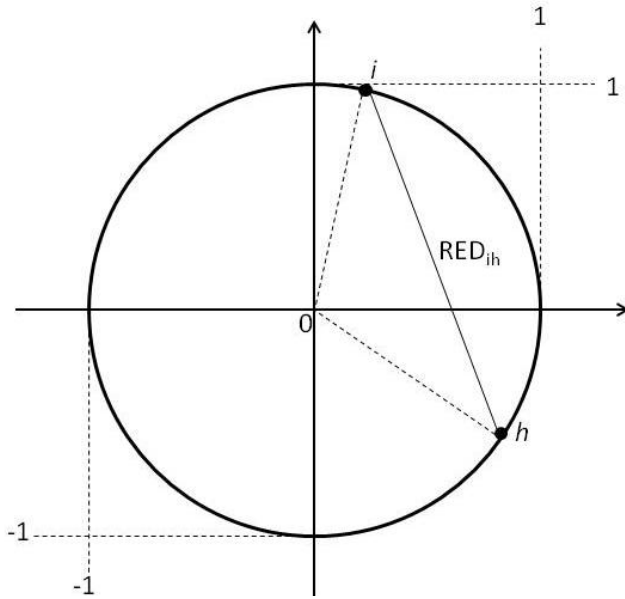
2.10.2 Relativ Euklidisk distans

Relativ Euklidisk distansmätt, RED, är en variant på Euklidisk distansmättet där värdena räknas om så att de hamnar på enhetscirkeln [12]. Distansen för RED blir sekanten mellan punkterna och beräknas enligt:

$$RED_{ih} = \sqrt{\sum_{j=1}^p \left[\left(\frac{a_{ij}}{\sqrt{\sum_{j=1}^p a_{ij}^2}} \right) - \left(\frac{a_{hj}}{\sqrt{\sum_{j=1}^p a_{hj}^2}} \right) \right]^2} \quad (2.10)$$

Då endast positiva värden finns i definitionsmängden för punkterna i och h placeras punkterna enligt RED metoden i enhetscirkeln första kvadrant. Det maximala avståndet mellan punkterna blir därför $\sqrt{2}$.

Då både positiva och negativa värden inkluderas i definitionsmängden för punkterna i och h utnyttjas hela enhetscirkeln med RED metoden enligt figur 2.5. Det maximala avståndet antas när punkterna är placerade diagonalt utmed enhetscirkeln och är därmed värdet två.



Figur 2.5: Visualisering av relativ Euklidisk distans som utnyttjar hela enhetscirkeln.

2.10.3 Kvadratisk relativ Euklidisk distans

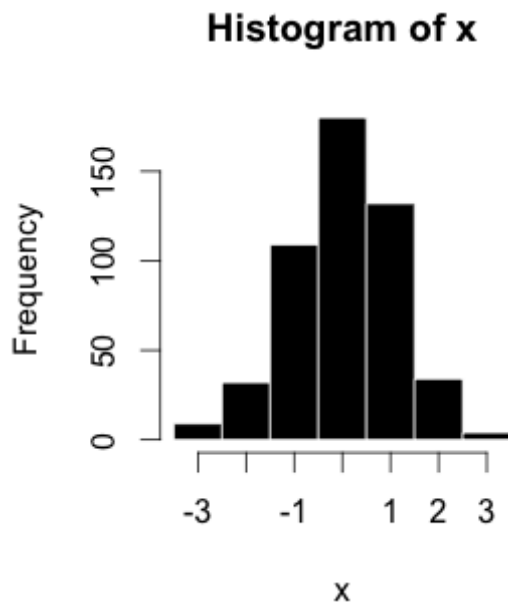
Kvadratisk relativ Euklidisk distans, SRED, är en modifikation av RED metoden. Genom att kvadrera RED uttrycket så läggs större vikt vid punkter som ligger långt ifrån varandra enligt:

$$SRED_{ih} = \sum_{j=1}^p \left[\left(\frac{a_{ij}}{\sqrt{\sum_{j=1}^p a_{ij}^2}} \right) - \left(\frac{a_{hj}}{\sqrt{\sum_{j=1}^p a_{hj}^2}} \right) \right]^2 \quad (2.11)$$

Precis som för RED metoden beror det maximala avståndet mellan dem två punkterna i och h av definitionsmängden för punkterna. SRED avståndet blir en kvadrat av det värdet RED antar.

2.11 Histogram

Ett histogram är ett stapeldiagram som beskriver antalet förekomster av en variabel inom ett visst intervall [13]. Antalet förekomster beskrivs med hjälp av stapelhöjden i diagrammet, se figur 2.6.



Figur 2.6: Bild avser ett exempel av ett histogram. Bild tagen från [13].

2.12 CSV-fil

Komma separerat värde (Comma Separated Value, CSV) är en formateringsstyp anpassad för export och import av data mellan olika mjukvaror [14]. Vanligast förekommande inom engelskspråkiga länder är att använda ett komma tecken för att separera olika kolumner av data eftersom de använder punkt istället för komma i decimaltal. I länder där det används kommatecken i decimaltal används istället semikolon för att separera kolumner av data, dock kan även andra tecken förekomma.

3. METOD

I detta kapitel presenterar projekts metodik översiktligt.

Innan utvärderingen av misfire konceptet påbörjades utfördes en litteraturstudie. Studien omfattade misfire i allmänhet, PAA samt Euklidiska distansmått. Därefter analyserades den Volvo Cars befintlig Matlabkod från Volvo för att utreda vad som var relevant för den kommande utvärderingen och vad som behövdes utvecklas.

3.1 PAA eller SAX?

Enligt [15] där både SAX metoden och PAA metoden utvärderas, fungerar PAA metoden minst lika bra som SAX- metoden dessutom har den bättre noggrannhet och är mindre komplex. Därför valdes i samråd med Volvo Cars att enbart använda PAA i detta arbete.

3.2 Val av normalisering metod och längd

Utvärdering av normaliseringslängd och metod sker med metoden NRMSE där de olika metodernas PAA representation jämförs med avseende på normaliseringslängd, glidande eller fixt normaliseringsfönster. Med glidande normaliseringsfönster innebär att normaliseringsfönstret glider ett sampel framåt för varje normalisering medans vid fixt normaliseringsfönster så görs normaliseringen för alla sampel i intervallet direkt.

3.3 Mönsterlängd bestämning

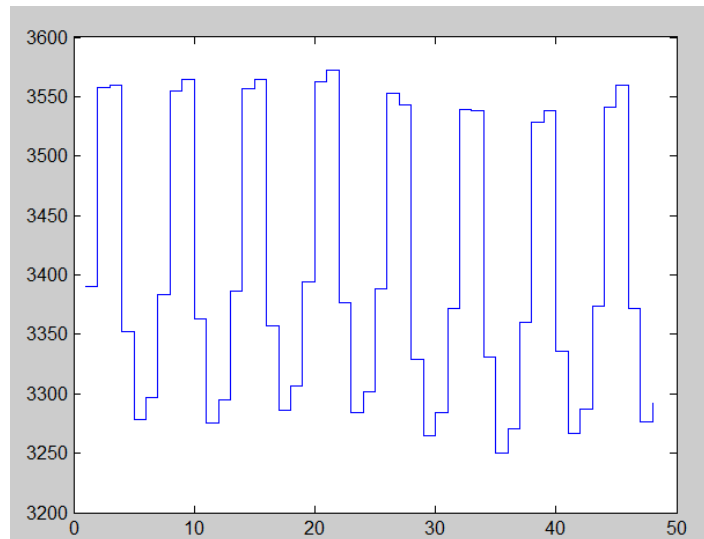
En metod har tagits fram [16] för att välja optimalt antal sampel som skall ingå i en PAA nivå enligt:

$$PAA \text{ segment storlek} = \frac{\text{Antalet datapunkter}}{\text{Antalet toppar}} \quad (3.1)$$

Om ekvation 3.1 appliceras för mönsterigenkänning på en fyrcylindrig motor med vevaxelgivare med samplingsperiod 30 grader (24 sampel) fås ekvation 3.2.

$$PAA \text{ segment storlek} = \frac{\text{Antalet datapunkter i sliding window}}{\text{Antalet toppar} \times n} \quad \text{då } 2 \leq n \leq 3 \quad (3.2)$$

I graf 3.1 visas hur normala förbränningar ser ut över flera cylindrar. I grafen kan man se att det finns en topp per cylinder alternativt sex sampel. I grafen syns åtta stycken toppar vilket då motsvarar åtta cylindrar.



Graf 3.1 . Normal förbränning. Y-axeln visar tiden mellan samplingarna i μs . X-axeln visar antal sampel där 6 st sampel motsvarar en cylinder. Varvtalet är 1500 varv/min och lasten 1,8 g/rev.

Vidare har n begränsats till $2 \leq n \leq 3$ på grund av komprimeringen av data. Då $n = 1$ komprimeras tidsserien inget vilket ger att PAA representationen är identisk med tidsserien. Då $n > 3$ blir komprimeringen av data hög och noggrannhet förloras. Detta då det inte finns något $n > 3$ där kvoten i (3.2) blir ett heltal då endast hela sampel betraktas i en tidsdiskret serie.

PAA mönsterstorleken bestäms genom att dividera antalet datapunkter i området med segment storleken enligt:

$$PAA \text{ mönsterstorlek} = \frac{\text{Datapunkter i sliding window}}{\text{Segment storlek}} \quad (3.3)$$

Genom att använda (3.2) och (3.3) fås värden i tabell 3.1. I tabellen ges sex möjliga testfall att utvärdera.

Tabell 3.1: Beskrivning av segment storlek och PAA mönsterstorlek där PAA segment storlek beräknas med ekvation 3.2 och PAA mönsterstorleken beräknas med ekvation 3.3

Antal betraktade cylindrar	1	1	2	2	4	4
Datapunkter i sliding window	6	6	12	12	24	24
Toppar i vid normal förbränning	1	1	2	2	4	4
N	2	3	2	3	2	3
PAA segmentstorlek	3	2	3	2	3	2
PAA mönsterstorlek	2	3	4	6	8	12

Fallet då tre cylindrar betraktas exkluderas av den anledningen att en fullständig tändningsserie på en fyrcylindrig motor inte går att dela jämt med tre cylindrar.

3.4 Mätdata

All mätdata denna rapport omfattar tillhandahålls utav Volvo Cars på en utav deras 4-cylindriga bensin motorer. Mätdata omfattar kontrollerade körningar i testmiljö med fixerade motor -last och –varvtal där misfire initieras enligt en förutbestämd sekvens. Detta görs genom att man avbryter insprutningen av bränsle eller avbryter tändningen i vald cylinder.

3.5 Mätsignaler och övningsdata

Följande mätsignaler används vid utvärderingen av metoden:

- T30: Mätsignal från vevaxelgivare
- IgnCut_1: Signal som indikerar tvingad avstängning av tändning i cylinder 1
- IgnCut_2: Signal som indikerar tvingad avstängning av tändning i cylinder 2
- IgnCut_3: Signal som indikerar tvingad avstängning av tändning i cylinder 3
- IgnCut_4: Signal som indikerar tvingad avstängning av tändning i cylinder 4
- InjCut_1: Signal som indikerar tvingad avstängning av insprutning i cylinder 1
- InjCut_2: Signal som indikerar tvingad avstängning av insprutning i cylinder 2
- InjCut_3: Signal som indikerar tvingad avstängning av insprutning i cylinder 3
- InjCut_4: Signal som indikerar tvingad avstängning av insprutning i cylinder 4
- Cylinder_ID: Identifierar cylinder

Övningsdatat som används i projektet kommer från en av Volvo Cars fyrcylindriga motorer med automatisk växellåda. Övningsdatat består av 88 stycken mätfiler med varierande last och varvtal. Varje mätfil består av i genomsnitt 80000 sampel och innehåller som cirka 2000 misfire.

3.6 Metodik för val av referensmönster

I detta stycke beskrivs metodiken för framtagning av referensmönster för mönsterigenkänning av misfire.

3.6.1 Definition av referensmönster

För fortsatt läsning behövs en definition av referensmönster som används i rapporten.

Definition: Ett referensmönster baseras på medelvärdet av förekomsten av en viss typ av förbränning inom en viss del eller hela last- och varvtalsområdet för en fyrcylindrig motor.

Syftet med ett eller flera referensmönster är att hitta ett mönster med god förmåga att identifiera en misfire och samtidigt bibehålla en god separation mellan misfire och normal förbränning.

3.6.2 Identifiering av misfire sekvenser

För identifikation av misfire i mätfilerna kontrolleras varje sampel mot IgnCut och InjCut signalerna med hänsyn till cylinder_ID. Om någon av IgnCut eller InjCut signalerna är aktiva samtidigt som matchande cylinder ges samplet identiteten "99" (misfire). Om IgnCut eller InjCut signalerna är aktiva men inte har en matchande cylinder ges samplet identiteten "10" (normal förbränning), samplet får även denna identitet om ingen av IgnCut eller InjCut är aktiva. Observera att värdet för identiteten kan väljas godtyckligt, enda syftet med identiteten är att lätt logiskt särskilja en misfire från en normal förbränning.

Varje PAA mönster identifieras med en lika lång sekvens av identiteter för en normalförbränning och misfire, i denna rapport kallat *mönster ID*. Exempel på ett mönster ID är vid normalförbränning med PAA mönsterstorlek på tre är '10 10 10' och vid en misfire '99 99 99'.

3.6.3 Unika mönster ID

För varje unikt mönster ID i en mätfil tas ett medelvärde för varje PAA segment som matchar det unika mönster ID. Därefter räknas antalet förekomster av varje unikt mönster ID i mätfilen. Data exporteras sedan till en textfil.

Syftet med exporten av data är att bestämma ett eller flera referensmönster samt för att verifiera Simulink modellen.

3.6.4 Behandling av data i MS excel

När valda mätfiler behandlats importerar datat från punkt 3.6.3 till Microsoft Office Excel. Data separeras med avseende på typ av förbränning; normal förbränning respektive misfire.

Ett medelvärde räknas ut för varje PAA segment beroende på typ av förbränning samt med avseende på viss del eller hela last- och varvtalsområdet. För att bedöma detta kontrolleras hur lika mönstren är med varandra och mellan dem olika typerna. Jämförelsen görs med S-RED.

Därefter beräknas ett eller flera medelvärden förr varje PAA segment beroende på tidigare jämförelse med S-RED. Medelvärdet utgör referensmönstret för specifik misfire typ och normalförbränning.

3.7 Metodik för verifiering av referensmönster

I detta stycke beskrivs metodiken för verifiering av referensmönster.

3.7.1 Histogram för visualisering av distansmått

Genom att använda antingen Euklidiskt distansmått, relativt Euklidiskt distansmått eller kvadratisk Euklidiskt distansmått räknas avståndet mellan varje PAA representation och referensmönster ut. Varje jämförelse med ett referensmönster visualiseras sedan med två histogram där den första utgör mönster ID som exakt matchar referensmönstrets mönster ID. Det andra histogrammet utgör alla andra mönster ID som inte matchar referensmönstrets mönster ID.

3.7.2 Bestämning av tröskelvärde

I ideal fallet är distansmättet i förhållande stort att en tydlig skillnad mellan matchande och ej matchande mönster kan ses histogrammen. Detta så att ett generellt tröskelvärde enkelt kan väljas för matchning av mönster.

För att utvärdera detta görs en visuell bedömning av varje histogram per mätfil och ett lämpligt tröskelvärde väljs.

3.8 Matlab program

I Matlab har tre stycken script programmeras för analys av mätfiler. Följande finns beskrivning av dem tre programmen. Scripten bygger på en importfunktion av mätfiler levererad av Volvo Cars.

3.8.1 Indata till programmet

Input parametrar till följande program är

- Normaliseringslängd
- Normaliseringsmetod
- PAA segment storlek
- PAA mönster storlek
- Exkludera PAA mönster ID med förekomster mindre än ett visst värde

3.8.2 Program 1: Export av unika PAA mönster

Programmets uppgift är att exportera alla unika PAA mönster per mätfil utifrån valda parametrar; normaliseringslängd, normaliseringsmetod, PAA sekvenslängd samt PAA mönsterlängd. Exporten sker till en CSV formaterad textfil.

Programmet börjar med att räkna antalet valda mätfiler och sedan påbörja behandling första mätfilen. Behandlingen börjar med normalisering av mätsignal T30, alla sampel identifieras som antingen misfire eller normalförbränning enligt stycke 3.6.1 och en PAA representation av mätsignal T30 görs. Därefter identifieras alla unika PAA mönster ID enligt beskrivning i

stycke 3.6.2 och resultatet exporteras till en CSV formaterad textfil. Därefter sker samma procedur med resterande mätfiler. Ett flödesschema för programmet kan ses i bilaga 1.

Den exporterade textfilen importeras in i MS excel och ett medelvärde för varje unikt PAA mönster ID bestäms. Detta medelvärde utgör referensmönster.

3.8.3 Program 2: Export av histogram

Programmets uppgift är att exportera Matlab histogram grafer baserat på jämförelse mellan referensmönster och mätfil utifrån valda parametrar; normaliseringsmetod, normaliseringslängd, PAA sekvenslängd samt PAA mönsterlängd. Framtagningen av referensmönster beskrivs i stycke 3.8.2.

Programmet börjar med att räkna antalet valda mätfiler och sedan påbörja behandling av första mätfilen. Behandlingen börjar med normalisering av mätsignal T30 och därefter en PAA representation av mätsignal T30. Därefter jämförs referensmönster med varje PAA mönster och ett distansmått fås. Därefter skapas ett histogram per referensmönster som visar fördelningen av distansmått jämfört med referensmönstret. Histogrammen sparas därefter och bearbetning av nästa mätfil påbörjas. Ett flödesschema för programmet kan ses i bilaga 1.

3.8.4 Program 3: Verifiering av algoritm

Programmets uppgift är att räkna antalet förekomster utav PAA mönster ID som matchar referensmönstret. Detta så att resultatet kan användas för att verifiera Simulink modellen. Programmet skall även exportera .mat fil med data för import i Simulink modellen.

Programmet börjar med att räkna antalet valda mätfiler och sedan påbörja behandling av första mätfilen. Behandlingen börjar med normalisering av mätsignal T30 och därefter en PAA representation av mätsignal T30. Därefter räknas antalet mönster som matchar referensmönsterna och resultatet exporteras till en CSV-fil.

Även mätsignalerna Cylinder_ID och T30 exporteras till en .mat fil för import i Simulink-modell. Ett flödesschema för programmet kan ses i bilaga 1.

3.9 Simulink-modell

En Simulink modell sätts upp som använder de exporterade mätsignalerna T30 samt Cylinder_ID från Matlab programmet Final_ver. Simulink modellen skall utföra samma beräkningar som Matlab programmen utfört med avseende på normalisering, PAA representation samt bestämning av distansmått. Simulink modellen kan ses i bilaga 2.

3.9.1 Nyckelblock

I detta stycke kommer en beskrivning av Simulink block som används i projektet.

Buffer

Blocket Buffer tar och sparar n samplade värden och skickar vidare datat som en array av storlek m, omvänd ordning kan också användas. Enligt Mathworks arbetar blocket med frame based processing [17]. Blocket arbetar då med att manipulera bildrutestorleken genom att antingen öka eller minska frame rate. Detta kan förenklas till att säga att bildrutestorleken bestämmer hur många sampel som skall skickas per simulerings steg.

Matlab Function

I ett Matlab function block kan Matlabkod användas för bearbetning av signalerna till blocket [18]. Detta är fördelaktigt då komplicerade beräkningar skall utföras eller lösningar som i annat fall kräver stora och komplicerade blockscheman.

From workspace

Importerar data från variabler i Simulink modellens workspace [19]. Krävs att data är formaterad med i matrisform med tidsstämpel eller sampelnummer i första kolumnen. I andra kolumnen finns en vektor med innehållande data för respektive tidsstämpel.

Compare to constant

Compare to constant blocket jämför en signal med en konstant [20]. Om signalen är lika med konstanten sätts utsignalen till ett annars är utsignalen noll.

Sample and hold

Genom att sätta signalen till blockets trigger port hålls insignalen värde på blockets ut port [21].

S-R Flip-Flop

Blocket S-R Flip-Flop är en Set-Reset vippa (S-R vippa) där reset porten är den dominerande insignalen [22]. Då reset porten ej är aktiverad hålls sätts blockets utsignal hög eller till användarens fördefinierade värde. När reset porten är aktiverad nollställs blockets utsignal.

Delay

Delay blocket ger användaren möjlighet att fördröja signalen med ett eller valfritt antal sampel [23]. Antalet sampel fördröjning kan antingen förbestämmas av användaren eller bestämmas av en andra insignal till blocket.

Mean

Blocket mean räknar ut medelvärdet över specificerad dimension av insignalen [24]. Dimensionen kan antingen räknas i rader eller kolumner eller av användaren specificerad dimension. Förenklat kan detta beskrivas som att medelvärdes beräkning sker i en $1 \times n$ matris eller $n \times 1$ matris eller matris definierad av användaren.

Counter

Blocket Counter används för att som en räknare vid förändring av insignalen till blocket [25]. Blocket räknar upp eller ner på bland annat positiv flank, negativ flank eller sampel som inte har värdet noll. Utsignalen är det räknade värdet.

Repeating Sequence Stair

Blocket repeterar en sekvens av värden definierade av användaren [26].

Switch

Blocket växlar mellan två insignaler beroende på vad styrsignalen till blocket [27]. Vilken insignal som passerar igenom blocket väljs beroende på om styrsignalen uppfyller dem villkor som sätts av användaren.

4. LÖSNINGSALTERNATIV

4.1 Val av testfall

I tabell 3.1 visas testfallen som var möjliga att utvärdera, denna kompletteras med vilka fall som är möjliga att detektera med avseende på PAA- segment storlek och mönsterstorlek.

Tabell 4.1: Komplimentering av tabell 3.1 med möjliga misfire kombinationer som kan detekteras med testfallet.

	Testfall 1	Testfall 2	Testfall 3	Testfall 4	Testfall 5	Testfall 6
Datapunkter i sliding window	6	6	12	12	24	24
Toppar i vid normal förbränning	1	1	2	2	4	4
N	2	3	2	3	2	3
PAA segment storlek	3	2	3	2	3	2
PAA mönsterstorlek	2	3	4	6	8	12
Möjligt att detektera	Single misfire Normal förbränning		Single misfire Sequel misfire Normal förbränning		Single misfire Sequel misfire Pair misfire First/last misfire Normal förbränning	

Valet av testfall väljs till testfall två samt testfall fem valts att utvärdera. Detta av anledningen att alla testfall inte kan utvärderas inom tidsbegränsningen för projektet.

Testfall 2 väljs av den anledningen att observera förbränningen över en cylinder, sex datapunkter gör det möjligt urskilja både normala förbränningar och misfire. Dock går det inte att urskilja vilken misfire typ det är bara genom att betrakta PAA mönstret över en cylinder.

Testfall 6 väljs för att det är möjligt att detektera alla typer av misfire kombinationer i samma mönster. Testfall 6 väljs även av den anledningen att PAA segmentstorleken på två ger en lägre komprimering av data vilket antas vara en fördel jämfört med testfall 6.

5. RESULTAT

5.1 Utvärdering av normaliseringsmetod m.h.a NRMSE

För att utvärdera lämplig normaliseringslängd och metod används samma mätsignal till alla testfall. Som referens används för varje normaliseringsmetod ett normaliseringsfönster på hela tidsserien, i detta fall 85736 sampel. Mätningen som används är för last 1,8 g/varv och varvtal -1500 varv/min. Resultatet visas både som NRMSE normaliserad och NRMSE PAA representation där sekvenslängden är tre.

Resultatet beskriver hur mycket den normaliserade tidsserien skiljer sig från referens tidsserien i procent. Ett lägre NRMSE värde har mer likheter med referens tidsserien jämfört med ett högre NRMSE värde.

Tabell 5.1: Resultatet av NRMSE för olika typer av normalisering metoder av data från mätfil med last 1,8g/varv och varvtal 1500 varv/min. PAA mönster storleken som används är satt till tre.

Normaliseringstyp	Fönstertyp	NRMSE Normaliserad			NRMSE PAA representation		
		24 sampel	12 sampel	6 sampel	24 sampel	12sampil	6 sampel
Standard score typ 1	Fix	164,82%	198,88%	246,82%	148,13%	190,39%	254,31%
Standard score typ 1	Glidande	185,24%	185,24%	96,54%	162,40%	214,71%	272,13%
Standard score typ 2	Fix	1,05%	1,22%	1,58%	1,20%	1,40%	1,54%
Standard score typ 2	Glidande	1,39%	1,44%	1,57%	1,45%	1,63%	1,78%
Standard score typ 3	Fix	0,03%	0,02%	0,01%	0,03%	0,02%	0,01%
Standard score typ 3	Glidande	0,02%	0,01%	0,01%	0,02%	0,01%	0,01%
Feature scaling typ 1	Fix	45,74%	45,92%	43,89%	67,21%	64,67%	41,38%
Feature scaling typ 1	Glidande	45,06%	47,03%	50,90%	40,15%	46,35%	49,65%
Feature scaling typ 2	Fix	22,87%	22,96%	21,94%	33,60%	32,33%	20,69%
Feature scaling typ 2	Glidande	22,53%	23,51%	25,45%	20,08%	23,18%	24,83%

Tabell 5.2: Beskrivningar av normaliseringstyper.

Normaliseringstyp	Beskrivning
Standard score typ 1	Variant på Standard score där man istället dividerar med variansen istället för standard avvikelsen
Standard score typ 2	Standard avvikelsen räknas om för varje normaliseringsfönster
Standard score typ 3	Standard avvikelsen sätts till 1
Feature scaling typ 1	Feature scaling i intervall 0-1
Feature scaling typ 2	Feature scaling i intervall -1 -1

Enligt tabell 5.1 visas att normalisering med standard score typ två samt typ tre särskiljer sig i egenskap av hur lika dem är referensen. Till detta ses även att standard score typ ett särskiljer sig som den i särklass sämsta normaliseringsmetoden i detta test.

Ytterligare ses att skillnaden mellan att ha ett fixt normaliseringsfönster jämfört med ett glidande normaliseringsfönster är överlag endast några procentenheter. Därför väljs ett fixt normaliseringsfönster då implementeringen i Simulink ger en mindre komplex modell.

5.2 Utvärdering av testfall

5.2.1 Utvärdering av testfall 6

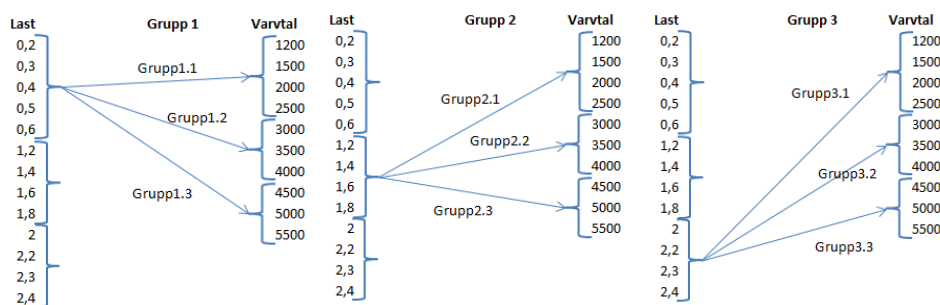
Testfall sex innebär ett glidande fönster på 24 sampel d.v.s. fyra cylindrar med en PAA mönsterstorlek på 12 samt en PAA segmentstorlek på två. Detta ger med ett fönster på 24 sampel nio unika mönster, tio om normal förbränning skall räknas med, möjliga att detektera enligt tabell 5.3.

Tabell 5.3 beskriver misfire kombinationer som kan detekteras i testfall 6

Misfire typ	cylinder 1	cylinder 2	cylinder 3	cylinder 4
Single	X			
Single		x		
Single			X	
Single				x
Consecutive	X	x		
Consecutive		x	X	
Consecutive			X	x
Pair	X		X	
Pair		x		x
First/last	X			x

Genom att använda dem 88 tillgängliga mätfilerna i övningsdata som används i projektet och beräkna PAA mönster för varje unikt mönster per mätfil ger detta sammanlagt 880-900 unika mönster.

För att förenkla och skapa färre mönster delades övningsdatat upp i nio stycken grupper enligt figur 5.1. För varje grupp togs medelvärdet av alla mönster för att få fram nio stycken unika mönster. Detta gav sammanlagt 88 stycken mönster.



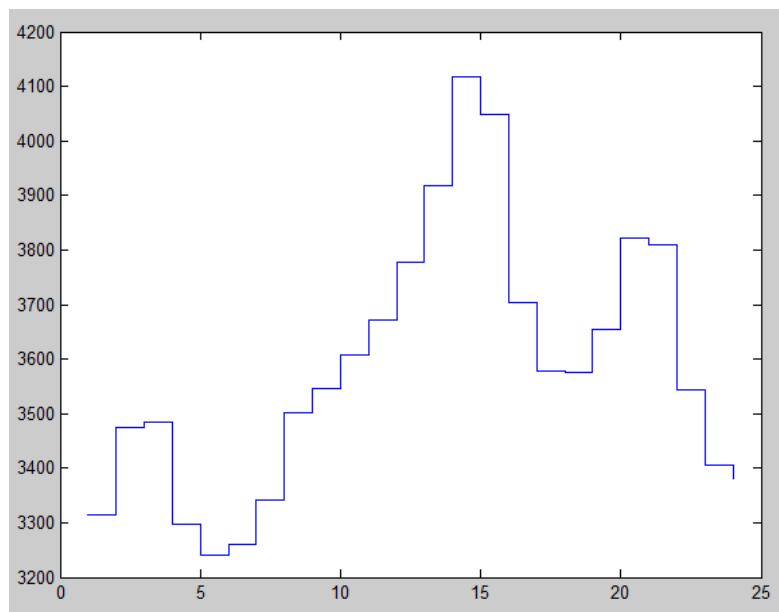
Figur 5.1: Visar hur last- och varvtalsområden delas upp i nio stycken grupper.

I detta skede valdes testfall fem bort på grund av komplexiteten som medförs av så många mönster. Detta skulle innebära att beroende på lösning att minst 9 stycken distansmått beräkningar skall ske parallellt. Detta ökar komplexiteten markant för Simulink modellen.

5.2.2 Utvärdering av testfall 2

Testfall två innebär ett glidande fönster på 6 sampel d.v.s. en cylinder. PAA segmentstorleken i testfallet är två samt PAA mönsterstorleken blir tre. Då endast en cylinder betraktas kan detta testfall till skillnad från testfall 6 endast detektera Single misfire. I graf 5.1 visas hur en Single misfire ser ut.

Graf 5.1 Single misfire. Y-axeln visar tiden mellan samplingarna i μs . X-axeln visar antal sampel där 6 st. sampel motsvarar en cylinder. Varvtalet är 1500 varv/min och lasten 1,8 g/rev.



Normaliseringsmetoderna utvärderas var för sig med respektive referensmönster som beräknats för varje metod. Därefter görs en visuell bedömning av histogrammen för att se om ett tydligt S-RED tröskelvärde kan bestämmas, en tydlig och god separation krävs för att testfallet skall bedömas som möjligt att detektera misfire i. I tabell 5.4 kan resultatet av utvärderingen ses för respektive normaliseringsmetod. I bilaga 5 finns detaljerat beskrivet vilka testfall som är möjliga att detektera misfire i respektive normaliseringsmetod.

Tabell 5.4: Visar antalet mätfiler det är möjligt att detektera misfire i per normaliseringsmetod. Observera att bedömningen är gjord visuellt genom att betrakta histogrammen för alla mätfiler i testdatat.

Möjlig detektion mätfiler	
Feature Scaling typ1	6
Feature Scaling typ2	7
Standard Score typ 1	30
Standard Score typ 2	10
Standard Score typ 3	6

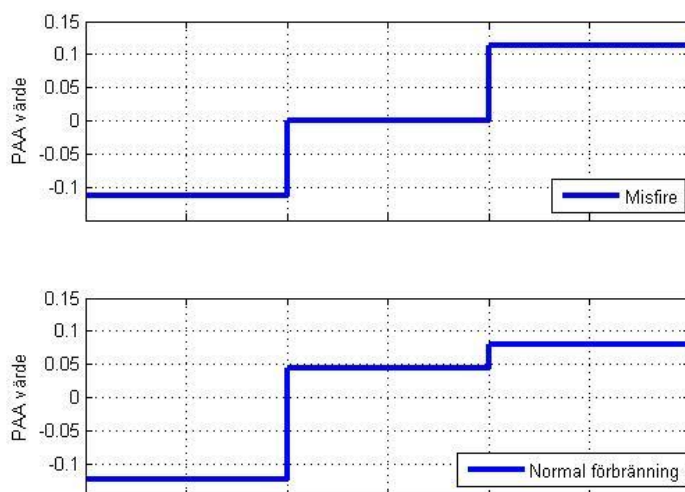
I tabell 5.4 kan ses att normalisering med avseende på variansen, normaliseringsmetod Standard score typ ett är den metod som möjligen skulle kunna detektera flest testfall. Detta resultat är i rak motsats till resultatet av NRMSE där Standard score typ ett är i särklass den metoden som förvanskar mätsignalen allra mest.

Dem referensmönster som beräknats med normalisering Standard score typ ett kan ses i tabell 5.5 och graf 5.2. Framtagningen av referensmönster kan ses i bilaga 7 samt bilaga 8.

Tabell 5.5: tabellen visar värdena för varje PAA segment som beräknats fram. Detta PAA mönster används sedan som referensmönster för detektering av singel misfire. Hur framtagningen har skett kan ses i bilaga 7 respektive 8.

Typ	PAA segment 1	PAA segment 2	PAA segment 3
Single misfire	-0,11371	0,000748	0,11296
Normal förbränning	-0,123366	0,0437963	0,07957

Graf 5.2: Visar grafiskt hur referensmönsterna för sigel misfire och normalförbränning ser ut. Värdena i grafen är samma som används i tabell 5.5.



Genom att visuellt bedöma alla histogram från mätfilerna där datat jämförs med referensmönstret valdes SRED tröskelvärde till två.

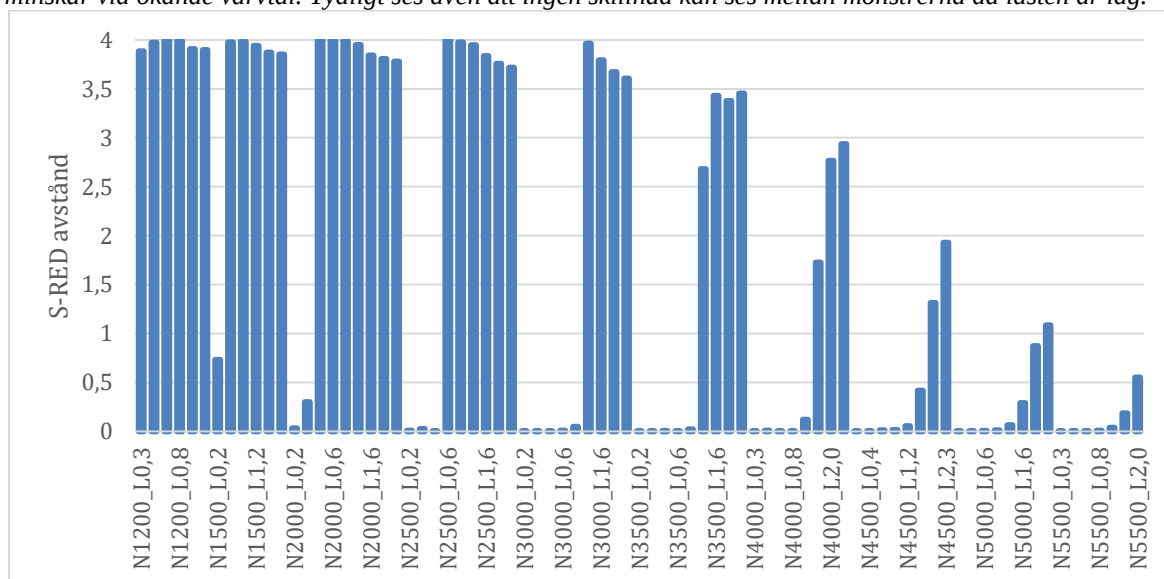
Med valt referensmönster, normaliseringsmetoden standard score typ 1 samt tröskelvärde blir det då möjligt att detektera misfire i 30 av 79 mätfiler beskrivet i tabell 5.6. Tabellen beskriver vilka mätfiler det är möjligt att detektera misfire i detta genom en visuell bedömning av histogrammen. Anledningen till varför datat separeras på detta sätt är p.g.a. om inte en god separation mellan misfire och normal förbränning visuellt kan ses i histogrammet kommer inte heller en god separation att uppnås med metoden. I bilaga 5 finns motsvarande tabeller för resterande normaliseringsmetoder.

Tabell 5.6: tabellen visar vilka mätfiler som är möjliga att detektera misfire i efter visuell bedömning av histogrammen för respektive mätfil. Villkoret är att en tydligt separation mellan misfire och normalförbränning skall synas tydligt i histogrammet för att det skall bedömas som möjligt att detektera misfire. Förkortningen "N" betyder att det ej är möjligt att visuellt separera misfire från normalförbränningar i histogrammet. Förkortningen "J" betyder att det visuellt är möjligt att separera misfire från normalförbränningar i histogrammet.

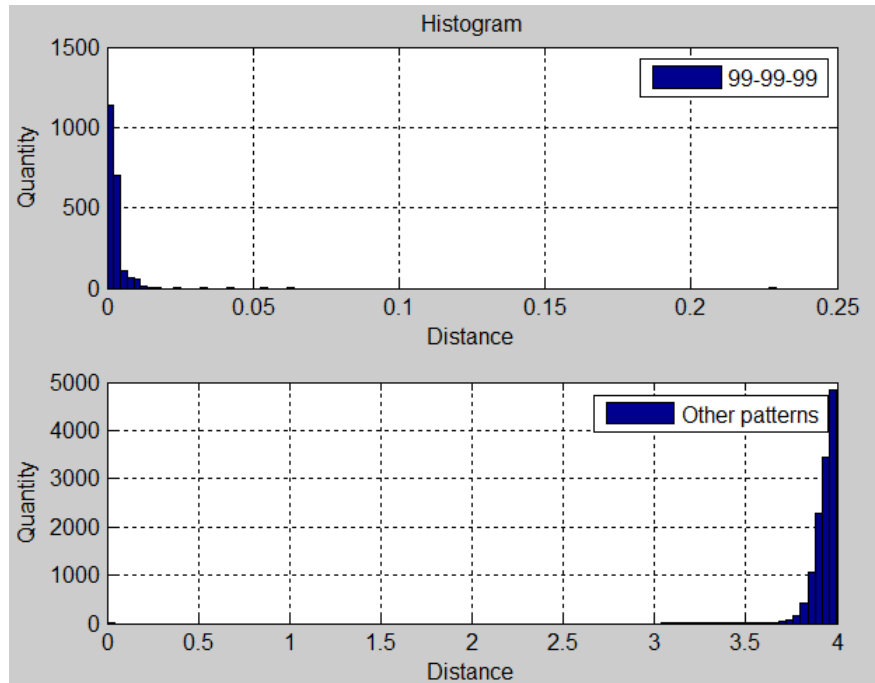
Varvtal [rpm]	1200	1500	2000	2500	3000	3500	4000	4500	5000	5500
Last [g/rev]										
0,2	-	N	N	N	N	N	-	-	-	-
0,3	J	-	N	N	N	N	N	N	N	N
0,4	J	N	J	N	N	N	N	N	N	N
0,6	J	-	J	J	N	N	-	N	N	N
0,8	J	J	J	J	N	N	N	N	N	N
1,2	J	J	J	J	J	N	N	N	N	N
1,4	J	-	-	-	-	-	-	-	-	-
1,6	-	J	J	J	J	J	N	N	N	N
1,8	-	J	-	-	-	-	-	-	-	-
2	-	-	J	J	J	J	N	N	N	N
2,2	-	-	-	-	-	-	-	-	N	-
2,3	-	-	J	J	-	J	N	N	-	-
2,4	-	-	-	-	J	-	-	-	-	-

Anledningen till varför detektionen endast funkar i fallen i tabell 5.6 är på grund av den ökande likheten mellan en misfire och normal förbränning med ökande last och varvtal. I graf 5.3 visas SRED avståndet mellan valt misfire mönster och medelvärdet för normalförbränningar.

Graf 5.3: Grafen visar distansmättet mellan referensmönstret för misfire och mönstret för en normal förbränning i varje mätfil. Värdeområdet för S-RED är 0 till 4 där värdet 4 indikerar att monstren har högsta möjliga olikhet och där 0 indikerar att mönstren är nära identiska. Observera att referensmönstret är ett medelvärde för alla misfire från alla mätfiler. Här kan tydligt ses att S-RED avståndet mellan monstren minskar vid ökande varvtal. Tydligt ses även att ingen skillnad kan ses mellan monstren då lasten är låg.

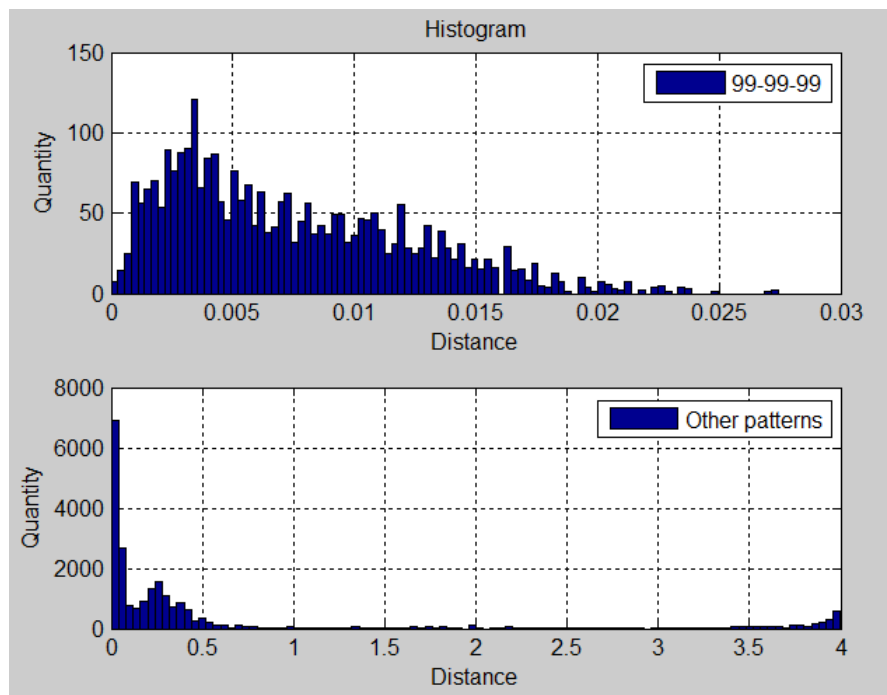


I graf 5.3 beskriver ett ökande SRED avstånd en ökande olikhet där noll indikerar en identisk likhet. Ett mönster kan betraktas i grafen där dem lägsta lasterna för varje varvtal är lika referensmönstret för en misfire. I graf 5.4 ses ett exempel på hur en detektion med god separation ser ut. Exemplet är hämtat från mätfil med varvtal 1500 varv/min samt last 1,2g/rev.



Graf 5.4: Histogram med distans. Varvtal 1500[rpm] samt Last 1.2 [g/rev]. Observera skillnaden i skala på x och y-axlarna. X axeln avser S-RED avståndet mellan datat och referensmönstret för en misfire. Y- axeln avser antalet förekomster. Övre histogrammet avser jämförelser mellan misfire referensmönster och kända misfire i mätfilen. Undre histogrammet avser jämförelse mellan misfire referensmönster och normala förbränningar.

Ytterligare exempel visas i graf 5.5 där motsatt fall gäller med data hämtat från mätfil med varvtal 5500 varv/minut samt last 0,8g/rev. I exemplet ses att det ej går att uppnå god separation mellan misfire och normalförbränning i testfallet.



Graf 5.5 Histogram med distans. Varvtal 5500[rpm] samt Last 0.8 [g/rev]. Observera skillnaden i skala på x och y-axlarna. X axeln avser S-RED avståndet mellan datat och referensmönstret för en misfire. Y-axeln avser antalet förekomster. Övre histogrammet avser jämförelser mellan misfire referensmönster och kända misfire i mätfilen. Undre histogrammet avser jämförelse mellan misfire referensmönster och normala förbränningar.

I graf 5.5 kan man tydligt se att merparten av dem normala förbränningarna liknar en misfire. Liknande utseende på histogrammen fås i resterande testfall i tabell 5.6 där detekteringen inte fungerar. Gemensamt för dessa testfall är att distansmåttet blir så litet att det blir omöjligt att särskilja misfire från normala förbränningar.

Detta pekar på resultatet av att använda flera referensmönster även hade gett en allt för dålig noggrannhet, detta då det med denna detekteringsalgoritm inte går att urskilja skillnaden mellan en misfire och en normal förbränning. Det ska tilläggas att utvärderingen för denna metod har gjorts med testdata inom vissa specifika last och varvtals områden. Testdata från en dynamisk körning med accelerationer samt retardationer har inte verifierats inom samma last och varvtalsområden och skulle därför kunna skilja sig mot resultatet ovan.

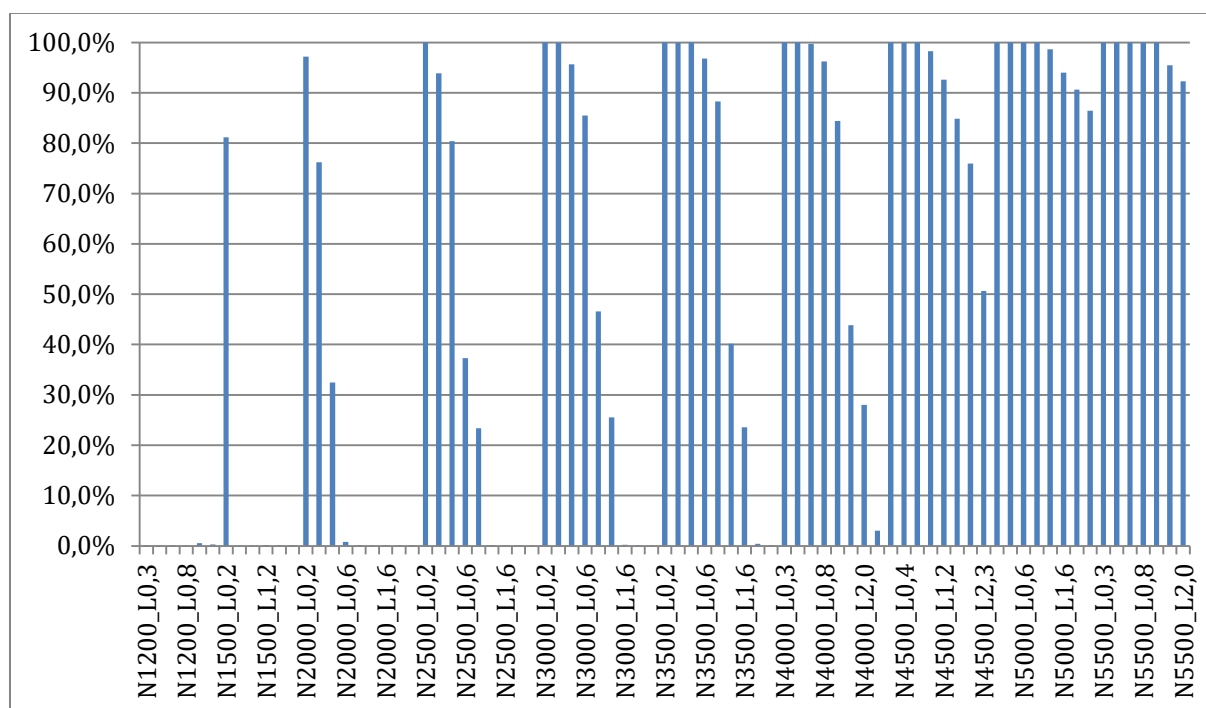
I tabell 5.7 visas noggrannheten i detekteringen av misfire. Tabellen visar medelvärdet i noggrannhet för alla mätfiler, fullständiga resultat per mätfil finns i bilaga 6. Korrekta detektioner misfire avser fall då det i datat finns en misfire och en misfire detekteras. Inkorrekta detektioner misfire avser fall då det i datat finns en misfire men inte detekteras. Falsklarm avser fall då det i datat detekteras en misfire trots att det inte finns någon. Ytterligare kan ses av tabellen att detektionsalgoritmen misslyckas att skilja på en misfire och en normal förbränning i ca 55 % utav fallen. Detta på grund av likheten mellan en misfire och en normal förbränning.

Tabell 5.7: Tabellen avser sannolikheten att för korrekt kategorisering av förbränningar. Värdena är ett medelvärde för alla mätfiler. I bilaga 6 kan värdena för specifika mätfiler ses.

	Medel
Korrekta detektioner misfire	99,99 %
Inkorrekta detektioner misfire	0,01 %
Falsklarm	54,99 %

I graf 5.6 visas falsklarm för olika mätfiler. Grafen visar andel falsklarm och tydligt kan ses att detektionen funkar mindre bra vid ökat varvtal. Ytterligare kan även ses att detektionen inte heller fungerar vid låga laster oavsett varvtal.

Graf 5.6: Grafen visar andel feldetektioner av normala förbränningar i procent per mätfil. Falsklarm avser fall då det i datat detekteras en misfire trots att det inte finns någon.



5.3 Implementering i Mathworks Simulink

En Simulink modell för att räkna ut totala antal single misfire har också tagits fram, Simulinkmodellen visas i bilaga 4. Modellen motsvarar samma beräkningar som Matlab koden använder sig utav. Modellen är tillskillnad från Matlab koden gjord för att användas i realtid. Insignalerna till blocket används från exporten beskriven i stycke 3.8.4 med hjälp av Simulink blocket From workspace.

Då metoden detekterar misfire i en cylinder åt gången måste även första samplet som bearbetas vara första samplet för en cylinder. För att göra detta används insignalen cylinder_ID_simulink som innehåller värden 0 till 23 som beskriver vilken cylinder samplet tillhör. Därmed är första samplet i cylindrarna 0,6,12 respektive 18 i tändningsföljden för cylindrarna 1,3,4,2. Genom att använda blocket Compare to constant sätts en S-R vippa då första samplet för någon utav cylindrarna kommer. S-R vippan sätter då i sin tur en switch som låter mätsignalen T30_simulink passera. Samtidigt har antalet sampel som gått sedan starten av simuleringen räknat tills S-R vippan sätts. Orsaken till detta är att en variabel fördröjning måste finnas för att buffer blocket Normalization_size skall fyllas med rätt sampel. Logiken för den variabla fördröjningen beskrivs i bilaga 4 som finns i Matlab Function blocket Delay synchronisation.

Den variabla fördröjningen och switchen låter en buffer fyllas med sex sampel avseende sampel från en hel cylinder, datan blir en 6*1 matris. Därefter normaliseras matrisen med metoden Standard score typ 1 i ett Matlab Function block. Koden för normaliseringen syns i bilaga 4

Den normaliserade T30_simulink signalen fylls därefter i en ny buffer, PAA_segment_size. Denna buffers uppgift är att omvandla 6*1 matrisen till tre stycken 2*1 matriser som skickas ut efter varandra. Detta avser storleken på ett PAA segment. Ett medelvärde tas för varje PAA segment och buffras sedan upp till en 3*1 matris i buffer PAA_pattern_length, detta avser PAA mönstret. Innan buffern måste varje PAA segment passera fördröjas med ett på grund av att buffer PAA_pattern_length skall fyllas med rätt värden.

Därefter utförs distansmått beräkningen med SRED metoden där PAA mönstret jämförs med ett referensmönster. Beräkningarna sker i Matlab Function blocket MATLAB S-RED. Om distansmålet är mindre än två har en misfire detekterats och skall registreras. För att göra detta används ett Repeating Sequence Stair block som repeterar sekvensen [0,1,2,3,4,5]. Sekvensen passerar igenom ett compare to constant block som ger en hög utsignal så länge insignalen är större än 0. Detta ger att genom att använda en S-R vippa och leda utsignalen från compare to constant blocket till reset ingången på vippan fås att vippan endast ger en hög utsignal var sjätte sampel. Det slutgiltiga och sista steget räknar antalet gånger S-R vippan sätts och därmed räknas antalet misfire som skett i simuleringen.

Motiveringen till varför beräkningarna sker i Matlab Function blocken är för att med koden från Matlab filerna kan återanvändas med ytterst liten modifikation på koden. För det andra medför lösningen att normaliseringsmetod och distansmått beräkning kan ändras lätt utan större justeringar av Simulink modellen.

Modellen kan även användas vid detektering av misfire på två eller fyra cylindrar utan större justeringar av modellen. Modellen kan även tillåta justering av längden på PAA mönstret som skall utnyttjas. I modellen som är uppsatt används en PAA mönsterlängd på tre.

Motiveringen till varför Simulink blocket Buffer används i modellen är på grund av enkelheten som medför i implementation av konceptet i Simulink. Genom att göra beräkningar på arrayer av data behövs inte hänsyn tas till antalet sampel som passerar igenom modellen. Utan beräkningarna sker på en array av data i taget och därmed alltid för sex sampel som motsvarar en förbränning i en cylinder. Komplexiteten i kod och modell kan därför hållas låg.

Modellen är mycket noggrann med en detektionsgrad på i genomsnitt 99,62 % jämfört med antalet detekterade misfire av Matlab koden. Enda fall som sticker ut är mätfil med varvtal 2500 varv/min och last 2,0 g/rev där detektionsgraden endast är 70,6 % utav dem misfire som Matlab koden detekterar. Observera att detta är en jämförelse om hur exakt Simulink modellen är i förhållande till tidigare beräkningar och detektion av misfire. Inte detektionsgraden av det faktiska antalet misfire i mätfilerna, för detta hänvisas läsaren till tabell 5.7.

6. Diskussion och slutsats

Grundtanken var att utvärdera om denna mönsterigenkänningsmetod är applicerbar på en fyrcylindrig motor samt se hur robust metoden är. Resultaten visade att denna metod inte fungerade bra på hela varvtals- och last spektret utan fungerade endast inom vissa intervall med en relativt liten felmarginal. Observera att testerna bara är utförda på statiska mätfiler från en motormodell. Det är inte säkert att samma resultat skulle kunna replikeras i en dynamisk körning i labbmiljö eller från verklig körning av fordonet.

Utvärderingen började först med att försöka detektera alla typer av misfire som nämnts i rapporten men förkastades på grund av komplexiteten som det medförde. Därför fortsatte utvärderingen med fokus på en cylinder istället och med fokus på single misfire och se om metoden fungerade bättre där. Man kan se i Simulink modellen vilken cylinder som är aktiv samt om misfire räknaren ökar. Önskvärt hade varit att spara ner dessa två värden i exempelvis ett textdokument för att se hur många misfire som uppstod på respektive cylinder efter en körning. Det kan ges som ett förslag till fortsatt arbete.

När resultaten inte visade sig fungera bra över fyra cylindrar och koncentrationen lades på att hitta enskilda single misfire så försvann ett av huvudmålen med att identifiera olika typer av misfire. För att kunna identifiera olika typer behövs ytterligare logik i Simulink modellen. Man skulle exempelvis kunna spara en sekvens innehållande fyra värden från signalen som aktiverar räknaren och jämföra dem mot olika mönster och på så sätt kunna avgöra vilken typ av misfire som inträffar. Detta skulle också kunna vara något man kan arbeta vidare på om resultaten anses vara tillräckligt goda för fortsatt utvärdering.

Normaliseringen gjordes tills en början genom att räkna ut en ny standardavvikelse för varje glidande fönster. Detta gjorde dock att signalen efter normaliseringen inte såg ut som en skalning av den ursprungliga signalen utan skiljde sig nämnvärt. Senare utfördes även tester med standardavvikelse 1 istället. Signalen såg då mycket mer lik ut insignalen. Skillnaderna på standardavvikelsen visade sig inte ha någon större påverkan på detektionen av misfire. Förvånande är att den normaliseringsmetod som visat sig fungera bäst vid detektion av misfire i konceptet är den metod som förvrider mätsignalen allra mest. Normaliseringsmetoden är en variant av Standard score där man istället för standard avvikelse använder variansen.

Valet av att använda kvadratisk relativ Euklidisk distans (SRED) motiveras av att denna metod skalar in avståndet i ett intervall istället för att endast ta avståndet mellan två punkter. Detta medför att avståndet alltid hamnar inom samma intervall tillskillnad från med att använda vanlig Euklidisk distans. Vanlig Euklidisk distans tar endast avståndet mellan två punkter i ett Euklidiskt rum, vilket medför att värdemängden blir alla positiva tal. Tillskillnad från SRED där värdemängden är 0-4. Detta gör det mycket enklare att bestämma ett gränsvärde för bedömning av misfire.

Resultatet visade att det skulle vara möjligt att detektera misfire i upp till 30 av 79 fall med detektion av förbränning i en cylinder. Skulle resultaten visat sig vara bättre med detektion i flera cylindrar är de inte alls säkert att de skulle fungera felfritt i verkligheten. En dynamisk körning skulle skilja sig mycket mot de testdata inom fasta intervall i last och varvtal som använts i denna utvärdering. I avgränsningar nämndes också att adaption av svänghjul utesluts vilket hade kunnat påverka resultaten vid en verklig körning. En annan faktor som kan påverka resultaten är vägens

ojämnheter. Detta eftersom mätsignalen som används är och vid ojämnheter kan motorn och vevaxel röra på sig och påverka tiden de tar för vevaxel att röra sig 30 grader.

En annan viktig aspekt som hade kunnat göra det svårt vid verklig körning är att det inte bara är helt uteblivna förbränningar som uppstår då utan även ofullständiga förbränningar, även kallat partiella förbränningar. Partiell förbränning hade då gjort att tidstappet på vevaxelsignalen inte hade blivit lika tydlig och hade försvårat misfire detektionen avsevärt.

Det finns viss potential för detektionsalgoritmen då det är önskvärt att köra bilen på låga varvtal för att minska bränsleförbrukningen. Enligt tabell 5.7 så fungerade metoden relativt bra på låga varvtal.

Skulle man i framtiden göra motorer med färre cylindrar som arbetar inom fasta varvtal och laster för att exempelvis fungera som generator till ett batteri i en hybridbil skulle de finnas betydligt högre potential för konceptet. En annan möjlighet för konceptet är implementering i elverk drivna av fossila bränslen. Exempel på detta är nödgeneratorer i samhällsviktiga infrastrukturer såsom sjukhus eller som portabla generatorer för konsumentmarknaden eller för större portabla generatorer för industribruk.

7. Rekommendation för framtida arbete

I samråd med Volvo Cars togs några punkter fram för fortsatt arbete för utvärdering av detektionsalgoritmen.

- Ta hänsyn till inverkan av misfire på efterkommande normala förbränningar.
- Testa annan tillgänglig mätsignal istället för tidsbaserad signal från vevaxelgivaren.
- Utvärdera med fokus på mönster för normalförbränning istället för fokus på misfire. Fokuset hamnar då på att identifiera mönster som avviker från det normala.
- Jämför med förbränning före och efter istället för att använda ett referensmönster. Ett alternativ kan vara att fylla ett FIFO-register med normalförbränningar och jämföra aktuell förbränning med.
- Eventuell viktning med hänsyn till last och varvtal.
- Plotta ut alla förbränningar per typ av misfire (enligt tabell 2.1) och normalförbränning ovanpå varandra i samma plot per mätfil för att jämföra skillnader.
- Utvärdera andra distansmått metoder, exempelvis klustring av data.

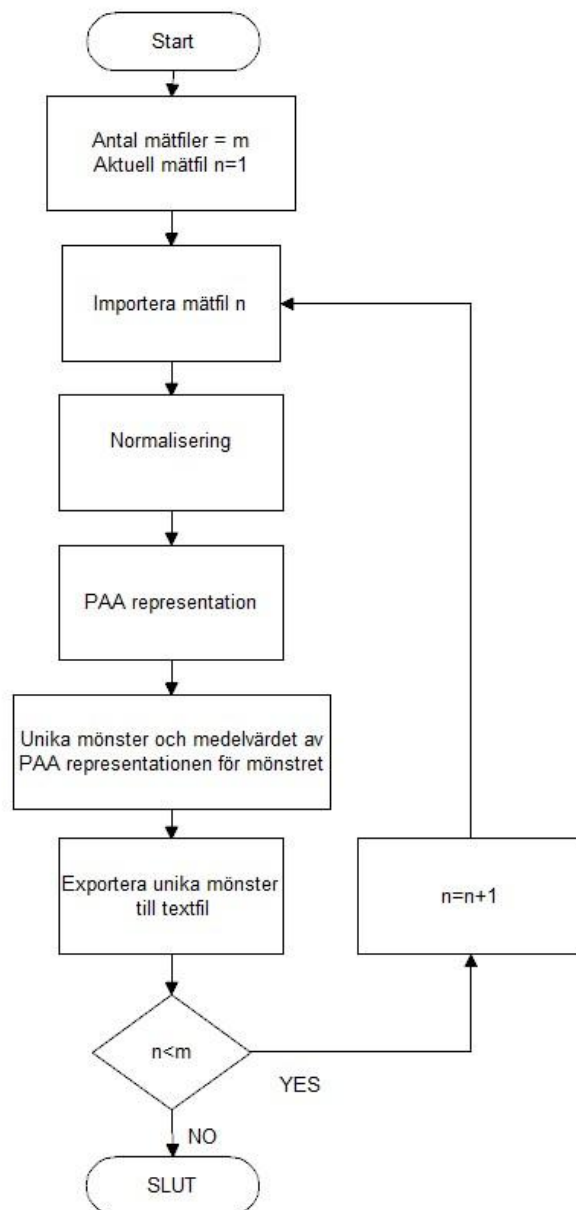
8. Referenser

- [1] California Environmental Protection Agency. (2016, May 18). On-Board Diagnostics (OBD) Program [Online]. Available:
<http://arbis.arb.ca.gov/msprog/obdprog/obdprog.htm#background>
- [2] D.Jung, L.Eriksson, E.Frisk, M Krysander, "Development of misfire detection algorithm using quantitative FDI performance analysis", *Control Engineering Practice*, vol. 34, pp.49-60, Nov 2014.
- [3] M.Matysiak. (2016, May 5). Proseminar - Data Stream Mining –Basic methods and techniques [Online]. Available:
http://dme.rwth-aachen.de/en/system/files/file_upload/course/12/elementary-data-mining-techniques/proseminar.pdf
- [4] Wikipedia. (2016, May 13). Standard score. [Online]. Available:
https://en.wikipedia.org/w/index.php?title=Standard_score&oldid=728827940
- [5] Wikipedia. (2016, May 13). Standard deviation. [Online]. Available:
https://en.wikipedia.org/w/index.php?title=Standard_deviation&oldid=736640403
- [6] Wikipedia. (2016, May 13). Feature scaling. [Online]. Available:
https://en.wikipedia.org/w/index.php?title=Feature_scaling&oldid=735723423
- [7] Wikipedia. (2016, May 20). Root-mean-square deviation. [Online]. Available:
https://en.wikipedia.org/w/index.php?title=Root-mean-square_deviation&oldid=731675441
- [8] E.Keogh, J.Lin, A.Fu, "HOT SAX: Finding the Most Unusual time Series Subsequence: Algorithms and Applications", *ICDM*. 2005.
- [9] E.Keogh, K.Chakrabarti, M.Pazzani, S.Mehrotra "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases", *Knowledge and Information Systems*, Vol.3, pp 263-286, Aug 2001.
- [10] Y.Khaokaq, S.Pukkawanna, "Time Series Anomaly Detection using Recessive Subsequence", *The International Conference on Information Network 2012*, pp.329-333, Feb 2012.
- [11] Wikipedia. (2016, May 16). Euclidean distance. [Online]. Available:
https://en.wikipedia.org/w/index.php?title=Euclidean_distance&oldid=736179287
- [12] B.McCune, J.B.Grace, D.L.Urban, "*Analysis of Ecological Communities*", Gleneden Beach Oregon USA, MjM Software, 2002.
- [13] Wikipedia. (2016, May 20). Histogram. [Online]. Available:
<https://sv.wikipedia.org/w/index.php?title=Histogram&oldid=32491011>
- [14] Wikipeida. (2016, May 21). Comma separated values. [Online]. Available:
https://en.wikipedia.org/w/index.php?title=Comma-separated_values&oldid=736139722
- [15] E.Staf, "A Study on Time Series Data Mining Techniques for Automotive Applications", Dep. Signals and Systems, Chalmers University of Technology, Gothenburg, 2016
Available:
<http://studentarbeten.chalmers.se/publication/238757-a-study-on-time-series-data-mining-techniques-for-automotive-applications>
- [16] B.Kulahcioglu, S.Ozdemir, B.Kumova. (2014, Feb). "Application of Symbolic Piecewise Aggregate Approximation (PAA) Analysis to ECG Signals". *Researchgate* [Online]. Available:
https://www.researchgate.net/publication/229011056_Application_of_Symbolic_Piecewise_Aggregate_Approximation_PAA_Analysis_to_ECG_Signals
- [17] Mathworks Inc. (2016, July 15). Buffer. [Online] Available:

- <http://se.mathworks.com/help/dsp/ref/buffer.html>
- [18] Mathworks Inc. (2016, July 15). Matlab Function. [Online] Available: <http://se.mathworks.com/help/simulink/slref/matlabfunction.html>
 - [19] Mathworks Inc. (2016, July 15). From workspace. [Online] Available: <http://se.mathworks.com/help/simulink/slref/fromworkspace.html>
 - [20] Mathworks Inc. (2016, July 15). Compare to constant. [Online] Available: <http://se.mathworks.com/help/simulink/slref/comparetoconstant.html>
 - [21] Mathworks Inc. (2016, July 15). Sample and hold. [Online] Available: <http://se.mathworks.com/help/dsp/ref/sampleandhold.html>
 - [22] Mathworks Inc. (2016, July 15). S-R Flip-Flop. [Online] Available: http://se.mathworks.com/help/simulink/ref_extras/srflipflop.html
 - [23] Mathworks Inc. (2016, July 15). Delay. [Online] Available: <http://se.mathworks.com/help/simulink/slref/delay.html>
 - [24] Mathworks Inc. (2016, July 15). Mean. [Online] Available: <http://se.mathworks.com/help/physmod/sps/powersys/ref/mean.html>
 - [25] Mathworks Inc. (2016, July 15). Counter. [Online] Available: <http://se.mathworks.com/help/dsp/ref/counter.html>
 - [26] Mathworks Inc. (2016, July 15). Repeating sequence stair. [Online] Available: <http://se.mathworks.com/help/simulink/slref/repeatingsequencestair.html>
 - [27] Mathworks Inc. (2016, July 15). Switch. [Online] Available: <http://se.mathworks.com/help/simulink/slref/switch.html>

Blaga 1: Flödesschema och matlabkod program 1

Flödesschema för program 1:



Matlabkod för program 1

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Program
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Matchar mätfilerna synkas till samma cylinder ID värde
n=1;
while (n<=24)
    if (X1([n])==0)
        break
    else
        n=n+1;
    end
end
%Tilldela variabler från Volvos data. OBS Volvos variabel namn är ersatta
%med X pga sekretes.
T30 = X2(n:end);
Ing_ign_cut_signal = X3(n:end);
cylinder_ID = X4(n:end);

IngCutCyl_1=X5(n:end);
IngCutCyl_2=X6(n:end);
IngCutCyl_3=X7(n:end);
IngCutCyl_4=X8(n:end);
InjCutCyl_1=X9(n:end);
InjCutCyl_2=X10(n:end);
InjCutCyl_3=X12(n:end);
InjCutCyl_4=X13(n:end);

%-----
%-----INPUT VARIABLE-----
%-----

timeserie_lenght =size(T30,1);
PAA_sample_size = 2; %segmet_size %%antalet sampel som skapar en PAA nivå
PAA_pattern_lenght = 3; % Pattern_size%%antalet PAA nivåer som utgör ett PAA-mönster
% ---PAA_sample_size*PAA_pattern_lenght = normalize_size---
normalize_size = 6;% data_point_window% %antalet sampel som ingår i normaliseringsintervallet
normalixation_type = 4;
% normalixation_type = 1 NORMALIZE Feature scaling MOVING WINDOW
% normalixation_type = 2 NORMALIZE Standard score MOVING WINDOW
% normalixation_type = 3 NORMALIZE Feature scaling FIX WINDOW
% normalixation_type = 4 NORMALIZE Standard score FIX WINDOW
exclude_patterns = 7; % Tar bort alla unika mönster som förekommer mindre än antalet
exclude_patterns

% Avrundning då PAA_sample_number ej är ett heltal
PAA_sample_number = floor((timeserie_lenght/PAA_sample_size)); %antalet PAA nivåer som finns i
tidsserien
normalize_number = floor((timeserie_lenght/normalize_size)); %antalet normaliserings intervall
som finns i tidserien
```

```

%-----INPUT SIGNAL-----

% skapa tomm array av samma storlek som T30 array
normT30=zeros(size(T30));
PAA_T30=zeros(size(T30));

%-----NORMALIZE Feature scaling MOVING WINDOW-----
%-----
if (normalisation_type==1)
    c=1; %positionsräknare
    while(c<=timeserie_lenght)
        if ((c-normalize_size)<=0)
            min_subsequence = min(T30(c:c));
            max_subsequence = max(T30(c:c));
        else
            min_subsequence = min(T30(c-normalize_size:c));
            max_subsequence = max(T30(c-normalize_size:c));
        end
        normT30([c])= ((T30([c])- min_subsequence)/(max_subsequence-min_subsequence));
        c=c+1;
    end

    %-----NORMALIZE Standard score MOVING WINDOW-----
    %-----
elseif (normalisation_type==2)
    c=1; %positionsräknare
    while (c<=timeserie_lenght)
        if ((c-normalize_size)<=0)
            mean_subsequence = mean(T30(c:c));
        else
            mean_subsequence = mean(T30(c-normalize_size:c));
        end
        sum_deviation =0;
        e=(c-normalize_size);
        while (e<=c)
            if (e<=0)
                break
            else
                deviation = (T30([e]) - mean_subsequence)^2; %avikelsen
                sum_deviation = deviation + sum_deviation; %summan avikelsen
                e=e+1;
            end
        end
        standard_deviation = (sum_deviation / normalize_size);
        normT30([c])= ((T30([c]) - mean_subsequence)/standard_deviation);
        c=c+1;
    end

    %-----NORMALIZE Feature scaling FIX WINDOW-----
    %-----
elseif (normalisation_type==3)
    c=0; %antalet normaliseringsintervall räknare
    while(c<normalize_number)

```

```

    %max och min värdena i normaliseringsintervallet
    min_subsequence = min(T30(c*normalize_size+1:c*normalize_size+normalize_size));
    max_subsequence = max(T30(c*normalize_size+1:c*normalize_size+normalize_size));
    e=1; %positionsräknare i normaliseringsintervallet
    while(e<=normalize_size)
        %normalisering med Feature scaling och max och min värdena i subsekvensen
        %normT30([c*normalize_size+e])= ((T30([c*normalize_size+e])-
min_subsequence)/(max_subsequence-min_subsequence));
        %normalisering med mean zero and [-1,1]
        normT30([c*PAA_sample_size+e])= 2*((T30([c*PAA_sample_size+e])-
min_subsequence)/(max_subsequence-min_subsequence))-1;
        e=e+1;
    end
    c=c+1;
end

%-----NORMALIZE Standard score-----
%-----
elseif (normalisation_type==4)
    c=0; %antalet normaliseringsintervall räknare
    while(c<normalize_number)
        %medelvärde i normaliseringsintervallet
        mean_subsequence = mean(T30(c*normalize_size+1:c*normalize_size+normalize_size));
        sum_deviation =0;
        e=1; %positionsräknare i normaliseringsintervallet
        while (e<=normalize_size)
            deviation = (T30([c*normalize_size+e])- mean_subsequence)^2; %avvikelsen
            sum_deviation = deviation + sum_deviation; %summan avvikelsen
            e=e+1;
        end
        standard_deviation = ((sum_deviation / normalize_size)); %standard avvikelsen
        % standard_deviation=1;
        e=1; %positionsräknare i normaliseringsintervallet
        while(e<=normalize_size)
            %normalisering med standard score
            normT30([c*normalize_size+e])= ((T30([c*normalize_size+e])-
mean_subsequence)/standard_deviation);
            e=e+1;
        end
        c=c+1;
    end
end

%-----PAA-algoritim-----
%-----
% PAA picewise aggregated approximation
c=0; %subsekvensräknare
while(c<PAA_sample_number)
    e=1; %positionsräknare i subsekvensen
    PAA = mean(normT30(c*PAA_sample_size+1:c*PAA_sample_size+PAA_sample_size)); %medelvärde
av subsekvensen
    PAA_sequence([c+1]) = PAA; %sparar medelvärde i array
    while(e<=PAA_sample_size)

```

```

        PAA_T30([c*PAA_sample_size+e])= PAA; %fyller array med medelvärde för plott
        e=e+1;
    end
    c=c+1;
end

PAA_matrix_rows = ceil(PAA_sample_number/PAA_pattern_lenght); %räknar ut antalet rader för
matrisen
PAA_sequence_matrix = zeros(PAA_matrix_rows, PAA_pattern_lenght+2); %fyller matrisen med
nollor.

c=0; %radräknare för matrisen
while(c<PAA_matrix_rows)
    e=1; % positionsräknare/kolumnräknare i matrisen
    while(e<=PAA_pattern_lenght)
        if (c*PAA_pattern_lenght+e <= PAA_sample_number)
            PAA_sequence_matrix(c+1, e) = PAA_sequence([c*PAA_pattern_lenght+e]); % fyller
matrisen rad för rad med värden från PAA_sequence array
            e=e+1;
        else
            break %hoppas ur array om antalet värden som lagts in i matrisen är uppnått,
jämfört med
        end
    end
    c=c+1;
end

%-----Förberedelse för att lagra data i CSV fil-----
%
CSV_matrix = zeros(timeserie_lenght, 15); %skapa matrix som skall skrivas ut
% följande rader ger rubriker i CSV fil
CSV_matrix(:,1) = T30;
CSV_matrix(:,2) = normT30;
CSV_matrix(:,3) = PAA_T30;
CSV_matrix(:,4) = cylinder_ID;
CSV_matrix(:,5) = IngCutCyl_1;
CSV_matrix(:,6) = IngCutCyl_2;
CSV_matrix(:,7) = IngCutCyl_3;
CSV_matrix(:,8) = IngCutCyl_4;
CSV_matrix(:,9) = InjCutCyl_1;
CSV_matrix(:,10) = InjCutCyl_2;
CSV_matrix(:,11) = InjCutCyl_3;
CSV_matrix(:,12) = InjCutCyl_4;

% radräknare i
i=1;
while(i<=timeserie_lenght)
    CSV_matrix(i,15) = i;
    i=i+1;
end

```

```

%-----Bedömning av vilken cylinder misfire sker på-----
-----
c=1;
while (c<=timeserie_lenght)
    % Misfire på cylinder 1
    if ((IngCutCyl_1([c])==1) || (InjCutCyl_1([c])==1)) && (cylinder_ID([c])<=5)
        CSV_matris(c,13) = 1;
    % Misfire på cylinder 3
    elseif
        ((IngCutCyl_3([c])==1) || (InjCutCyl_3([c])==1)) && ((cylinder_ID([c])>=6) && (cylinder_ID([c])<=11)
        )
            CSV_matris(c,13) = 3;
    % Misfire på cylinder 4
    elseif
        ((IngCutCyl_4([c])==1) || (InjCutCyl_4([c])==1)) && ((cylinder_ID([c])>=12) && (cylinder_ID([c])<=17
        ))
            CSV_matris(c,13) = 4;
    % Misfire på cylinder 2
    elseif
        ((IngCutCyl_2([c])==1) || (InjCutCyl_2([c])==1)) && ((cylinder_ID([c])>=18) && (cylinder_ID([c])<=23
        ))
            CSV_matris(c,13) = 2;
    % Ingen misfire
    else
        CSV_matris(c,13) = 0;
    end
    c=c+1;
end

%-----Bedömning av misfire typ-----
-----
c=1;
while (c<=timeserie_lenght)
    %      % ----Enkel misfire-----
    if (CSV_matris(c,13)>0)
        CSV_matris(c,14) = 99;
    else
        % ----Normal-----
        CSV_matris(c,14) = 10;
    end
    c=c+1;
end

%-----Gör mönster matris-----
-----
c=0; %räknare för while loop

while (c<(timeserie_lenght/PAA_sample_size))
    %tmp_array är en temporär lagring för omstrukturering av data
    tmp_array(c+1,1) = CSV_matris(c*PAA_sample_size+1,3); %sparar värdet från CSV_matris i
    vald kolum och rad

```

```

        tmp_array(c+1,2) = CSV_matris(c*PAA_sample_size+1,14); %sparar värdet från CSV_matris i
        vald kolum och rad
        tmp_array(c+1,3) = CSV_matris(c*PAA_sample_size+1,13); %sparar värdet från CSV_matris i
        vald kolum och rad
        c=c+1; %räknare för while loop
    end

    d=0; %räknare för while loop
    while (d < timeserie_lenght/(PAA_sample_size*PAA_pattern_lenght))
        for e=1:PAA_pattern_lenght
            if (e+PAA_pattern_lenght*d > timeserie_lenght/PAA_sample_size )
                break; %avbryter for-loop om man försöker nå en rad som inte finns, out of
                dimension of matris
            else
                monster_matris(d+1,e) = tmp_array(e+PAA_pattern_lenght*d,1); %sparar värdet i
                monster_matris från tmp_array
                monster_matris(d+1,e+PAA_pattern_lenght) = tmp_array(e+PAA_pattern_lenght*d,2);
                %sparar värdet i monster_matris från tmp_array
                monster_matris(d+1,e+PAA_pattern_lenght*2) = tmp_array(e+PAA_pattern_lenght*d,3);
                %sparar värdet i monster_matris från tmp_array
            end
        end
        d=d+1;
    end

    %-----HITTA UNIKA MÖNSTER-----
    -
    unique_matris = unique(monster_matris(:,PAA_pattern_lenght+1:2*PAA_pattern_lenght),'rows');

    %-----MEDELVÄRDE FÖR MÖNSTER-----
    ---
    unique_matris= wextend('ac','zpd',unique_matris,PAA_pattern_lenght+1,'r'); %öka matrisen med
    nollor för kommande beräkning

    for k=1:size(unique_matris,1) % loopar igenom alla rader i unique_matrix och kör forloopen på
        div_count=0;
        for m=1:size(monster_matris,1) %
            if monster_matris(m,PAA_pattern_lenght+1:2*PAA_pattern_lenght) ==
            unique_matris(k,1:PAA_pattern_lenght) %letar upp mönster som matchar referensmönstret som är
            valt i while loopen
                unique_matris(k,PAA_pattern_lenght+1:PAA_pattern_lenght*2) =
                unique_matris(k,PAA_pattern_lenght+1:PAA_pattern_lenght*2) +
                monster_matris(m,1:PAA_pattern_lenght); %summerar valda kolumner och sparar undan
                div_count = div_count+1; % räknar antalet matchningar och används för att räkna ut
                medelvärde

            end
        end
        unique_matris(k,PAA_pattern_lenght+1:PAA_pattern_lenght*2) =
        unique_matris(k,PAA_pattern_lenght+1:PAA_pattern_lenght*2)/ div_count; %räknar ut medelvärdet
        för referensmönstret som väljs i while loopen
        unique_matris(k,PAA_pattern_lenght*2+1) = div_count; % skriv ut antal mönstermatchningar
    end

```

```

%----- Antal misfire per cykel unique_matris-----
for k=1:size(unique_matris,1)
    count_misf =0;
    for m=1:PAA_pattern_lenght
        if unique_matris(k,m) == 99
            count_misf= count_misf +1;
        end
    end
    unique_matris(k,PAA_pattern_lenght*2+2)= count_misf/(6/PAA_sample_size);    % 6st segment
per 180grader
end
%----- Antal misfire per cykel monster_matris-----
for k=1:size(monster_matris,1)
    count_misf =0;
    for m=1:PAA_pattern_lenght
        if monster_matris(k,PAA_pattern_lenght+m) == 99
            count_misf= count_misf +1;
        end
    end
    monster_matris(k,PAA_pattern_lenght*3+1)= count_misf/(6/PAA_sample_size);    % 6st segment
per 180grader
end
%
% %-----Skapa CSV fil-----
string_name = s.FileList{FileNbr};
fname = 'unique_matris_logg.txt';
fileID = fopen(fname, 'a+');
i = 1;

formatSpec = '%f';
while (i <= size(unique_matris,1))
    j=1;
    while (j<=size(unique_matris,2))
        fprintf(fileID,formatSpec,unique_matris(i,j));
        j=j+1;
    end
    fprintf(fileID,string_name);
    fprintf(fileID,'\n');
    i=i+1;
end
fclose(fileID);

% -----Rensar matriser inför nästa filinläsning-----
clear CSV_matris IngCutCyl_1 IngCutCyl_2 IngCutCyl_3 IngCutCyl_4 Ing_ign_cut_signal
InjCutCyl_1 InjCutCyl_2 InjCutCyl_3 InjCutCyl_4 PAA PAA_T30 PAA_matrix_rows PAA_pattern_lenght
PAA_sample_number PAA_sample_size PAA_sequence PAA_sequence_matrix T30 ans ax1 ax2 cylinder_ID
d deviation div_count e histogram_array1 histogram_array2 i j k m mean_subsequence
monster_matris n nlag normT30 normalixation_type normalize_number normalize_size rows
square_distance standard_deviation string_filename string_name sum_deviation timeserie_lenght
tmp_array unique_matris unique_matris_name vecs;
end
toc
disp('-----');

```

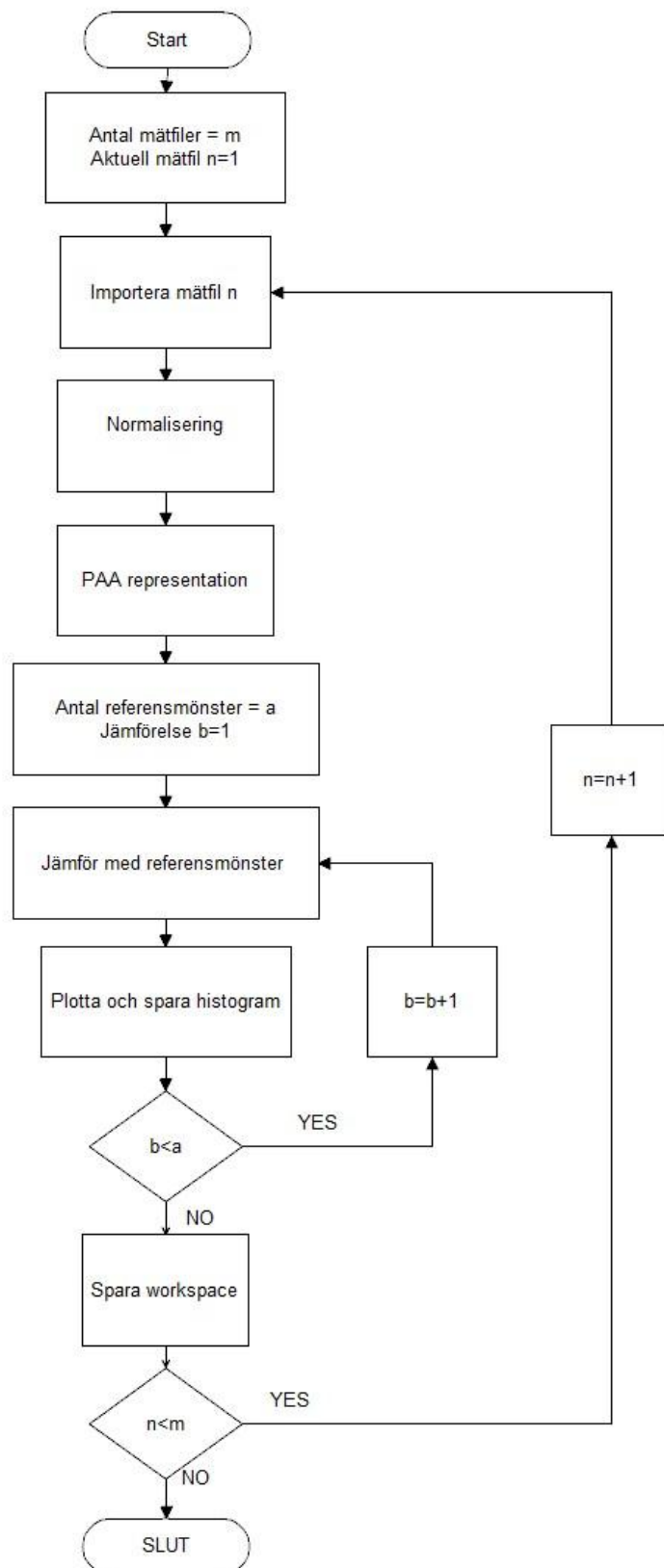
```
disp('-----KLAR-----');  
disp('
```

.....

Published with MATLAB® R2014a

Bilaga 2: Flödesschema och matlabkod program 2

Flödesschema för program 2:



Matlabkod för program 2:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Program
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Matchar mätfilerna synkas till samma cylinder ID värde
n=1;
while (n<=24)
    if (X1([n])==0)
        break
    else
        n=n+1;
    end
end
%Tilldela variabler från Volvos data. OBS Volvos variabel namn är ersatta
%med X pga sekretes.
T30 = X2(n:end);
Ing_ign_cut_signal = X3(n:end);
cylinder_ID = X4(n:end);

IngCutCyl_1=X5(n:end);
IngCutCyl_2=X6(n:end);
IngCutCyl_3=X7(n:end);
IngCutCyl_4=X8(n:end);
InjCutCyl_1=X9(n:end);
InjCutCyl_2=X10(n:end);
InjCutCyl_3=X12(n:end);
InjCutCyl_4=X13(n:end);

%-----
%-----INPUT VARIABLE-----
%-----
timeserie_lenght =size(T30,1);
PAA_sample_size = 2; %segmet_size %%antalet sampel som skapar en PAA nivå
PAA_pattern_lenght = 3; % Pattern_size%%antalet PAA nivåer som utgör ett PAA-mönster
% ---PAA_sample_size*PAA_pattern_lenght = normalize_size---
normalize_size = 6;% data_point_window% %antalet sampel som ingår i normaliseringsintervallet
normalixation_type = 4;
% normalixation_type = 1 NORMALIZE Feature scaling MOVING WINDOW
% normalixation_type = 2 NORMALIZE Standard score MOVING WINDOW
% normalixation_type = 3 NORMALIZE Feature scaling FIX WINDOW
% normalixation_type = 4 NORMALIZE Standard score FIX WINDOW
exclude_patterns = 7; % Tar bort alla unika mönster som förekommer mindre än antalet
exclude_patterns

% Avrundning då PAA_sample_number ej är ett heltal
PAA_sample_number = floor((timeserie_lenght/PAA_sample_size)); %antalet PAA nivåer som finns i
tidsserien
```

```

normalize_number = floor((timeserie_lenght/normalize_size)); %antalet normaliserings intervall
som finns i tidserien

% skapa tomm array av samma storlek som T30 array
normT30=zeros(size(T30));
PAA_T30=zeros(size(T30));

%-----NORMALIZE Feature scaling MOVING WINDOW-----
%-----
if (normalixation_type==1)
    c=1; %positionsräknare
    while(c<=timeserie_lenght)
        if ((c-normalize_size)<=0)
            min_subsequence = min(T30(c:c));
            max_subsequence = max(T30(c:c));
        else
            min_subsequence = min(T30(c-normalize_size:c));
            max_subsequence = max(T30(c-normalize_size:c));
        end
        normT30([c])= ((T30([c]) - min_subsequence)/(max_subsequence-min_subsequence));
        c=c+1;
    end

    %-----NORMALIZE Standard score MOVING WINDOW-----
    %-----
elseif (normalixation_type==2)
    c=1; %positionsräknare
    while (c<=timeserie_lenght)
        if ((c-normalize_size)<=0)
            mean_subsequence = mean(T30(c:c));
        else
            mean_subsequence = mean(T30(c-normalize_size:c));
        end
        sum_deviation =0;
        e=(c-normalize_size);
        while (e<=c)
            if (e<=0)
                break
            else
                deviation = (T30([e]) - mean_subsequence)^2; %avikelsen
                sum_deviation = deviation + sum_deviation; %summan avikelsen
                e=e+1;
            end
        end
        standard_deviation = (sum_deviation / normalize_size);
        normT30([c])= ((T30([c]) - mean_subsequence)/standard_deviation);
        c=c+1;
    end

    %-----NORMALIZE Feature scaling FIX WINDOW-----
    %-----
elseif (normalixation_type==3)

```

```

c=0; %antalet normaliseringsintervall räknare
while(c<normalize_number)
    %max och min värdena i normaliseringsintervallet
    min_subsequence = min(T30(c*normalize_size+1:c*normalize_size+normalize_size));
    max_subsequence = max(T30(c*normalize_size+1:c*normalize_size+normalize_size));
    e=1; %positionsräknare i normaliseringsintervallet
    while(e<=normalize_size)
        %normalisering med Feature scaling och max och min värdena i subsekvensen
        % normT30([c*normalize_size+e])= ((T30([c*normalize_size+e])-
min_subsequence)/(max_subsequence-min_subsequence));
        %normalisering med mean zero and a standard deviation of one
        normT30([c*PAA_sample_size+e])= 2*((T30([c*PAA_sample_size+e])-
min_subsequence)/(max_subsequence-min_subsequence))-1;
        e=e+1;
    end
    c=c+1;
end

%-----NORMALIZE Standard score-----
%
elseif (normalisation_type==4)
c=0; %antalet normaliseringsintervall räknare
while(c<normalize_number)
    %medelvärde i normaliseringsintervallet
    mean_subsequence = mean(T30(c*normalize_size+1:c*normalize_size+normalize_size));
    sum_deviation =0;
    e=1; %positionsräknare i normaliseringsintervallet
    while (e<=normalize_size)
        deviation = (T30([c*normalize_size+e])- mean_subsequence)^2; %avvikelsen
        sum_deviation = deviation + sum_deviation; %summan avvikelsen
        e=e+1;
    end
    standard_deviation = sqrt(sum_deviation / normalize_size); %standard avvikelsen
    %
    standard_deviation=1;
    e=1; %positionsräknare i normaliseringsintervallet
    while(e<=normalize_size)
        %normalisering med standard score
        normT30([c*normalize_size+e])= ((T30([c*normalize_size+e])-
mean_subsequence)/standard_deviation);
        e=e+1;
    end
    c=c+1;
end
end

%-----PAA-algoritim-----
%
% PAA picewise aggregated approximation
c=0; %subsekvensräknare
while(c<PAA_sample_number)
    e=1; %positionsräknare i subsekvensen
    PAA = mean(normT30(c*PAA_sample_size+1:c*PAA_sample_size+PAA_sample_size)); %medelvärde
    av subsekvensen

```

```

PAA_sequence([c+1]) = PAA; %sparar medelvärdet i array
while(e<=PAA_sample_size)
    PAA_T30([c*PAA_sample_size+e])= PAA; %fyller array med medelvrde för plott
    e=e+1;
end
c=c+1;
end

PAA_matrix_rows = ceil(PAA_sample_number/PAA_pattern_lenght); %räknar ut antalet rader för
matrisen
PAA_sequence_matrix = zeros(PAA_matrix_rows, PAA_pattern_lenght+2); %fyller matrisen med
nollor.

c=0; %radräknare för matrisen
while(c<PAA_matrix_rows)
    e=1; % positionsräknare/kolumnräknare i matrisen
    while(e<=PAA_pattern_lenght)
        if (c*PAA_pattern_lenght+e <= PAA_sample_number)
            PAA_sequence_matrix(c+1, e) = PAA_sequence([c*PAA_pattern_lenght+e]); % fyller
matrisen rad för rad med värden från PAA_sequence array
            e=e+1;
        else
            break %hoppas ur array om antalet värden som lagts in i matrisen är uppnått,
jämfört med
        end
    end
    c=c+1;
end

%-----Förberedelse för att lagra data i CSV fil-----
%
CSV_matrix = zeros(timeserie_lenght, 15); %skapa matrix som skall skrivas ut
% följande rader ger rubriker i CSV fil
CSV_matrix(:,1) = T30;
CSV_matrix(:,2) = normT30;
CSV_matrix(:,3) = PAA_T30;
CSV_matrix(:,4) = cylinder_ID;
CSV_matrix(:,5) = IngCutCyl_1;
CSV_matrix(:,6) = IngCutCyl_2;
CSV_matrix(:,7) = IngCutCyl_3;
CSV_matrix(:,8) = IngCutCyl_4;
CSV_matrix(:,9) = InjCutCyl_1;
CSV_matrix(:,10) = InjCutCyl_2;
CSV_matrix(:,11) = InjCutCyl_3;
CSV_matrix(:,12) = InjCutCyl_4;

% radräknare i
i=1;
while(i<=timeserie_lenght)
    CSV_matrix(i,15) = i;
    i=i+1;
end

```

```

end

%-----Bedömning av vilken cylinder misfire sker på-----
%-----
c=1;
while (c<=timeserie_lenght)
    % Misfire på cylinder 1
    if ((IngCutCyl_1([c])==1) || (InjCutCyl_1([c])==1)) && (cylinder_ID([c])<=5)
        CSV_matris(c,13) = 1;
    % Misfire på cylinder 3
    elseif
    ((IngCutCyl_3([c])==1) || (InjCutCyl_3([c])==1)) && ((cylinder_ID([c])>=6) && (cylinder_ID([c])<=11)
    )
        CSV_matris(c,13) = 3;
    % Misfire på cylinder 4
    elseif
    ((IngCutCyl_4([c])==1) || (InjCutCyl_4([c])==1)) && ((cylinder_ID([c])>=12) && (cylinder_ID([c])<=17)
    ))
        CSV_matris(c,13) = 4;
    % Misfire på cylinder 2
    elseif
    ((IngCutCyl_2([c])==1) || (InjCutCyl_2([c])==1)) && ((cylinder_ID([c])>=18) && (cylinder_ID([c])<=23)
    ))
        CSV_matris(c,13) = 2;
    % Ingen misfire
    else
        CSV_matris(c,13) = 0;
    end
    c=c+1;
end

%-----Bedömning av misfire typ-----
%-----
c=1;
while (c<=timeserie_lenght)
    %      % ----Enkel misfire-----
    if (CSV_matris(c,13)>0)
        CSV_matris(c,14) = 99;
    else
        % ----Normal-----
        CSV_matris(c,14) = 10;
    end
    c=c+1;
end

%-----Gör mönster matris-----
%-----
c=0; %räknare för while loop

while (c<(timeserie_lenght/PAA_sample_size))
    %tmp_array är en temporär lagring för omstrukturering av data

```

```

    tmp_array(c+1,1) = CSV_matris(c*PAA_sample_size+1,3); %sparar värdet från CSV_matris i
    vald kolum och rad
    tmp_array(c+1,2) = CSV_matris(c*PAA_sample_size+1,14); %sparar värdet från CSV_matris i
    vald kolum och rad
    tmp_array(c+1,3) = CSV_matris(c*PAA_sample_size+1,13); %sparar värdet från CSV_matris i
    vald kolum och rad
    c=c+1; %räknare för while loop
end

d=0; %räknare för while loop
while (d < timeserie_lenght/(PAA_sample_size*PAA_pattern_lenght))
    for e=1:PAA_pattern_lenght
        if (e+PAA_pattern_lenght*d > timeserie_lenght/PAA_sample_size )
            break; %avbryter for-loop om man försöker nå en rad som inte finns, out of
dimension of matrix
        else
            monster_matris(d+1,e) = tmp_array(e+PAA_pattern_lenght*d,1); %sparar värdet i
monster_matris från tmp_array
            monster_matris(d+1,e+PAA_pattern_lenght) = tmp_array(e+PAA_pattern_lenght*d,2);
%sparar värdet i monster_matris från tmp_array
            monster_matris(d+1,e+PAA_pattern_lenght*2) = tmp_array(e+PAA_pattern_lenght*d,3);
%sparar värdet i monster_matris från tmp_array
        end
    end
    d=d+1;
end

%----- Antal misfire per cykel monster_matris-----
for k=1:size(monster_matris,1)
    count_misf =0;
    for m=1:PAA_pattern_lenght
        if monster_matris(k,PAA_pattern_lenght+m) == 99
            count_misf= count_misf +1;
        end
    end
    monster_matris(k,PAA_pattern_lenght*3+1)= count_misf/(6/PAA_sample_size); % 6st segment
per 180grader
end

% % %-----squared Relativt euklidiskt distansmått (S_RED)-----
---
monster_matris= wextend('ac','zpd',monster_matris,11,'r');
for m=1:size(monster_matris,1)
    sum_monster_pattern = sqrt(sum(monster_matris(m,1:PAA_pattern_lenght).^2));
    sum_distance_RED =0;
    compare_patter_matrix = [99 99 99 -0.113708544      0.000748392      0.112960152; 10 10
10 0.031122129      0.017810161      -0.048932258];
    for n=1:size(compare_patter_matrix,1)
        if n==1
            type_comb =0;
            minimun_dist=0;
        end
    end
end

```

```

        sum_distance_RED=0;
        sum_pattern =
sqrt(sum(compare_patter_matrix(n,PAA_pattern_lenght+1:PAA_pattern_lenght*2).^2));
        for e=1:PAA_pattern_lenght
            distance_RED = ((monster_matrix(m,e)/ sum_monster_pattern)-
(compare_patter_matrix(n,PAA_pattern_lenght+e)/sum_pattern))^2;
            sum_distance_RED= sum_distance_RED +distance_RED;

        end
        monster_matrix(m,PAA_pattern_lenght*3+n)=sum_distance_RED;
    end

end

% Matris med namn enligt mönster från unique_matrix, som legends till
% histogram plottarna ex: '10-10-10-10-91-91-10-10'
for k=1:size(compare_patter_matrix,1)
    compare_patter_matrix_name(k,1) =
cellstr(strrep(num2str(compare_patter_matrix(k,1:PAA_pattern_lenght)), ' ','-'));
end

%-----Skapa mapp för workspace och plottar för EN fil-----
% ---- lägg mapp i plottar_last_varvtal som har namn med last och varvtal,
% en för varje fil
string_name = s.FileList{FileNbr};
mkdir('Plottar_Last_Vartal',string_name(1:end-4)); % utan filändelse och sökväg för inre
mappen
    histogram_array1=0;
    histogram_array2=0;
for k=1:size(compare_patter_matrix,1)
    clear histogram_array1 histogram_array2
    histogram_array1=0;
    histogram_array2=0;
    for m=1:size(monster_matrix,1)
        if monster_matrix(m,PAA_pattern_lenght+1:PAA_pattern_lenght*2) ==
compare_patter_matrix(k,1:PAA_pattern_lenght)
            histogram_array1(end+1,1) = monster_matrix(m,PAA_pattern_lenght*3+k);
        else
            histogram_array2(end+1,1) = monster_matrix(m,PAA_pattern_lenght*3+k);
        end
    end
end

ax1=subplot(2,1,1);
hist(histogram_array1(:,1),100);
grid on;
legend(compare_patter_matrix_name(k,1));
xlabel('Distance');
ylabel('Quantity');
title('Histogram');
%xlim([0 4]);
ax2=subplot(2,1,2);
hist(histogram_array2(:,1),100);
grid on;
legend('Other patterns');
xlabel('Distance');
ylabel('Quantity');

```

```

%linkaxes([ax2,ax1],'xy'); %Synkar x- och y-axel skala
%xlim([0 4]);

% -----Spara histogramplottar i mapp för last-varvtal för varje unikt
% mönster samt sparar undan workspacet.
string_filename =strrep(num2str(compare_patter_matrix(k,1:PAA_pattern_lenght)),' ','-
');
    saveas(gcf,fullfile(pathname(1:end-1),'Plottar_Last_Vartal',string_name(1:end-
4),string_filename),'fig');
    close all;

end

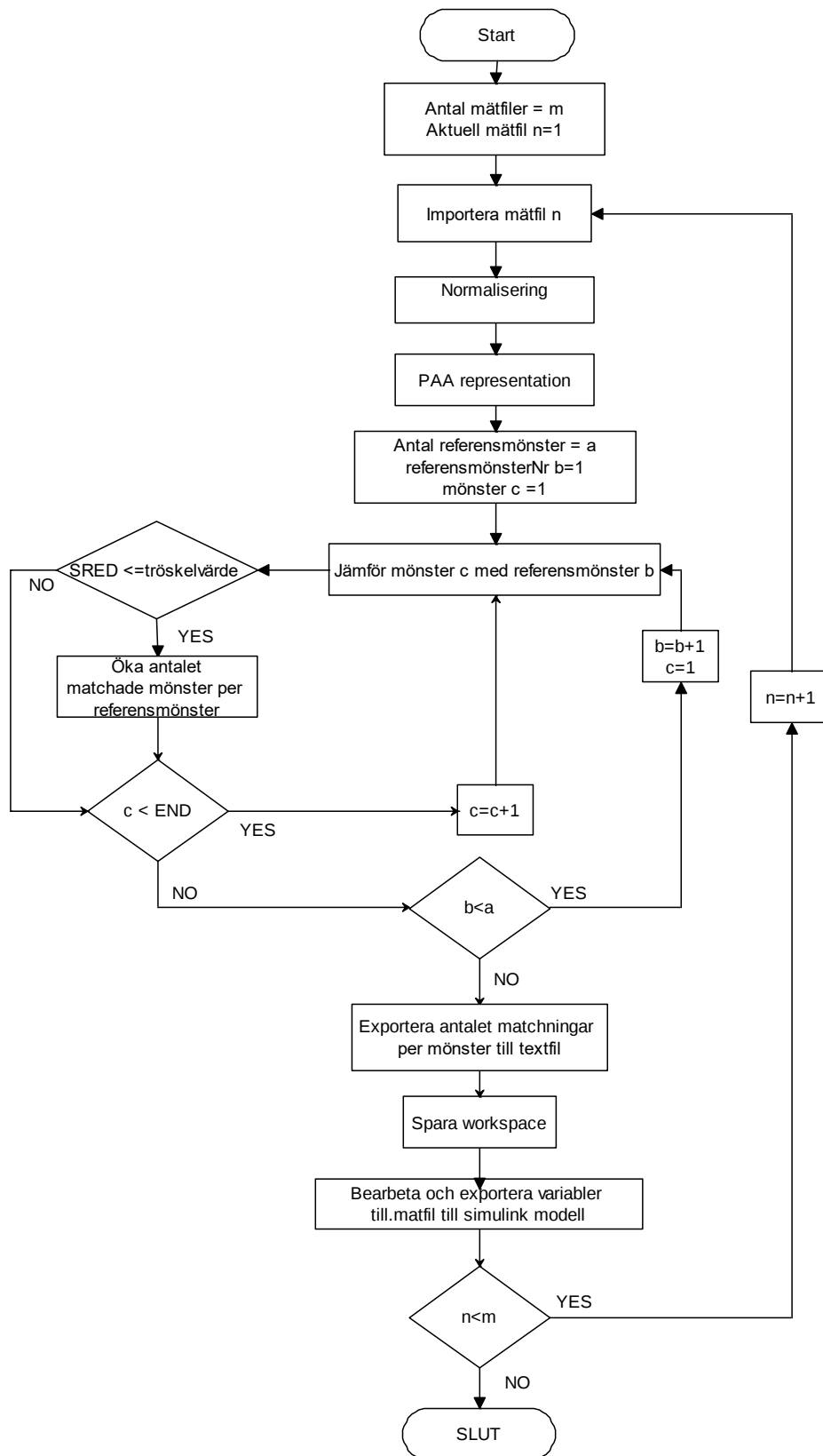
% -----Sparar hela wokspacet-----
% save(fullfile(pathname(1:end-1),'Plottar_Last_Vartal',M.TextFileName(13:end-
4),'workspace.mat'));
% -----Sparar utvalda matriser -----
save(fullfile(pathname(1:end-1),'Plottar_Last_Vartal',string_name(1:end-
4),'workspace.mat'),'compare_patter_matrix','normalixation_type',
'normalize_size','PAA_sample_size','PAA_pattern_lenght','exclude_patterns');

% -----Rensar matriser inför nästa filinläsning-----
clear CSV_matris IngCutCyl_1 IngCutCyl_2 IngCutCyl_3 IngCutCyl_4 Ing_ign_cut_signal
InjCutCyl_1 InjCutCyl_2 InjCutCyl_3 InjCutCyl_4 PAA PAA_T30 PAA_matrix_rows PAA_pattern_lenght
PAA_sample_number PAA_sample_size PAA_sequence PAA_sequence_matrix T30 ans ax1 ax2 cylinder_ID
d deviation div_count e histogram_array1 histogram_array2 i j k m mean_subsequence
monster_matris n nlag normT30 normalixation_type normalize_number normalize_size rows
square_distance standard_deviation string_filename string_name sum_deviation timeserie_lenght
tmp_array unique_matris unique_matris_name vecs;
end
toc
disp('-----');
disp('-----KLAR-----');
disp('-----');

```

Bilaga 3: Flödesschema och matlabkod program 3

Flödesschema för program 3



Matlabkod för program 3:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Program
3%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Matchar mätfilerna, synkas till samma cylinder ID värde
n=1;
while (n<=24)
    if (X3([n])==0)
        break
    else
        n=n+1;
    end
end
%Tilldela variabler från Volvos data. OBS Volvos variabel namn är ersatta
%med X pga sekretes.
T30 = X1(n:end);
Ing_ign_cut_signal = X2(n:end);
cylinder_ID = X3(n:end);

IngCutCyl_1=X4(n:end);
IngCutCyl_2=X5(n:end);
IngCutCyl_3=X6(n:end);
IngCutCyl_4=X7(n:end);
InjCutCyl_1=X8(n:end);
InjCutCyl_2=X9(n:end);
InjCutCyl_3=X10(n:end);
InjCutCyl_4=X11(n:end);
rpm = X12(n:end);
load = X13(n:end);

%-----
%-----INPUT VARIABLE-----
%-----
timeserie_lenght =size(T30,1);
PAA_sample_size = 2; %segmet_size %%antalet sampel som skapar en PAA nivå
PAA_pattern_lenght = 3; % Pattern_size%%antalet PAA nivåer som utgör ett PAA-mönster
% ---PAA_sample_size*PAA_pattern_lenght = normalize_size---
normalize_size = 6;% data_point_window% %antalet sampel som ingår i normaliseringsintervallet
normalixation_type = 4;
% normalixation_type = 1 NORMALIZE Feature scaling MOVING WINDOW
% normalixation_type = 2 NORMALIZE Standard score MOVING WINDOW
% normalixation_type = 3 NORMALIZE Feature scaling FIX WINDOW
% normalixation_type = 4 NORMALIZE Standard score FIX WINDOW
exclude_patterns = 7; % Tar bort alla unika mönster som förekommer mindre än antalet
exclude_patterns

% Avrundning då PAA_sample_number ej är ett heltal
PAA_sample_number = floor((timeserie_lenght/PAA_sample_size)); %antalet PAA nivåer som finns i
tidsserien
normalize_number = floor((timeserie_lenght/normalize_size)); %antalet normaliserings intervall
som finns i tidserien

% skapa tomm array av samma storlek som T30 array
normT30=zeros(size(T30));
```

```

PAA_T30=zeros(size(T30));

%-----NORMALIZE Feature scaling MOVING WINDOW-----
%-----
if (normalisation_type==1)
    c=1; %positionsräknare
    while(c<=timeserie_lenght)
        if ((c-normalize_size)<=0)
            min_subsequence = min(T30(c:c));
            max_subsequence = max(T30(c:c));
        else
            min_subsequence = min(T30(c-normalize_size:c));
            max_subsequence = max(T30(c-normalize_size:c));
        end
        normT30([c])= ((T30([c]) - min_subsequence)/(max_subsequence-min_subsequence));
        c=c+1;
    end

%-----NORMALIZE Standard score MOVING WINDOW-----
%-----
elseif (normalisation_type==2)
    c=1; %positionsräknare
    while (c<=timeserie_lenght)
        if ((c-normalize_size)<=0)
            mean_subsequence = mean(T30(c:c));
        else
            mean_subsequence = mean(T30(c-normalize_size:c));
        end
        sum_deviation =0;
        e=(c-normalize_size);
        while (e<=c)
            if (e<=0)
                break
            else
                deviation = (T30([e]) - mean_subsequence)^2; %avvikelsen
                sum_deviation = deviation + sum_deviation; %summan avvikelsen
                e=e+1;
            end
        end
        standard_deviation = (sum_deviation / normalize_size);
        normT30([c])= ((T30([c]) - mean_subsequence)/standard_deviation);
        c=c+1;
    end

%-----NORMALIZE Feature scaling FIX WINDOW-----
%-----
elseif (normalisation_type==3)
    c=0; %antalet normaliseringsintervall räknare
    while(c<normalize_number)
        %max och min värdena i normaliseringsintervallet
        min_subsequence = min(T30(c*normalize_size+1:c*normalize_size+normalize_size));
        max_subsequence = max(T30(c*normalize_size+1:c*normalize_size+normalize_size));
        e=1; %positionsräknare i normaliseringsintervallet

```

```

        while(e<=normalize_size)
            %normalisering med Feature scaling och max och min värdena i subsekvensen
            normT30([c*normalize_size+e])= ((T30([c*normalize_size+e])-
min_subsequence)/(max_subsequence-min_subsequence));
            %normalisering med mean zero and a standard deviation of one
            %normT30([c*PAA_sample_size+e])= 2*((T30([c*PAA_sample_size+e])-
min_subsequence)/(max_subsequence-min_subsequence))-1;
            e=e+1;
        end
        c=c+1;
    end

%-----NORMALIZE Standard score-----
%-----
elseif (normalisation_type==4)
    c=0; %antalet normaliseringsintervall räknare
    while(c<normalize_number)
        %medelvärde i normaliseringsintervallet
        mean_subsequence = mean(T30(c*normalize_size+1:c*normalize_size+normalize_size));
        sum_deviation =0;
        e=1; %positionsräknare i normaliseringsintervallet
        while (e<=normalize_size)
            deviation = (T30([c*normalize_size+e]) - mean_subsequence)^2; %avvikelsen
            sum_deviation = deviation + sum_deviation; %summan avvikelsen
            e=e+1;
        end
        standard_deviation = (sum_deviation / normalize_size); %standard avvikelsen
        e=1; %positionsräknare i normaliseringsintervallet
        while(e<=normalize_size)
            %normalisering med standard score
            normT30([c*normalize_size+e])= ((T30([c*normalize_size+e])-
mean_subsequence)/standard_deviation);
            e=e+1;
        end
        c=c+1;
    end
end

%-----PAA-algoritim-----
%-----
% PAA picewise aggregated approximation
c=0; %subsekvensräknare
while(c<PAA_sample_number)
    e=1; %positionsräknare i subsekvensen
    PAA = mean(normT30(c*PAA_sample_size+1:c*PAA_sample_size+PAA_sample_size)); %medelvärde
av subsekvensen
    PAA_sequence([c+1]) = PAA; %sparar medelvärde i array
    while(e<=PAA_sample_size)
        PAA_T30([c*PAA_sample_size+e])= PAA; %fyller array med medelvrde för plott
        e=e+1;
    end
    c=c+1;
end
end

```

```

PAA_matrix_rows = ceil(PAA_sample_number/PAA_pattern_lenght); %räknar ut antalet rader för
matrisen
PAA_sequence_matrix = zeros(PAA_matrix_rows, PAA_pattern_lenght+2); %fyller matrisen med
nollor.

c=0; %radräknare för matrisen
while(c<PAA_matrix_rows)
    e=1; % positionsräknare/kolumnräknare i matrisen
    while(e<=PAA_pattern_lenght)
        if (c*PAA_pattern_lenght+e <= PAA_sample_number)
            PAA_sequence_matrix(c+1, e) = PAA_sequence([c*PAA_pattern_lenght+e]); % fyller
matrisen rad för rad med värden från PAA_sequence array
            e=e+1;
        else
            break %hoppas ur array om antalet värden som lagts in i matrisen är uppnått,
jämfört med
        end
    end
    c=c+1;
end

%-----Förberedelse för att lagra data i CSV fil-----
%
CSV_matris = zeros(timeserie_lenght, 15); %skapa matris som skall skrivas ut
% följande rader ger rubriker i CSV fil
CSV_matris(:,1) = T30;
CSV_matris(:,2) = normT30;
CSV_matris(:,3) = PAA_T30;
CSV_matris(:,4) = cylinder_ID;
CSV_matris(:,5) = IngCutCyl_1;
CSV_matris(:,6) = IngCutCyl_2;
CSV_matris(:,7) = IngCutCyl_3;
CSV_matris(:,8) = IngCutCyl_4;
CSV_matris(:,9) = InjCutCyl_1;
CSV_matris(:,10) = InjCutCyl_2;
CSV_matris(:,11) = InjCutCyl_3;
CSV_matris(:,12) = InjCutCyl_4;

% radräknare i
i=1;
while(i<=timeserie_lenght)
    CSV_matris(i,15) = i;
    i=i+1;
end

%-----Bedömning av vilken cylinder misfire sker på-----
-----
c=1;

```

```

while (c<=timeserie_lenght)
    % Misfire på cylinder 1
    if ((IngCutCyl_1([c])==1) || (InjCutCyl_1([c])==1)) && (cylinder_ID([c])<=5)
        CSV_matris(c,13) = 1;
    % Misfire på cylinder 3
    elseif
    ((IngCutCyl_3([c])==1) || (InjCutCyl_3([c])==1)) && ((cylinder_ID([c])>=6) && (cylinder_ID([c])<=11)
    )
        CSV_matris(c,13) = 3;
    % Misfire på cylinder 4
    elseif
    ((IngCutCyl_4([c])==1) || (InjCutCyl_4([c])==1)) && ((cylinder_ID([c])>=12) && (cylinder_ID([c])<=17
    ))
        CSV_matris(c,13) = 4;
    % Misfire på cylinder 2
    elseif
    ((IngCutCyl_2([c])==1) || (InjCutCyl_2([c])==1)) && ((cylinder_ID([c])>=18) && (cylinder_ID([c])<=23
    ))
        CSV_matris(c,13) = 2;
    % Ingen misfire
    else
        CSV_matris(c,13) = 0;
    end
    c=c+1;
end

%-----Bedömning av misfire typ-----
%
c=1;
while (c<=timeserie_lenght)
    % ----Enkel misfire-----
    if (CSV_matris(c,13)>0)
        CSV_matris(c,14) = 99;
    else
        % ----Normal-----
        CSV_matris(c,14) = 10;
    end
    c=c+1;
end

%-----Gör mönster matris-----
%
c=0; %räknare för while loop

while (c<(timeserie_lenght/PAA_sample_size))
    %tmp_array är en temporär lagring för omstrukturering av data
    tmp_array(c+1,1) = CSV_matris(c*PAA_sample_size+1,3); %sparar värdet från CSV_matris i
    vald kolum och rad
    tmp_array(c+1,2) = CSV_matris(c*PAA_sample_size+1,14); %sparar värdet från CSV_matris i
    vald kolum och rad
    tmp_array(c+1,3) = CSV_matris(c*PAA_sample_size+1,13); %sparar värdet från CSV_matris i
    vald kolum och rad
    c=c+1; %räknare för while loop
end

```

```

d=0; %räknare för while loop
while (d < timeserie_lenght/(PAA_sample_size*PAA_pattern_lenght))
    for e=1:PAA_pattern_lenght
        if (e+PAA_pattern_lenght*d > timeserie_lenght/PAA_sample_size )
            break; %avbryter for-loop om man försöker nå en rad som inte finns, out of
dimension of matrix
        else
            monster_matrix(d+1,e) = tmp_array(e+PAA_pattern_lenght*d,1); %sparar värdet i
monster_matrix från tmp_array
            monster_matrix(d+1,e+PAA_pattern_lenght) = tmp_array(e+PAA_pattern_lenght*d,2);
            %sparar värdet i monster_matrix från tmp_array
            monster_matrix(d+1,e+PAA_pattern_lenght*2) = tmp_array(e+PAA_pattern_lenght*d,3);
            %sparar värdet i monster_matrix från tmp_array
        end
    end
    d=d+1;
end

%----- Antal misfire per cykel monster_matrix-----
for k=1:size(monster_matrix,1)
    count_misf =0;
    for m=1:PAA_pattern_lenght
        if monster_matrix(k,PAA_pattern_lenght+m) == 99
            count_misf= count_misf +1;
        end
    end
    monster_matrix(k,PAA_pattern_lenght*3+1)= count_misf/(6/PAA_sample_size); % 6st segment
per 180grader
end

%-----
monster_matrix= wextend('ac','zpd',monster_matrix,11,'r');
for m=1:size(monster_matrix,1)
    raknare =0;
    sum_monster_pattern = sqrt(sum(monster_matrix(m,1:PAA_pattern_lenght).^2));
    sum_distance_RED =0;
    compare_patter_matrix = [99 99 99 -0.123136 0.000453 0.123033 ; 10 10 10 0.028472 0.008217 -0.038091];
    for n=1:size(compare_patter_matrix,1)
        if n==1
            type_comb =0;
            minimun_dist=0;
        end
        sum_distance_RED=0;
        sum_pattern =
sqrt(sum(compare_patter_matrix(n,PAA_pattern_lenght+1:PAA_pattern_lenght*2).^2));
        for e=1:PAA_pattern_lenght
            distance_RED = ((monster_matrix(m,e)/ sum_monster_pattern)-
(compare_patter_matrix(n,PAA_pattern_lenght+e)/sum_pattern))^2;
            sum_distance_RED= sum_distance_RED +distance_RED;
        end
        monster_matrix(m,PAA_pattern_lenght*3+n)=sum_distance_RED;
    end
end

```

```

end
compare_matrix= wextend('ac','zpd',compare_patter_matrix,2,'r');
for n=1:size(compare_patter_matrix,1)
    counter = 0;
    for m=1:size(monster_matrix,1)
        if monster_matrix(m,PAA_pattern_lenght*3+n)<2 %TRÖSKELVÄRDE
            counter = counter+1;
        end
    end
    compare_matrix(n,PAA_pattern_lenght*2+1) = counter;
end

% Matris med namn enligt mönster från unique_matrix, som legends till
% histogram plottarna ex: '10-10-10-10-91-91-10-10'
for k=1:size(compare_patter_matrix,1)
    compare_patter_matrix_name(k,1) =
cellstr(strrep(num2str(compare_patter_matrix(k,1:PAA_pattern_lenght)),' ','-'));
end

%-----Skapa mapp för workspace och plottar för EN fil-----
% ---- lägg mapp i plottar_last_varvtal som har namn med last och varvtal,
% en för varje fil

string_name = s.FileList{FileNbr};
fname = 'comp_matrix_logg.txt';
fileID = fopen(fname, 'a+');
i = 1;

formatSpec = '%f';
while (i <= size(compare_matrix,1))
    j=1;
    while (j<=size(compare_matrix,2))
        fprintf(fileID,formatSpec,compare_matrix(i,j));
        j=j+1;
    end
    fprintf(fileID,string_name);
    fprintf(fileID,'\n');
    i=i+1;
end
fclose(fileID);

string_name = s.FileList{FileNbr};
fname = 'unique_matrix_logg.txt';
fileID = fopen(fname, 'a+');
i = 1;

T30_simulink = zeros(size(X1));
T30_simulink = wextend('ac','zpd',T30_simulink,1,'r');
cylinder_ID_simulink = zeros(size(X3));
cylinder_ID_simulink = wextend('ac','zpd',cylinder_ID_simulink,1,'r');

for n=1:size(X1,1)
    T30_simulink(n,1) = n;
    cylinder_ID_simulink(n,1) = n;
end

```

```

end

T30_simulink(:,2) = X1(1:end);
cylinder_ID_simulink(:,2) = X3(1:end);

mkdir('Plottar_Last_Vartal',string_name(1:end-4));
% -----Sparar hela wokspacet-----
%save(fullfile(pathname(1:end-1),'Plottar_Last_Vartal',M.TextFileName(13:end-4),'workspace.mat'));
% -----Sparar utvalda matriser -----
save(fullfile(pathname(1:end-1),'Plottar_Last_Vartal',string_name(1:end-4),'workspace.mat'),'compare_patter_matrix','normalixation_type','normalize_size','PAA_sample_size','PAA_pattern_lenght','exclude_patterns');
save(fullfile(pathname(1:end-1),'Plottar_Last_Vartal',string_name(1:end-4),'to_simulink.mat'),'T30_simulink','cylinder_ID_simulink');

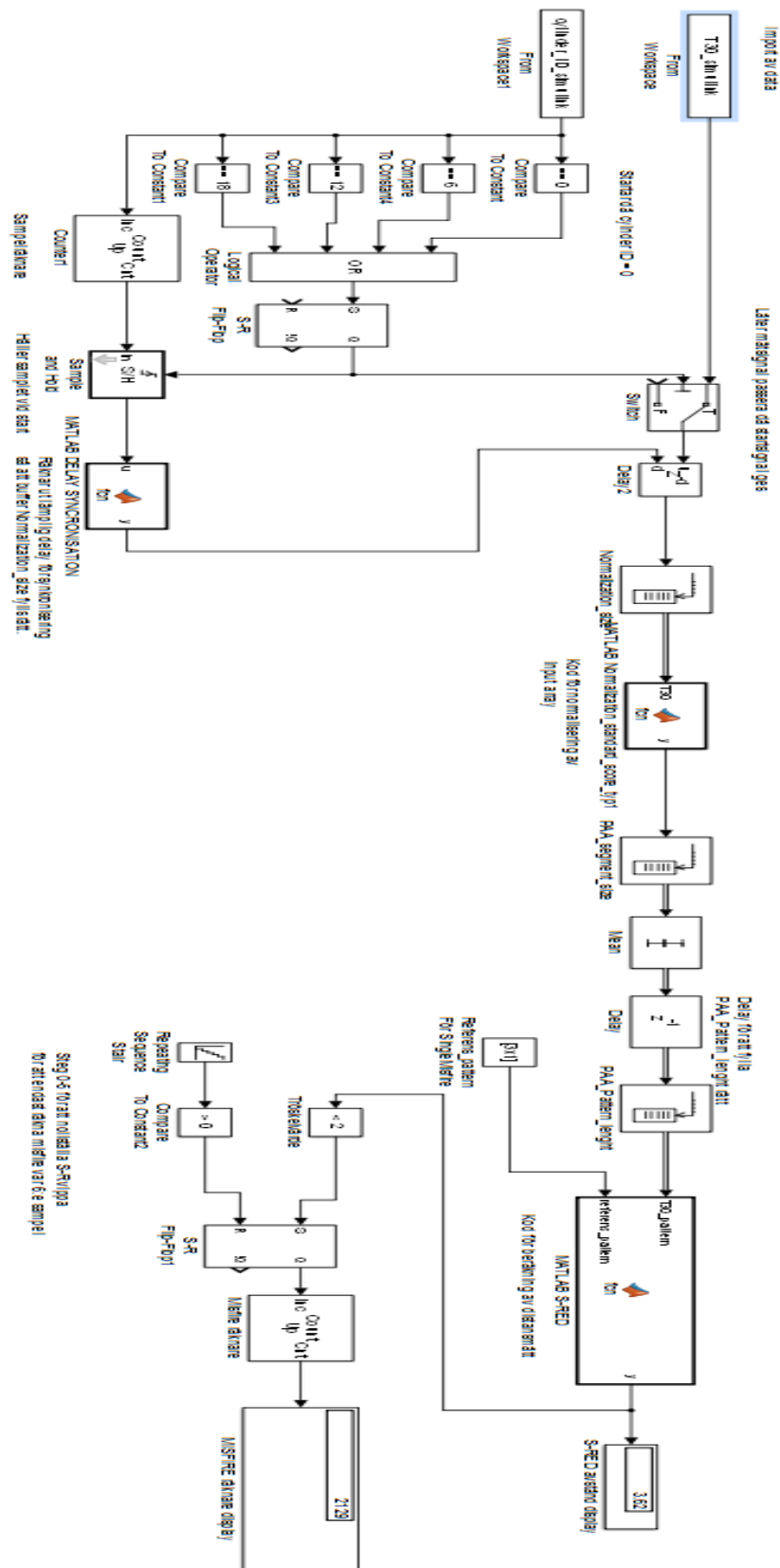
% -----Rensar matriser inför nästa filinläsning-----
clear CSV_matris IngCutCyl_1 IngCutCyl_2 IngCutCyl_3 IngCutCyl_4 Ing_ign_cut_signal
InjCutCyl_1 InjCutCyl_2 InjCutCyl_3 InjCutCyl_4 PAA PAA_T30 PAA_matrix_rows PAA_pattern_lenght
PAA_sample_number PAA_sample_size PAA_sequence PAA_sequence_matrix T30 ans ax1 ax2 cylinder_ID
d deviation div_count e histogram_array1 histogram_array2 i j k m mean_subsequence
monster_matris n nlag normT30 normalixation_type normalize_number normalize_size rows
square_distance standard_deviation string_filename string_name sum_deviation timeserie_lenght
tmp_array unique_matris unique_matris_name vecs T30_simulink cylinder_ID_simulink;
end
toc
disp('-----');
disp('-----KLAR-----');
disp('-----');

```

.....

Published with MATLAB® R2014a

Bilaga 4: Simulink modell



Specifikation av Matlab Function block: MATLAB DELAY SYNCRONISATION

```
%-----MATLAB Delay synchronisation-----  
function y = fcn(u)  
delay=0;  
if u==0;  
    delay=0;  
    %räknar ut lämplig fördröjning/delay då insignalen u  
    %uppfyller följande villkor  
elseif (6*1-u)>=0  
    delay=(6*1-u);  
elseif (6*2-u)>=0  
    delay=(6*2-u);  
elseif (6*3-u)>=0  
    delay=(6*3-u);  
elseif (6*4-u)>=0  
    delay=(6*4-u);  
end  
y = delay; %utsignal framräknad fördröjning
```

Specifikation av Matlab Function block: MATLAB Normalization_standard_score_typ1

```
%-----MATLAB Normalisation_standard_score_typ1-----  
function y = fcn(T30)  
%#Initiera variabler  
sum_deviation = 0;  
variance = 0;  
deviation_datapoints = 0;  
  
mean_sequece = mean(T30); %Medelvärdet av insignal T30  
for e=1:size(T30,1) %räknar ut summan av avvikelsen för varje  
värde i insignalen  
    deviation_datapoints = (T30(e,1) - mean_sequece)^2;  
    sum_deviation = sum_deviation + deviation_datapoints;  
end  
variance = (sum_deviation/size(T30,1)); %variansen  
normT30 = zeros(size(T30,1),1); %skapa en array av samma storlek  
som insignal  
for e=1:size(T30,1) %räknar ut det normaliserade värdet för aktuel  
datapunkt i arrayen  
    normT30(e,1)=(T30(e,1) - mean_sequece)/variance;  
end  
y = normT30; %utsignal
```

Specifikation av Matlab Function block: MATLAB S-RED

```
%-----MATLAB S-RED-----  
function y = fcn(T30_pattern, referens_pattern)  
compare_pattern = sqrt(sum((referens_pattern).^2));  
PAA_pattern = sqrt(sum((T30_pattern).^2));  
distance=0;  
for c=1:size(T30_pattern,1) %räknar ut SRED avståndet  
    a = ((T30_pattern(c,1)/PAA_pattern) - (referens_pattern(c,1) /  
        (compare_pattern))^2;  
    distance = distance+a;  
end  
y = distance; %utsignal SRED avstånd
```

Bilaga 5: Möjlig detektion i följande testfall

Feature scaling typ1

Varvtal [rpm]	1200	1500	2000	2500	3000	3500	4000	4500	5000	5500
Last [g/rev]										
0,2	-	-	N	N	N	N	-	-	-	-
0,3	N	N	N	N	N	N	N	N	N	N
0,4	N	N	N	N	N	N	N	N	N	N
0,6	N	-	J	N	N	N	-	N	N	N
0,8	N	N	J	N	N	N	N	N	N	N
1,2	N	N	J	N	N	N	N	N	N	N
1,4	N	-	-	N	-	-	-	-	-	-
1,6	-	N	J	N	N	N	N	N	N	N
1,8	-	N	-	-	-	-	-	-	-	-
2	-	-	J	N	N	N	N	N	N	N
2,2	-	-	-	-	-	-	-	-	N	-
2,3	-	-	J	N	-	N	N	N	-	-
2,4	-	-	-	-	N	-	-	-	-	-

Feature scaling typ2

Varvtal [rpm]	1200	1500	2000	2500	3000	3500	4000	4500	5000	5500
Last [g/rev]										
0,2	-	N	N	N	N	N	-	-	-	-
0,3	N	-	N	N	N	N	N	N	N	N
0,4	N	N	N	N	N	N	N	N	N	N
0,6	J	-	N	N	N	N	-	N	N	N
0,8	J	N	N	N	N	N	N	N	N	N
1,2	J	N	N	N	N	N	N	N	N	N
1,4	N	-	-	N	-	-	-	-	-	-
1,6	-	N	N	N	N	N	N	N	N	N
1,8	-	N	-	-	-	-	-	-	-	-
2	-	-	N	J	J	N	N	N	N	N
2,2	-	-	-	-	-	-	-	-	N	-
2,3	-	-	N	J	-	N	N	N	-	-
2,4	-	-	-	-	J	-	-	-	-	-

Standard score typ 2

Varvtal [rpm]	1200	1500	2000	2500	3000	3500	4000	4500	5000	5500
Last [g/rev]										
0,2	-	N	N	N	N	N	-	-	-	-
0,3	N	-	N	N	N	N	N	N	N	N
0,4	N	N	N	N	N	N	N	N	N	N
0,6	N	-	j	N	N	N	-	N	N	N
0,8	N	N	j	N	N	N	N	N	N	N
1,2	N	N	j	j	N	N	N	N	N	N
1,4	N	-	-	N	-	-	-	-	-	-
1,6	-	N	j	j	N	N	N	N	N	N
1,8	-	N	-	N	-	-	-	-	-	-
2	-	-	j	j	N	N	N	N	N	N
2,2	-	-	-	-	-	-	-	-	N	-
2,3	-	-	j	j	-	N	N	N	-	-
2,4	-	-	-	-	N	-	-	-	-	-

Standard score typ 3

Varvtal [rpm]	1200	1500	2000	2500	3000	3500	4000	4500	5000	5500
Last [g/rev]										
0,2	-	N	N	N	N	N	-	-	-	-
0,3	N	-	N	N	N	N	N	N	N	N
0,4	N	N	N	N	N	N	N	N	N	N
0,6	N	-	j	N	N	N	-	N	N	N
0,8	N	N	j	k	N	N	N	N	N	N
1,2	N	N	j	k	N	N	N	N	N	N
1,4	N	-	-	-	-	-	-	-	-	-
1,6	-	N	j	k	N	N	N	N	N	N
1,8	-	N	-	k	-	-	-	-	-	-
2	-	-	j	k	N	N	N	N	N	N
2,2	-	-	-	-	-	-	-	-	N	-
2,3	-	-	j	k	-	N	N	N	-	-
2,4	-	-	-	-	N	-	-	-	-	-

Bilaga 6: Noggrannhet i detektion per mätfil

Varvtal	Last	Typ	Procent
2000	0,6	Korrekta detektioner misfire	100,00%
2000	0,6	Inkorrekta detektioner misfire	0,00%
2000	0,6	Falsklarm	0,82%
2000	0,8	Korrekta detektioner misfire	100,00%
2000	0,8	Inkorrekta detektioner misfire	0,00%
2000	0,8	Falsklarm	0,17%
2000	1,2	Korrekta detektioner misfire	100,00%
2000	1,2	Inkorrekta detektioner misfire	0,00%
2000	1,2	Falsklarm	0,12%
2000	1,6	Korrekta detektioner misfire	100,00%
2000	1,6	Inkorrekta detektioner misfire	0,00%
2000	1,6	Falsklarm	0,12%
2000	2,0	Korrekta detektioner misfire	100,00%
2000	2,0	Inkorrekta detektioner misfire	0,00%
2000	2,0	Falsklarm	0,11%
2000	2,3	Korrekta detektioner misfire	99,95%
2000	2,3	Inkorrekta detektioner misfire	0,05%
2000	2,3	Falsklarm	0,13%
2500	0,2	Korrekta detektioner misfire	100,00%
2500	0,2	Inkorrekta detektioner misfire	0,00%
2500	0,2	Falsklarm	99,96%
2500	0,3	Korrekta detektioner misfire	100,00%
2500	0,3	Inkorrekta detektioner misfire	0,00%
2500	0,3	Falsklarm	93,86%
2500	0,4	Korrekta detektioner misfire	100,00%
2500	0,4	Inkorrekta detektioner misfire	0,00%
2500	0,4	Falsklarm	80,40%
2500	0,6	Korrekta detektioner misfire	100,00%
2500	0,6	Inkorrekta detektioner misfire	0,00%
2500	0,6	Falsklarm	37,29%

2500	0,8	Korrekta detektioner misfire	100,00%
2500	0,8	Inkorrekta detektioner misfire	0,00%
2500	0,8	Falsklarm	23,36%
2500	1,2	Korrekta detektioner misfire	100,00%
2500	1,2	Inkorrekta detektioner misfire	0,00%
2500	1,2	Falsklarm	0,15%
2500	1,6	Korrekta detektioner misfire	100,00%
2500	1,6	Inkorrekta detektioner misfire	0,00%
2500	1,6	Falsklarm	0,10%
2500	2,0	Korrekta detektioner misfire	100,00%
2500	2,0	Inkorrekta detektioner misfire	0,00%
2500	2,0	Falsklarm	0,10%
2500	2,3	Korrekta detektioner misfire	99,94%
2500	2,3	Inkorrekta detektioner misfire	0,06%
2500	2,3	Falsklarm	0,12%
3000	0,2	Korrekta detektioner misfire	100,00%
3000	0,2	Inkorrekta detektioner misfire	0,00%
3000	0,2	Falsklarm	100,00%
3000	0,3	Korrekta detektioner misfire	100,00%
3000	0,3	Inkorrekta detektioner misfire	0,00%
3000	0,3	Falsklarm	99,92%
3000	0,4	Korrekta detektioner misfire	100,00%
3000	0,4	Inkorrekta detektioner misfire	0,00%
3000	0,4	Falsklarm	95,65%
3000	0,6	Korrekta detektioner misfire	100,00%
3000	0,6	Inkorrekta detektioner misfire	0,00%
3000	0,6	Falsklarm	85,51%
3000	0,8	Korrekta detektioner misfire	100,00%
3000	0,8	Inkorrekta detektioner misfire	0,00%
3000	0,8	Falsklarm	46,59%
3000	1,2	Korrekta detektioner misfire	100,00%
3000	1,2	Inkorrekta detektioner misfire	0,00%
3000	1,2	Falsklarm	25,50%

3000	1,6	Korrekte detektioner misfire	100,00%
3000	1,6	Inkorrekte detektioner misfire	0,00%
3000	1,6	Falsklarm	0,20%
3000	2,0	Korrekte detektioner misfire	100,00%
3000	2,0	Inkorrekte detektioner misfire	0,00%
3000	2,0	Falsklarm	0,11%
3000	2,4	Korrekte detektioner misfire	100,00%
3000	2,4	Inkorrekte detektioner misfire	0,00%
3000	2,4	Falsklarm	0,11%
3500	0,2	Korrekte detektioner misfire	100,00%
3500	0,2	Inkorrekte detektioner misfire	0,00%
3500	0,2	Falsklarm	100,00%
3500	0,3	Korrekte detektioner misfire	100,00%
3500	0,3	Inkorrekte detektioner misfire	0,00%
3500	0,3	Falsklarm	100,00%
3500	0,4	Korrekte detektioner misfire	100,00%
3500	0,4	Inkorrekte detektioner misfire	0,00%
3500	0,4	Falsklarm	99,99%
3500	0,6	Korrekte detektioner misfire	100,00%
3500	0,6	Inkorrekte detektioner misfire	0,00%
3500	0,6	Falsklarm	96,81%
3500	0,8	Korrekte detektioner misfire	100,00%
3500	0,8	Inkorrekte detektioner misfire	0,00%
3500	0,8	Falsklarm	88,28%
3500	1,2	Korrekte detektioner misfire	100,00%
3500	1,2	Inkorrekte detektioner misfire	0,00%
3500	1,2	Falsklarm	40,18%
3500	1,6	Korrekte detektioner misfire	100,00%
3500	1,6	Inkorrekte detektioner misfire	0,00%
3500	1,6	Falsklarm	23,55%
3500	2,0	Korrekte detektioner misfire	100,00%
3500	2,0	Inkorrekte detektioner misfire	0,00%
3500	2,0	Falsklarm	0,40%

3500	2,3	Korrekte detektioner misfire	100,00%
3500	2,3	Inkorrekte detektioner misfire	0,00%
3500	2,3	Falsklarm	0,13%
4000	0,3	Korrekte detektioner misfire	100,00%
4000	0,3	Inkorrekte detektioner misfire	0,00%
4000	0,3	Falsklarm	100,00%
4000	0,4	Korrekte detektioner misfire	100,00%
4000	0,4	Inkorrekte detektioner misfire	0,00%
4000	0,4	Falsklarm	100,00%
4000	0,6	Korrekte detektioner misfire	100,00%
4000	0,6	Inkorrekte detektioner misfire	0,00%
4000	0,6	Falsklarm	99,73%
4000	0,8	Korrekte detektioner misfire	100,00%
4000	0,8	Inkorrekte detektioner misfire	0,00%
4000	0,8	Falsklarm	96,21%
4000	1,2	Korrekte detektioner misfire	100,00%
4000	1,2	Inkorrekte detektioner misfire	0,00%
4000	1,2	Falsklarm	84,39%
4000	1,6	Korrekte detektioner misfire	100,00%
4000	1,6	Inkorrekte detektioner misfire	0,00%
4000	1,6	Falsklarm	43,87%
4000	2,0	Korrekte detektioner misfire	100,00%
4000	2,0	Inkorrekte detektioner misfire	0,00%
4000	2,0	Falsklarm	28,02%
4000	2,3	Korrekte detektioner misfire	100,00%
4000	2,3	Inkorrekte detektioner misfire	0,00%
4000	2,3	Falsklarm	3,02%
4500	0,3	Korrekte detektioner misfire	100,00%
4500	0,3	Inkorrekte detektioner misfire	0,00%
4500	0,3	Falsklarm	100,00%
4500	0,4	Korrekte detektioner misfire	100,00%
4500	0,4	Inkorrekte detektioner misfire	0,00%
4500	0,4	Falsklarm	100,00%

4500	0,6	Korrekt detekteret misfire	100,00%
4500	0,6	Inkorrekt detekteret misfire	0,00%
4500	0,6	Falsklarm	100,00%
4500	0,8	Korrekt detekteret misfire	100,00%
4500	0,8	Inkorrekt detekteret misfire	0,00%
4500	0,8	Falsklarm	98,25%
4500	1,2	Korrekt detekteret misfire	100,00%
4500	1,2	Inkorrekt detekteret misfire	0,00%
4500	1,2	Falsklarm	92,61%
4500	1,6	Korrekt detekteret misfire	100,00%
4500	1,6	Inkorrekt detekteret misfire	0,00%
4500	1,6	Falsklarm	84,82%
4500	2,0	Korrekt detekteret misfire	100,00%
4500	2,0	Inkorrekt detekteret misfire	0,00%
4500	2,0	Falsklarm	75,97%
4500	2,3	Korrekt detekteret misfire	100,00%
4500	2,3	Inkorrekt detekteret misfire	0,00%
4500	2,3	Falsklarm	50,65%
5000	0,3	Korrekt detekteret misfire	100,00%
5000	0,3	Inkorrekt detekteret misfire	0,00%
5000	0,3	Falsklarm	100,00%
5000	0,4	Korrekt detekteret misfire	100,00%
5000	0,4	Inkorrekt detekteret misfire	0,00%
5000	0,4	Falsklarm	100,00%
5000	0,6	Korrekt detekteret misfire	100,00%
5000	0,6	Inkorrekt detekteret misfire	0,00%
5000	0,6	Falsklarm	100,00%
5000	0,8	Korrekt detekteret misfire	100,00%
5000	0,8	Inkorrekt detekteret misfire	0,00%
5000	0,8	Falsklarm	100,00%
5000	1,2	Korrekt detekteret misfire	100,00%
5000	1,2	Inkorrekt detekteret misfire	0,00%
5000	1,2	Falsklarm	98,65%

5000	1,6	Korrekt detekteret misfire	100,00%
5000	1,6	Inkorrekt detekteret misfire	0,00%
5000	1,6	Falsklarm	93,98%
5000	2,0	Korrekt detekteret misfire	100,00%
5000	2,0	Inkorrekt detekteret misfire	0,00%
5000	2,0	Falsklarm	90,64%
5000	2,2	Korrekt detekteret misfire	100,00%
5000	2,2	Inkorrekt detekteret misfire	0,00%
5000	2,2	Falsklarm	86,46%
1200	0,3	Korrekt detekteret misfire	100,00%
1200	0,3	Inkorrekt detekteret misfire	0,00%
1200	0,3	Falsklarm	0,14%
1200	0,4	Korrekt detekteret misfire	99,81%
1200	0,4	Inkorrekt detekteret misfire	0,19%
1200	0,4	Falsklarm	0,15%
1200	0,6	Korrekt detekteret misfire	99,81%
1200	0,6	Inkorrekt detekteret misfire	0,19%
1200	0,6	Falsklarm	0,14%
1200	0,8	Korrekt detekteret misfire	99,95%
1200	0,8	Inkorrekt detekteret misfire	0,05%
1200	0,8	Falsklarm	0,13%
1200	1,2	Korrekt detekteret misfire	100,00%
1200	1,2	Inkorrekt detekteret misfire	0,00%
1200	1,2	Falsklarm	0,55%
1200	1,4	Korrekt detekteret misfire	100,00%
1200	1,4	Inkorrekt detekteret misfire	0,00%
1200	1,4	Falsklarm	0,31%
1500	0,2	Korrekt detekteret misfire	100,00%
1500	0,2	Inkorrekt detekteret misfire	0,00%
1500	0,2	Falsklarm	81,19%
1500	0,4	Korrekt detekteret misfire	100,00%
1500	0,4	Inkorrekt detekteret misfire	0,00%
1500	0,4	Falsklarm	0,13%

1500	0,8	Korrekte detektioner misfire	99,95%
1500	0,8	Inkorrekte detektioner misfire	0,05%
1500	0,8	Falsklarm	0,12%
1500	1,2	Korrekte detektioner misfire	100,00%
1500	1,2	Inkorrekte detektioner misfire	0,00%
1500	1,2	Falsklarm	0,14%
1500	1,6	Korrekte detektioner misfire	100,00%
1500	1,6	Inkorrekte detektioner misfire	0,00%
1500	1,6	Falsklarm	0,14%
1500	1,8	Korrekte detektioner misfire	99,86%
1500	1,8	Inkorrekte detektioner misfire	0,14%
1500	1,8	Falsklarm	0,14%
2000	0,2	Korrekte detektioner misfire	100,00%
2000	0,2	Inkorrekte detektioner misfire	0,00%
2000	0,2	Falsklarm	97,21%
2000	0,3	Korrekte detektioner misfire	100,00%
2000	0,3	Inkorrekte detektioner misfire	0,00%
2000	0,3	Falsklarm	76,20%
2000	0,4	Korrekte detektioner misfire	100,00%
2000	0,4	Inkorrekte detektioner misfire	0,00%
2000	0,4	Falsklarm	32,45%
5500	0,3	Korrekte detektioner misfire	100,00%
5500	0,3	Inkorrekte detektioner misfire	0,00%
5500	0,3	Falsklarm	100,00%
5500	0,4	Korrekte detektioner misfire	100,00%
5500	0,4	Inkorrekte detektioner misfire	0,00%
5500	0,4	Falsklarm	100,00%
5500	0,6	Korrekte detektioner misfire	100,00%
5500	0,6	Inkorrekte detektioner misfire	0,00%
5500	0,6	Falsklarm	100,00%
5500	0,8	Korrekte detektioner misfire	100,00%
5500	0,8	Inkorrekte detektioner misfire	0,00%
5500	0,8	Falsklarm	100,00%

5500	1,2	Korrekte detektioner misfire	100,00%
5500	1,2	Inkorrekte detektioner misfire	0,00%
5500	1,2	Falsklarm	99,98%
5500	1,6	Korrekte detektioner misfire	100,00%
5500	1,6	Inkorrekte detektioner misfire	0,00%
5500	1,6	Falsklarm	95,47%
5500	2,0	Korrekte detektioner misfire	100,00%
5500	2,0	Inkorrekte detektioner misfire	0,00%
5500	2,0	Falsklarm	92,26%

Bilaga 7: Referensmönster misfire

MönsterID_1	MönsterID_2	MönsterID_3	PAA1	PAA2	PAA3	Antal förekomster i mätfil	Varvtal	Last
99	99	99	-0,024627	-0,007321	0,031948	2107	N1200	L0,3
99	99	99	-0,020259	-0,004347	0,024605	2108	N1200	L0,4
99	99	99	-0,014586	-0,000473	0,015059	2106	N1200	L0,6
99	99	99	-0,011403	0,000453	0,01095	2111	N1200	L0,8
99	99	99	-0,008904	0,000458	0,008447	2107	N1200	L1,2
99	99	99	-0,008634	0,000417	0,008217	2107	N1200	L1,4
99	99	99	-0,054633	-0,005415	0,060048	2109	N1500	L0,2
99	99	99	-0,033721	-0,003728	0,037449	2110	N1500	L0,4
99	99	99	-0,022139	0,000031	0,022108	2110	N1500	L0,8
99	99	99	-0,016169	0,000508	0,015661	2111	N1500	L1,2
99	99	99	-0,014289	0,0004	0,013889	2112	N1500	L1,6
99	99	99	-0,014053	0,000364	0,013689	2110	N1500	L1,8
99	99	99	-0,078517	-0,004258	0,082775	2107	N2000	L0,2
99	99	99	-0,074975	-0,00448	0,079454	2111	N2000	L0,3
99	99	99	-0,077837	-0,000613	0,07845	2107	N2000	L0,4
99	99	99	-0,062853	0,000821	0,062032	2111	N2000	L0,6
99	99	99	-0,050988	0,000782	0,050206	2107	N2000	L0,8
99	99	99	-0,037698	0,000708	0,03699	2110	N2000	L1,2
99	99	99	-0,03272	-0,000054	0,032774	2112	N2000	L1,6
99	99	99	-0,030286	-0,000438	0,030724	2108	N2000	L2,0
99	99	99	-0,028638	-0,000763	0,029401	2110	N2000	L2,3
99	99	99	-0,099257	-0,004501	0,103758	2911	N2500	L0,2
99	99	99	-0,097652	-0,006214	0,103866	2908	N2500	L0,3
99	99	99	-0,100626	-0,001807	0,102432	2908	N2500	L0,4
99	99	99	-0,095654	0,001257	0,094397	2912	N2500	L0,6
99	99	99	-0,085812	0,001232	0,08458	2910	N2500	L0,8
99	99	99	-0,070942	0,001252	0,06969	2907	N2500	L1,2
99	99	99	-0,064097	-0,001201	0,065298	2911	N2500	L1,6
99	99	99	-0,05471	-0,003984	0,058693	2910	N2500	L2,0
99	99	99	-0,049252	-0,004769	0,054021	1809	N2500	L2,3
99	99	99	-0,127699	-0,000087	0,127787	2909	N3000	L0,2
99	99	99	-0,119835	-0,003198	0,123033	2910	N3000	L0,3
99	99	99	-0,123755	-0,002384	0,126139	2910	N3000	L0,4
99	99	99	-0,120739	-0,001297	0,122036	2908	N3000	L0,6
99	99	99	-0,114169	0,00105	0,113119	2906	N3000	L0,8
99	99	99	-0,100986	0,002305	0,098682	2910	N3000	L1,2
99	99	99	-0,091823	0,00189	0,089932	2911	N3000	L1,6
99	99	99	-0,08557	0,00191	0,08366	2910	N3000	L2,0
99	99	99	-0,078521	0,000538	0,077983	1807	N3000	L2,4
99	99	99	-0,140592	0,000412	0,140179	2836	N3500	L0,2
99	99	99	-0,137131	-0,001509	0,13864	2908	N3500	L0,3

99	99	99	-0,138733	0,000238	0,138495	2911	N3500	L0,4
99	99	99	-0,14055	0,004645	0,135905	2907	N3500	L0,6
99	99	99	-0,142293	0,004665	0,137628	2908	N3500	L0,8
99	99	99	-0,133074	0,004686	0,128388	2910	N3500	L1,2
99	99	99	-0,123136	0,004714	0,118422	2908	N3500	L1,6
99	99	99	-0,110913	0,003961	0,106953	1959	N3500	L2,0
99	99	99	-0,104105	0,003328	0,100777	1882	N3500	L2,3
99	99	99	-0,155581	-0,002719	0,1583	3107	N4000	L0,3
99	99	99	-0,154602	-0,000348	0,15495	3110	N4000	L0,4
99	99	99	-0,154721	0,004779	0,149942	3110	N4000	L0,6
99	99	99	-0,157617	0,006172	0,151445	3109	N4000	L0,8
99	99	99	-0,150876	0,005004	0,145872	3108	N4000	L1,2
99	99	99	-0,142599	0,003481	0,139118	3111	N4000	L1,6
99	99	99	-0,129183	0,002551	0,126633	2109	N4000	L2,0
99	99	99	-0,128557	0,001754	0,126803	1812	N4000	L2,3
99	99	99	-0,174525	-0,002587	0,177112	3110	N4500	L0,3
99	99	99	-0,173974	-0,00077	0,174744	3110	N4500	L0,4
99	99	99	-0,171504	0,00498	0,166524	3107	N4500	L0,6
99	99	99	-0,172525	0,007929	0,164596	3110	N4500	L0,8
99	99	99	-0,16989	0,006653	0,163237	3109	N4500	L1,2
99	99	99	-0,164109	0,003347	0,160762	3107	N4500	L1,6
99	99	99	-0,156848	0,000944	0,155903	1811	N4500	L2,0
99	99	99	-0,152178	-0,000659	0,152837	1808	N4500	L2,3
99	99	99	-0,189138	-0,005468	0,194606	3109	N5000	L0,3
99	99	99	-0,189631	-0,005023	0,194654	3108	N5000	L0,4
99	99	99	-0,188904	0,000071	0,188833	3108	N5000	L0,6
99	99	99	-0,188182	0,006876	0,181306	3113	N5000	L0,8
99	99	99	-0,187757	0,00935	0,178408	3113	N5000	L1,2
99	99	99	-0,18473	0,009953	0,174777	2711	N5000	L1,6
99	99	99	-0,185806	0,013635	0,172171	1809	N5000	L2,0
99	99	99	-0,187227	0,013746	0,173481	1810	N5000	L2,2
99	99	99	-0,197841	-0,011572	0,209414	3107	N5500	L0,3
99	99	99	-0,198873	-0,010245	0,209118	3107	N5500	L0,4
99	99	99	-0,199824	-0,002738	0,202562	3107	N5500	L0,6
99	99	99	-0,199668	0,00464	0,195028	3107	N5500	L0,8
99	99	99	-0,200914	0,008575	0,192339	3113	N5500	L1,2
99	99	99	-0,202165	0,005069	0,197096	3109	N5500	L1,6
99	99	99	-0,203471	0,001559	0,201912	2208	N5500	L2,0

Referensmönster (medelvärde)	-0,113709	0,0007484	0,11296
------------------------------	-----------	-----------	---------

Bilaga 8: Referensmönster normal förbränning

MönsterID_1	MönsterID_2	MönsterID_3	PAA1	PAA2	PAA3	Antal förekomster i mätfil	Varvtal	Last
10	10	10	0,038412	-0,013351	-0,025062	12053	N1200	L0,3
10	10	10	0,021826	-0,003907	-0,017919	11711	N1200	L0,4
10	10	10	0,012469	-0,001058	-0,011412	11475	N1200	L0,6
10	10	10	0,008012	0,000755	-0,008767	13306	N1200	L0,8
10	10	10	0,004441	0,001929	-0,00637	12911	N1200	L1,2
10	10	10	0,004097	0,001901	-0,005998	12072	N1200	L1,4
10	10	10	-0,019609	-0,090015	0,109624	14102	N1500	L0,2
10	10	10	0,046187	-0,008095	-0,038091	12704	N1500	L0,4
10	10	10	0,016544	0,002552	-0,019097	14002	N1500	L0,8
10	10	10	0,009484	0,003124	-0,012608	12447	N1500	L1,2
10	10	10	0,006401	0,003485	-0,009886	12221	N1500	L1,6
10	10	10	0,005836	0,003507	-0,009343	12166	N1500	L1,8
10	10	10	-0,114892	-0,026026	0,140918	14745	N2000	L0,2
10	10	10	-0,044789	-0,049104	0,093893	14282	N2000	L0,3
10	10	10	0,050612	0,002933	-0,053545	13508	N2000	L0,4
10	10	10	0,069768	0,001887	-0,071656	13709	N2000	L0,6
10	10	10	0,052368	0,006591	-0,058958	14218	N2000	L0,8
10	10	10	0,028472	0,008392	-0,036863	13725	N2000	L1,2
10	10	10	0,016841	0,01066	-0,027501	14127	N2000	L1,6
10	10	10	0,012308	0,009098	-0,021407	13527	N2000	L2,0
10	10	10	0,009958	0,008217	-0,018176	13672	N2000	L2,3
10	10	10	-0,141156	-0,016133	0,157289	15616	N2500	L0,2
10	10	10	-0,158094	-0,029898	0,187992	15404	N2500	L0,3
10	10	10	-0,108852	0,003331	0,105522	16305	N2500	L0,4
10	10	10	0,03999	0,001604	-0,041594	18870	N2500	L0,6
10	10	10	0,073505	0,014017	-0,087521	15971	N2500	L0,8
10	10	10	0,059573	0,017909	-0,077482	16250	N2500	L1,2
10	10	10	0,034169	0,022241	-0,05641	16990	N2500	L1,6
10	10	10	0,021892	0,019551	-0,041443	15886	N2500	L2,0
10	10	10	0,018056	0,01861	-0,036665	13656	N2500	L2,3
10	10	10	-0,16494	-0,004424	0,169364	22198	N3000	L0,2
10	10	10	-0,204614	0,008919	0,195695	16801	N3000	L0,3
10	10	10	-0,213095	-0,006485	0,21958	17360	N3000	L0,4
10	10	10	-0,145017	-0,016467	0,161484	22849	N3000	L0,6
10	10	10	-0,019633	-0,005598	0,025231	16282	N3000	L0,8
10	10	10	0,07714	0,018918	-0,096058	18051	N3000	L1,2
10	10	10	0,057733	0,044963	-0,102696	17932	N3000	L1,6
10	10	10	0,037707	0,045247	-0,082954	16964	N3000	L2,0
10	10	10	0,028331	0,041517	-0,069847	15894	N3000	L2,4
10	10	10	-0,175023	-0,000003	0,175026	20472	N3500	L0,2
10	10	10	-0,198739	0,009084	0,189655	18368	N3500	L0,3
10	10	10	-0,250668	0,019876	0,230792	18265	N3500	L0,4

10	10	10	-0,27439	0,008735	0,265654	18096	N3500	L0,6
10	10	10	-0,264098	0,043573	0,220524	20859	N3500	L0,8
10	10	10	-0,013594	0,075455	-0,061861	18576	N3500	L1,2
10	10	10	0,039	0,098584	-0,137584	20751	N3500	L1,6
10	10	10	0,031869	0,095625	-0,127493	16433	N3500	L2,0
10	10	10	0,031785	0,074709	-0,106494	16112	N3500	L2,3
10	10	10	-0,203338	0,005798	0,19754	19580	N4000	L0,3
10	10	10	-0,234741	0,023531	0,21121	28117	N4000	L0,4
10	10	10	-0,293845	0,019073	0,274772	18627	N4000	L0,6
10	10	10	-0,302932	0,02048	0,282452	24824	N4000	L0,8
10	10	10	-0,194926	0,068721	0,126205	20987	N4000	L1,2
10	10	10	-0,082889	0,134767	-0,051878	20615	N4000	L1,6
10	10	10	-0,021594	0,160362	-0,138768	17692	N4000	L2,0
10	10	10	-0,005174	0,147713	-0,142539	16867	N4000	L2,3
10	10	10	-0,216167	0,003631	0,212537	22076	N4500	L0,3
10	10	10	-0,234506	0,01351	0,220996	21409	N4500	L0,4
10	10	10	-0,294843	0,037146	0,257698	21260	N4500	L0,6
10	10	10	-0,345693	0,047663	0,29803	20640	N4500	L0,8
10	10	10	-0,324016	0,0804	0,243616	22537	N4500	L1,2
10	10	10	-0,214513	0,133685	0,080827	21408	N4500	L1,6
10	10	10	-0,157054	0,193733	-0,036679	18532	N4500	L2,0
10	10	10	-0,10267	0,195127	-0,092456	19839	N4500	L2,3
10	10	10	-0,221631	-0,002191	0,223822	22116	N5000	L0,3
10	10	10	-0,237116	0,0057	0,231416	21170	N5000	L0,4
10	10	10	-0,283811	0,022143	0,261668	20923	N5000	L0,6
10	10	10	-0,336021	0,04318	0,292841	20954	N5000	L0,8
10	10	10	-0,418856	0,111464	0,307393	21383	N5000	L1,2
10	10	10	-0,395101	0,207374	0,187727	21540	N5000	L1,6
10	10	10	-0,328263	0,302576	0,025687	20328	N5000	L2,0
10	10	10	-0,30484	0,324757	-0,019918	18555	N5000	L2,2
10	10	10	-0,225073	-0,010067	0,23514	22721	N5500	L0,3
10	10	10	-0,237498	-0,003016	0,240513	22458	N5500	L0,4
10	10	10	-0,270412	0,017027	0,253385	24283	N5500	L0,6
10	10	10	-0,314361	0,033876	0,280485	25387	N5500	L0,8
10	10	10	-0,417901	0,088895	0,329006	25011	N5500	L1,2
10	10	10	-0,501439	0,2138	0,28764	22435	N5500	L1,6
10	10	10	-0,478295	0,342114	0,136181	20785	N5500	L2,0

Referensmönster (medelvärde)	-0,123366	0,0437963	0,07957
---------------------------------	-----------	-----------	---------