

Umeå universitet
Institutionen för datavetenskap
Examensarbete C, 10p

Utveckling av fläktvalsprogram för FUMEX AB

Andreas Lindroth
ens04alb@cs.umu.se

November 2005

Intern handledare:
Stefan Johansson

Extern handledare:
Lars Hedlund

Sammanfattning

Denna rapport redovisar ett 10 poängs examensarbete i datavetenskap utfört för Fumex AB. Arbetat har gått ut på att utveckla och implementera ett fläktnöjningsprogram utifrån en kravspecifikation. Rapporten beskriver de olika stadierna i utvecklingsarbetet och hur resonemanget har gått kring olika problem som uppkommit och dess lösningar. När Fumex AB i Skellefteå idag ska välja ut en lämplig fläkt till sina produkter får de för hand läsa ur en graf vilken som passar bäst. Denna metod är både omständlig och skapar problem för de som inte har tillräckliga kunskaper om hur detta görs. För att underlätta detta arbete har en applikation utvecklats, en Java-Applet. Applikationen kommer att användas vid offertförfrågningar och beställningar från kunder. Man räknar även med att den nya applikationen kommer att underlätta arbetet för distributörer och leverantörer som inte har samma kunskap om fläktar.

Abstract

This report presents a thesis for Bachelor of Science in Computing Science. The purpose of the thesis has been to develop a computer application which was implemented with the purpose to facilitate every day-work for the company Fumex AB in Skellefteå. The thesis describes the different stages in the development process and discusses the problems that have occurred and the solutions of the very same. The company in question works with industrial fans, a business where the fans under many years have been selected according to readings from graphs. The purpose of the developed program is to make this procedure easier for the employees as well as for the suppliers and distributors by choosing a fan from certain requirements. The application will also be used when submitting an offer and for customer orders. The application has been implemented as a Java-Applet.

Innehåll

1. Inledning.....	1
1.1 Företag och varumärken	1
1.2 Fumex AB	1
1.3 Bakgrund och uppgift.....	1
2. Fläktar	3
3. Fördjupningsstudie.....	5
3.1 Kurvanpassning.....	5
3.2 Fläktval.....	7
3.3 Felkällor	9
3.4 Java-Applets	9
4. Fläktväljaren	11
4.1 Gränssnittet.....	11
4.2 Testning	11
4.3 Företagsvinster	12
5. Framtida arbete och slutsats.....	15
6. Referenser	17

1. Inledning

I denna rapport redovisas ett 10 poängs examensarbete i datavetenskap. Arbetets mål har varit att utveckla ett fläktnvalsprogram och anpassa det efter kundens önskemål. Kunden i det här fallet är Fumex AB i Skellefteå.

1.1 Företag och varumärken

Företagsnamn som förekommer i rapporten är, Fumex AB i Skellefteå som kommer att använda sig av det färdiga programmet och Ventur tekniska AB i Göteborg tillverkar fläktarna som Fumex använder i sina installationer. Varumärken som förekommer är, MATLAB, registrerat varumärke av The MathWorks, Inc. samt Java och HotJava, registrerade varumärken av Sun Microsystems, Inc.

1.2 Fumex AB

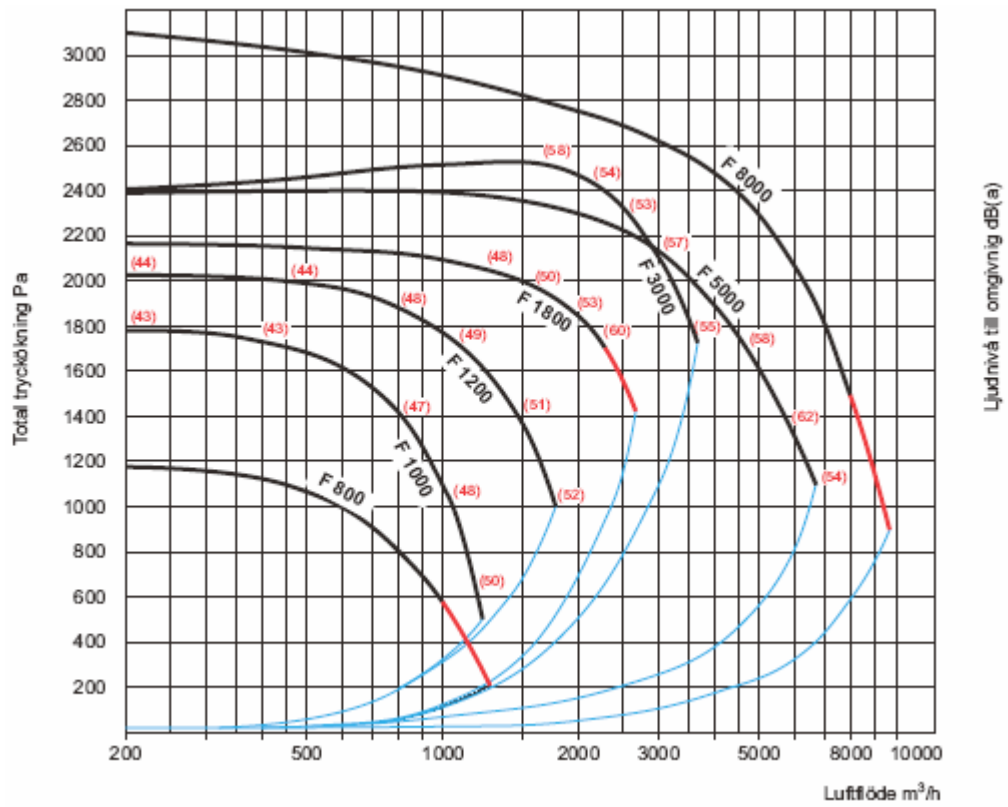
Företaget Fumex AB [1] i Skellefteå har funnits i dryga 60 år och var från början ett plåtslageri. I dag består produktsortimentet från Fumex av lokala utsugsfläktar för stationära arbetsplatser samt ett antal mobila filter för flexibla applikationer. I det omfattande sortimentet finns lösningar för verkstadsindustrin, alla typer av laboratorier samt för utsugning av avgaser från motorfordon. Största marknader utanför Sverige är USA och länderna inom EU. Fläktarna är av typen radialfläkt och tillverkas av Ventur tekniska AB i Göteborg.

1.3 Bakgrund och uppgift

Huvuduppgiften är att utveckla och implementera ett fläktnvalsprogram, för att göra detta måste först ett antal delproblem lösas. Dessa problem beskrivs ingående i kapitel 3.

I dagsläget går det till på följande sätt när en lämplig fläkt skall väljas ut. Man börjar med att mäta upp och beräkna den totala tryckökningen och luftflödet för en lokal. Med hjälp av de två parametrarna går det sedan att ur en graf med fläktarnas prestandakurvor (se figur 1) bestämma vilken av fläktarna som är bäst lämpad för just denna lokal och situation. Fumex har kommit fram till att det skulle underlätta om denna process kunde automatiseras med hjälp av ett fläktnvalsprogram, dels skulle det underlätta för distributörer och leverantörer som inte besitter de specifika kunskaperna om fläktar dels skulle på ett snyggt och professionellt sätt kunna användas vid offertförfrågningar och beställning från kunder.

Till en början var kravspecifikationen ganska vagt formulerad. Kraven som fanns vid examensarbetets början var att med två indatavariabler så skulle en lämplig fläkt väljas och ges som utdata. Vid ett senare möte, cirka fyra veckor in i arbetet, med Lars Hedlund och hans kollega Andreas Hedlund på Fumex framkom det mer konkreta önskemål om funktioner och egenskaper som den färdiga applikationen skulle ha. Dessa lades då till i kravspecifikationen. Tiden att färdigställa arbetet var tio veckor så för att det skulle bli möjligt att hinna färdigt på utsatt tid så prioriterades vissa funktioner bort. Dessa kommer att implementeras vid ett senare tillfälle utanför examensarbetet.

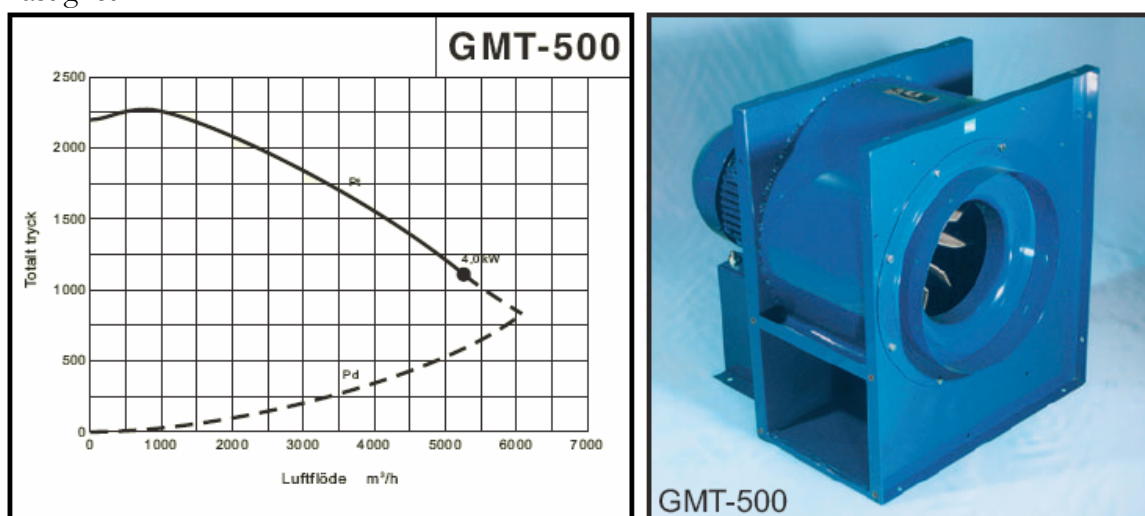


Figur 1. Prestandakurvorna för fläktarna i modellserien GMT som ingår i Venturs produktsortiment [2].

2. Fläktar

Fläktarna som Fumex använder sig av är typen radialfläkt där fläkthjulet är monterat direkt på drivaxeln. Att fläkten är av radial typ innebär att luften kommer in axiellt och lämnar fläkten radiellt. Med axiellt menas att luften kommer in i fläkten längs en tänkt linje rakt in från drivaxeln och lämnar fläkten vinkelrätt mot drivaxeln, radiellt. De är avsedda att transportera luft eller andra gaser i ventilations- och industrianläggningar. Skovlarna på fläkthjulen som används är raka eller framåtböjda.

Till varje fläkt hör en prestandakurva som visar fläktens arbetsområde (se figur 2). För varje prestandakurva finns också en underliggande kurva som visar det dynamiska trycket. Det dynamiska trycket mäts i pascal (Pa) och är lite förenklat uttryckt ”luftens hastighet”.



Figur 2. Prestandakurvan och bild för fläkten GMT-500 från Venturs produktkatalog [2].

För att en fläkt skall kunna väljas ut så måste två variabler bestämmas, luftflöde och total tryckökning. De bestäms med avseende på vad fläkten skall ingå i för applikation, transportera bort avgaser från en anläggning hos bilprovningen eller rök vid en lödstation. Luftflödet beror på fläkthjulets bredd där skovlarna sitter, bredare hjul desto mer luft kan fläkten transportera. Den totala tryckökningen beror på fläkthjulets radie och på rotationshastigheten. Alla fläktar har dock ett maximum som främst har att göra med att motorn som driver fläkten har en varvtalsbegränsning.

3. Fördjupningsstudie

I kapitel 1.3 beskrevs de två variablerna som behövs för att bestämma vilken fläkt som passar bäst. Eftersom det inte fanns några tillgängliga mätdata från fläkttillverkaren [2, 3], då de inte vill ge ut några, kunde data endast inhämtas från produktkatalogen. Genom att ta ut mätpunkter från prestandakurvan för varje fläkt med linjal gick det att få fram tillfredställande värden för x- och y-koordinater för respektive kurva. Prestandakurvorna som är presenterade i produktkatalogen visar endast den del av kurvan som befinner sig i första kvadranten vilket medförde att ett antagande om vilken grad kurvorna är av behövdes tas. Det bestämdes att det skulle räcka om kurvorna approximerades med ett andragsgrads polynom. Fördjupningsstudien består av flera mindre delar. En del har varit att ta fram en modell för kurvorna vilket resulterar i ett linjärt minsta kvadratproblem. Vid omskrivning med normalekvationen med stora decimaltal blev det efter ett antal steg ohållbart att räkna för hand då talen närmade sig 10^{16} . Efter en kortare instuderingsperiod så visade sig MATLAB [7] kunna hjälpa till med att lösa dessa problem. Ett annat delproblem har varit instudering av de grafiska avsnitten i Java. Programmering av Applets (kapitel 3.4) och användandet av grafikpaketen krävde inläsning i ämnet och studerande av olika exempel.

3.1 Kurvanpassning

Grundprincipen är att värdena för luftflödet och totala tryckökningen (som i fortsättningen kommer att kallas för ”punkterna”) ska ligga så nära en fläkts prestandakurva (som i fortsättningen kommer att kallas ”kurvan”) som möjligt. Avståndet mellan alla punkter och kurva ska alltså vara så litet som möjligt. Detta ger upphov till att ett linjärt minsta kvadratproblem måste lösas [4, 5, 6].

Ett linjärt minsta kvadratproblem finner det p som minimerar residualen $r = b - Fp$ med avseende på 2-normen, dvs.

$$\min_p \|b - Fp\|_2$$

där

$$F = \begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_n(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_n(x_2) \\ \vdots & & & \\ f_1(x_m) & f_2(x_m) & \cdots & f_n(x_m) \end{bmatrix}, \quad b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \text{och} \quad p = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}.$$

Matrisen F har fler rader än kolumner, dvs. $m > n$. Ekvationen $Fp \approx b$ är ett så kallat överbestämt ekvationssystem. Funktionen $F(x) = p_1 f_1(x) + p_2 f_2(x) + \dots + p_n f_n(x)$ kallas för modellfunktionen och $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ är de uppmätta mätvärdena.

För att lösa ett linjärt minsta kvadratproblem finns det (i huvudsak) två metoder. Antingen med hjälp av normalekvationen

$$F^T Fx = F^T b$$

där $F^T F$ blir kvadratisk och x har en entydig lösning, eller med QR-faktorisering, se [5, 6]. Att lösa ett minsta kvadratproblem för hand när talen är stora och har flera decimaler blir komplicerat men mest väldigt tidskrävande. Så för att underlätta arbetet används MATLAB¹[7] till att ta fram en tillfredställande approximation för varje kurva. I MATLAB löses problemet lättast med hjälp av \-operatorn vilken använder sig av QR-faktorisering för att ta fram en optimal lösning.

För att ta fram goda approximationer för ekvationerna till varje kurva så valdes modellfunktionen $F(x) = a + bx + cx^2$. Då blir

$$F = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_m & x_m^2 \end{bmatrix} \text{ och } b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

där x_1, \dots, x_m är luftflödet och y_1, \dots, y_m är totala tryckökningen i de m stycken uppmätta mätpunkterna i prestandakurvan. Lösningen

$$p = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

är det som söks.

För att lösa ett minsta kvadratproblem i MATLAB så definieras två vektorer, en med värdena för luftflödet och den andra med värdena för det totala lufttrycket, därefter bildas matrisen F och lösningen $p = [a \ b \ c]^T$ fås med \-operatorn. Här följer ett exempel på hur en modell tas fram i MATLAB för fläkten F5000, givet åtta uppmätta mätpunkter i prestandakurvan.

Bildar vektorn x med luftflödet i varje mätpunkt.

```
>> x = [0 500 100 1500 2000 3000 4000 5000]'
```

Skapar matrisen F .

```
>> F = [ones(8,1) x x.^2]
```

Bildar höger-led med det totala tryckökningen i varje mätpunkt.

```
>> b = [2212.5 2275 2262.5 2175 2087.5 1843.7 1550 1212.5]'
```

Löser minsta kvadratproblemet. Resultaten sparas i vektorn p .

```
>> p = F\b
```

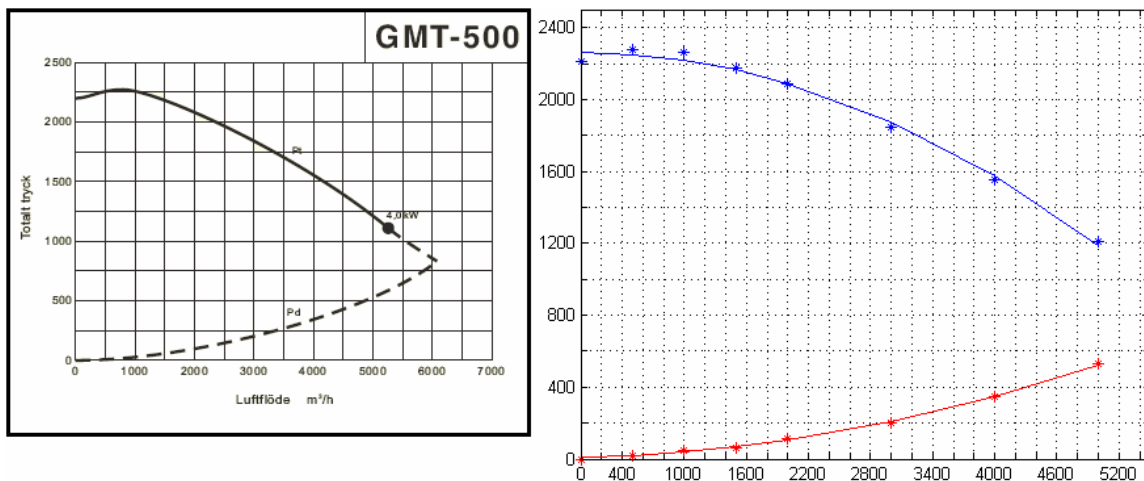
¹ MATLAB är ett programmeringsspråk och program som utvecklades under slutet av 1970-talet och används till matematiska och tekniska beräkningar men även till manipulering av bilder.

$$p = \begin{bmatrix} 2260.44988795080 \\ 0.00012463519 \\ -0.00004290466 \end{bmatrix}$$

Modellfunktionen till den sökta kurvan blir alltså $F(x) = 2260.44988795080 + 0.00012463519x - 0.00004290466x^2$, se även figur 3. På samma sätt har alla prestandakurvor med tillhörande kurvor för dynamiskttryck genererats, se tabell 1.

Fläktmodell Ventur	Fläktmodell Fumex	a	b	c
GMT-80	F800	1178.04	0.0697916	-0.000523958
GMT-100	F1000	1772.05	0.433300	-0.00110946
GMT-130	F1200	2032.42	0.170229	-0.000723705
GMT-200	F1800	2125.29	0.0940487	-0.000269631
GMT-300	F3000	2243.75	0.135257	-0.000150895
GMT-500	F5000	2260.45	0.000124635	-0.0000429047
GMT-700	F8000	3415.77	-0.168336	-0.0000115720
GSF-2-146/62	C600	573.511	0.160985	-0.000588730

Tabell 1. Implementerade fläktmodeller med de beräknade parametrarna a, b och c i modellfunktionen $F(x) = a + bx + cx^2$ angivna med sex värdesiffror.

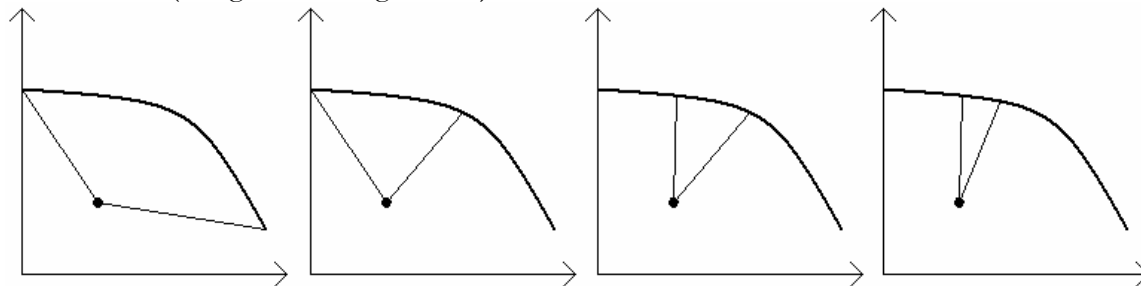


Figur 3. Prestandakurva för fläkten GMT-500 och den beräknade minsta kvadratanpassningen för mätdatat.

3.2 Fläktval

Nästa problem att lösa och det viktigast i hela projektet var att ta fram en algoritm för att bestämma minsta avståndet mellan en punkt, de beräknade värdena, och en kurva, prestandakurvan för en fläkt, som ej var för komplicerad och möjlig att implementera i ett imperativt språk. Att matematiskt bestämma minsta avståndet är på pappret inte svårt men såg ut att bli väl komplicerat för att implementera i kod då det inbegriper första derivatan och hitta rötterna till denna.

Då ekvationerna för varje fläktkurva och tillhörande kurva för det dynamiska trycket är bestämda enligt kapitel 3.1 återstår att utarbeta algoritmen för att finna minsta avståndet mellan en punkt och en kurva. Algoritmen fungerar på följande sätt att från punkten utgår två vektorer till kurvans respektive ändpunkter, längden på dessa beräknas. Vektorernas längder jämförs och den vektor som är längst flyttas till kurvans mitt. Proceduren upprepas och den vektor som är längst flyttas till den kvarvarande delen av kurvans mitt (se figur 4 och algoritm 1).



Figur 4. Kort grafisk beskrivning av algoritmen för att finna minsta avståndet steg för steg. Första steget visas i figuren längst till vänster.

Algoritm 1. Algoritm för att finna avståndet mellan en kurva och en punkt(x,y).

Sätt *vektor1* = längden från punkten p till kurvans vänstra ändpunkt

Sätt *vektor2* = längden från punkten p till kurvans högra ändpunkt

För (antal iterationer)

om (längden av *vektor1* \geq längden av *vektor2*)

sätt x = halva kvarvarande längden av kurvan

sätt y = det värde som kurvans ekvation ger för x -värdet

sätt *vektor1* = längden från punkten p till punkten (x, y) på kurvan

annars (längden av *vektor1* $<$ längden av *vektor2*)

sätt x = till halva kvarvarande längden av kurvan

sätt y = det värde som kurvans ekvation ger för x -värdet

sätt *vektor2* = längden från punkten p till punkten (x, y) på kurvan

Antalet iterationer är genom testning satt till 20 gånger. Då är de tre första decimalerna lika för vänster (*vektor1*) respektive höger (*vektor2*) vektor. Teoretiskt skulle det vara möjligt att iterera ett nästan oändligt antal gånger då det inte finns krav på att applikationen skall vara snabb, men resultatet kommer ej att ändras. I och med att kurvornas x^2 -variabel varierar mellan -0,0011 och -0,0000115, dvs. kurvornas krökning är liten, så approximeras mitten på kurvan till halva x -variabeln i varje steg. Det finns två alternativ i algoritmen beroende på om punkten har sitt minsta avstånd till kurvans vänstra eller högra halva. Det avgörs initialt av vilken vektor som är längst.

Endast de kurvor som uppfyller kravet, att punkten ligger under och inte längre bort än kurvans maximala x -värde. Dessa fungerar i en installation, övriga kurvor sållas bort. De kurvor som uppfyller kraven sorteras med avseende på kortaste avståndet till punkten.

Denna metod att finna minsta avståndet har dock en nackdel i att den bara fungerar för kurvor av andra gradens polynom. Detta medförde att fläkten C600 vars kurva har karakteristiken av ett tredjegradspolynom inte var applicerbar på algoritmen. Då prestandakurvan ligger mycket lägre i y-led än resterande kurvor och inte ingår i några beräkningar som kurvan för det dynamiska trycket gör så har istället kurvan approximerats ett andragradspolynom.

3.3 Felkällor

De felkällor som förekommer beror till stor del av att alla mätningar är gjorda för hand med linjal, inget mätvärde är 100%-igt men målet har hela tiden varit att vara så nära verkligheten som möjligt. Det har varit den mänskliga faktorn som har avgjort och uppskattat mätvärden från mätning till mätning och kan ligga till grund för diskussion. Metoden att ta fram en modell till prestandakurvorna genererar endast en god approximation av de i ett laboratorium uppmätta värden som ligger till grund för kurvan. Det spelar dock inte så stor roll i det här fallet, då avstånden mellan kurvorna i y-led är stora och felmarginalerna inte har någon inverkan på valet av fläkt.

3.4 Java-Applets

Java utvecklades av Sun Microsystems [8] i början av 1990-talet. Målet var att utveckla ett plattformsoberoende programmeringsspråk. För att kunna vara plattformsoberoende så finns en virtuell maskin där all Java-kod kompileras. Det var när Sun introducerade sin webbläsare HotJava i mitten av 1990-talet som Applets[9] började dyka upp. En Applet är ett litet program som kan ingå i en webbsida och köras via en webbläsare som förstår Java eller med hjälp av en Applet Viewer.

4. Fläktväljaren

Resultatet av allt arbete som har lagts ner har utmynnat i en Java-Applet (se figur 5) som tar två inparametrar och sedan presenterar fakta för en lämplig fläkt. Appleten använder sig av algoritm 1 i kapitel 3.2 som beräknar minsta avståndet till en kurva och sedan presenterar fakta. En viktig bit som funnits med i projektet har varit att den slutgiltiga applikationen skulle vara enkel att underhålla och att nya fläktar med lätthet skulle kunna läggas till och tas bort. Detta har tyvärr inte gått att följa fullt ut då en Applet normalt inte får läsa filer från hårddisken och problem skulle kunna uppstå vid distribution av programmet. Därför har istället viss fläktdata blivit hårdkodad. Det har gjort applikationen statisk, men då samma fläktar har använts och kommer att användas i flera år framöver så borde det inte skapa några problem. I ett initialt skede fanns det flera olika idéer om vilket programspråk som skulle användas och hur en färdig applikation kunde implementeras. Php, Java-script, Java, C++ och SQL var några av teknikerna som diskuterades. Då den färdiga applikationen ska fungera utan särskilda tillägg, insticksprogram och användas och distribueras till Fumex distributörer samt att vara enkel att installera så valdes till slut en implementation i Java. Den slutgiltiga applikationen kommer dock att kräva att användaren har Java installerat, minst Java 2 version 1.3.1. Då Java fritt kan laddas ner från Suns hemsida anses det inte finnas några problem med en sådan lösning. Algoritm 1 i kapitel 3.2 för att beräkna minsta avståndet är en egen klass och användargränssnittet är implementerat som en Java-Applet. Huvudklassen Appleten består av, Starter, som innehåller gränssnittet, som i sin tur har tre inre klasser som lyssnar på knapptryckningarna i användargränssnittet. Klassen Starter kommunicerar i sin tur med klasserna CalcDist och DynamicPressure. Klassen CalcDist innehåller algoritmen för att beräkna minsta avståndet mellan punkt och kurva. Dessutom innehåller den viss hårdkodad fläkt data. Klassen DynamicPressure innehåller hårdkodad fläktdata för att beräkna det dynamiska trycket.

4.1 Gränssnittet

Användargränssnittet är implementerat som en Java-Applet och är utformad för att vara så enkel och användarvänlig som möjlig, se figur 5. Tanke med att använda sig av Java är att man kan dra nytta av de fördefinierade objekten/metoder som finns i Java, så som objekt för att ta emot indata och presentera bilder och fakta. Det underlättar även för den ovane datoranvändaren att programmet körs i en webbläsare. Programmet fungerar gemon att först fylls värden för luftflödet och tryckökning i. Användaren väljer sedan om enheten för luftflödet är i m^3/h eller l/s och klickar sedan på knappen FLÄKTVAL! ,se figur 6. Rullistan uppdateras nu med lämpliga fläktar där den översta är bästa alternativet. Användaren väljer önskad fläkt och klickar på knappen VISA INFO! för att få fram tillgänglig data om fläkten, se figur 7.

4.2 Testning

Under hela projektets gång har främst algoritm 1 i kapitel 3.2 testas och provkörts med indatavariabler som ger känd utdata. Algoritmen har även testats emot de grafiska prestandakurvorna. Resterande delar i projektet har testats och provats under arbetets gång. En första version av programmet har även skickats till den externe handledaren på Fumex för att där testats och utvärderas. Visst problem uppkom då flera datorer hos Fumex inte hade stöd för Java, men detta kommer snarast att åtgärdas.

FUMEX®

Luftflöde

m³/h l/s

Total tryckökning (Pa)

▼

Dynamiskt tryck (Pa)

Figur 5. Användargränssnittet vid programstart.

4.3 Företagsvinster

En mindre del har varit att undersöka vilka vinster som Fumex AB gör vid användande av ett fläktvalsprogram. Användandet innebär inga reella vinster tidsmässigt utan att vinsterna ligger i att programmet kommer att användas som ett säljstöd för distributörerna av Fumex produkter. Det ger också en möjlighet att på ett snyggt och professionellt sätt redovisa fakta om fläktar och kan även ingå som underlag i en offert.

FUMEX®

Luftflöde

m³/h l/s

Total tryckökning (Pa)

▼

Dynamiskt tryck (Pa)

Figur 6. Användargränssnittet då användaren har fyllt i uppgifter om luftflöde och tryckökning.

FUMEX®

Luftflöde

m³/h l/s

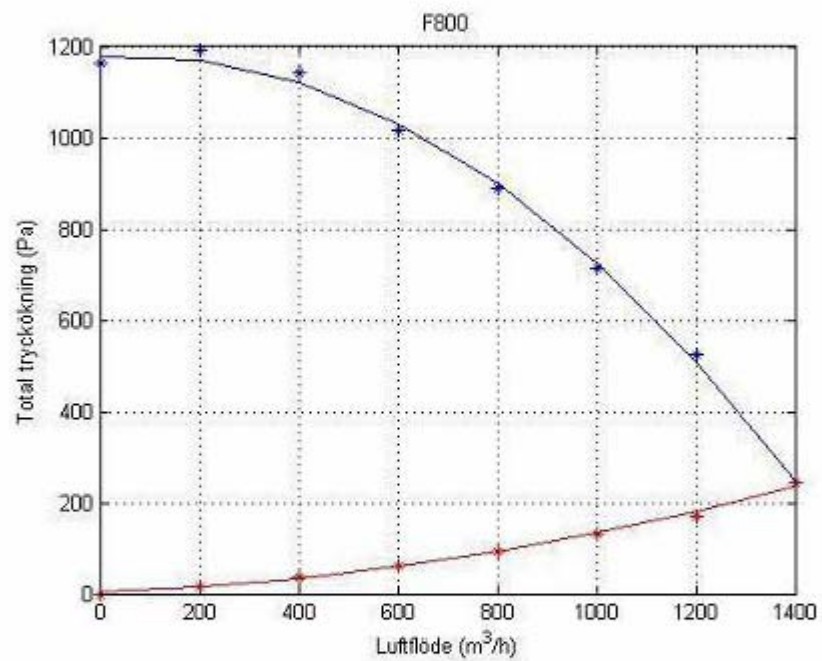
Total tryckökning (Pa)

Fläktval!

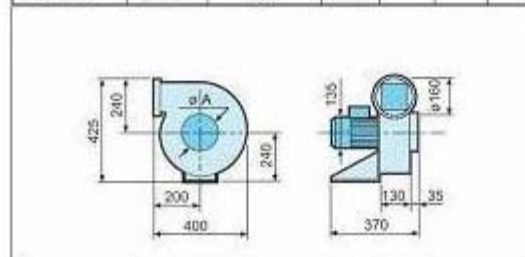
F800 ▼

Visa info!

Dynamiskt tryck (Pa)



Beteckning	Varvtal rpm	Märkeffekt kW	Märkström		Spänning V	Vikt Kg
			A 1-fas 230 V	A 3-fas 230 V 400 V		
F 800 -3	2800	0,37		1,68	230/400	11
F 800 -1	2800	0,37	2,75	0,97	230	12



Figur 7. Användargränssnittet då visa info är valt. Fläktens data och dynamiska tryck presenteras.

5. Framtida arbete och slutsats

Problemspecifikationen var tvungen att bantas ned för att projektet skulle passa in i tidsramen vilket medförde att den slutgiltiga applikationen som är avsedd för examensarbetet inte möter alla krav. Funktioner som ska läggas till är bland annat att implementera ett användargränssnitt för den amerikanska marknaden, där främst enheterna ska konverteras. Ytterligare är att nya bilder på fläktarna i 3-d och ljuddata skall läggas till. Detta är data som i dagsläget inte finns tillgängliga. Fumex vill även att den sökta punkten ska komma upp som en markör i prestandagrafen. Arbetet med att slutföra applikationen med den utökade kravspecifikationen och fortsatt utveckling av den för Fumex räkning kommer att fortsätta inom en snar framtid. Hela projektet har varit oerhört utvecklande och stimulerande då arbetet verkligen har känts på riktigt. Vissa moment under projektets gång tog lite längre tid än förväntat då viss inläsning krävdes och en del testning behövdes göras. I det stora hela så har arbetet varit enormt lärorikt och gett insikter i hur ett program, om än ett litet sådant, kan tas fram och vilka hinder som kan uppkomma.

6. Referenser

- [1] Webbadress till Fumex, <http://www.fumex.com>.
- [2] Ventur. Fläktar 2004, produktkatalog.
- [3] Webbadress till Ventur, <http://www.ventur.se/mainsite.asp>.
- [4] Robert A. Adams. *Calculus: A complete course*. 3:e uppl. Addison-Wesley, 1995.
- [5] Anders Barrlund. *Kompendium i grundläggande Numerisk Analys*. Inst. för datavetenskap, Umeå universitet, 2001.
- [6] Peter Pohl. *Grundkurs i Numeriska Metoder*. Liber, 2005.
- [7] Webbadress till Wikipedia, <http://sv.wikipedia.org/wiki/MATLAB>
Sidan besökt 10/29/2005.
- [8] Webbadress till Java, <http://java.sun.com>.
- [9] Aaron Walsh, Justin Couch och Daniel H. Steinberg . *Java 2 Bibeln*. IDG Books, 2000.