



Norges miljø- og
biovitenskapelige
universitet

Masteroppgave 2016 30 stp
Fakultet for Miljøvitenskap og Teknologi
Institutt for Matematiske Realfag og Teknologi

Praktisk og teoretisk evaluering av regenerativ gassturbinsyklus med ekstern forbrenning

Practical and theoretical evaluation of regenerative
gas turbine cycle with external combustion

Fredrik Nerol Kleven
Maskin, Prosess og Produktutvikling

FORORD

Jeg vil gjerne benytte anledningen til å takke min hovedveileder Odd-Ivar Lekang for tålmodigheten og tiden han har satt av til å hjelpe meg med dette prosjektet og oppgaveskrivingen, og for all tid han har brukt på å hjelpe meg til å vitenskapeliggjøre mine Reodor-Felgen løsninger og metoder.

Arbeidet med denne masteroppgaven har bestått av utallige oppturer og nedturer. Jeg vil rette en stor takk til min far Ole Bjørn Kleven, min bror Knut Gunnar Nerol Kleven, og min søster Åse Margrethe Nerol Kleven som har jublet sammen med meg ved oppturene, og bannet sammen i kor når en ukes arbeid brenner opp foran øynene våre på noen sekunder. Det er alltid en glede å forske/leke med dere selv om ting ikke alltid går som planlagt! Jeg vil også takke min mor Anne Marit Nerol som alltid er der for meg for alle problemene som ikke er av ingeniørmessig art. Men mest av alt vil jeg takke Kjæresten min, som har holdt ut alle de sene kveldene jeg tilbringer på master-rommet, og ukene jeg har tilbragt i Hallingdal for å fullføre forsøkene.

Og en aller siste takk til Pål Brigte Olsen og Åse Margrete! Jeg er imponert over at dere har orket å lese igjennom denne oppgaven på jakt etter de utallige skrivefeilene jeg har klart å pådra meg oppgjennom skrivningen.

Ås, 12.05.2016

.....
Fredrik Nerol Kleven

SAMMENDRAG

Denne oppgaven tar for seg en praktisk og teoretisk tilnærming for å benytte biomasse som energikilde i en gassturbinsyklus. I en verden med et eksponentielt økende energibehov er det stadig viktigere at vi utforsker muligheten til å benytte oss av alle fornybare energikilder som en har tilgjengelig.

Løsningen som evalueres er en regenerativ gassturbinsyklus med eksternt brennkammer som muliggjør forbrenning av biomasse i en gassturbinsyklus.

Oppgaven har to hovedmål:

- Restaurere og videreutvikle en selvbygd prototyp
- Utvikle et simuleringsprogram for gassturbinsykluser for å evaluere løsningen

Prototypen består av et selvkonstruert brennkammer og roterende varmeveksler tilkoblet en turbo brukt som en lavtrykks radial-gassturbin. Prototypen ble restaurert og det ble konstruert to nye brennkammer for propan, samt en 750W direktekoblet generatormodul for testing av elektrisk virkningsgrad med et tilhørende forsøksopplegg. Testingen ble igangsatt men ikke fullført på grunn av problemer med turbinmodulen og brennkammer. Gassturbinen mistet oljetrykket og påfølgende lagerslitasje førte til store vibrasjoner som ville forårsaket skader på andre komponenter hvis forsøket hadde fortsatt. I tillegg ble det avdekket en feildimensjonering av brennkammeret.

Simuleringsprogrammet ble utviklet «langsgående» med det praktiske arbeidet, men ble ikke ferdigstilt før arbeidet med prototypen var over. Etersom forsøket med prototypen ikke lot seg fullføre ble den teoretiske evalueringen av syklusen basert på et innsamlet datasett for en Capstone C30 regenerativ mikroturbin. Programmet ble også brukt til å finne problematiske elementer ved prototypen.

Det blir estimert at en Capstone C30 konfigurert for eksternt forbrenning kan gi mellom 20.5-22.7% elektrisk virkningsgrad med 26-30kw elektrisk effekt levert avhengig av tapene i syklusen. Et slikt system vil ha en produksjonspris på hhv 153 øre/kWh ved forbrenning av pellets, 97 øre/kWh ved forbrenning av flis og -38 øre/kWh ved forbrenning av restavfall. I pris-utregningene er det ikke tatt hensyn til at eksosvarmen kan levere 74 kW til vannbåren varme gjennom et CHP anlegg, som vil redusere energikostnaden ytterligere.

ABSTRACT

This thesis presents a practical and theoretical approach for using biomass as an energy source in a gas turbine cycle. In a world with exponentially growing energy-requirements, it is increasingly important that we explore the possibility to take advantage of all renewable energy sources available.

The solution being evaluated is a regenerative gas turbine cycle with an external combustion chamber, enabling the combustion of biomass in a gas turbine cycle.

The thesis has two main goals:

- Restore and further develop a self-built prototype
- Develop a simulation program for gas turbine cycles to evaluate our solution

The prototype consists of a self-constructed combustion chamber and rotary heat exchanger connected to a turbo used as a low pressure radial gas-turbine. The prototype was restored and it was constructed two new combustion chambers for propane, and a 750W generator module for testing electrical efficiency with a corresponding experimental procedure. Testing was initiated but not completed due to problems with the turbine module and combustion chamber. The gas turbine lost oil pressure and subsequent bearing wear led to major vibrations that would have caused damage to other components if the experiment had continued. In addition, it was discovered a miscalculation in the dimensioning of the combustion chamber.

The simulation program was developed along with the practical work, but was not completed before work on the prototype was finished. Since the experiment with the prototype could not be completed, the theoretical evaluation of the cycle based on a collected data set for a Capstone C30 regenerative micro turbine. The program was also used to find the problematic elements of the prototype.

It is estimated that a Capstone C30 configured for external combustion yield between 20.5 to 22.7% electrical efficiency, with 26-30kw electric power supplied depending on the losses in the cycle. Such a system will have a production cost respectively 18,7 cents / kWh by combustion of pellets, 11.9 cents / kWh by burning wood chips and -4,7 cents / kWh by incineration of garbage. Price calculations are not taking into account that the exhaust heat can deliver 74 kW through hot water in a CHP configuration, which will further reduce the costs of produced energy.

INNHALDSFORTEGNELSE

FORORD	I
SAMMENDRAG.....	III
ABSTRACT	V
FIGUR-LISTE.....	IX
TABELL-LISTE	XI
1. INNLEDNING.....	1
1.1. BAKGRUNN	1
1.2. MÅLSETNING.....	1
1.3. BEGRENSNINGER.....	1
1.4. PROGRAMVARE BRUKT I OPPGAVEN	2
2. GASSTURBIN- LITTERATUR OG TEORI	3
2.1. TERMINOLOGI	3
2.2. SYMBOLER OG ENHETER	4
2.3. FORMLER.....	5
2.4. GASSTURBINEN	7
2.4.1. Aksial luftstrøm	7
2.4.2. Radial Luftstrøm	8
2.4.3. Eroderende Drivstoff	8
2.5. ENKEL BRAYTON-SYKLUS	10
2.5.1. Endring i entropi	11
2.5.2. Ideell virkningsgrad for enkel syklus	12
2.6. REGENERATIV BRAYTON-SYKLUS	13
2.6.1. Ideell virkningsgrad for regenerativ syklus.....	13
2.7. TAP I BRAYTONS VARMESYKLUS.....	15
2.7.1. Kompressor / Turbin virkningsgrad	15
2.7.2. Trykkfall	16
2.7.1. Kompressor og turbinkart	17
2.8. KONSTRUKSJON AV BRENNKAMMER.....	18
2.9. VARMEVEKSLERE OG CHP	19
3. UTVIKLING AV EGEN PROTOTYP – ARES TURBINE.....	21
3.1. TEORETISK VIRKNINGSGRAD REGENERATIV SYKLUS MED EKSTERN FORBRENNING	22
3.2. ORIGINAL PROTOTYP OG TIDLIGERE ARBEID	23
3.2.1. Roterende Varmevexler.....	24
3.3. FORBEREDELSE AV PROTOTYP - EGET ARBEID.....	25
3.3.1. Utgangspunkt	25
3.3.2. Generatormodul	26
3.3.3. Brennkammer.....	28
4. TESTPROSEDYRE PROTOTYPE.	33
4.1. TESTOPPSETT	33
4.1.1. Temperatur.....	33
4.1.2. Trykkmåling og turtall	34
4.1.3. Effekt Ut	35
4.1.4. Tilførsel av drivstoff	35
4.2. FREMGANGSMÅTE TEST.....	36

5.	UTVIKLING OG VERIFISERING AV SIMULERINGSPROGRAM	37
5.1.	METODE PROGRAM.....	38
5.1.1.	Flowchart av fremgangsmåte for simulering	39
5.2.	CASE STUDY 1 – GAS TURB OG VTB	40
5.2.1.	Grunnparameter	40
5.2.2.	Resultat	41
5.3.	CASE STUDY 2 – SSS-15 TURBIN	42
5.3.1.	Grunnparameter	42
5.3.2.	Resultat	43
6.	INNHENTING AV DATA OG SIMULERING AV SYKLUS	45
6.1.	VALG AV SAMMENLINKNINGSTURBIN FOR REGENERATIV SYKLUS.....	45
6.1.1.	Innsamling av data	45
6.2.	SIMULERING AV VARMESYKLUS MED EKSTERN FORBRENNING (ARES TURBIN)	47
7.	RESULTAT	49
7.1.	SIMULERING AV CAPSTONE C30, «BENCHMARKTURBIN»:	49
7.2.	SIMULERING AV ARES TURBIN (EKSTERN FORBRENNING)	50
7.2.1.	Sammenlinkning med «benchmarkturbin»	50
7.2.2.	Simulering med variable parameter.	51
7.3.	EVALUERING AV PROTOTYPE OG TESTPROSEDYRE	54
7.3.1.	Verifisering av virkningsgrad.	54
7.3.1.	Nye komponenter	54
7.3.2.	Testprosedyre.....	54
8.	DISKUSJON	55
8.1.	VURDERING AV EKSTERN FORBRENNING FRA SIMULERING	55
8.2.	SIMULERINGSPROGRAM	57
8.3.	UTVIKLING AV PROTOTYP OG TESTOPPLEGG	58
8.3.1.	Prototyp og brennkammer:	58
8.3.2.	Generator.	58
8.3.3.	Effektmåling – Variabel last	59
9.	VIDERE ARBEID	61
9.1.1.	Prototype og testprosedyre.....	61
9.1.2.	Program og simulering.....	61
10.	KONKLUSJON	63
11.	REFERANSER.....	65
	VEDLEGGSLISTE	69

Figurliste

Figur 2.1 Typisk aksialturbin [3]	7
Figur 2.2 Typisk radialturbin [4, p. 3]	8
Figur 2.3 Damp turbin syklus, damp turbinen går i en lukket syklus der all damp kondenseres og gjenbrukes i syklusen [6]	9
Figur 2.4 Pris ekskludert MVA på forskjellige energikilder uavhengig av virkningsgrad ved forbrenning, *Vi har ikke funnet norske data på kullpris, og bruker derfor europeisk spotpris uten frakt-påslag. [8] [9] [10] [11] [12] Utregninger i Vedlegg D	9
Figur 2.5 trykk/volum (PV) og temperatur/entropi (TS) diagram av en brayton varmesyklus [19]	10
Figur 2.6 Forenklet skisse av en enkel syklus gassturbin [8]	10
Figur 2.7 Virkningsgrad i forhold til kompresjonsforhold i en gassturbin. Basert på: [3, p. 60]	12
Figur 2.8 Chrystler gassturbin med roterende varmeveksler [5]	13
Figur 2.9 Turbec gassturbin med stasjonær varmeveksler [19]	13
Figur 2.10 Temperatur/entalpi diagram for gassturbin med varmeveksler, [5, p. 93]	13
Figur 2.11 Forenklet skisse av gassturbin med regenerator [5, p. 93]	13
Figur 2.12 Virkningsgraden til en regenerativ varmesyklus i forhold til kompresjonsforholdet r_p ved flere temperaturforhold t. [20, p. 41]	14
Figur 2.13 Tap i syklusen grunnet virkningsgrad til kompressor og turbin i et forenklet PV-TS diagram hvor $\eta_c = \eta_t = 100\%$ (heltrukket), 90%, 80%, 70% basert på [22, p. 104]	15
Figur 2.14 Trykkfallets påvirkning på PV-TS diagrammet. trykkfallet minker energien tatt ut over turbin, og hever dermed utløpstemperaturen. Her vises et trykktap på hhv 0% (Heltrukket) 10, 20 og 30%. basert på [24, p. 181]	16
Figur 2.15 kompressor-virkningsgradskart [25]	17
Figur 2.16 Forenklet tubulært diffusjonsbrennkammer	18
Figur 2.17 Skjematisk oppstilling av varmeveksler, varmeveksleren har en varmside T_h , hvor varm luft strømmer inn, og en kaldside T_c hvor kald luft strømmer inn. Q er energien overført fra varmside til kaldside.	19
Figur 3.1 Forenklet skisse av gassturbin med regenerativ syklus og eksternt brennkammer. Basert på [16, p. 93]	21
Figur 3.2 PV-TS diagram for ideell regenerativ varmesyklus med ekstern forbrenning. Stiplet linje viser overføring av varme via varmeveksler	22
Figur 3.3 Første prototype med pellets-brennkammer [31]	23
Figur 3.4 Konseptskisse av Ares Turbinen med tall-henvisninger til Figur 3.1	23
Figur 3.5 Roterende varmeveksler, designet og produsert av Ares Turbine AS [12]	24
Figur 3.6 Spyling av roterende varmeveksler. Bildet T.H viser nærbilde av spyleprosessen på varmsiden av varmeveksleren	24
Figur 3.7 Turbinens tilstand da den ble gravd frem på låven	25
Figur 3.8 Tversnitt av generatormodul.	26
Figur 3.9 Printing og utprøving av de konstruerte delene.	26
Figur 3.10 Detalj bilde av oppspenning for generator og maskinering av ytterrør.	27
Figur 3.11 Generatormodul montert på gassturbin for testing	27
Figur 3.12 Første forsøk med katalytisk brennkammer. Katalysatoren ble forvarmet med en propanflamme og propan tilført gjennom et hull i messingrøret. Kompressoren ble brukt som luftkilde	28
Figur 3.13 Konstruksjon av katalytisk brennkammer med radiale luftinntak for å sikre at en uniform gjennomstrømming i substratet.	29
Figur 3.14 Test av katalytisk brennkammer og bilder av oppbrent katalysator.	29
Figur 3.15 Forenklet CFD analyse med standard mesh, 1.5kg/min volumstrøm og 2 bars mottrykk uten varmetilførsel, ettersom Autocad ikke støtter simulering av forbrenning	30
Figur 3.16 Maskinering av flammerør og bakplate i brennkammeret	31
Figur 3.17 Modifisering av tennplugg for enkel tenning med propantilførsel gjennom perforert kobberør. Komplette flammerør ses i bildet T.H	31

Figur 3.18 Utpøving av brennkammer. Det kan observeres at flammen er sentrert i flammerøret og ikke berører ytterveggene i flammerøret.	32
Figur 3.19 Turbo kjørt som enkelsyklus gassturbin.	32
Figur 4.1 Skjematisk testoppsett av gassturbin basert på [16, p. 93] brukt i Figur 3.1	33
Figur 4.2 Tilkobling av K elementer på varmeveksler. Benetech GM1312, instrumentet har to innganger og kan avlese J, K, T, E, N, R og S elementer med 1,5% nøyaktighet.	34
Figur 4.3 Analogt RaceTech Manometer montert på kontrollpanelet til gassturbin. Og Garrett turteller via induktiv måler i kompressorhuset.	34
Figur 4.4 Detaljert kretsdiagram effektmåling, utregninger finnes i vedlegg A	35
Figur 4.5 Rendring av variabel last (reostat).	35
Figur 5.1 Turbin med ekstern kraft-turbin. Separeringen av de to turbinene lar de rotere uavhengig av hverandre og med forskjellig hastighet avhengig av vinkel på turbinbladene. [34]	40
Figur 5.2 Skjematisk oppsett av SSS-15 enkelsyklus turbin	42
Figur 6.1 Capstone C30. gassturbin. Komplette enhet med varmeveksler og brennkammer montert.. [36]	45
Figur 6.2 Capstone C30 turbinkjerne demontert fra turbin vist i Figur 6.1 [37]	46
Figur 6.3 Sammenlikning av de to syklusene, Capstone-turbinen T.V og Ares turbinen T.H	47
Figur 7.1 PV-TS diagram for Capstone C30	49
Figur 7.2 PV-TS diagram for regenerativ syklus med ekstern forbrenning	50
Figur 7.3 Effekt ut og virkningsgrad for forskjellige kompresjonsforhold ved fast innløpstemperatur på 871 °C, stiplede linje er Capstone Turbin mens heltrukne linje er Ares Turbin.	51
Figur 7.4 Konturplott av hvordan effekt ut og virkningsgrad forandrer seg ved variasjon i trykkforhold og temperatur, stiplede røde linje er maksimal virkningsgrad, og stiplede svarte linje er maksimal effekt ut	52
Figur 7.5 Variasjon i effekt ut og elektrisk virkningsgrad ved forandring i virkningsgrad varmeveksler og trykkforhold. Stiplede røde linje er optimalt trykkforhold for en gitt virkningsgrad for varmeveksler.	52
Figur 7.6 Påvirkning av trykktap i brennkammer og varmeveksler. Alle andre parameter holdes konstant.	53
Figur 7.7 Effekt ut og total virkningsgrad i forhold til variasjon i virkningsgrad til kompressor og turbin. Alle andre parameter holdes konstant.	53
Figur 8.1 Pris per kwh strøm produsert med Ares Turbin ved 20,5% el virkningsgrad, mot capstoneturbin ved 26% el virkningsgrad [8] [9] [10] [11] [12] Utregninger i Vedlegg D	56
Figur 8.2 Forslag til modifisering av last	59

Tabell Liste

<i>Tabell 3.1 Dimensjonering av flammerør for å fordele luftstrøm i brennkammer</i>	30
<i>Tabell 5.1 Komponent og funksjonsnavn i analysebiblioteket.</i>	38
<i>Tabell 5.2 [32, p. 51] Grunnparameter spesifisert for simulering tallverdiene i «Komponent» henviser til Figur 5.1</i>	40
<i>Tabell 5.3 [32, p. 51] Etter simuleringen ble resultatene slått sammen med tabellen gitt i kilden. Simuleringen med vårt program er markert med grå bakgrunn i raden benevnt som Ares Turb. De uthevede verdiene er variable parametere programmet kalkulerer for å tilfredsstille grunnparameterne.</i>	41
<i>Tabell 5.4 basert på [35, p. 3] Grunnparametre spesifisert for SSS-15 turbin. Programmet vil kalkulere effekt ut basert på de opplyste verdiene. Tallhenvisninger til Figur 5.2</i>	42
<i>Tabell 5.5 Resultat fra simulering med tallhenvisning til Figur 5.2.</i>	43
<i>Tabell 6.1 Innsamlet data for capstone C30 fra manual, artikler og lærebøker.</i>	46
<i>Tabell 6.2 Parameterintervaller for komponenter hvor vi mangler data.</i>	46
<i>Tabell 7.1 Komplette datasett for «benchmarkturbin» basert på innsamlede data i Tabell 6.1 og egne simuleringer.</i>	49
<i>Tabell 2 Utreknet datasett for Ares Turbin med parametere fra simulering av Capstone Turbin</i>	50

1. Innledning

1.1. Bakgrunn

Gassturbinen er et gammelt konsept kjent allerede 50 år etter Kristi fødsel som Heros motor. Likevel skulle det gå nesten 2000 år før det ble konstruert en gassturbin som produserte mer energi enn den brukte internt. For mange er det en overraskelse at denne bragden er utført av en nordmann med navn Jens William Ægadius Elling i 1903. Han la med dette grunnlaget for den videre utviklingen av gassturbinen slik vi kjenner den i dag. [1, p. 70]

Fra den første gassturbinen produsert til dagens systemer har det vært svært store fremskritt i metallurgi og optimalisering av komponentene som har økt virkningsgrad og kraftproduksjon betraktelig. Men felles for alle småskala gassturbiner i dag, er at de fortsatt benytter de samme raffinerte og ikke fornybare energikildene som Ægadius Elling brukte i sin første gassturbin. Det finnes alternativer som bioetanol, biodiesel og bio-olje, men disse har fortsatt ikke nådd et prisnivå som kan konkurrere med ikkefornybare kilder. [2, p. 7]

Dette betyr at en strømproduserende gassturbin som går på fornybare kilder må kunne benytte et brennstoff som er konkurransedyktig på pris, i tillegg til å ha en tilstrekkelig høy virkningsgrad for å opprettholde prisfordelen.

1.2. Målsetning

Denne oppgaven vil i hovedsak ta for seg en gassturbin som kan produsere elektrisitet fra fast biomasse og vil bestå av to hovedmål:

- Restaurere og videreutvikle eksisterende prototype for testing av elektrisk virkningsgrad
 - o Restaurere eksisterende prototype
 - o Konstruere generator for gassturbin
 - o Opprette testopplegg for elektrisk virkningsgrad
- Utvikle et program for simulering av gassturbin sykluser
 - o Evaluere regenerativ Brayton-syklus med ekstern forbrenning

1.3. Begrensninger

Grunnet økonomiske hindringer som oppsto ved årsskiftet har vi satt følgende begrensninger på konstruksjon og forsøk.

- Produksjon foregår i eget verksted med materialer og verktøy som vi har tilgjengelig
- Tilgang på måleapparater er begrenset til utstyr vi har tilgjengelig.

1.4. Programvare brukt i oppgaven

Cad tegning og design:

- Solidworks 2013 – 3D printede deler
- Autodesk Inventor Professional 2016 Student Version, Build 138

Simulering:

- Autodesk CFD 2016, V16.0

3D- Printing:

Slicer 1.1.7

Printrun 1.10.19

Grafisk design:

- Inkscape 0.91
- Gimp 2.8.10

Programmering:

- Spyder 2.3.8 via Continium Anaconda distribusjon.

Innhenting av materialdatasett:

- CES edupack 2013 V.12.2.13

2. Gassturbin- Litteratur og Teori

2.1. Terminologi

Forkortelse	Beskrivelse
CHP	Combined Heat and Power
TIT	Turbine Inlet Temperature
CFD	Computer Fluid Dynamics
MVA	Merverdiavgift
TS	Temperatur- Entropi
PV	Trykk Volum
RPM	Omdreining per minutt
HE, H-E	Varmeveksler
CO	Karbonmonoksid
T.H / T.V	Til høyre/venstre
HHV	Henholdsvis
V	Volt
A	Ampere

2.2. Symboler og Enheter

Symbol	Beskrivelse	Enhet
η	Virkningsgrad	%
c_p	Spesifikk varmekapasitans 1.005 for luft	$\frac{kJ}{kg K}$
$T_{\#}$	Temperatur ved gitt nivå	$^{\circ}K$ eller $^{\circ}C$
γ	Adiabatisk eksponent.	c_p/c_v
r_p	Kompresjonsforhold	p_1/p_0
q Q	Varmeenergi	Joule
t	Temperaturforhold	T_2/T_1
$p_{\#}$	Trykk ved forskjellige nivåer	N/m^2
Δs	Endring i entropi	$\frac{kJ}{kg K}$
R	Ideell gasskonstant 0.28705 for luft	$\frac{kJ}{kg K}$
f_d	Dimensjonsløs faktor for friksjon i rør	
e	Ruhet i overflate	mm
D	Diameter	mm
Re	Reynolds Tall	
T_h, T_c	Temprturnivå i varmeveksler varmside og kaldside	$^{\circ}K$ eller $^{\circ}C$
g	Tyngdekraftskonstanten , 9.81	m/s^2

2.3. Formler

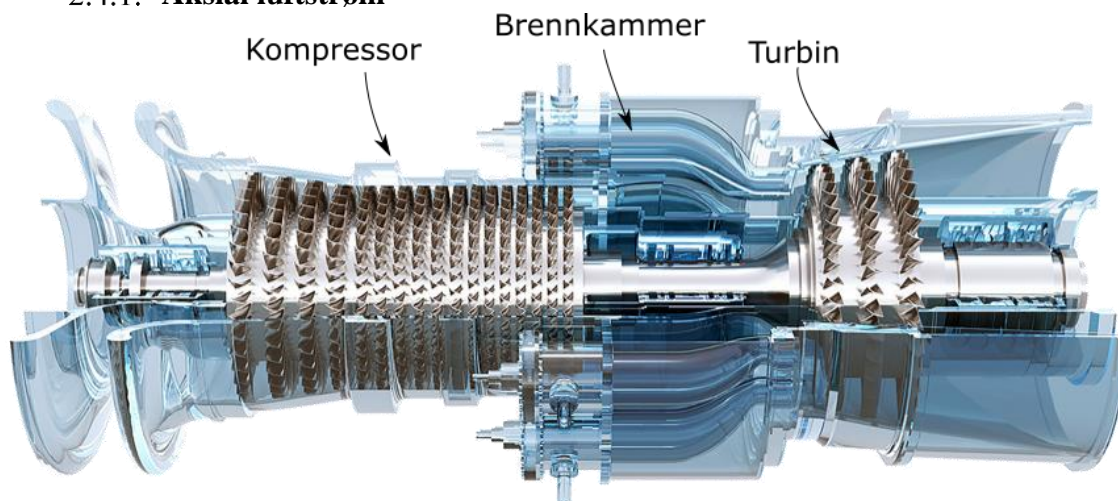
Nummer	Beskrivelse	Formel
Formel 2.1	Isentropisk varmeøkning fra trykkforandring	$T_0 + T_0 * \left(\left(\frac{p_1}{p_0} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right)$
Formel 2.2	Volumøkning fra forandring i trykk og temperatur	$V_1 = V_0 \frac{p_0}{p_1} \frac{T_1}{T_0}$
Formel 2.3	Energi brukt på oppvarming av gass	$\Delta q = Cp \Delta T = Cp * (T_1 - T_0)$
Formel 2.4	Forandring av entalpi	$\Delta s = c_p \log_e \frac{T_2}{T_1} - R \log_e \frac{p_2}{p_1}$
Formel 2.5	Beregning av virkningsgrad fra endring i temperatur i en enkel Brayton syklus	$\eta = \frac{c_p(T_3 - T_4) - c_p(T_2 - T_1)}{c_p(T_3 - T_2)}$
Formel 2.6	Beregning av virkningsgrad fra kompresjonsforhold i en enkel Brayton syklus	$\eta = 1 - \left(\frac{1}{r_p} \right)^{(\gamma-1)/\gamma}$
Formel 2.7	Beregning av virkningsgrad fra endring i temperatur i en regenerativ Brayton syklus	$\eta = \frac{c_p(T_4 - T_5) - c_p(T_2 - T_1)}{c_p(T_4 - T_3)}$
Formel 2.8	Beregning av virkningsgrad fra kompresjonsforhold og temperaturforhold i varmeveksler i en regenerativ Brayton-syklus	$\eta = 1 - \frac{r_p^{(\gamma-1)/\gamma}}{t} = 1 - \frac{r_p^{0,283}}{t}$
Formel 2.9	Isentropisk varmeøkning grunnet virkningsgrad på kompressor/turbin	$\Delta T_{tap} = \frac{T_0}{\eta} \left(\left(\frac{P_1}{P_0} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right)$
Formel 2.10	Trykktap i rør	$\Delta P = f_d \frac{L}{D} \frac{\rho V^2}{2g}$
Formel 2.11	Dimensjonsløs tapsfaktor for rørtap	$f_d = \frac{0.25}{\log \left(\frac{e}{3.7 D} + \frac{5.74}{Re^{0.9}} \right)^2}$
Formel 2.12	Utregning av reynolds tall	$Re = \frac{\rho V D}{\mu}$

Formel 2.13	Virkningsgrad for varmeveksler	$\eta = \frac{Q_{reell}}{Q_{max}}$
Formel 2.14	Energi overført i varmeveksler basert på temperatur inn/ut	$Q_{reell} = \eta C_p (T_{h1} - T_{c1})$
Formel 2.15	ΔT i varmeveksler	$\Delta T = \frac{Q_{reell}}{\eta C_p}$
Formel 3.1	Virkningsgrad for regenerativ syklus med ekstern forbrenning	$\eta = \frac{c_p(T_3 - T_4) - c_p(T_2 - T_1)}{c_p(T_5 - T_4)} =$

2.4. Gassturbinen

Felles for alle gassturbiner er tre hovedkomponenter; en kompressor, en turbin, og et brennkammer. Oppbygningen av disse kan være svært forskjellig fra turbin til turbin, men grunnprinsippet er det samme for alle: kompressoren komprimerer luften, før den tilsettes drivstoff som forbrennes i brennkammeret, og til slutt ekspanderer luften gjennom turbinen hvor energien hentes ut. Dette forklares mer inngående i Kap. 2.5.

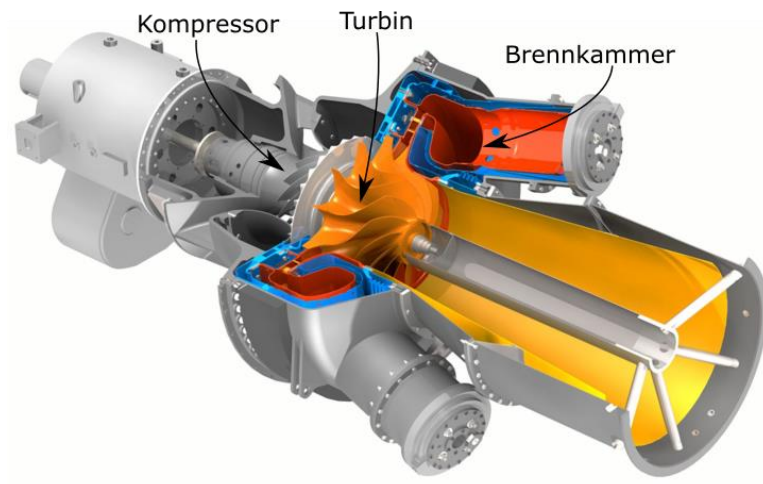
2.4.1. Aksial luftstrøm



Figur 2.1 Typisk aksialturbin [3]

Aksial luftstrøm er vanligst i store gassturbiner. Dette kommer av at flere kompressortrinn kan plasseres i serie, og man oppnår et mye høyere kompresjonsforhold enn ved ett enkelt trinn, i tillegg kan turbinbladene lages hule slik at de kjøles ned under drift og dermed tåle høyere temperaturer ut fra brennkammeret. Generelt sett er disse gassturbine dyrere enn radialturbiner, ettersom de krever betraktelig mer maskinering på de mange kompressortrinnene.

2.4.2. Radial Luftstrøm



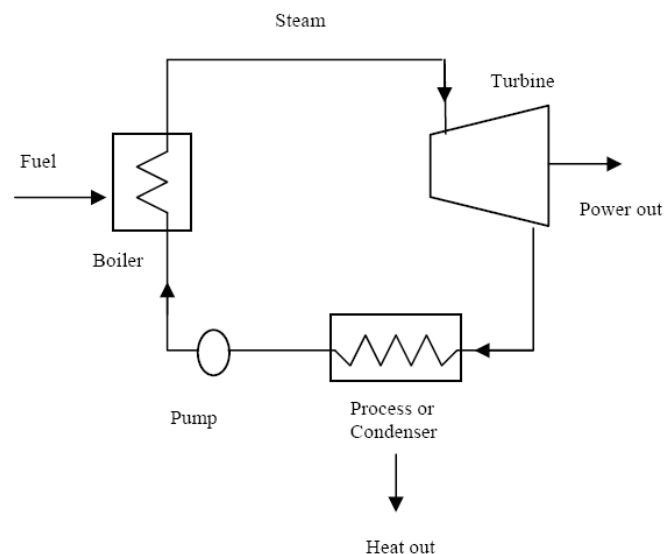
Figur 2.2 Typisk radialturbin [4, p. 3]

Turbiner med radial luftstrøm er vanlig i kompakte gassturbiner under 1000kw. Fordelen med radialturbiner er at de har færre komponenter i kompressoren, og man kan oppnå et akseptabelt kompresjonsforhold med ett enkelt kompressortrinn. Generelt består kompressoren og turbinen av en enkelt maskinert del. Ulempen er at det er krevende å seriekoble flere kompressorer. Det er derfor vanskelig å oppnå kompresjonsforhold som kan konkurrere med aksialturbiner.

2.4.3. Eroderende Drivstoff

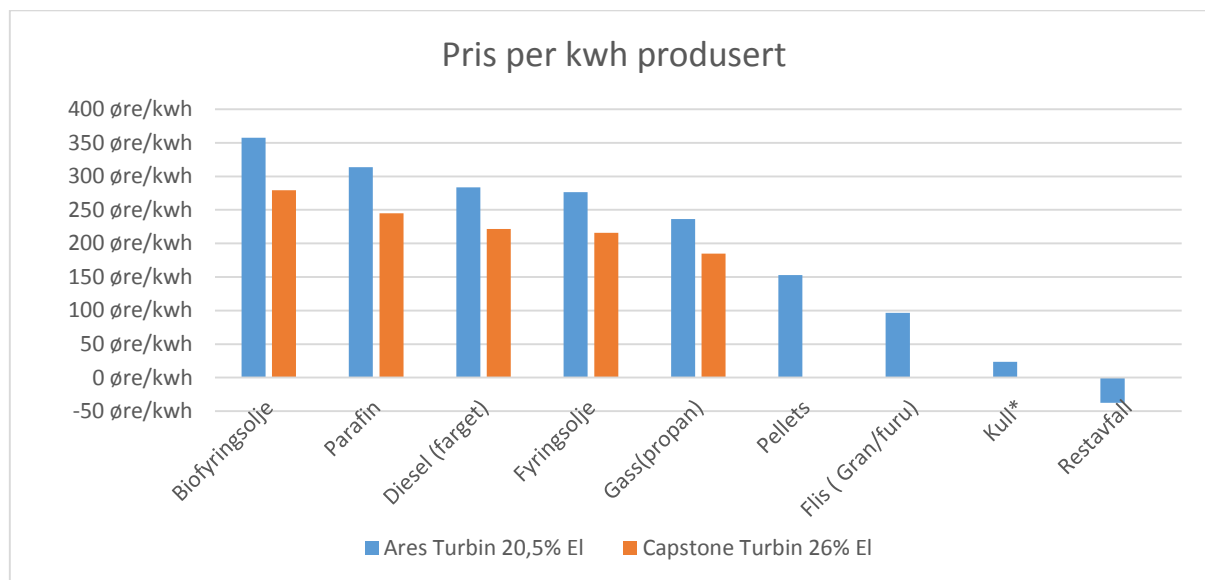
Gassturbinen er en svært allsidige forbrenningsmotor, og kan designes til å forbrenne tilnærmet alle flytende drivstoff og brennbare gasser såfremt de har et lavt askeinnhold [5, p. 436]. Problemet med drivstoff som produserer større mengder askepartikler under forbrenningen opptrer først når askepartikler føres gjennom turbinen. Bladene roterer nær lydens hastighet, og en kontinuerlig strøm av aske gjennom turbinen vil raskt forårsake erosjon.

Den vanlige løsningen for å forbrenne drivstoff med høyt askeinnhold er å bytte ut kompressoren med en dampkjele. I stedet for å komprimere luften for deretter å tilføre energi i et brennkammer, varmes vann opp til damp via en varmeveksler. Energien tas ut i en turbin før dampen nedkjøles, slik det er vist i Figur 2.3. Ulempen med denne prosessen er at dampen må kjøles ned til under 100°C før den kan gjenbrukes i varmesyklusen. Dette krever et stort vannreservoar, i tillegg til et kjøleanlegg som kondenserer dampen tilnærmet like raskt som den varmes opp, og gjør dampturbiner lite egnet for småskala-systemer.



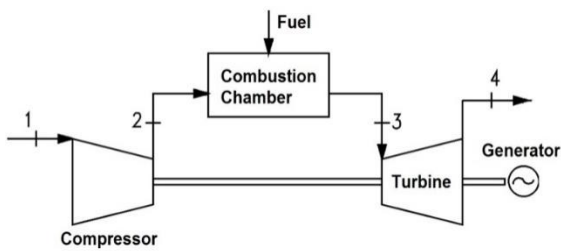
Figur 2.3 Dampturbin syklus, dampturbinen går i en lukket syklus der all damp kondenseres og gjenbrukes i syklusen [6]

Den virkelige fordelen med partikulært drivstoff som kull, restavfall, pellets og flis er at det er langt billigere enn drivstoff med høyere raffineringsgrad slik vi kan se fra Figur 2.4. Fra det samme diagrammet ser vi også årsaken til at størstedelen av verdens strømproduksjon fortsatt foregår ved hjelp av kullkraft. [7]

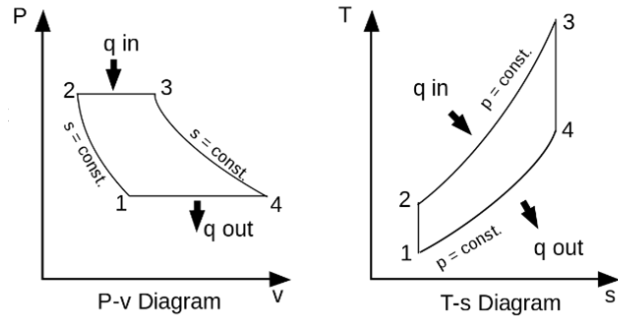


Figur 2.4 Pris ekskludert MVA på forskjellige energikilder uavhengig av virkningsgrad ved forbrenning, *Vi har ikke funnet norske data på kullpris, og bruker derfor europeisk spotpris uten frakt-påslag. [8] [9] [10] [11] [12] Utregninger i Vedlegg D

2.5. Enkel Brayton-syklus



Figur 2.6 Forenklet skisse av en enkel syklus gassturbin [8]



Figur 2.5 trykk/volum (PV) og temperatur/entropi (TS) diagram av en brayton varmesyklus [19]

En ordinær gassturbin består av tre hovedkomponenter: en kompressor, en turbin, og et brennkammer. For å illustrere funksjonen til hver av disse komponentene bruker vi Braytons varmesyklus fra Figur 2.5, og gjennomgår de fire stegene som ideelle prosesser med henvisning til nivåene i Figur 2.6 og Figur 2.5.

1 → 2 Isentropisk kompresjon: Kompressoren øker lufttrykket i systemet med 2-15 bar avhengig av kompresjonsforholdet. Med kompresjonen følger en økning av lufttemperaturen. [5, p. 59]. Gjennom trykkforandringen vil temperaturen følge formelen for isentropisk kompresjon:

$$T_0 + T_0 * \left(\left(\frac{p_1}{p_0} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right) \quad \text{Formel 2.1}$$

[13, p. 564]

Mens den respektive endringen av volum vil følge endringen i temperatur og trykk ved:

$$V_1 = V_0 \frac{p_0 T_1}{p_1 T_0} \quad \text{Formel 2.2}$$

[14, p. 227]

Energien som kreves for å komprimere gassen til det spesifikke trykket kan enkelt utregnes fra temperaturøkningen til gassen gjennom kompresjonen:

$$\Delta q = Cp \Delta T = Cp * (T_1 - T_0) \quad \text{Formel 2.3}$$

[15, p. 581]

Denne energien er det totale energiforbruket, og tilsvarer all energi som har gått med til kompresjonen. Dette inkluderer også energi som har gått tapt i mekanisk oppvarming av gassen grunnet friksjon og dårlig virkningsgrad på kompressoren hvis prosessen ikke er ideell.

2→3 Isobar oppvarming: Komprimert luft ledes inn i brennkammeret og tilføres drivstoff som antennes. Varmen av forbrenningen fører til en volumøkning under tilnærmet konstant trykk, med en påfølgende økning av hastigheten på luftstrømmen. [5, p. 59] Volumforandringen kan utregnes med *Formel 2.2* hvor $p_0 \approx p_1$. Energien som går med til oppvarmingen av gassen utregnes ved *Formel 2.3*

3→4 Isentropisk ekspansjon: Den oppvarmede luften ekspanderer gjennom turbinen. Denne prosessen frigir en andel av energien som ble tilført i brennkammeret. Ettersom turbinen er koblet sammen med kompressoren, vil 55-60% av energien [16, p. 53] gå med til å drive kompressoren.

Den overskytende energien kan tas ut som elektrisitet ved hjelp av en generator. [5, p. 59] Temperatur og volumendring og energiuttak over turbinen kan utregnes ved det samme formelverket som for isentropisk ekspansjon (*Formel 2.1*, *Formel 2.2* og *Formel 2.3*)

4→1 Isobar varmeoverføring: Resterende energi (*Formel 2.3*) frigis i atmosfæren og tilsvarer tapet i Brayton-syklusen. Volum og temperatur returnerer til utgangspunktet. [5, p. 59]

2.5.1. Endring i entropi

Det eneste som mangler for å lage TS diagrammet for denne syklusen er entropinivået som kan utregnes fra forandringen i temperatur og trykk:

$$\Delta s = c_p \log_e \frac{T_2}{T_1} - R \log_e \frac{p_2}{p_1} \quad \text{Formel 2.4}$$

[17, p. 264]

Δs er forandringen i entropi, c_p er termisk varmekapasitet, $T_{\#}$ er temperaturnivå, R er gasskonstanten, $p_{\#}$ er trykket.

2.5.2. Ideell virkningsgrad for enkel syklus

Virkningsgraden til den ideelle Brayton varmesyklusen kan utregnes fra arbeidet som kreves for å øke temperaturen på gassen i Ts diagrammet (Figur 2.5). Ved å anta at virkningsgraden til den totale syklusen er arbeid ut, delt på varme tilført, samt at den spesifikke varmekapasiteten til gassen c_p er konstant gjennom syklusen, vil vi få den forenklete formelen:

$$\eta = \frac{\text{Arbeid ut}}{\text{Varme inn}} = \frac{q_t - q_c}{q_b} = \frac{c_p(T_3 - T_4) - c_p(T_2 - T_1)}{c_p(T_3 - T_2)} \quad \text{Formel 2.5}$$

$$= \frac{(T_3 - T_4) - (T_2 - T_1)}{T_3 - T_2} \quad [18, \text{ p. } 39]$$

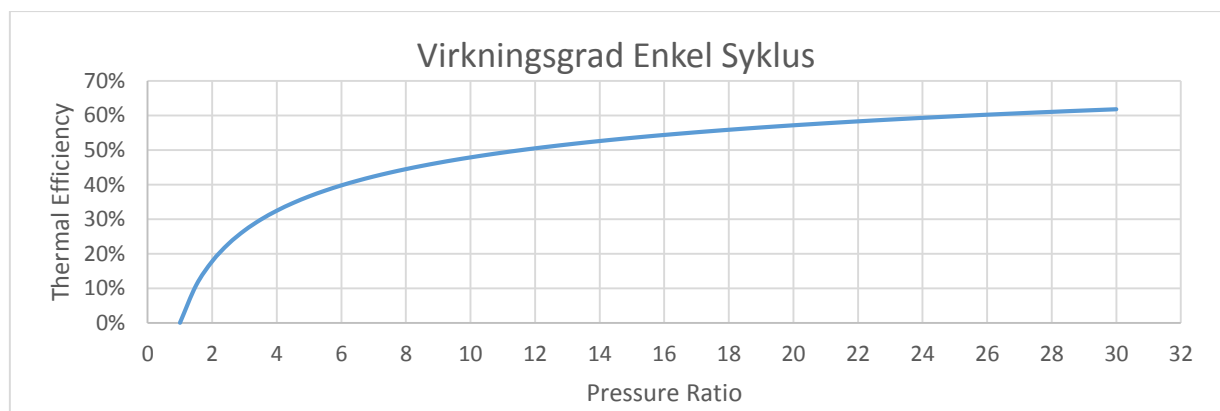
Hvor q_t er energien tatt ut i turbinen, minus q_c energien brukt til å komprimere gassen, og q_b er energien tilført i brennkammeret.

Uten å beskrive matematikken inngående, vil det isentropiske forholdet mellom trykk og temperatur i Braytons ideelle varmesyklus føre til at den totale virkningsgraden kan utregnes ved hjelp av kompresjonsforholdet i kompressoren alene [18, pp. 38-40]:

$$\eta = 1 - \left(\frac{1}{r_p}\right)^{\frac{\gamma-1}{\gamma}} = 1 - \left(\frac{1}{r_p}\right)^{0,283} \quad \text{Formel 2.6}$$

[18, p. 39]

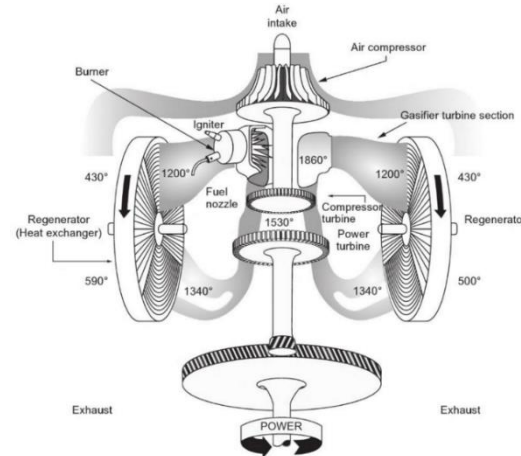
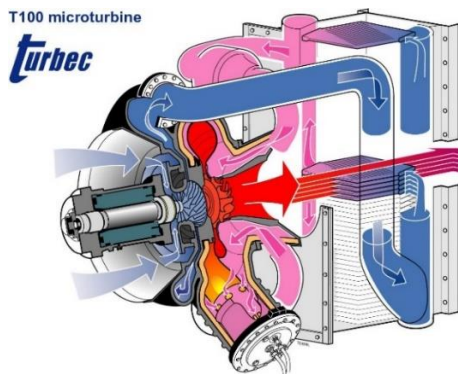
Dette betyr at virkningsgraden til en ideell enkelsyklus gassturbin kun er avhengig av kompresjonsforholdet. Denne trenden kommer klart frem av plottet mellom virkningsgraden og kompresjonsforholdet vist i Figur 2.7, og er årsaken til at produsenter av enkel-syklus gassturbiner etterstreber så høyt kompresjonsforhold som mulig.



Figur 2.7 Virkningsgrad i forhold til kompresjonsforhold i en gassturbin. Basert på: [3, p. 60]

2.6. Regenerativ Brayton-syklus

I realiteten vil det ikke alltid være mulig å oppnå et ettertraktet kompresjonsforhold. Trykket generert i kompressoren er proporsjonalt med periferihastigheten på kompressorbladene. På mindre gassturbiner er diameteren på kompressoren så liten at den må rotere med flere hundre tusen RPM for å oppnå et trykk tilsvarende større gassturbiner oppnår med bare et par tusen RPM.

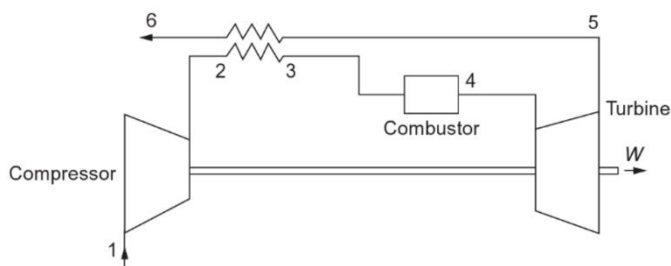


Figur 2.9 Turbec gassturbin med stasjonær varmeveksler [19] Figur 2.8 Chrysler gassturbin med roterende varmeveksler [5]

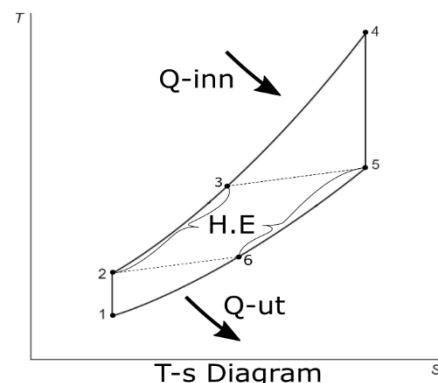
For å kompensere for det lave kompresjonsforholdet i mindre gassturbiner, er det vanlig å gjenbruke eksosvarmen fra turbinen som ellers ville gått til spille, og dermed oppnå en høyere virkningsgrad. Dette gjøres som regel ved hjelp av en stasjonær varmeveksler (Figur 2.9), men kan også gjøres med en roterende varmeveksler (Figur 2.8).

2.6.1. Ideell virkningsgrad for regenerativ syklus

Oppbygningen av en regenerativ gassturbin er tilnærmet lik en ordinær turbin. Hovedforskjellen er varmeveksleren som varmer opp luftstrømmen mellom kompressor og brennkammer ved hjelp av eksos fra turbinen.



Figur 2.11 Forenklet skisse av gassturbin med regenerator [5, p. 93]



Figur 2.10 Temperatur/entalpi diagram for gassturbin med varmeveksler, [5, p. 93]

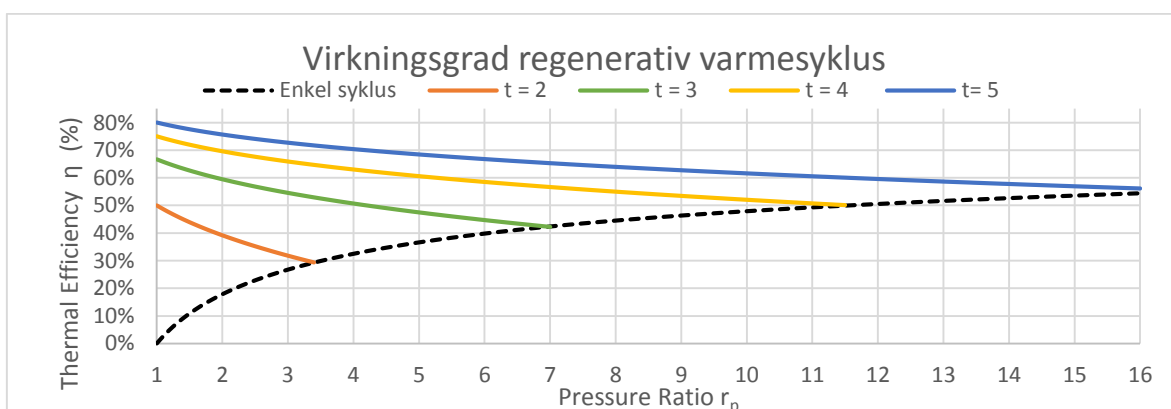
I en vanlig gassturbin beskrevet i kap. 2.5 må brennkammeret stå for den totale oppvarmingen T_2 til T_4 ($T_{\#}$ henviser til Figur 2.10). En regenerativ turbin benytter eksosvarmen til å forvarme luftstrømmen fra T_2 , til T_3 , og brennkammeret varmer luften fra T_3 til T_4 . I en ideell regenerativ gassturbin vil kun energien som går med til kompresjonen gå tapt i denne syklusen, og temperaturdifferansen mellom T_5 (utløpstemperatur) og T_2 (temperatur etter kompresjon) blir gjenvunnet. Virkningsgraden for dette systemet blir dermed:

$$\eta = \frac{\text{Arbeid ut}}{\text{Varme inn}} = \frac{q_t - q_c}{q_b} = \frac{c_p(T_4 - T_5) - c_p(T_2 - T_1)}{c_p(T_4 - T_3)} \quad \text{Formel 2.7} \\ = \frac{(T_4 - T_5) - (T_2 - T_1)}{T_4 - T_3} \quad [18, \text{p. 40}]$$

Om vi ser for oss en ideell varmeveksler uten tap, hvor $T_3 = T_5$, og et isentropisk forhold mellom trykk og temperatur, kan uttrykket over substitueres til:

$$\eta = 1 - \frac{r_p^{(\gamma-1)/\gamma}}{t} = 1 - \frac{r_p^{0,283}}{t} \quad \text{Formel 2.8} \\ [18, \text{p. 40}]$$

t er forholdet mellom temperatur inn i turbin T_4 og temperatur inn i kompressor T_1 , mens r_p er kompresjonsforholdet i kompressoren. I motsetning til en enkel Brayton syklus nevnt i kap. 2.5.2, vil et høyt kompresjonsforhold i en regenerativ syklus gi lavere virkningsgrad. Dette kommer av at kompresjonen øker temperaturen ut av kompressor T_2 , samtidig som temperatur ut av turbin T_5 forblir den samme, dermed minker temperaturdifferansen varmeveksleren kan overføre. Derimot vil en høy temperatur inn i brennkammeret heve utløpstemperaturen T_5 mens kompresjonstemperaturen T_2 forblir den samme, dette fører til en økning av turbinens virkningsgrad.



Figur 2.12 Virkningsgraden til en regenerativ varmesyklus i forhold til kompresjonsforholdet r_p ved flere temperaturforhold t . [20, p. 41]

Figur 2.12 viser det inverse forholdet mellom økende kompresjonsforhold og virkningsgrad for den regenerative varmesyklusen. Den stiplede linjen viser kompresjonsforholdet der en regenerativ syklus ikke vil lønne seg. Etter dette punktet vil utløpstemperaturen fra turbinen T_5 være kaldere enn temperaturen etter kompresjon T_2 , og varmeveksleren vil kjøle ned luftstrømmen før brennkammeret og virkningsgraden vil bli lavere enn om eksosen ble ført ut i friluft [18, p. 41]. Hovedfordelen av å benytte en regenerativ gassturbin er at vi kan bruke relativt billige radialturbiner med lavt kompresjonsforhold, samtidig som vi opprettholder en høy virkningsgrad.

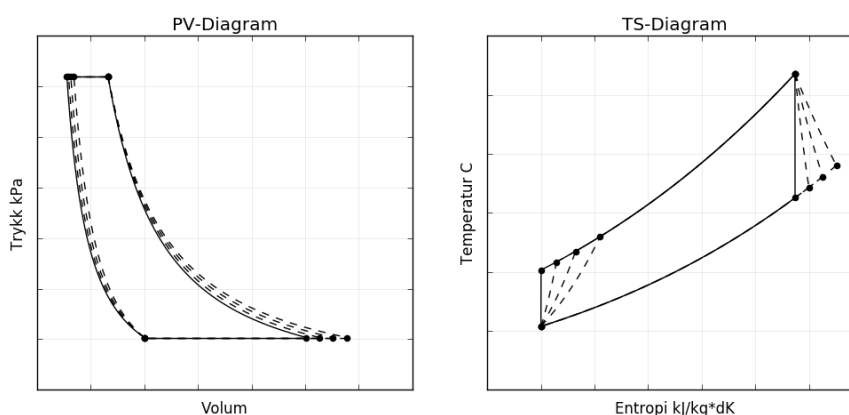
2.7. Tap i Braytons varmesyklus

2.7.1. Kompressor / Turbin virkningsgrad

I den ideelle varmesyklusen tar vi forbehold om at all varmeoverføring og gasstransport er ideell og uten tap. Et av de mest opplagte og viktigste tapene er et resultat av virkningsgraden til kompressor og turbin. En dårlig virkningsgrad i kompressoren fører til en oppvarming av luften som komprimeres og drar energi fra turbinen tilsvarende varmeøkningen. Tilsvarende vil en dårlig virkningsgrad i turbinen varme opp luften på vei ut, som fører til tap av energi, og dermed et lavere kraftuttak på turbinaksel. Temperaturøkningen grunnet dette tapet kan utregnes ved hjelp av formelen for isentropisk kompresjon og illustreres i Figur 2.13

$$\Delta T_{tap} = \frac{T_0}{\eta} \left(\left(\frac{P_1}{P_0} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right) \quad \text{Formel 2.9} \quad [21]$$

For kompressor vil $\eta = \eta_c$ og for turbin vil $\eta = 1/\eta_t$. Påvirkningen av trykktapet vises i Figur 2.13



Figur 2.13 Tap i syklusen grunnet virkningsgrad til kompressor og turbin i et forenklet PV-TS diagram hvor $\eta_c = \eta_t = 100\%$ (heltrukket), 90%, 80%, 70% basert på [22, p. 104]

2.7.2. Trykkfall

Et annet viktig tap kommer fra trykkfall gjennom syklusen. Når gass presses gjennom varmeveksler, brennkammer og lyddempingsanretninger for eksosen, vil friksjon i vegger og hindringer i gassbanen forårsake deler av trykkfallet som ellers ville skjedd over turbinen. Dette fører til at mindre energi kan tas ut i turbinen og som en lavere total virkningsgrad. Trykkfallet over rørlengder kan estimeres med Darcy-Weisbach-formelen

$$\Delta P = f_d \frac{L}{D} \frac{\rho V^2}{2g} \quad \text{Formel 2.10} \quad [23]$$

Hvor L er rørlengden, D er hydraulisk rørdiameter, ρ er egenvekt til mediet, V er hastigheten til mediet og f_d er en dimensjonsløs faktor som kan utregnes ved:

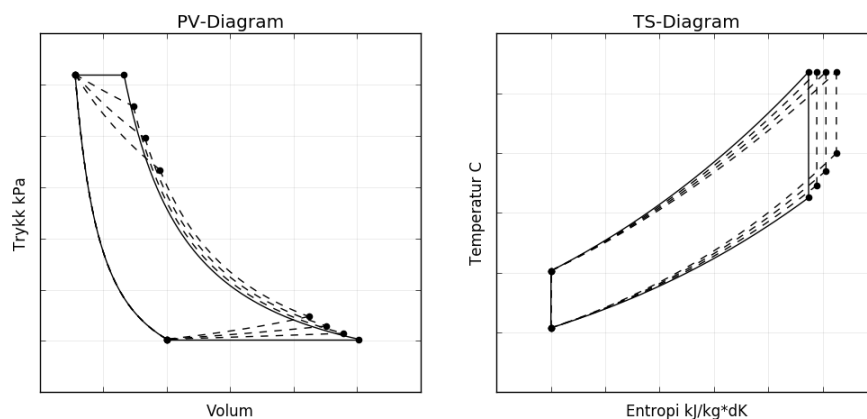
$$f_d = \frac{0.25}{\log\left(\frac{e}{3.7D} + \frac{5.74}{Re^{0.9}}\right)^2} \quad \text{Formel 2.11} \quad [23]$$

Hvor e er overflateruheten og reynoldstallet Re som utregnes:

$$Re = \frac{\rho V D}{\mu} \quad \text{Formel 2.12} \quad [23]$$

Hvor μ er absolutt viskositet for mediet. Formel 2.11 vil gjelde for en overflatefaktor $10^{-6} < \frac{e}{D} < 0.01$ og et reynolds tall $5000 < Re < 3 * 10^8$

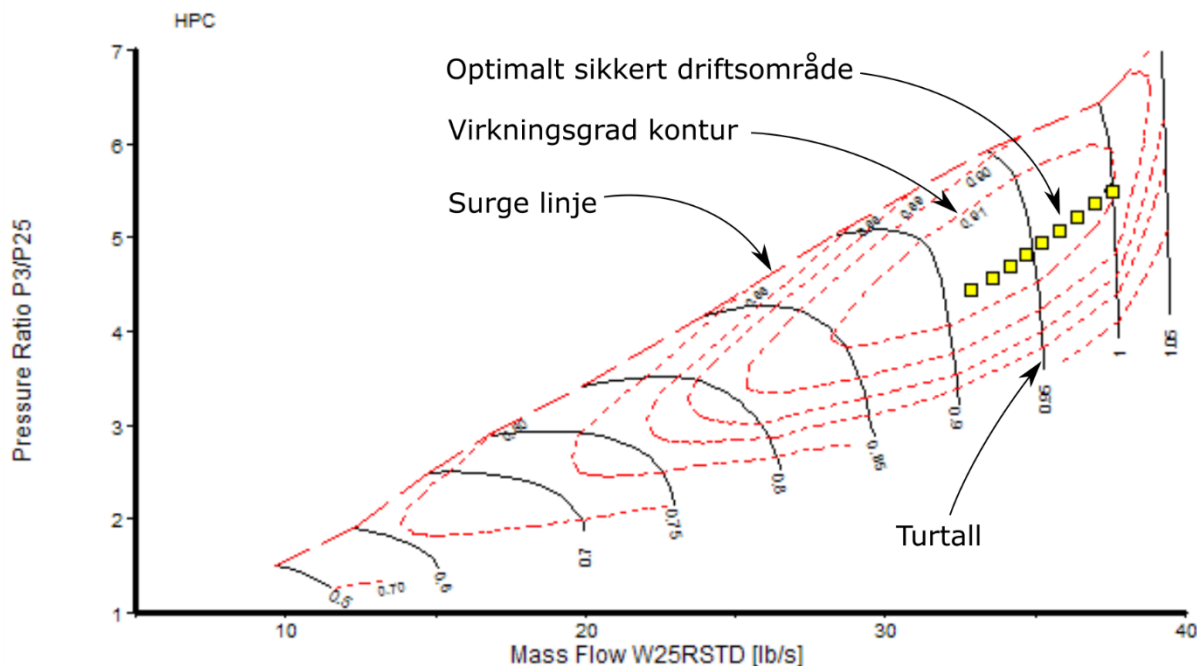
Trykktapet over brennkammeret er vanskelig å estimere uten bruk av CFD programmer, men vil vanligvis ligge mellom 2-10% av det statisk trykket [16, p. 439].



Figur 2.14 Trykkfallets påvirkning på PV-TS diagrammet. trykkfallet minker energien tatt ut over turbin, og hever dermed utløpstemperaturen. Her vises et trykktap på hhv 0% (Heltrukket) 10, 20 og 30%. basert på [24, p. 181]

2.7.1. Kompressor og turbinkart

Alle kompressorer og turbiner vil ha et optimalt driftsturtall og trykkforhold hvor virkningsgraden til komponenten er på sitt maksimum med en korresponderende gass-strøm. Dette området kan avleses fra et kompressor-virkningsgradskart vist i Figur 2.15. og vil være unikt for hvert kompressordesign.



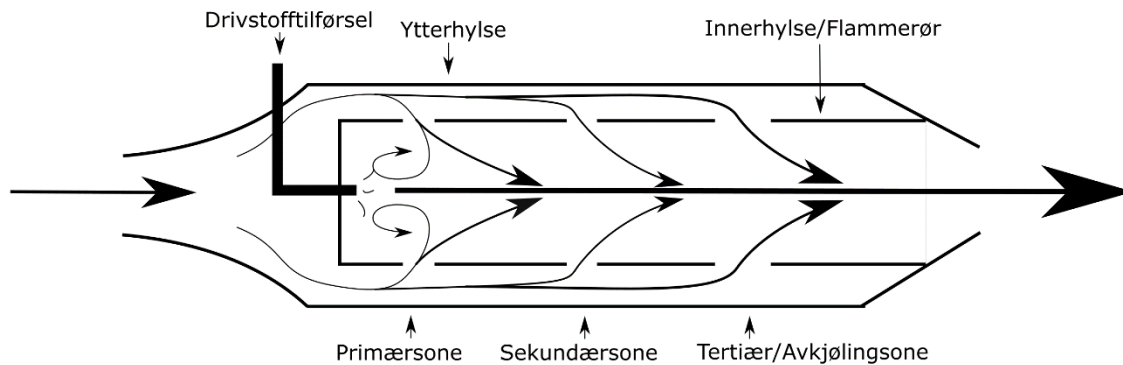
Figur 2.15 kompressor-virkningsgradskart [25]

Det topografiske kartet tilsvarende virkningsgraden til kompressoren, de svarte strekene er turgtallet kompressoren må holde i det gitte området. Den stipulerte linjen øverst på kartet er et forhold av rotasjonshastighet og massestrøm hvor kompressoren vil gå i «Surge» [26, p. 105]. «Compressor surge» er et fenomen der kompressoren ikke lenger klarer å opprettholde et mottrykk og massestrømmen vil reverseres i kompressoren. [16, p. 328], dette kan være svært destruktiv for en gassturbin. For å forsikre seg mot en slik hendelse vil driftsområdet normalt legges litt under optimalpunktet for å gi en sikkerhetsbuffer mot «surge» som vises i form av de gule merkene på figuren.

Denne informasjonen er erfaringsvis meget vanskelig å anskaffe fra produsenter av gassturbiner, men det er mulig å estimere kartet ved hjelp av programvare for simulering av gassturbiner. En utfordring er at disse programmene er svært dyre.

2.8. Konstruksjon av Brennkammer

Det finnes flere forskjellige variasjoner av brennkammer for gassturbiner, hvor den enkleste av konstruksjonene er et «Tubulært» diffusjonsbrennkammer [27].



Figur 2.16 Forenklet tubulært diffusjonsbrennkammer

Dette brennkammeret består av to konsentriske rør, hvor forbrenningen foregår i det innerste røret. Innerrøret deles opp i tre soner som har forskjellige oppgaver [27].

Primærzone:

Hovedforbrenning av det tilførte drivstoffet, i tillegg er det ønskelig med turbulens som holder en konstant tenning av drivstoffet bakerst i brennkammeret.

Sekundærzone:

En mindre mengde luft tilfører nytt oksygen som opprettholder forbrenningen og forhindrer dannelse av CO.

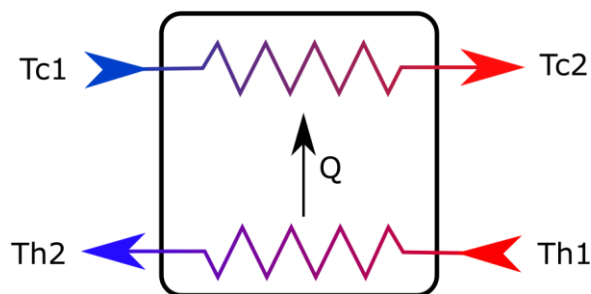
Tertiærzone /Avkjølingszone:

Temperaturen i brennkammeret må nedkjøles før den kan sendes inn i turbinen, luften blander seg med forbrenningsgassene og reduserer temperaturen til et akseptabelt nivå.

Vi har funnet forskjellige anbefalinger når det kommer til strømningsfordelingen i de forskjellige sonene. En kilde anbefaler en 30/70 fordeling av luftstrømmen mellom primær og sekundær/tertiærzone samlet [16], mens en annen kilde anbefaler at 40-60% av luftstrømmen går til tertierzonen alene. [27]

2.9. Varmevekslere og CHP

I en gassturbin brukes varmevekslere i flere konfigurasjoner for å overføre energi fra ett sted i syklusen til et annet. De tre tilfellene hvor en varmeveksler er nødvendig er ved regenerering av varme (Kap 2.6), intercoolere for å kjøle ned gasstrømmen mellom kompressortrinn, og CHP systemer hvor eksosvarmen benyttes et annet sted enn i turbinsyklusen, for eksempel i fjernvarmeanlegg. [28, p. 241]



Figur 2.17 Skjematisk oppstilling av varmeveksler, varmeveksleren har en varmside Th , hvor varm luft strømmer inn, og en kaldside Tc hvor kald luft strømmer inn. Q er energien overført fra varmside til kaldside.

Energioverføringen fra varm til kaldside vil ikke være 100% effektiv. Nøyaktig utregning av virkningsgraden til en teoretisk varmeveksler er omfattende, men i vårt tilfelle vil det være nok å anta at virkningsgraden er:

$$\eta = \frac{Q_{reell}}{Q_{max}} \quad \text{Formel 2.13} \quad [29]$$

Ved å sette energiberegningen fra Formel 2.3 inn i Formel 2.13 kan vi estimere energioverføringen i varmeveksleren med tap som

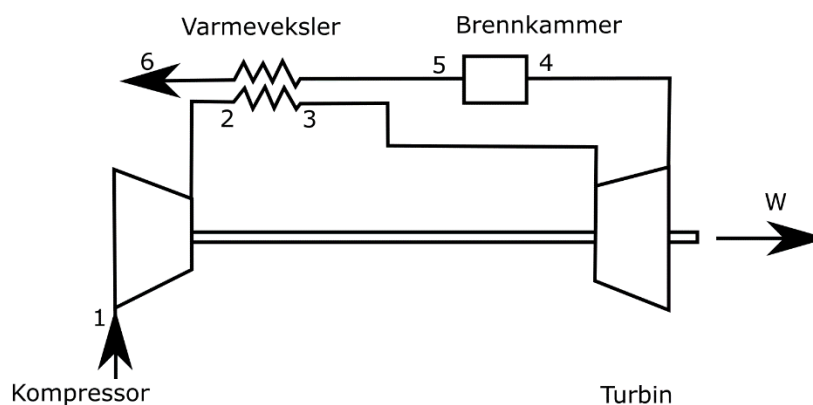
$$Q_{reell} = \eta Q_{max} = \eta C_p (Th1 - Tc1) \quad \text{Formel 2.14} \quad [29]$$

Temperaturdifferansen over varmeveksleren blir da

$$\Delta T = \frac{Q_{reell}}{\eta C_p} \quad \text{Formel 2.15}$$

3. Utvikling av egen prototyp – Ares Turbine

Som nevnt i kap. 2.4.3, er det problematisk å benytte drivstoff som genererer store askepartikler i gassturbiner da dette fører til slitasje av turbinen. Siden opprettelsen av familieselskapet Ares Turbine AS vært jeg vært delaktig i patententering [30] og utviklingen av en løsning på dette problemet. Patentet går ut på å flytte brennkammeret utenfor turbinsyklusen, og benytte en roterende varmeveksler for å overføre all varmeenergien til luftstrømmen mellom kompressor og turbin. Det fører til at eksosen og asken fra forbrenningen aldri passerer gjennom turbinen, og forårsaker dermed ikke skade på turbinbladene. En forenklet skisse av dette konseptet kan sees under i Figur 3.1

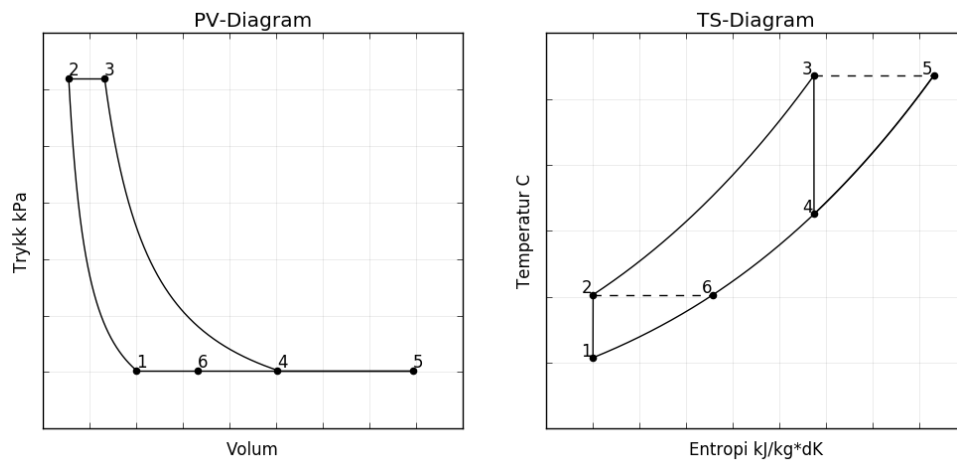


Figur 3.1 Forenklet skisse av gassturbin med regenerativ syklus og eksternt brennkammer. Basert på [16, p. 93]

I motsetning til prosessen nevnt i kap 2.4.3, er Ares Turbinen et lavtrykk- system med luft som medium. Dette betyr at en punktering av turbin-syklusen kun fører til at gassturbinen slutter å fungere. I en dampturbin vil en punktering av rørsystemet føre til en «eksplosjon» av 500°C varm damp med 30-50 bars trykk. Det vil derfor være forsvarlig å benytte dette systemet innendørs som en erstatning for eksisterende olje, flis og pellets baserte oppvarmingssystemer.

Det interessante med denne syklusen er at luften ut fra turbinen ikke har gjennomgått en forbrenning, og fortsatt inneholder oksygen. Den oppvarmede luften kan dermed føres tilbake til brennkammeret, og benyttes på nytt i forbrenningsprosessen. Dette gir en effektiv forbrenning, i tillegg til at varmen resirkuleres.

3.1. Teoretisk virkningsgrad regenerativ syklus med ekstern forbrenning



Figur 3.2 PV-TS diagram for ideell regenerativ varmesyklus med ekstern forbrenning. Stiplet linje viser overføring av varme via varmeveksler

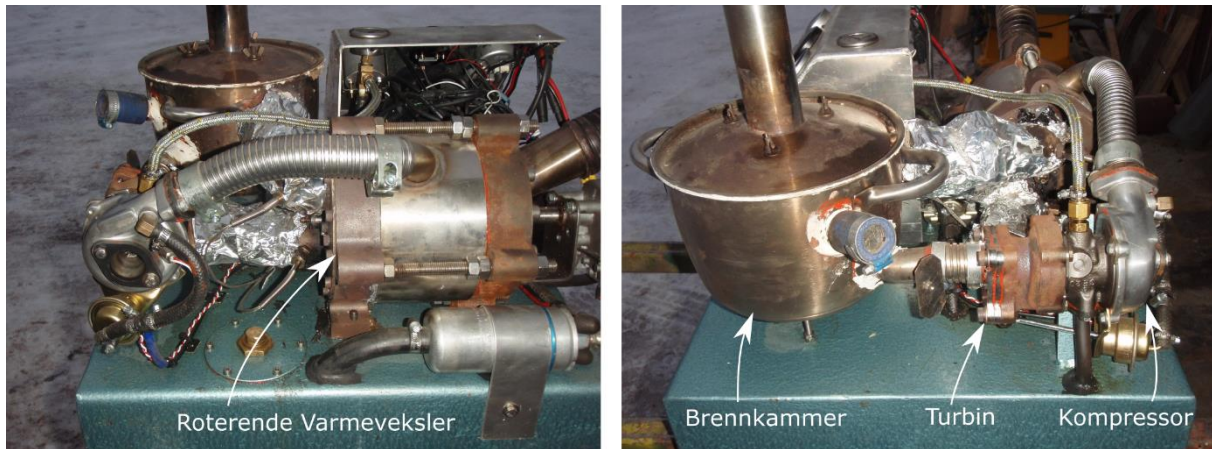
I Figur 3.2 ser vi at energitilførselen mellom punkt 4→5 skjer ved atmosfærisk trykk og blir en rett strek ut fra diagrammet. I denne syklusen vil varmevekslingen foregå mellom punkt 5→3. Her vil varmeveksleren overføre all energi fra *utløpet på brennkammeret til innløpet til turbin*, i motsetning til i en vanlig regenerativ turbin (kap 2.6) hvor varmevekslingen skjer fra *utløpet på turbin til innløpet av brennkammeret*. I en ideell gassturbin vil ikke dette ha noen innvirkning, og de to syklusene vil ha identisk teoretisk virkningsgrad beskrevet ved hjelp av Formel 5, i tillegg til å kunne utregnes med:

$$\eta = \frac{\text{Arbeid ut}}{\text{Varme inn}} = \frac{q_t - q_c}{q_b} = \frac{c_p(T_3 - T_4) - c_p(T_2 - T_1)}{c_p(T_5 - T_4)} \quad \text{Formel 3.1}$$

$$= \frac{(T_3 - T_4) - (T_2 - T_1)}{T_5 - T_4}$$

Hvor q_t er arbeid tatt ut i turbinen, q_c er arbeid brukt i komprimering av lufte, q_b er energi tilført i brennkammeret og T_x er temperaturnivået i TS diagrammet.

3.2. Original Prototyp og tidligere arbeid

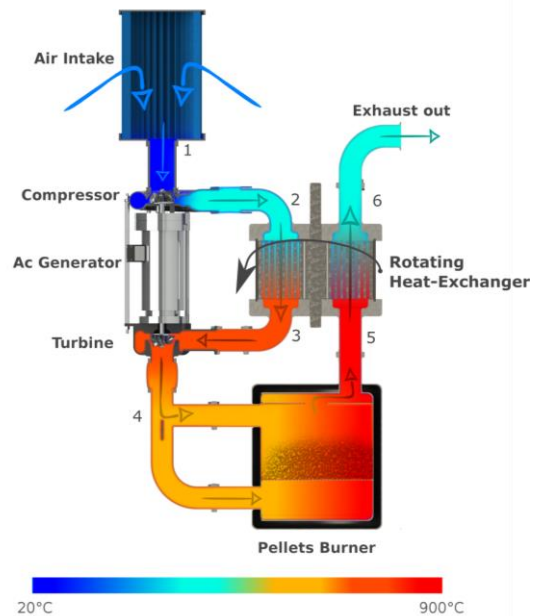


Figur 3.3 Første prototype med pellets-brennkammer [31]

I 2009 ble det konstruert en funksjonell prototype for å teste konseptet. Prototypen benytter seg av en turbo som erstatning for en lavkompresjons gassturbin, med egenprodusert roterende varmeveksler og brennkammer for pellets. Oppsettet til prototypen vist i Figur 3.3 kan være vanskelig å følge, men følger samme oppsett som konseptskissen fra Figur 3.4. og det skjematiske oppsettet i Figur 3.1

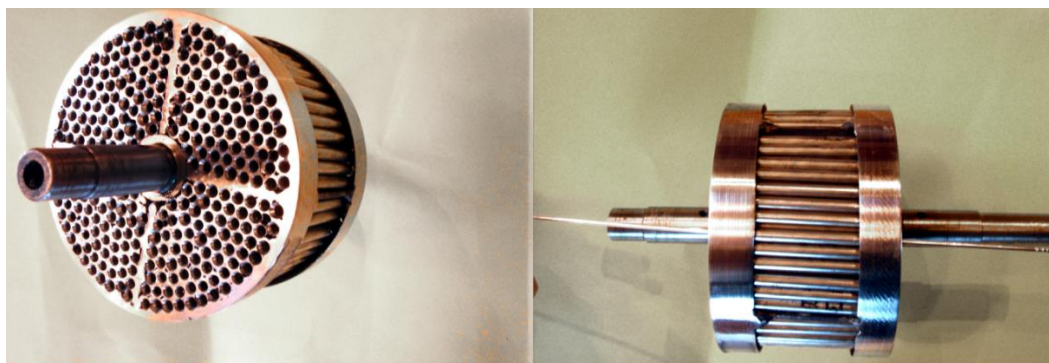
Hovedforskjellen mellom Figur 3.1 og Figur 3.4, er at luftstrømmen kan føres utenom brennkammeret for nøyaktig og hurtig kontroll av temperatur ettersom forbrenningen av pellets har svært høy temperatur-hysterese.

Prototypen ble satt i drift uten last på turbin, men før vi fikk gjennomgått målinger av virkningsgrad på systemet ble prototypen delvis ødelagt grunnet et urelatert uhell, og prosjektet satt på vent i 2010.



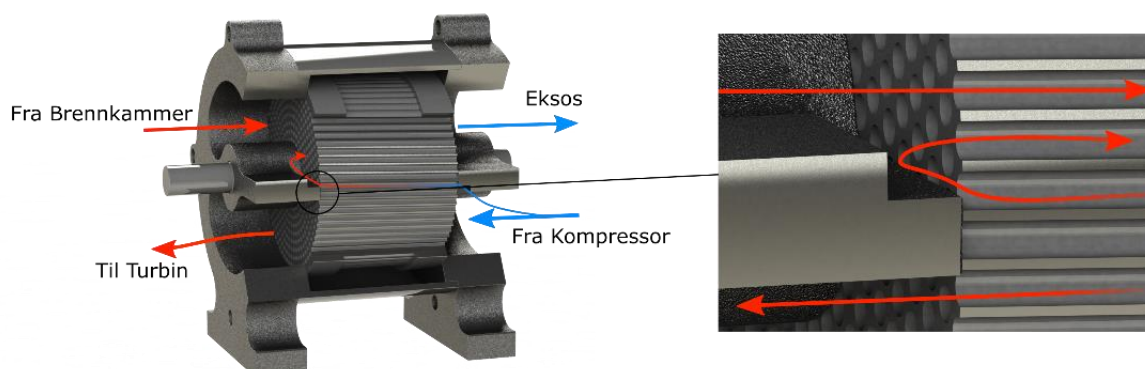
Figur 3.4 Konseptskisse av Ares Turbinen med tall-henvisninger til Figur 3.1

3.2.1. Roterende Varmeveksler



Figur 3.5 Roterende varmeveksler, designet og produsert av Ares Turbine AS [12]

Etttersom konseptet er konstruert for å benytte pellets som drivstoff har vi valgt å benytte en roterende varmeveksler (Figur 3.5). Forbrenningen av pellets gir en klebende aske som kan etterlate et isolerende lag i varmeveksleren. Dette er et kjent problem i pelletsovner og kaminer hvor varmeveksleren jevnlig må rengjøres for å opprettholde en god virkningsgrad.

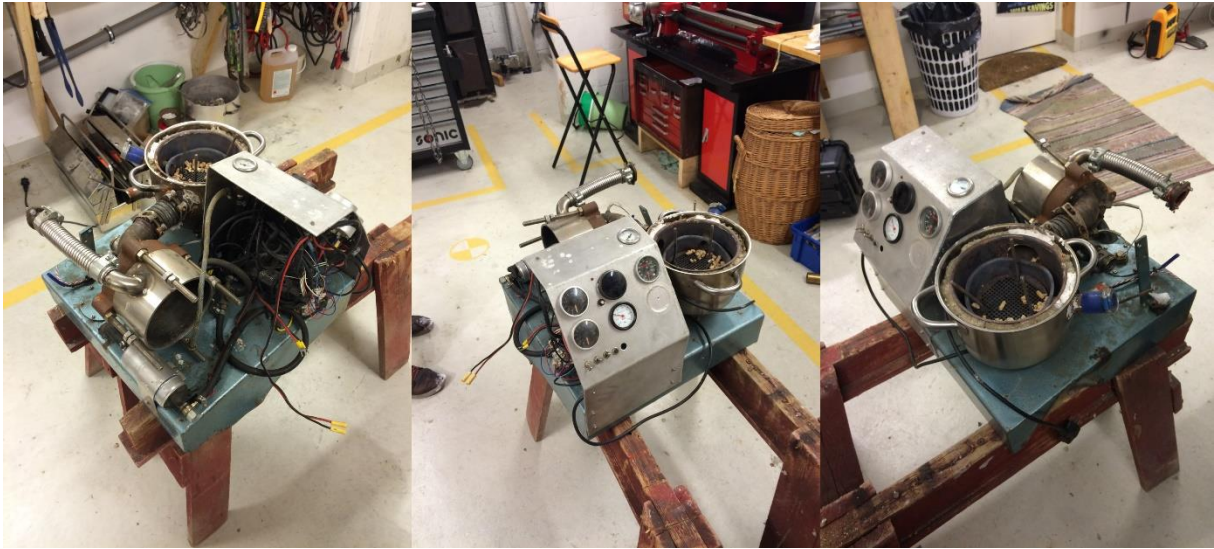


Figur 3.6 Spyling av roterende varmeveksler. Bildet T.H viser nærbilde av spyleprosessen på varmsiden av varmeveksleren

Ved å benytte en roterende varmeveksler vil luftstrømmen alternere for hver halve omdreining. Det påfølgende termiske sjokket løsner askepartikler som har festet seg til metalloverflaten. Rørene i varmeveksleren er montert i en vinkel på ca 5 grader. Dette fører til at askepartiklene som løsner vil spyles tilbake til forbrenningssiden og forsvinner ut med eksosen (Figur 3.6). Luften som går med til spylingen blir oppvarmet av varmeveksleren, men en andel av dette energitapet blir gjenvunnet når den passerer varmeveksleren på nytt (T.H Figur 3.6). Spylingen vil resultere i et varmetap, men det vil kunne lønne seg sett i forhold til reduksjon av vedlikehold og tap av virkningsgrad grunnet askeoppbygning.

3.3. Forberedelse av prototyp - Eget arbeid

3.3.1. Utgangspunkt



Figur 3.7 Turbinens tilstand da den ble gravd frem på låven

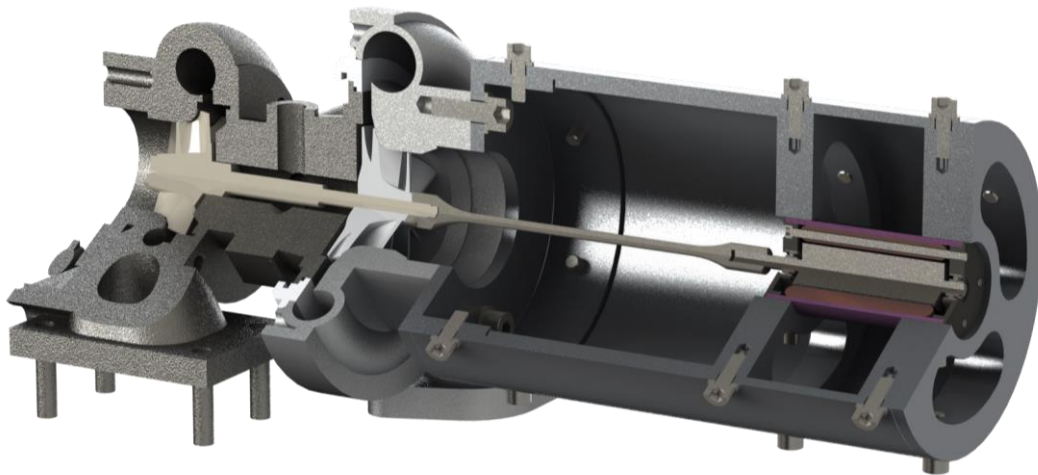
Prototypen ble i seks år lagret i en låve, og er av den grunn relativt medtatt av omgivelsene. Dette krevde restaurering av varmeveksler, turbo, elektronikk, ramme og oljereservoar. I tillegg var størstedelen av ledningsnettets dratt fra hverandre, og måtte settes opp på nytt.

Pelletsbrennkammeret ble også restaurert, men etter testing ble det avgjort at det var uhensiktsmessig å bruke pellets under evalueringen av gasturbinen. På bakgrunn av dette ble det konstruert et «propanbrennkammer» for å gi bedre kontroll over drivstofftilførsel, samt gjøre det enklere å starte og stoppe syklusen under test.

Den originale prototypen hadde heller ikke energiuttak på aksel. Det var derfor nødvendig å konstruere en generatormodul for å verifisere den elektriske virkningsgraden.

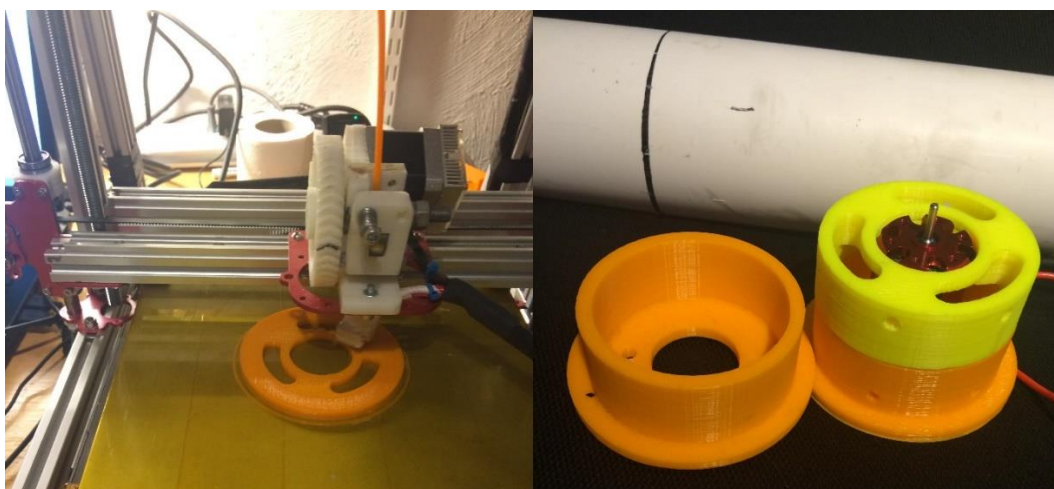
3.3.2. Generatormodul

Generatoren ble konstruert med grunnlag i tekniske tegninger og utregninger fra mitt eget forprosjekt i faget TIP300. Det originale designet var tiltenkt en 1.5kw permanentmagnetmotor med maskinerte aluminiumsdeler, men den endelige konstruksjonen ble en 750W 5000kv børsteløs generator konstruert i plastmaterialene ABS og POM grunnet tilgjengelighet av materialer og komponenter.



Figur 3.8 Tversnitt av generatormodul.

Oppspenningen av motoren/generatoren ble produsert ved hjelp av en MendelMax 2.0 3D-printer. Komponentene ble printet i ABS-plast med 70% fyllingsgrad og fire ytterlag. Det ble noe forandring i de originale plantegningene for å forenkle montering, forbedre kjøling av motor, og bedre festingen av selve generatoren. I tillegg ble kontaktflatene til de 3D-printede delene dreid ned for å forhindre akseforskyvning mellom turbin og generatormodul.



Figur 3.9 Printing og utprøving av de konstruerte delene.

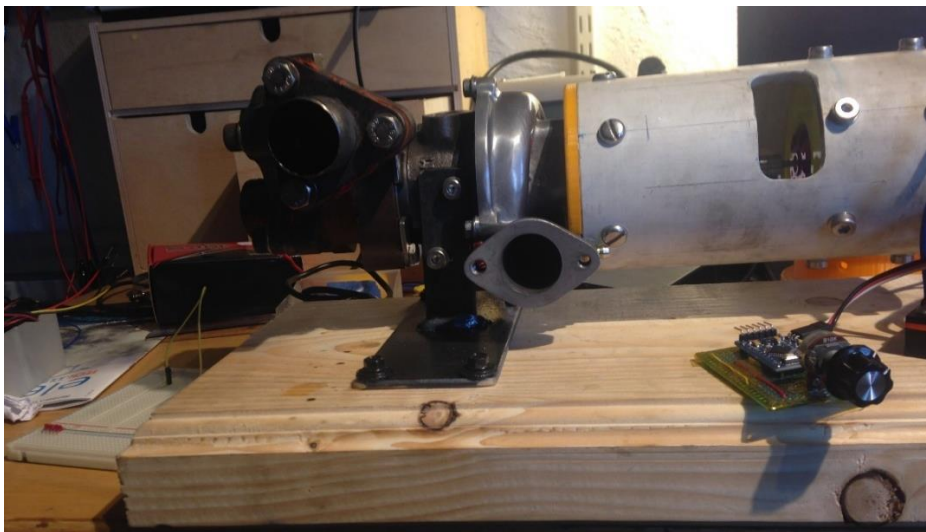


Figur 3.10 Detaljbilde av oppspenning for generator og maskinering av ytterrør.

Ytterrøret ble konstruert av et 105mm POM plastrør med dreide kontaktflater (Figur 3.10). Og modulen blir skrudd sammen med M6 skruer og setskruer for å feste motoren slik det er vist i Figur 3.8.

Styringselektronikken er en 7-cellers børsteløs motorkontroller med en Arduino Pro Mini som kontrollenhet. Det er konstruert en 50A likeretterbro av RHRG5060 dioder. Dette for å likerette trefasesrømmen når motoren kobles over fra startmotor til generator etter at gassturbinen produserer mer energi enn den forbruger.

Det endelige resultatet vist i Figur 3.11 ble vellykket, men med litt større vibrasjoner enn ønskelig. På tross av dette er sluttresultatet overraskende bra, med tanke på at det er en plastkonstruksjon.



Figur 3.11 Generatormodul montert på gassturbin for testing

3.3.3. Brennkammer

Ettersom generatormodulen er konstruert i plast, må vi ha muligheten til å starte og stanse gassturbinen så raskt som mulig for å forhindre overoppheting av plastrammen. I tillegg vil en test av virkningsgraden kreve nøyaktig måling av forbrukt drivstoff. På bakgrunn av dette valgte vi å konstruere et brennkammer for propan.

Den første versjonen ble konstruert etter et sideeksperiment, hvor vi prøvde å forbrenne propan med en palladium/platina katalysator fra en «flameout preventor». Det forenklede forsøket vist i Figur 3.12 resulterte i god forbrenning av propanen over et bredt register av luftstrøm og propantilførsel. En annen fordel med et slikt brennkammer er at katalysatoren tenner propanen på nytt ved korte avbrudd i drivstofftilførsel.



Figur 3.12 Første forsøk med katalytisk brennkammer. Katalysatoren ble forvarmet med en propanflamme og propan tilført gjennom et hull i messingrøret. Kompressoren ble brukt som luftkilde

På bakgrunn av dette konstruerte vi et fullskala brennkammer fra en katalysatormodul. Katalysatoren kom ferdig opprullet i et stålrør, og var derfor relativt enkelt å ombygge til å fungere som et brennkammer. Det ble gjort få beregninger på det første brennkammeret, ettersom dette var mer en funksjonstest enn et endelig produkt: I tillegg var vi begrenset av de opprinnelige dimensjonene til katalysatorhuset.

Delene som ble sveist på katalysatoren var rustfrie rørdeler for utløpet og rullede stålplater for forlengelsen av kammeret. Det er ikke optimalt å blande rustfritt og konstruksjonsstål sammen med tanke på korrosjon. I tillegg vil stålplatene «flaske» etter oppvarming, men brennkammeret vil ikke bli kjørt lenge nok til at dette er et alvorlig problem.



Figur 3.13 Konstruksjon av katalytisk brennkammer med radially luftinntak for å sikre at en uniform gjennomstrømning i substratet.

Brennkammeret ble testet med propan. Under testen valgte vi å tilføre propanen i luftinnførselen med 50cm korrugert slange før brennkammeret for å forsikre en uniform blanding av luft og propan inn i brennkammeret. Gassen ble tent ved å forvarme katalysatoren med en propanflamme gjennom et hull i brennkammeret så selve katalysatoren antente gassen da den ble tilført

Resultatet av testen var en total ødeleggelse av katalysatoren. Et interessant element med brennkammeret var at temperatur ut var svært stabil og meget lett å kontrollere over et stort spenn av volumstrøm mellom 600-1200 °C. Problemet opptrådte først da vi skulle teste makstemperaturen ut av brennkammeret. Med maks luftstrøm og tilnærmet full propanstrøm ble utløpet hvitglødende, og i forsøket på å kjøle ned katalysatoren med luftstrømmen virket det som at selve substratet tok fyr da propanen ble kuttet ut. Dette kan observeres i bilde nummer 2 Figur 3.14, hvor gnistspruten står ut av det rødglødende brennkammeret.



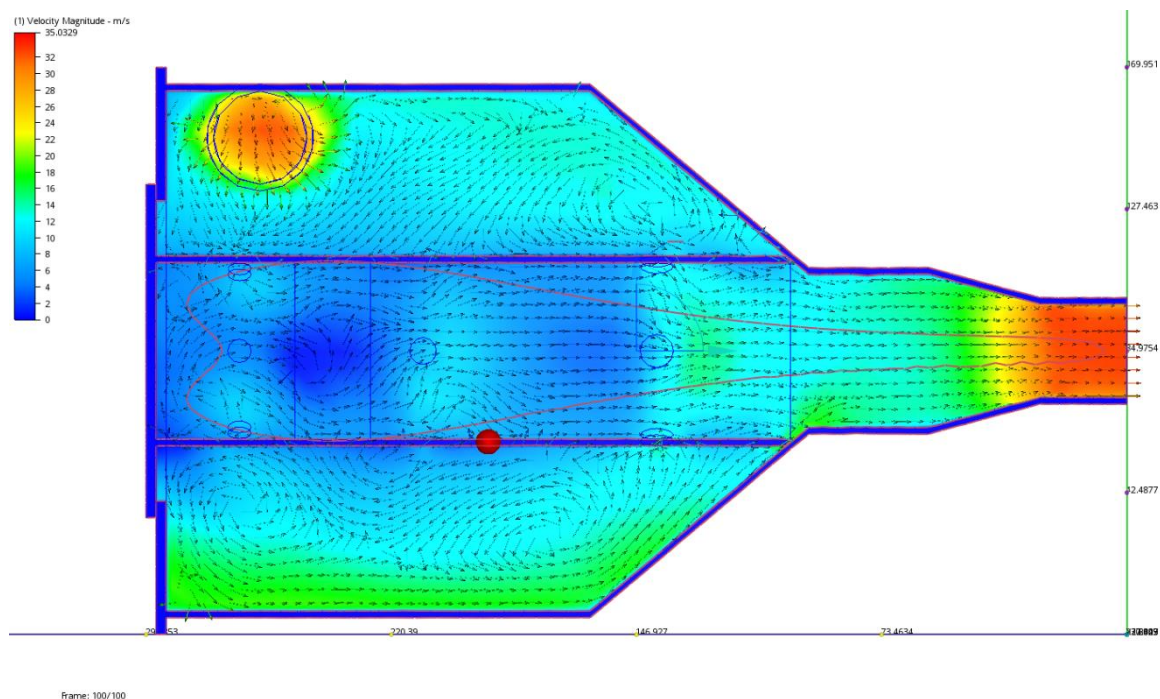
Figur 3.14 Test av katalytisk brennkammer og bilder av oppbrent katalysator.

Som konsekvens av dette måtte vi konstruere et nytt brennkammer. Vi valgte å bruke det modifiserte huset fra katalysatoren som grunnlag for et konvensjonelt diffusjonsbrennkammer. Designgrunnlaget for dimensjoneringen ble hentet ut i fra Gas Turbine Engineering Handbook [16]. En kjapp CFD analyse av designet i AutoDesk CFD 2016 verifiserte at hastigheten i flammerøret lå under 24m/s. Som er viktig for å unngå flammeutblåsning [16, p. 435]

Tabell 3.1 Dimensjonering av flammerør for å fordele luftstrøm i brennkammer

Sone	Hulldiameter (mm)	Antall Hull	Areal (mm ²)	Luftstrøm
Primær	7	6	231	30 %
Sekundær	8	3	151	19 %
Tertiær	10	5	393	51 %
Sum			774	100 %

Fordelingen av hullene ble valgt for å få en strømingsfordeling på HHV 30% i primærsonen, 19% i sekundærsonen, og 51% i avkjølingsone og sluttforbrenning i flammerøret. Vi valgte å beholde den radielle luftstrømmen inn for å skape en roterende luftstrøm ut som forhindrer flammen å komme i kontakt ytterveggen i flammerøret.



Figur 3.15 Forenklet CFD analyse med standard mesh, 1.5kg/min volumstrøm og 2 bars mottrykk uten varmetilførsel, ettersom Autocad ikke støtter simulering av forbrenning

Fra Figur 3.15 ser vi at hastighet i flammeområdet (rødt område på figuren) ligger mellom 7-12m/s. Man kan også observere at det oppstår en virvel bakerst i brennkammeret, dette er fordelaktig for å opprettholde en pilotflamme som sikrer kontinuerlig tenning av gassen. Etter simuleringen konstruerte vi brennkammeret.



Figur 3.16 Maskinering av flammerør og bakplate i brennkammeret

I det nye brennkammeret vil flammen hovedsakelig påvirke innersiden av flammerøret. På grunn av dette valgte vi å konstruere flammerøret i rustfritt stål med avstandsstykker for å unngå direkte kontakt med bakplaten vist i bilde 4 Figur 3.17. For å forenkle tenningen bruker vi en modifisert tennplugg (Figur 3.17 bilde 1 og 2) med en høyspentskilde som slår gnister mot kobberøret hvor propanen tilføres (Figur 3.17 bilde 3).



Figur 3.17 Modifisering av tennplugg for enkel tenning med propantilførsel gjennom perforert kobberør. Komplette flammerør ses i bildet T.H

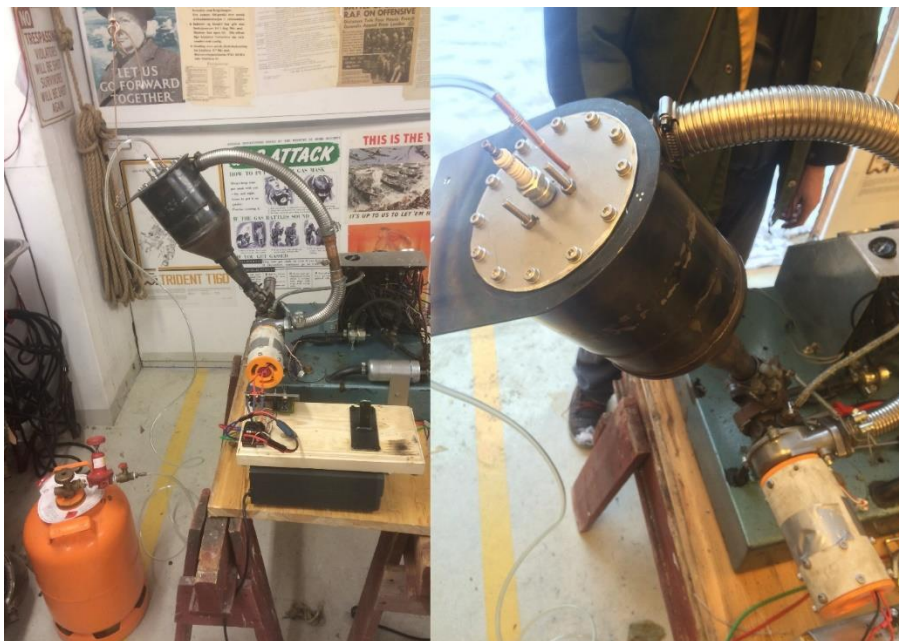
Dette brennkammeret har blitt kjørt flere ganger sammenhengende i 30 minutters intervaller med luft fra en vifte med $1.5\text{m}^3/\text{min}$ luftstrøm og en maksimal utløpstemperatur på over 1240°C uten skade på flammerøret. (Temperaturmåler makser ut ved 1240°C så vi får ikke verifisert høyere temperatur). Gasstilførsel kontrolleres med en regulator og nåleventil.



Figur 3.18 Utprøving av brennkammer. Det kan observeres at flammen er sentrert i flammerøret og ikke berører ytterveggene i flammerøret.

Figur 3.18 viser testene av brennkammer. Det første bildet viser hvordan den roterende flammen ikke kommer i kontakt med ytterveggen før den har forlatt brennkammeret. Temperaturen på utsiden av brennkammeret er ca 150 grader, mens temperaturmåling med et infrarødt termometer av stålørret som er tredd over eksos-dysen viste over 1000 grader.

For å teste brennkammer i en reell driftsituasjon ble gassturbinen prøvekjørt som i enkelsyklus konfigurasjon jamfør kap. 2.5. Vi fikk start på turbinen og kjørte ca 20 minutter, men testen ble avsluttet grunnet tap av oljetrykk.

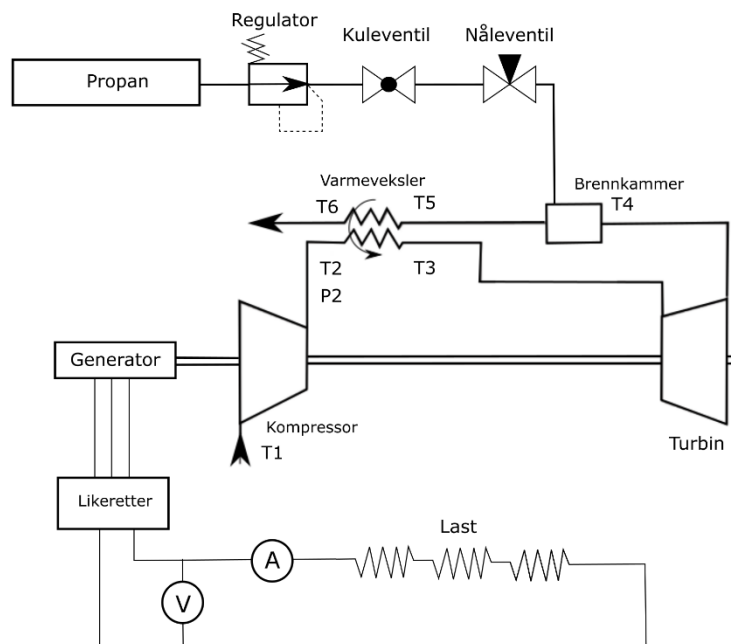


Figur 3.19 Turbo kjørt som enkelsyklus gassturbin.

4. Testprosedyre prototype.

4.1. Testoppsett

Skjematisk oppstilling av testen vises i Figur 4.1 og består av 5 måleparametere: Temperatur før og etter hver komponent (T1, T6), trykk etter kompressjon (P2), effekt levert fra generator (V,A), gram propan tilført brennkammer under måleperioden, og turtall på turbin.



Figur 4.1 Skjematisk testoppsett av gassturbin basert på [16, p. 93] brukt i Figur 3.1

Ideelt skal vi kunne kalkulere virkningsgraden basert på mengden drivstoff inn i forhold til effekt levert fra generatoren. Samtidig vil temperatur og trykkmålingene være essensielle for å fastslå virkningsgraden til de enkelte komponentene i gassturbinen, i tillegg til når det skal utføres teoretisk evalueringen av syklusen og fastslå potensielle forbedringspunkter ved oppsettet.

4.1.1. Temperatur

Temperaturmålingene blir utført ved hjelp av type K termoelementer innkapslet i rustfritt stål. Elementene kan måle temperaturer mellom -200°C til 1250°C . Målingen blir utført på punktene spesifisert som T# fra Figur 4.1. Element T1 er plassert i innløpet til generatormodulen. Element T4 er plassert i innløpet til brennkammeret. Elementene T2, T3, T5 og T6 er plassert på innløp og utløp på hver side av varmeveksleren slik det er vist i Figur 4.2.



Figur 4.2 Tilkobling av K elementer på varmeveksler. Benetech GM1312, instrumentet har to innganger og kan avlese J, K, T, E, N, R og S elementer med 1,5% nøyaktighet.

Avlesningen av elementene blir utført ved hjelp av tre Benetech GM1312 Thermocouple avlesere, Disse har en påstått nøyaktighet på 1,5%. Hvert måleinstrument kan lese av to temperaturelementer samtidig. Ettersom måleinstrumentet ikke har noen form for loggingsfunksjon vil alle instrumentene filmes under forsøket sammen med en synkronisert klokke.

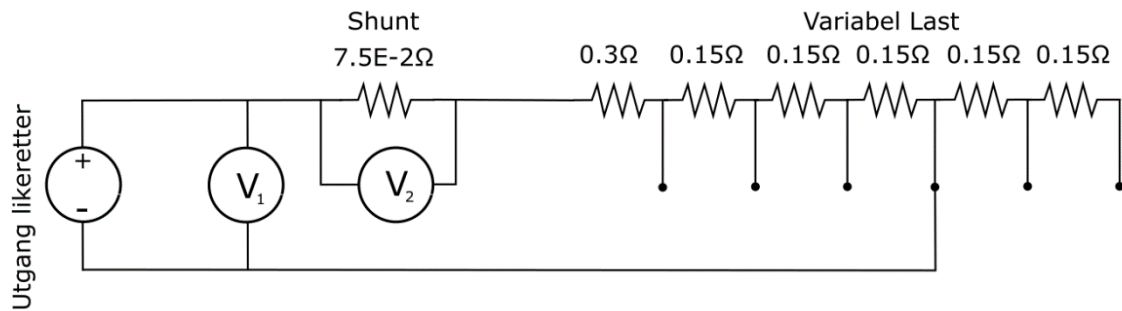
4.1.2. Trykkmåling og turtall



Figur 4.3 Analogt RaceTech Manometer montert på kontrollpanelet til gassturbin. Og Garrett turteller via induktiv måler i kompressorhuset.

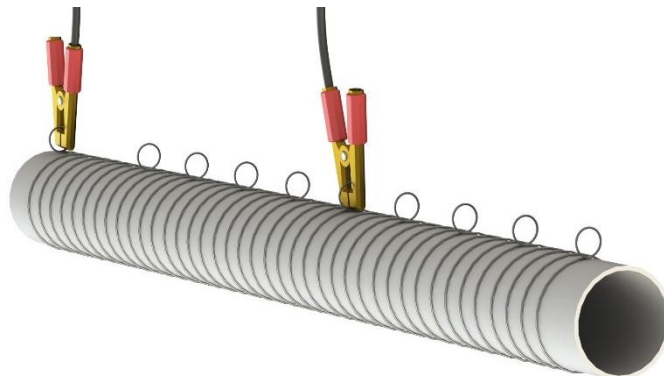
Trykkmåling P2 blir avlest analogt via et manometer fra et tilkoblingspunkt på utløpet til kompressor, omgivelsestrykket P1 blir utregnet fra høyde over havet på testlokasjon. Turteller er et Garrett tachometer som logges sammen med trykkmåling på det samme kontrollpanelet ved hjelp av et kamera og synkronisert klokke. Magnetsensoren er plassert på kompressorsiden.

4.1.3. Effekt Ut



Figur 4.4 Detaljert kretsdiagram effektmåling, utregninger finnes i vedlegg A

Effekt ut kalkuleres ved hjelp av to simultane spenningsmålinger hvor en er i parallell med utgangen til likeretterbroen. Den andre målingen er over en 100A/75mv shuntmotstand koblet i serie med en improvisert reostat. Spenningen over shuntmotstanden tilsvarer 1 A for hver 0.75 mV avlest. Reostaten lages av 10m med 1,2mm ståltråd spunnet rundt et plastrør. Lasten vil være variabel mellom 150w til 750w med en spenning på 15V ved å feste klemmer på reostaten vist i Figur 4.5. De to spenningsene logges ved hjelp av et Picoscope 3000A USB oscilloskop med 12 bit oppløsning.



Figur 4.5 Rendring av variabel last (reostat).

4.1.4. Tilførsel av drivstoff

Ettersom vi ikke har et flowmeter tilgjengelig, blir tilførselen av drivstoffet målt ved hjelp av vekten på propanbeholderen. Beholderen må veies i det testen igangsettes, og propantilførselen må umiddelbart stanses etter fullført testperiode. Vektdifferanse før og etter test vil tilsvare den totale massetilførselen av drivstoff.

4.2. Fremgangsmåte test

For å få et nøyaktig resultat må gassturbinen stabilisere seg før testing kan settes i gang. Dette innebærer at turbinen må gå til alle indre komponenter er har nådd temperaturlikevekt med omgivelsene og alle målepunktene har stabilisert seg. Dette krever at generatoren kjører med last.

Oppstart av turbin:

1. Oljepumpe startes og det bekreftes at oljetrykket ligger mellom 2-4 bar.
2. Motor til varmeveksler startes.
3. Lukk nåleventilen på propantilførsel.
4. Generatoren brukes som startmotor, spoler opp turbinen til 60'000 RPM. i inkremerter på 10'000 RPM.
5. Kuleventilen på propantilførselen åpnes mens høyspentskilden er slått på og nåleventilen åpnes sakte til det første tenningsblaffet høres.
6. Når forbrenning er bekreftet og temperaturen ut av varmeveksler overstiger 900 grader kobles generatoren fra styringselektronikken og over i likeretteren uten last.
7. Propantilførselen økes sakte med nåleventilen frem til turtallet når 70'000 RPM
8. Last kobles til

Når likevekt er nådd og temperaturene holder seg stabile over et minutt kan forsøket påbegynnes:

1. Kameraene som logger temperatur, trykk og turtall på turbin starter opptak
2. Vekten nullstilles og propanbeholderen plasseres på vekten, testen startes i det vekten på propanbeholderen noteres ned sammen med et klokkeslett
3. Vekten nullstilles hvert minutt og propanbeholderen veies på nytt med klokkeslett.
4. Testen avsluttes ved en av følgende hendelser:
 - a. Temperatur på kompressorhuset overstiger 60 grader, testen må da avsluttes for å forhindre overoppheting av generatormodulen.
 - b. Oljetemperaturen overstiger 100 grader.
 - c. Det oppstår store forandringer i måleparametrene.
 - d. Etter 30 minutter.
5. Kuleventil på propan lukkes samtidig som tidspunkt noteres.
6. Propanflaske veies på nytt og test avsluttes.

5. Utvikling og verifisering av simuleringsprogram

Utviklingen av simuleringsprogrammet ble påbegynt da vi hadde behov for å demonstrere tapene i TS diagrammene i kap 2.7. Dette ble først gjennomført i Excel, men det kom raskt frem at Excel er lite tilrettelagt for dette formålet. Dette fordi hver eneste endring av parametere i syklusen krevde en total omgjøring av Excel formlene. Formelverket ble derfor overført til programmeringsspråket Python. Ettersom vi ønsket å sjekke flere forskjellige sykluser opp mot hverandre, ble funksjonene generalisert for å kunne bruke de samme funksjonene uavhengig av syklus. Den første versjonen av programmet brukte formelverket beskrevet i kapittel 2 direkte for å simulere ideelle sykluser, men etter gjentatte oppdateringer av funksjonene ble det nødvendig å gjøre beregningene av virkningsgrad og effekt numerisk.

Ettersom programmet først ble utviklet for å generere PV-TS diagrammer, ble utregningen av virkningsgrader og andre energiberegninger integrert fra de samme matrisene som utgjør diagrammet. Dette gjorde det mye enklere å ivareta variabler som ikke forandrer seg lineært i forhold til temperatur gjennom syklusen. Det krever imidlertid at hvert ledd i syklusen må utregnes med minimum 50 punkter mellom start og sluttposisjon for å minimere utregningsfeil.

Det endelige programmet ble et generelt utregningsverktøy for evaluering av teoretiske gassturbinsykluser, plotting av PV-TS diagrammer og optimalisering av syklus-parametere.

5.1. Metode program

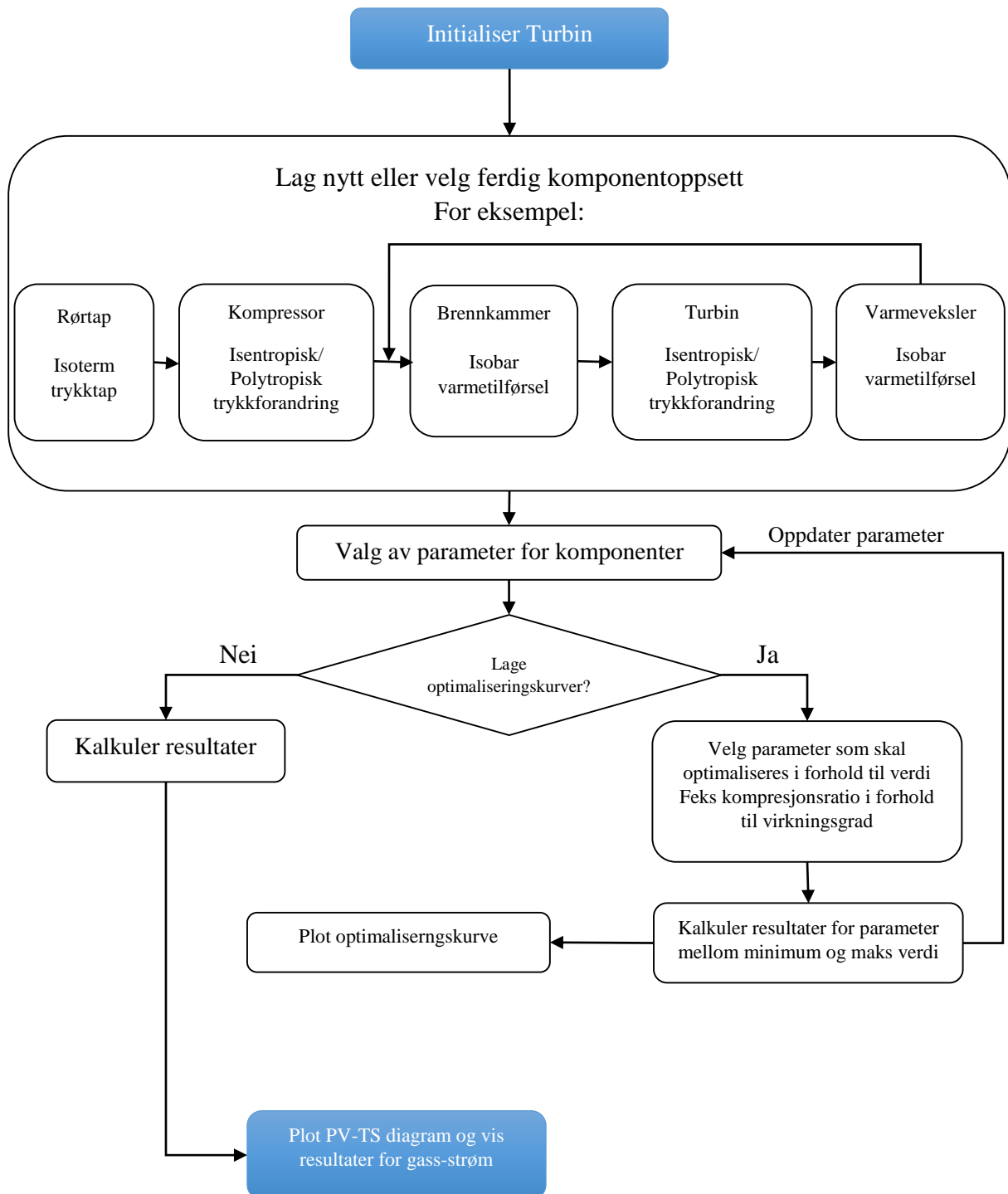
Program-biblioteket består av syv ferdige moduler som brukes til å lage en gassturbinsimulering. De må plasseres lineært etter hverandre, slik at hver funksjon henter inn en matrise fra den foregående funksjonen. Syntax for oppbygning av en syklus vises i vedlegg B, Eksempler på simuleringer i vedlegg C og kildekode i vedlegg E

Tabell 5.1 Komponent og funksjonsnavn i analysebiblioteket.

Komponent	Funksjonsnavn	Virkemåte
Innløp	Pipeloss()	Tar for seg rørtap i gassturbinen og settes mellom to komponenter. Kan enten ta inn et spesifisert trykktap, tap i prosent, eller rørlengde, diameter og ruhet for å kalkulere rørtapet iht kap 2.7.2
Kompressor/ Turbin	Compressor() Turbine()	Komprimerer/ekspanderer gassen med isentropisk eller polytropisk virkningsgrad.
Brennkammer	CombChamb()	Varmer opp gasstrømmen til spesifisert temperatur, kan innføre trykktap over brennkammer spesifisert i kPa, vannsøyle, eller prosent.
Varmeveksler	HeatExchCold() HeatExchWarm()	Overfører energi fra ett sted i syklusen til et annet. Kaldside må stå foran et brennkammer med spesifisert utløpstemperatur. Virkningsgrad kan spesifiseres ved temperaturdifferanse eller i prosent. Kan innføre trykktap spesifisert i kPa, vannsøyle, eller prosent.
Intercooler	Intercooler()	Kjøler ned gasstrømmen mot en gitt temperatur. Virkningsgrad kan spesifiseres ved temperaturdifferanse eller i prosent og kan innføre trykktap spesifisert i kPa, vannsøyle, eller prosent
Eksos	Exhaust()	Returnerer gassen til omgivelses-temperatur og trykk, kan innføre trykktap spesifisert i kPa, vannsøyle, eller prosent.

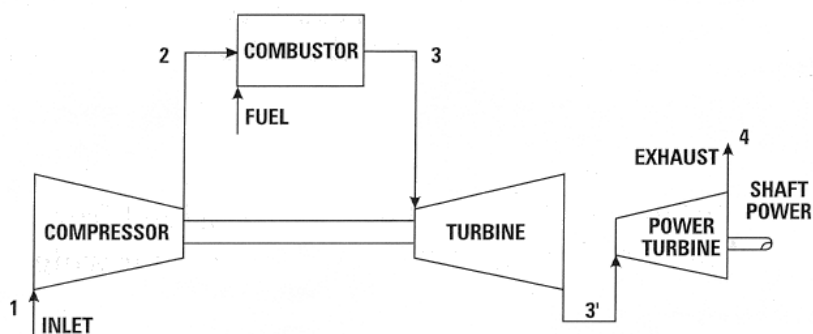
5.1.1. Flowchart av fremgangsmåte for simulering

Flowchartet under viser den generelle fremgangsmåten for å simulere en gassturbin med biblioteket. Det er mulig å simulere de fleste Brayton-sykluser, men det kreves noen modifiseringer for å evaluere avanserte turbinkonfigurasjoner.



5.2. Case Study 1 – GasTurb og VTB

For å verifisere at programmet faktisk stemmer overens med virkeligheten, ble det kjørt to simuleringer opp mot et annet simuleringsprogram, samt en ekte gassturbin. Den første casen kommer fra en masteroppgave [32] skrevet om en dynamisk simuleringsmetode VTB, som er testet opp mot et kjent simuleringsprogram for gassturbiner GasTurb [33] .



Figur 5.1 Turbin med ekstern kraft-turbin. Separeringen av de to turbinene lar de rotere uavhengig av hverandre og med forskjellig hastighet avhengig av vinkel på turbinbladene. [34]

Den aktuelle gassturbinen har adskilt energiuttak. Dette betyr at den første turbinen 3→3' skal dimensjoneres til å kun drive kompressoren, mens den overskytende energien tas ut i kraft-turbinen 3' → 4. Programmet skal da kalkulere hvilke trykkforhold turbinene må operere ved for å produsere 1187kW, samtidig som grunnparametere overholdes

5.2.1. Grunnparameter

Tabell 5.2 [32, p. 51] Grunnparameter spesifisert for simulering tallverdiene i «Komponent» henviser til Figur 5.1

Komponent		Parameter	Verdi
1-1	Innløp	Lufttrykk	kPa
		Lufttemp	K
		Trykkforhold	
1-2	Kompressor	Kompressor utløpstemp	K
		Massestrøm	kg/s
2-3	Brennkammer	Utløps Temp	k
		Massestrøm Drivstoff	kg/s
		Trykktap	%
3-3'	Turbin	Turbin Utløpstemp	K
3'-4	Powerturbin	Utløpstemp	K
		Kraft levert	kW
4-1	Eksos	Trykktap	kPa

5.2.2. Resultat

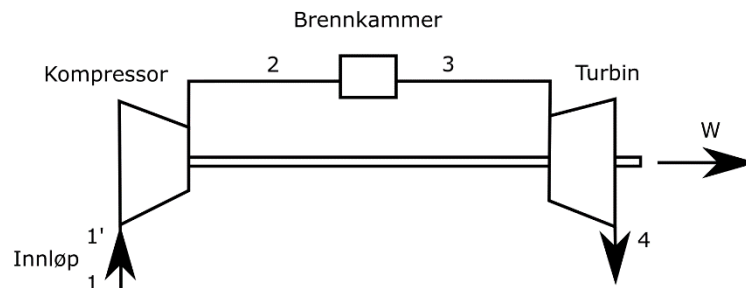
Programmet beregnet de manglende verdiene for gassturbinen, som deretter ble plottet inn i tabellen gitt i masteroppgaven. Det er opplyst en gjennomsnittlig feilprosent mellom mitt program og de to modellene. I de kalkulerte parameterne ligger mitt program under 0.5% feil, med unntak av generert effekt ut som ligger -1.03% under de andre metodene.

Tabell 5.3 [32, p. 51] Etter simuleringen ble resultatene slått sammen med tabellen gitt i kilden. Simuleringen med vårt program er markert med grå bakgrunn i raden benevnt som Ares Turb. De uthevede verdiene er variable parametere programmet kalkulerer for å tilfredsstille grunnparameterne.

Komponent	Parameter	Enhet	GasTurb	VTB	Ares Turb.	Feil
1-2 Kompressor	Inlet.Mass Flow.	kg/s	3,50	3,50	3,50	-
	Inlet.Temperature	°K	288,15	288,15	288,15	-
	Inlet.Pressure	kPa	101,33	101,33	101,33	-
	Exit.Mass Flow.	kg/s	3,50	3,49	3,50	-
	Exit.Temperature	°K	657,99	658,03	658,00	-
	Exit.Pressure	kPa	1317,23	1316,07	1317,23	-
2-3 Brennkammer	Pressure	Ratio	13,00	12,99	13,00	-
	Exit.Mass Flow.	kg/s	3,58	3,57	3,58	-
	Exit.Temperature	°K	1450,00	1452,79	1450,00	-
	Exit.Pressure	kPa	1277,71	1276,44	1277,70	-
	Fuel	kg/s	0,08	0,08	0,08	-
3-3' Turbin	Exit.Mass Flow.	kg/s	3,58	3,57	3,58	-
	Exit.Temperature	°K	1148,62	1152,45	1149,18	0,12 %
	Exit.Pressure	kPa	382,91	383,39	383,53	-0,10 %
	Pressure Ratio		3,34	3,33	3,33	0,09 %
3'-4 Power Turbin	Exit.Mass Flow.	kg/s	3,58	3,57	3,57	-
	Exit.Temperature	°K	868,22	871,66	867,99	0,22 %
	Exit.Pressure	kPa	102,34	102,34	102,33	0,01 %
	Shaft Power Delivered	KW	1187,10	1188,89	1175,74	-1,03 %
	Pressure Ratio		3,74	3,75	3,75	-0,16 %
4-1 Eksos	Exit.Temperature	°K	288,15	288,15	288,15	-
	Exit.Pressure	kPa	101,33	101,33	101,33	-

5.3. Case Study 2 – SSS-15 Turbin

Den andre sammenlikningen er en vurdering av en enkel-syklus gassturbin (kap 2.5) hvor vi fant et komplett datasett for gassturbinen med interne tap [35]. I denne simuleringen bygges gassturbinen opp med de angitte komponentene og kjøres direkte med de spesifiserte parameterne i Tabell 5.4.



Figur 5.2 Skjematisk oppsett av SSS-15 enkelsyklus turbin

5.3.1. Grunnparametre

Tabell 5.4 basert på [35, p. 3] Grunnparametre spesifisert for SSS-15 turbin. Programmet vil kalkulere effekt ut basert på de opplyste verdiene. Tallhenvisninger til Figur 5.2

Komponent	Parameter	Enhet	Verdi	
1-1'	Innløp	Airflow	kg/s	0,481
		Inlet pressure loss	%	0,8
1'-2	Kompressor	Compressor Pr		11,10
		Compressor Eff	%	78,5
2-3	Brennkammer	Combustor Press Drop	%	2
		Combustor Eff	%	98
		Fuel flow	kgs/s	0,01
3-4	Turbin	Turbine Inlet Temp	°K	1344
		Turbine Press Ratio		10,10
		Turbine Eff	%	88,3
4-1	Eksos	Exhaust press loss	%	6,23
Andre Tap		Parasitic loss	kw	2,8
		Gearbox Loss	kw	3,5
		Net output	kw	111,85

5.3.2. Resultat

Programmet estimerte en netto effekt ut på 112,85kW, altså 0.89% høyere enn den reelle turbinen.

Tabell 5.5 Resultat fra simulering med tallhenvisning til Figur 5.2.

komp	Press1(kPa)	Press2(kPa)	Temp1(°K)	Temp2(°K)	Volume1	Volume2
1-1'	101,3	100,5	288,2	288,2	1,0	1,0
1'-2	100,5	1115,7	288,2	679,6	1,0	0,2
2-3	1115,7	1093,4	679,6	1344,0	0,2	0,4
3-4	1093,4	115,5	1344,0	818,9	0,4	2,5
4-1'	115,5	108,3	818,9	818,9	2,5	2,7
1'-1	108,3	101,3	818,9	288,2	2,7	1,0

Beregnet Kw	112,85
Opplyst Kw	111,85
Feilmargin	0,89 %

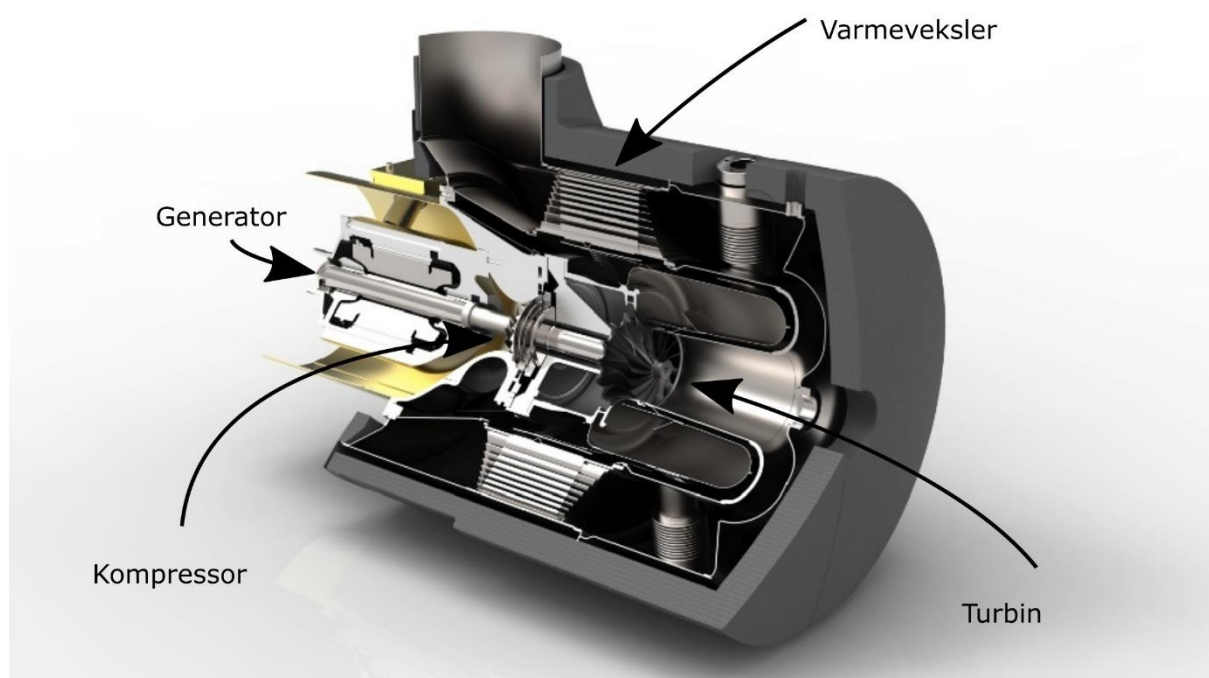
6. Innhenting av data og simulering av syklus

6.1. Valg av sammenlikningsturbin for regenerativ syklus

Ettersom vi ikke fikk forsøksdata vil vi sammenlikne vår syklus med en vanlig regenerativ gassturbin, er det behov for et komplett datasett før analysen kan utføres. I forsøket på å anskaffe data for en «benchmarkturbin» i 30-40kw klassen, har vi kontaktet Capstone og Elliott Energy System. Det har vist seg å være svært vanskelig å oppdrive et komplett datasett ettersom produsentene ser på disse dataene som meget sensitive i med tanke på konkurrenter, og henviste derfor til håndbøkene for turbinen som kun opplyser om eksterne parameter. Dette betyr at data må samles inn fra forskjellige lærebøker, artikler og bruksanvisninger.

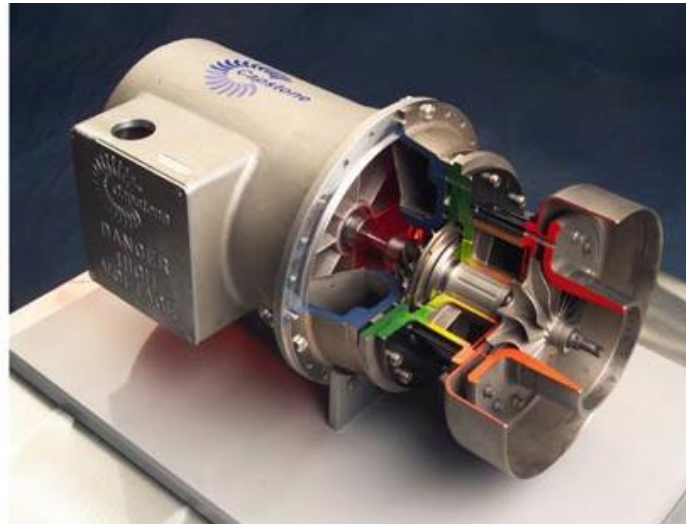
6.1.1. Innsamling av data

Turbinen vi har funnet flest data for er en Capstone C30 regenerativ mikroturbin. Denne turbinen er godt dokumentert med tanke på ytre parameter, men det er lite informasjon angående indre tap i gass-syklusen.



Figur 6.1 Capstone C30. gassturbin. Komplette enhet med varmeveksler og brennkammer montert.. [36]

En fordel med denne gassturbinen vist i Figur 6.1 er at den har en turbinkjerne som potensielt kan fjernes fra gassturbinen (Figur 6.2) og ombygges til en regenerativ gassturbin med ekstern forbrenning. Dette vil gi store besparelser i forhold til å utvikle en turbin fra grunnen av hvis det skal produseres en fullskala prototyp i fremtiden.



Figur 6.2 Capstone C30 turbinkjerne demontert fra turbin vist i Figur 6.1 [37]

Tabell 6.1 Innsamlet data for capstone C30 fra manual, artikler og lærebøker.

Parameter	Verdi	Kilder
Kompresjons ratio	3.9	[24, p. 210]
Turbin temp inløp	871°C	[38]
Turbin temp utløp	600-620 °C	[20]
Massestrøm Utløp	0.31 kg/s	[38]
Eksostemp utløp	274 °C	[38], [24, p. 210]
Elektrisk Virkningsgrad	26% (+/- 1%)	[38]
Elektrisk Effekt ut	30kw (-/ 1kW)	[38]
Tap innløp, utløp	~0%	[38]

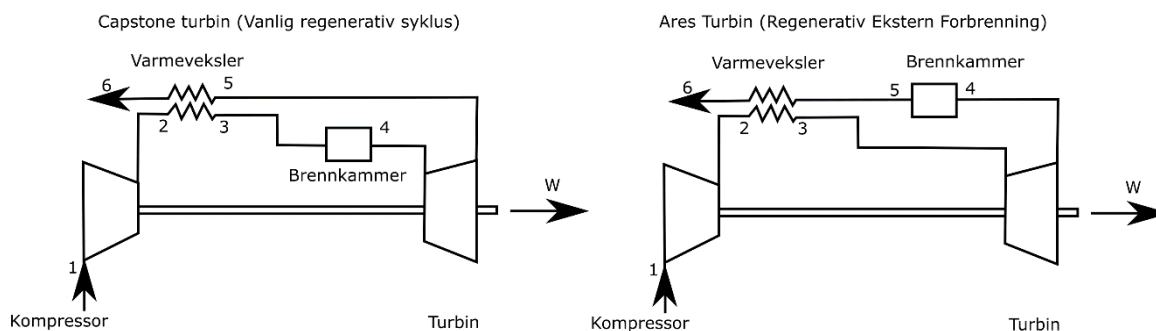
For å simulere denne gassturbinen mangler vi fortsatt en del parameter gitt i Tabell 6.2, disse må estimeres ved hjelp av simuleringprogrammet, hvor vi bruker parameterne gitt i Tabell 6.1 som fasit.

Tabell 6.2 Parameterintervaller for komponenter hvor vi mangler data.

Parameter	Antagelse
Virkningsgrad Kompressor	70-80%
Virkningsgrad Turbin	75-85%
Virkningsgrad Varmeveksler	70-90%
Trykktap over varmeveksler	4-8%
Trykktap over brennkammer	2-10%
Virkningsgrad generator og mek. tap	90-98%
Omgivelses temp	15°C
Omgivelses trykk	101.3 kPa

6.2. Simulering av varmesyklus med ekstern forbrenning (Ares Turbin)

Capstone turbinen simuleres med oppsettet merket Capstone Turbin i Figur 6.3 Mens den eksterne varmesyklusen heretter benevnt som Ares Turbinen vil bli simulert med konfigurasjonen merket Ares Turbin i samme figur.



Figur 6.3 Sammenlikning av de to syklusene, Capstone-turbinen T.V og Ares turbinen T.H

Ettersom rekkefølgen på komponentene i Capstone turbinen og Ares Turbinen er forskjellig, kreves det en litt annerledes simulering av varmeveksleren for å oppnå et sammenliknbart resultat.

Capstone rekkefølge:

Kompressor → **Varmeveksler** → **Brennkammer** → **Turbin** → Varmeveksler → Eksos

I simuleringen av Capstone-turbinen (Figur 6.3 T.V) tilfører varmeveksleren varme til gassstrømmen før brennkammeret. Da vil temperaturen inn i brennkammeret være variabel i forhold til eksostemperaturen på turbinen, mens temperatur ut av brennkammer og inn i turbin er satt som en fast verdi.

Ares Turbin rekkefølge:

Kompressor → **Varmeveksler** → **Turbin** → **Brennkammer** → Varmeveksler → Eksos

I syklusen med ekstern forbrenning (Ares-Turbinen), blir temperaturen ut av varmeveksler satt som den faste verdien og temperaturen ut fra brennkammer blir utregnet med hensyn til varmevekslerens virkningsgrad for å gi konstant temperatur inn i turbin. Dette innebærer at de to syklusene vil ha et brennkammer og varmeveksler som er dimensjonert til hver syklus for å oppnå samme virkningsgrad og tapsparameter i komponenten.

Simuleringen av Ares-Turbinen bruker tap og varmeparameter fra simuleringen av Capstone turbinen for å kunne sammenlikne de to turbinkonfigurasjonene opp mot hverandre.

Det er også satt noen begrensninger ved simuleringen av syklus:

- All gass passerer gjennom brennkammer
- Vi antar ingen lekkasje fra kald til varmsiden på roterende varmeveksler.
- Ingen lekkasje ellers i systemet.
- Null varmetap fra komponenter til omgivelsene.
- Regner ikke med massetilførsel fra drivstoff i energiberegninger.
- Tar ikke i betraktning luftfuktighet.
- Antar konstant virkningsgrad for kompressor/turbin, og konstant massestrøm over det angitte trykkområdet.

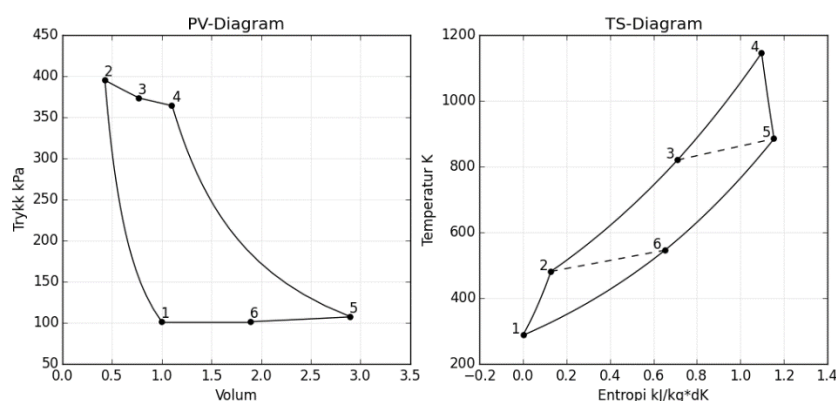
I hoved-simuleringen vil alle parameterne være like for regenerativ og ekstern syklus.

Dessuten kjøres noen simuleringer for parameterintervallene i Tabell 6.2, ettersom parameterne i «benchmarkturbinen» er estimert uten garanti for at de er korrekte.

7. Resultat

7.1. Simulering av Capstone C30, «Benchmarkturbin»:

Det er mulig å automatisere denne prosessen, men det ville vært tidkrevende å sette opp en numerisk analyse med så mange variabler, det ble derfor satt opp en turbinsimulering hvor vi prøvde oss frem i de ukjente intervallene. Etter gjentatte simuleringer med forskjellige variabler kom vi frem til en turbinkonfigurasjon som holder seg innenfor de gitte parameterne i Tabell 6.1. De utregnede Parameterne i Tabell 7.1 blir referansen for simuleringen av Ares Turbinen.



Figur 7.1 PV-TS diagram for Capstone C30

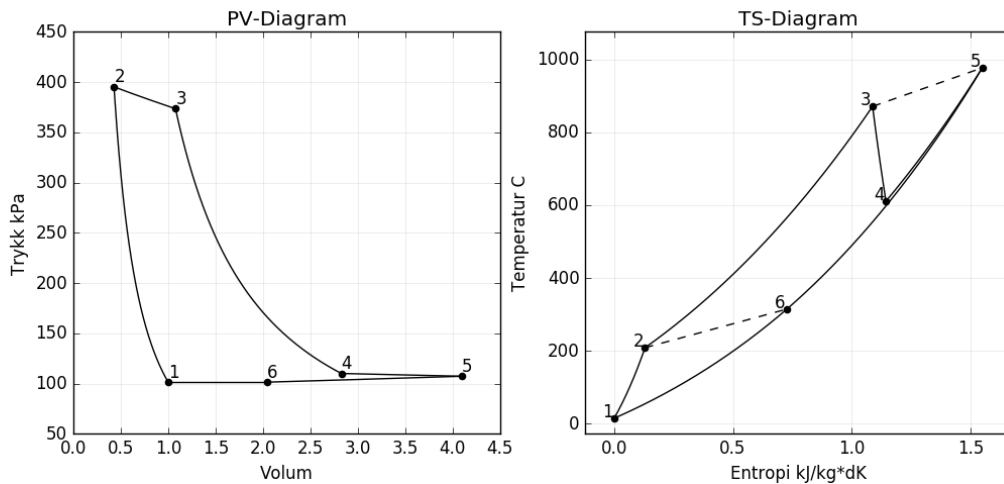
Tabell 7.1 Komplette datasett for «benchmarkturbin» basert på innsamlede data i Tabell 6.1 og egne simuleringer.

Komponent		Parameter	Enhet	Verdi
1-2	Kompressor	Airflow	kg/s	0,31
		Compressor Pr. Ratio		3,9
		Compressor Eff	%	75,1
2-3	Varmeveksler Kald side	He- Inlet Temp	°C	208
		He- Press loss	%	5,5
		He- Efficiency	%	84
3-4	Brennkammer	Combustor Inlet Temp	°C	546
		Combustor Press Drop	%	2.5
		Combustor Eff	%	99
4-5	Turbin	Turbine Inlet Temp	°C	871
		Turbine Eff	%	84
		Turbine Outlet temp	°C	611,8
5-6	Varmeveksler Varm side	He inlet temp	°C	611
		He- outlet = eksos	°C	273
Andre Virk.		Generator inkl mek tap	%	96
		Net el output	kW	29,96
		Net el efficiency	%	26,0
		Input heat	kW	111,5
		He- energy transfer	kW	114

7.2. Simulering av Ares Turbin (Ekstern Forbrenning)

7.2.1. Sammenlinkning med «benchmarkturbin»

I denne simuleringen er alle tapsparametere like i Ares Turbinen som i Capstone turbinen. Simuleringen av syklusen gir en total elektrisk virkningsgrad på 22.7% med en elektrisk effekt ut på 30 kw.



Figur 7.2 PV-TS diagram for regenerativ syklus med ekstern forbrenning

Tabell 2 Utregnet datasett for Ares Turbin med parametere fra simulering av Capstone Turbin

Komponent		Parameter	Enhet	Verdi
1-2	Kompressor	Airflow	kg/s	0,31
		Compressor Pr. Ratio		3,9
		Compressor Eff	%	75,1
2-3	Varmeveksler	He- Inlet Temp	°C	208
		He- Press loss	%	5,5
		He- Efficiency	%	84
3-4	Turbin	Turbine Inlet Temp	°C	871
		Turbine Eff	%	84
		Turbine Outlet temp	°C	610,8
4-5	Brennkammer	Combustor outlet Temp	°C	976
		Combustor Press Drop	%	2.5
		Combustor Eff	%	99
5-6	Varmeveksler Varmside	He inlet temp	°C	977
		He- outlet = eksos	°C	315
Andre Virk.		Generator inkl mek tap	%	96

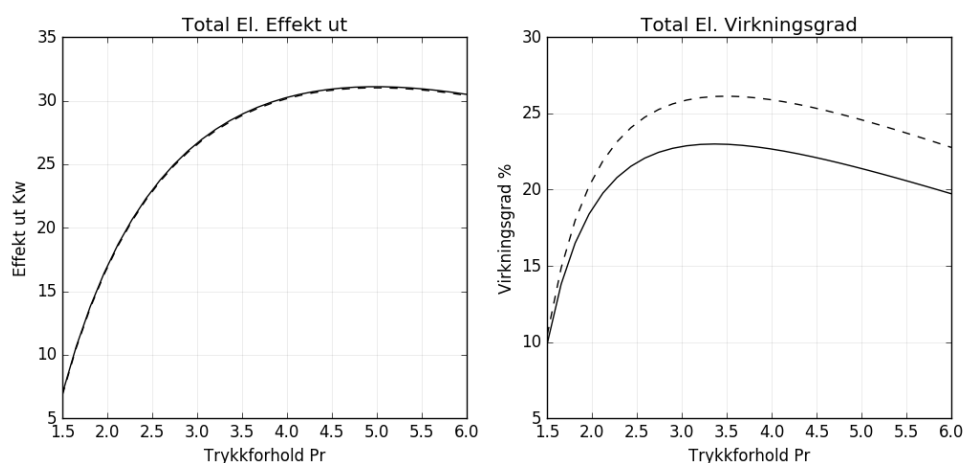
Net el output	kW	30,03
Net el efficiency	%	22,70
Input heat	kW	130,8
He- energy transfer	kW	230,1

Tabell 2 viser det utregnede datasettet for Ares Turbinen, det som er verdt å merke seg er at temperatur ut av brennkammer må være 100 grader høyere enn benchmarkturbinen, Dette kommer av at temperatur inn i turbin holdes konstant som igjen betyr at tapet i varmeveksleren kompenseres med en høyere forbrenningstemperatur, dette gjøres mulig siden vi anntar et brennkammer og varmeveksler optimalisert for hver syklus. Det andre som er verdt å merke seg er at varmeveksleren må overføre 230 kw med en inntemperatur på 977°C for å opprettholde en temperatur inn i brennkammeret på 871°C. Med en eksostemperatur på 315 grader og en massestrøm på 0.31 kg/s, med en luft til vann varmeveksler som har 80% virkningsgrad. Vil energien som kan benyttes i et CHP anlegg være:

$$\eta C_p m_{luft} (T_{h1} - T_{c1}) = 0,8 * 1,005 \frac{kJ}{kg K} * (315^{\circ}C - 15^{\circ}C) * 0,31 kg/s = 74kw$$

7.2.2. Simulering med variable parameter.

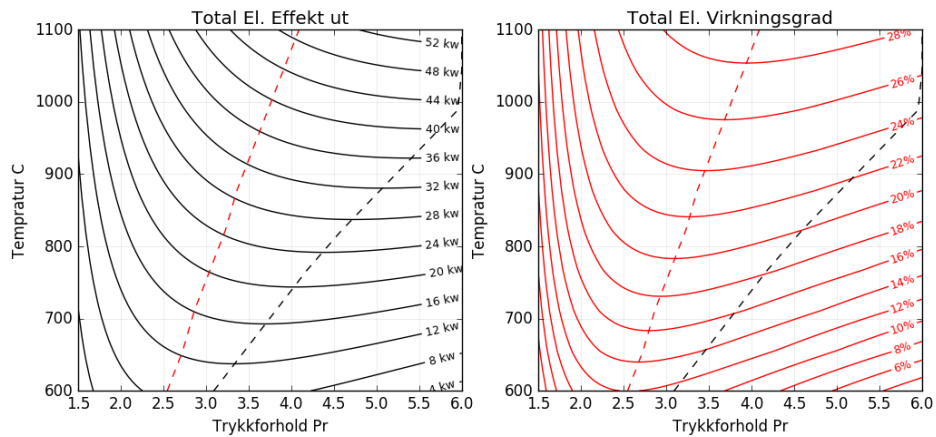
Ettersom datasettet fra «benchmarkturbinen» er estimert, ble det også simulert hvordan virkningsgrad og effekt ut vil respondere på variasjon i parameterne for Ares Turbinen. Simuleringene endrer kun på parameterne nevnt i aksene, og det resulterende plottet er et resultat av 200 simuleringer i akseretning for 2d plot, og 100x100 simuleringer for 3d plot. Disse simuleringene vil også være viktige for å finne designparametere, samt se trender i syklusen.



Figur 7.3 Effekt ut og virkningsgrad for forskjellige kompresjonsforhold ved fast innløpstemperatur på 871°C, stiplet linje er Capstone Turbin mens heltrukket linje er Ares Turbin.

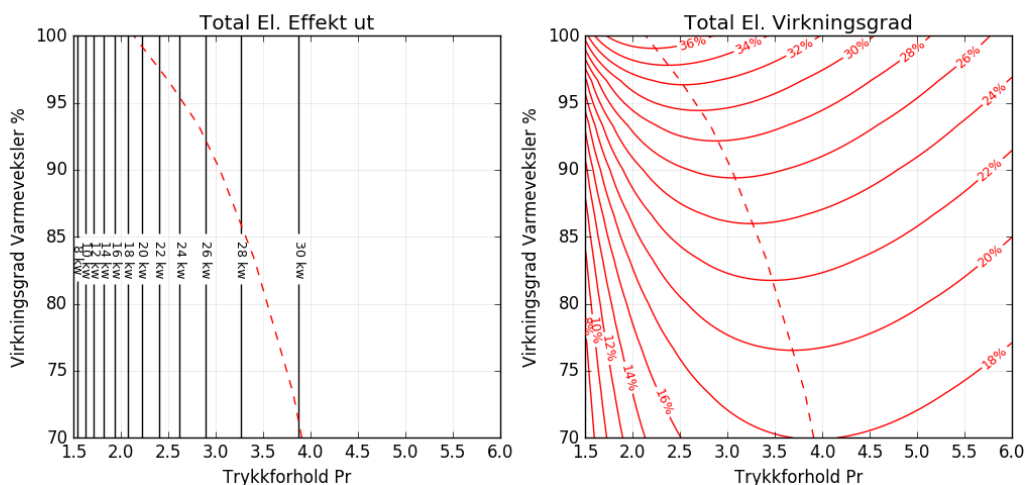
Figur 7.3 viser hvordan syklusen reagerer på varierende trykkforhold ut av kompressor mens alle andre parametere holdes konstant. Det kan observeres at maksimalnivået for effekt ut ligger ved et trykkforhold på 4,7 mens virkningsgrad har toppunktet ved 3,4.

For å finne ut om optimalpunktet for virkningsgrad og effekt ut også er avhengig av temperatur ble det plottet et 3d konturplott der effekt og virkningsgrad vises som en funksjon av temperatur og trykkforhold. Konturplottet er et resultat av 100 plot tilsvarende Figur 7.3 for temperaturnivå fra 600-1100 °C.



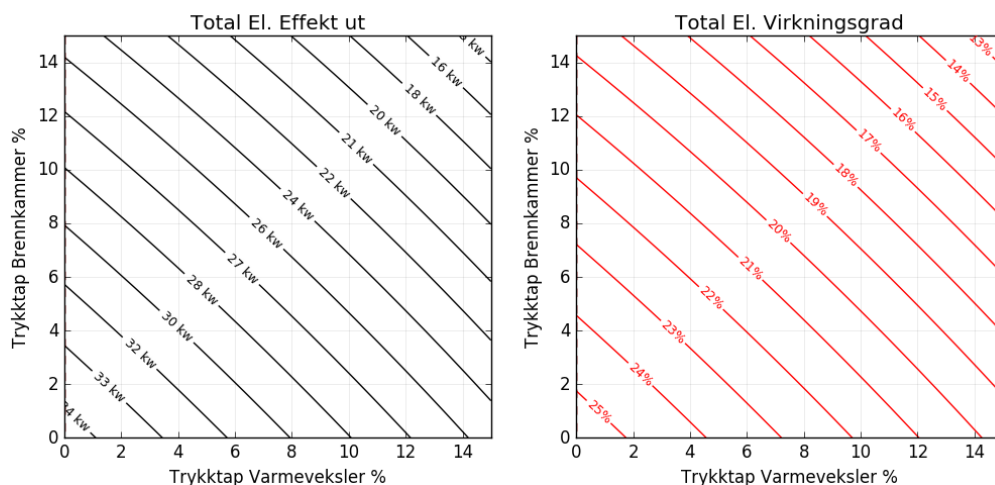
Figur 7.4 Konturplott av hvordan effekt ut og virkningsgrad forandrer seg ved variasjon i trykkforhold og temperatur, stiplet rød linje er maksimal virkningsgrad, og stiplet svart linje er maksimal effekt ut

Optimalpunktet for elektrisk virkningsgrad vil også variere med virkningsgraden til varmeveksler, altså den prosentmessige overføringen av energi fra kald til varmside i en varmeveksler som er dimensjonert for syklusen. Det ble da simulert 100 virkningsgrader fra 70% til 80% med 100 trykkforhold fra 1.5 til 6.0.



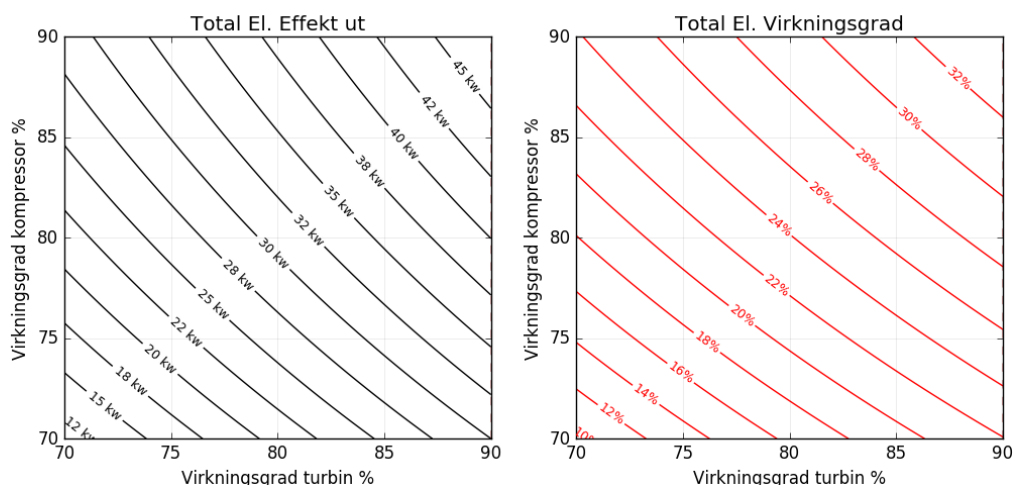
Figur 7.5 Variasjon i effekt ut og elektrisk virkningsgrad ved forandring i virkningsgrad varmeveksler og trykkforhold. Stiplet rød linje er optimalt trykkforhold for en gitt virkningsgrad for varmeveksler.

Fra Figur 7.2 ser vi at volumstrømmen gjennom varmeveksler 4→5 og brennkammer 5→6 er mye større enn i den regenerative syklusen, økningen av volum er direkte relatert til gass hastigheten gjennom komponentene og det vil da bli feil å bruke det samme trykktapet for begge syklusene. Derfor plottet vi total effekt ut og virkningsgrad som et forhold av trykktapet i varmeveksler og brennkammer så dette kan kompenseres for i ettertid.



Figur 7.6 Påvirkning av trykktap i brennkammer og varmeveksler. Alle andre parameter holdes konstant.

Gitt at verdiene for virkningsgraden til kompressor og turbin ikke er estimert fullstendig korrekt har vi også laget et konturplot som kompenserer for varierende virkningsgrader for disse komponentene.



Figur 7.7 Effekt ut og total virkningsgrad i forhold til variasjon i virkningsgrad til kompressor og turbin. Alle andre parameter holdes konstant.

7.3. Evaluering av prototype og testprosedyre

7.3.1. Verifisering av virkningsgrad.

Da forsøket skulle igangsettes ville ikke turbinen gå over 40'000 rpm grunnet kraftige vibrasjoner i gassturbinen som ville ødelegge generatormodulen hvis forsøket hadde fortsatt. Vi trodde dette skyldtes egensvingninger i akselen til generatormodulen, men etter testing med trykkluft uten generator oppstod det fortsatt svært store vibrasjoner som hindrer turbinen i å komme opp i driftsturtall. Ettersom det ikke var mulig for oss økonomisk sett å anskaffe en erstatningsturbin på kort varsel, ble forsøket avbrutt.

7.3.1. Nye komponenter

Brennkammer:

Brennkammeret fungerte godt montert som enkelsyklus gassturbin. Når det ble kjørt i regenerativ syklus med ekstern forbrenning (Ares Turbin Konfigurasjon) var det problematisk å opprettholde stabil forbrenning ettersom det var fort gjort at brennkammeret fikk «flamout» Dette tydet på at strømningshastigheten i brennkammeret var for høy til å opprettholde flamme. I tillegg var det vanskelig å holde en stabil temperatur ut fra brennkammeret.

Generator:

Generatormodulen fungerte overraskende bra teknisk sett, men det var tungvint å måtte skifte over fra styringselektronikken til diodebroen under drift. Når turbinen ble utprøvd som enkelsyklus turbin ble temperaturen på kompressorhuset og overgangen til generatorhuset målt til 50 grader. Spenningen ut fra generator ble målt til å være ca 3V under forventet spenning over hele turtallsregisteret.

7.3.2. Testprosedyre.

Måleparameter i testprosedyren ble utprøvd uavhengig av selve forsøket. Måleparameterne var gjennomførbare og fungerte tilfredsstillende, men det ble avdekt et problem i designet til den variable lasten for effektmåling. Når den ble tilkoblet en spenningskilde på 12V på første tilkoblingspunkt (0.3Ω) ble effekten ut umiddelbart målt til 470W, men når elementet varmet seg opp sank effekten ut frem til temperaturen kom i likevekt med omgivelsene ved 350W. Forsøket ble også utført med elementet nedsenket i vann for nedkjøling av ståltråden, og samme trend ble observert men i lavere grad. Dette problemet var også mye mindre ved lav belastning. Et annet problem med effektmålingen er at kretsen åpnes når effekt forandres, dette medfører at generatoren vil drives uten last frem til koblingen slutes på nytt. Ved høy effekt vil generatoren gå fra ingen til full last umiddelbart, dette kan være skadelig for aksel og turbin.

8. Diskusjon

8.1. Vurdering av ekstern forbrenning fra simulering

Fra resultatene ser vi at virkningsgraden for denne syklusen blir 22,7% ved 30kw. Dette resultatet krever i midlertid en varmeveksler som har over dobbelt så høy kapasitet (230 kw) som den eksisterende varmeveksleren i capstoneturbinen (114 kw), i tillegg til at den må tåle en temperatur inn på nesten 975 grader. Dette er høye verdier, men er ikke umulig å få til. Det finnes roterende keramiske varmevekslere for salg som klarer disse temperaturene med opptil 5 bars trykkdifferanse mellom varm og kaldside [39].

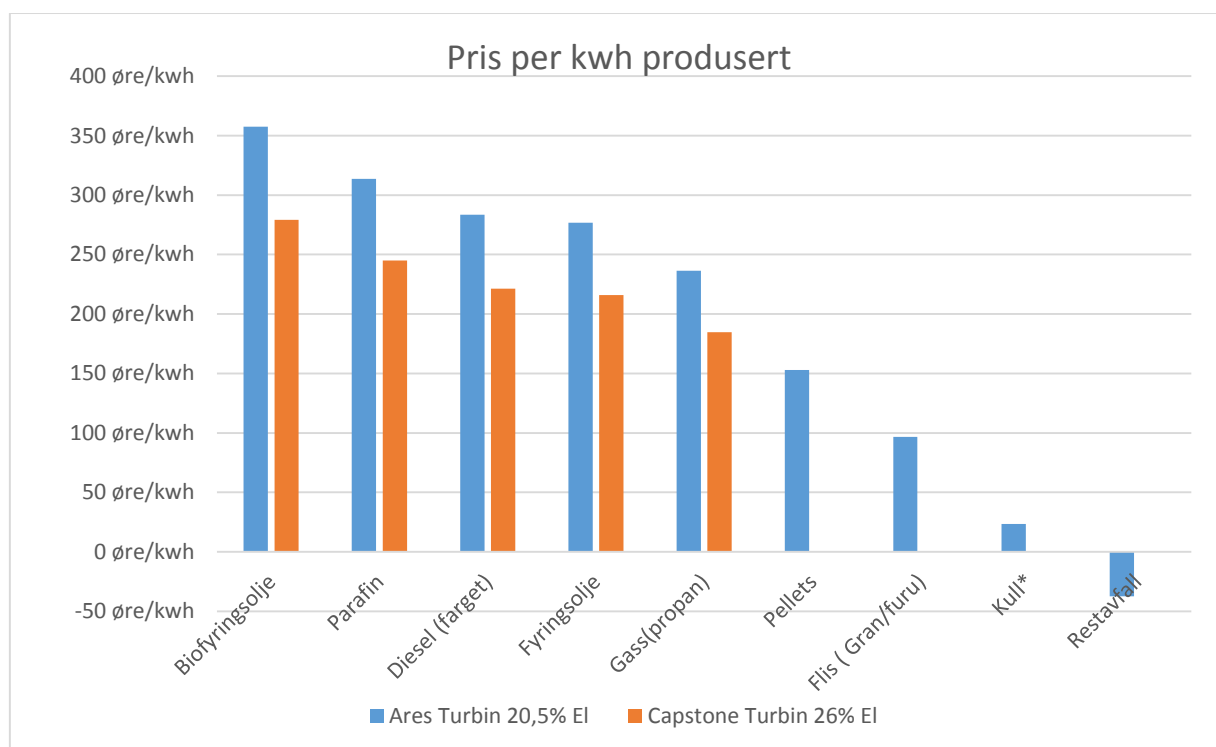
Det er funnet 2 studier som har sett på liknende oppsett som Aresturbinen, Begge basert ut i fra turbec turbinkjerne og platevarmevekslere. Den teoretiske vurderingen skrevet av Martin Kautz et all [40] konkluderer med en potensiell elektrisk virkningsgrad på 27.8% ved en elektrisk effekt ut på 100kW. Artikkelen skrevet av Alberto Traverso et all [41] er et praktisk forsøk som benytter to platevarmevekslere og en elektrisk virkningsgrad på 28% ved 80kw ut. Årsaken til at virkningsgraden for simuleringen og forsøket er høyere enn ved vår simulering, er mest sannsynlig en kombinasjon av flere faktorer. Innløpstemperaturen i Turbecturbinen er satt til 950°C i motsetning til 870°C i vårt tilfelle. Varmeveksleren brukt i forsøkene har nærmere 90% virkningsgrad i motsetning til 84% som vi har brukt. I tillegg har større gassturbiner generelt bedre virkningsgrad for kompressor og turbin enn små gassturbiner. Sammenlagt vil dette gi en høyere virkningsgrad enn for 30kw systemet som ble simulert.

Fra Figur 7.3 og Figur 7.4 ser vi hvordan virkningsgraden varierer med trykkforholdet til kompressoren. For å oppnå en optimal virkningsgrad for syklusen burde trykkforholdet reduseres til 3.5, noe som ville gitt en virkningsgrad på litt over 23.5 prosent. Denne økningen av virkningsgrad vil gå på bekostning av effekt levert fra gassturbinen som reduseres til 29kw. Det må nevnes at forandringen i trykkforhold i simuleringen ikke tar hensyn til varierende massetransport og virkningsgrad for den gitte kompressoren ettersom vi ikke har kompressorkart tilgjengelig.

PV diagrammet i Figur 7.2 viser en stor økning i gassvolum gjennom varmeveksler og brennkammer fra 4→5→6. Gjennom varmeveksleren er volumet på gasstømmen mer en doblet i forhold til den regenerative syklusen vist i Figur 7.1. I henhold til kap 2.7.2 vil en dobling i strømningshastigheten gjennom det samme tverrsnittet føre til en firedobling av trykkfallet. Dette kan kompenseres for ved å bruke en større varmeveksler, og bygge et brennkammer optimalisert for volumstrømmen.

På bakgrunn av dette vil det være et større trykkfall enn i den vanlige regenerative syklusen, hvis vi antar en dobling av trykkfallet vil dette tilsvare hhv 5% og 9% over brennkammer og varmeveksler. Den resulterende totale virkningsgraden blir da 20,5% med en strømproduksjon på 26kw i henhold til Figur 7.6. Dette er mest sannsynlig en mer realistisk verdi for virkningsgraden til syklusen.

Den virkelige fordelene med denne turbinen er det faktum at vi kan bruke alle drivstoff, inkludert partikulære drivstoff som restavfall, flis, pellets og kull. Dette kan medføre en betraktelig kostnads-reduksjon i produksjon av elektrisitet i mikrogassturbiner. Bare ved å kunne bruke pellets istedenfor propan kan det resultere i en 17% reduksjon av drivstoffkostnader basert på tallene i Figur 8.1. Ved forbrenning av flis vil drivstoffkostnader reduseres 47%, og ved forbrenning av restavfall vil man få betalt 38 øre per kwh for å ta imot brennstoffet.



Figur 8.1 Pris per kwh strøm produsert med Ares Turbin ved 20,5% el virkningsgrad, mot capstoneturbin ved 26% el virkningsgrad [8] [9] [10] [11] [12] Utrekninger i Vedlegg D

Det må nevnes at prisene i Figur 8.1 ikke tar hensyn til at varmen ut fra gassturbinen kan gjenvinnes i en CHP konfigurasjon (kap 2.9), for oppvarming av vann i et vannbårent varmeanlegg. En håndberegning anslår at eksosen fra gassturbinen leverer 74kW i varmeenergi som kan brukes til oppvarming.

8.2. Simuleringsprogram

Utviklingen av simuleringsprogrammet var den mest tidkrevende prosessen i hele oppgaven, men ga god innsikt i hvordan samspillet mellom de ulike komponentene i en gassturbin fungerer. I tillegg var simuleringsresultatene i overenstemmelse med virkelige turbiner og tilgjengelig simuleringer fra annen programvare.

Men programmet er ikke uten feil og mangler. Det mest åpenbare er at simuleringene baserer seg på endringer i selve kildekoden for å simulere forskjellige sykluser. Dette gjør det tungvint å sette opp nye sykluser uten kjennskap til hvordan funksjonene skrives og hvilke variabler som må defineres for å få en kjørbare syklus, men det er flere ferdiglagde turbinsykluser og parameteroppsett i programmet som viser hvordan en syklus er oppbygd. Den beste løsningen hadde vært et diagrambasert brukergrensesnitt der modulene kunne dras rundt i forskjellige konfigurasjoner.

Det er også funksjoner det ikke ble mulighet til å legge inn grunnet tidsmangel, blant annet har ikke programmet mulighet til å ta hensyn til luftfuktighet i energiberegningene. Det er heller ingen måte å simulere lekkasje av luft i syklusen. Det ble forsøkt å estimere kompressorkart, men det ga ikke tilfredsstillende resultater og funksjonen ble ikke brukt i simuleringene.

8.3. Utvikling av prototyp og testopplegg

8.3.1. Prototyp og brennkammer:

Turbinen som ikke ville gå over 40'000 rpm er mest sannsynlig et resultat av lagerslitasje som oppstod grunnet varmgang og tap av oljetrykk da turbinen ble kjørt som enkelsyklus gassturbin vist i Figur 3.19. Det ble observert klare slitasjemerker i bronselageret og turbinakselen hadde slingringsmonn på nesten 0.5 mm i lageret. Den eneste løsningen på dette er å kjøpe en ny turbin.

Under forsøket ble det også klart at brennkammeret var vanskeligere å kontrollere når det var satt opp i ekstern forbrenning. Volumøkningen som er ble observert i simuleringen kan også være en viktig faktor i hvorfor prototypen ikke ville gå i denne konfigurasjonen. Datagrunnlaget for brennkammeret var laget med hensyn til intern forbrenning, og designet vi lagde ble simulert med 2 bars mottrykk og temperatur direkte fra kompressor. Fra resultatene ser vi at gasstrømmen ut av turbin er i friluft med en temperatur på over 600 grader. Dette gir en volumstrøm mer enn 6 ganger større enn i simuleringen. Det er egentlig et under at brennkammeret overhodet opprettholdt en forbrenning med en så stor strømningshastighet.

I ettertid ser jeg derfor at det hadde vært en stor fordel og laget simuleringsprogrammet og kjørt simuleringene før utvikling av prototyp

8.3.2. Generator.

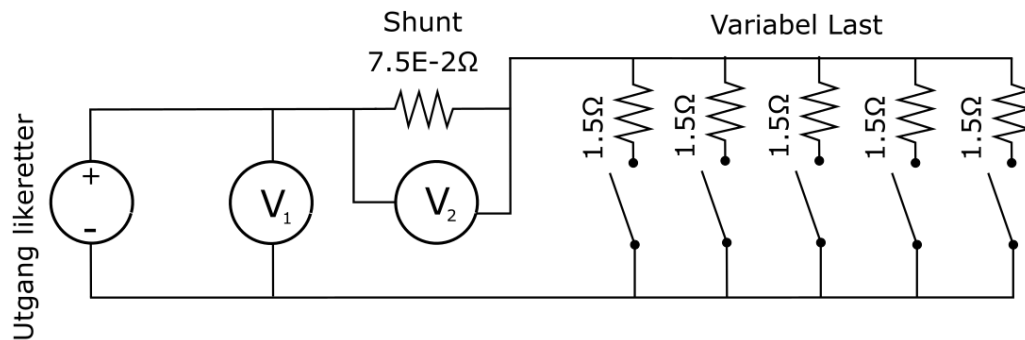
Faren for overoppheting er et reelt problem for overgangen mellom kompressorhuset og generatorhuset. Overgangen er konstruert i ABS-plast med en arbeidstemperatur på under 70°C som medfører at temperaturen må overvåkes og forsøket avbrytes hvis temperaturen blir for høy. Hvis overgangen blir konstruert i aluminium vil den bli nedkjølt av luftstrømmen som passerer gjennom generatormodulen, i tillegg til at ytterrøret er konstruert av POM, med en arbeidstemperatur opp mot 90°C. En slik endring vil øke tidsrommet turbinen kan kjøres før nedkjøling er nødvendig.

Det ble observert en lavere spenning ut fra generator enn forventet, dette spenningsfallet på 3V tilsvarer diodespenningen til likeretterbroen på 2x1,5V for RHRG5060. I forhold til testopplegget er dette egentlig uproblematisk ettersom vi kan måle spenningstapet over diodebroen og medregne dette sammen med effekt ut. Om det er ønskelig kan spenningsfallet også forminskes ved å benytte schottky dioder som har en mye lavere diodespenning, for eksempel VS-80CPQ020PBF som har en diodespenning på 0.3V

8.3.3. Effektmåling – Variabel last

Det ble observert en fallende effekt når ståltråden i den variable lasten nevnt i kapittel 4.1.3 ble oppvarmet. Dette er problematisk hvis vi skal ha mulighet til å kjøre turbinen med maks last ettersom lasten da må starte på et mye høyere nivå som stabiliserer seg på målnivået når last er oppvarmet. Generatoren må da produsere høyere effekt enn den er produsert for frem til lasten stabiliserer seg, dette kan skade generator og aksel.

Fallet i effekt kommer av at motstanden til stål øker med temperaturen. Reostaten øker effekten ut ved å bruke en kortere andel av ståltråden som minker motstanden. Den økte effekten vil være fordelt over en kortere lengde ståltråd som varmes opp i større grad ved stor effekt ut. Ved lav effekt ut vil ståltråden være lengre og en lavere effekt vil fordele seg over en større lengde ståltråd som gir en mindre temperaturøkning, derav en lavere økning av motstand.



Figur 8.2 Forslag til modifisering av last

Dette kan unngås ved å benytte en parallellkobling istedenfor en seriellkobling av lasten. Det benyttes fem elementer på $1,5 \Omega$ som kobles inn i parallell vist i Figur 8.2, som vil gi en variabel last mellom $150W$ - $750W$ ved 15 volt inn. Fordelen med denne løsningen er at hvert element vil gi ut $150W$ over hele lengden av ståltråden. I tillegg vil denne metoden gi en gradvis økning av lasten ettersom ett og ett element kan kobles inn ved hjelp av knivbrytere.

9. Videre Arbeid

Det ble avdekket viktige elementer som må forbedres eller utvikles på nytt i alle delene av utviklingen.

9.1.1. **Prototype og testprosedyre**

- Kjøpe inn ny turbo for bruk i prototypen.
- Konstruere nytt brennkammer optimalisert for forbrenning i atmosfæretrykk med temperatur ut fra turbin.
- Konstruere overgang fra kompressor til generatormodul i aluminium. .
- Konstruere ny effektmåler basert på den modifiserte lasten nevnt i kap 8.3.3.

9.1.2. **Program og simulering**

- Oppdatere funksjon for energiberegninger til å ta hensyn til fuktighetsnivå i luften.
- Programmet har ingen måte å simulere lekkasje i systemet. Dette er et viktig element og som bør tas med i beregningene
- Innhente eller estimere kompressor og turbinkart for turbiner som skal simuleres. Programmet har allerede funksjoner for å uthente data fra kompressor/turbinkart, men vi har ingen verdier for disse parametrene
- Lage et grafisk brukergrensesnitt som forenkler oppsett av turbinsykluser.

10. Konklusjon

En Capstone C30 som ombygges til regenerativ ekstern forbrenning gir mulighet til å forbrenne drivstoff med partikulær aske som pellets, flis, kull og avfall. Et slikt oppsett vil ha en teoretisk virkningsgrad mellom 20.5-22.7% med 26-30kw elektrisk effekt basert på innhentede og estimerte data fra håndbøker og andre kilder. Et slikt system vil kunne produsere elektrisitet med kostnad på hhv 153 øre/kWh ved forbrenning av pellets, 97 øre/kWh ved forbrenning av flis og -38 øre/kWh ved forbrenning av restavfall. I pris-utregningene er det ikke tatt hensyn til at eksosvarmen kan levere 74 kW til vannbåren varme gjennom et CHP anlegg, som vil redusere energikostnaden ytterligere

Programmet laget for å verifisere virkningsgraden har potensiale for å forbedres, men fungerer godt for grunnleggende analyser av flere forskjellige gassturbinsykluser.

Det er utviklet en prototyp som skulle brukes til å verifisere teorien brukt i vurderingen men forsøket ble ikke gjennomført grunnet tekniske problemer. I tillegg er det foreslått et testopplegg for verifisering av elektrisk virkningsgrad samt innhenting av andre parameter for forbedringer og evaluering.

11. Referanser

- [1] D. Eckardt, «Gas Turbine Powerhouse: The Development of the Power Generation Gas Turbine 2.nd edition,» Oldenbourg Wissenschaftsverlag GMBH, Munchen, 2014.
- [2] P. B. L. B. Halfdan Wiig, «Vurdering av biodrivstoff i transportsektoren,» Insa Institutt for strategisk Analyse, Ski, 2009.
- [3] I. Gas Turbine Parts & Services, «<http://www.gtps-inc.com/>,» Gas Turbine Parts & Services, Inc. , 2016. [Internett]. Available: <http://www.gtps-inc.com/>. [Funnet 26 Jan 2016].
- [4] O. T. AS, «<http://www.opraturbines.com/en/>,» Opra Technologies AS, 2015. [Internett]. Available: http://www.opraturbines.com/upload/products/9070_corporate_opra07.pdf. [Funnet 28 Jan 2016].
- [5] M. P. Boyce, Gas Turbine Engineering Handbook - 2.nd edition, Gulf Professional Publishing, 2001.
- [6] R. Reeves, «MassEngineers,» MassEngineers, [Internett]. Available: http://www.massengineers.com/turbines%20CPH/steam_turbines_used_for_chp.htm. [Funnet 28 Jan 2016].
- [7] International Energy Agency, «Key World Energy Statistics,» Chirat, France, 2015.
- [8] K. F. M. S. H. Eivind Magnus, «Status for energiutnyttelse av avfall i Norge,» Avfall Norge, 2014.
- [9] Energi1Olje, «<http://energi1olje.no/>,» energi1olje, [Internett]. Available: <http://energi1olje.no/>. [Funnet 20 April 2016].
- [10] Shell, «www.shell.no,» Shell, 28 April 2016. [Internett]. Available: <http://www.shell.no/products-services/solutions-for-businesses/services/fuel-pricing.html>. [Funnet 29 April 2016].
- [11] Statoil, «www.statoil.com,» Statoil, 1 mars 2016. [Internett]. Available: <http://www.statoil.com/no/InvestorCentre/AnalyticalInformation/InternalGasPrice/Pages/default.aspx>. [Funnet 29 April 2016].
- [12] Norsk Gartnerforbund, «BIOBRENSLER,» Norsk Gartnerforbund, 2014.
]
- [13] P. E. B. Richard Gentle, Mechanical engineering systems, Oxford: Butterworth Heinemann, 2001.
- [14] J. R. W. A. Allan D. Kraus, Introduction to thermal and fluid engineering, Boca Raton: CRC Press, 2012.

- [15 J. B. F. E. John Finnemore, Fluid Mechanics with engineering application 10th. Edition,
] New York: Mc Graw Hill, 2009.
- [16 M. p. Boyce, Gas Turbine Engineering Handbook 4th. Edition, Waltham: Butterworth
] Heinemann, 2012.
- [17 R. K. Rajput, Thermal Engineering 4th. edition, New Delhi: Laxmi Publication LTD,
] 2010.
- [18 G. F. C. R. H. S. H. Cohen, Gas Turbine Theory, Essex, England: Longman Group
] Limited, 1996.
- [19 PACT, «www.pact.ac.uk,» PACT , 2013. [Internett]. Available:
] <http://www.pact.ac.uk/facilities/PACT-Core-Facilities/Gas-Turbine/gas-turbine-system>.
[Funnet 25 Januar 2016].
- [20 M. A. R. d. Nascimento, L. d. O. Rodrigues, E. C. d. Santos, E. E. B. Gomes, F. L. G.
] Dias, E. I. G. Velásques og R. A. M. Carrillo, Micro Gas Turbine Engine: A Review ISBN
978-953-51-1166-5, InTech, 2013.
- [21 N. Hall, «www.grc.nasa.gov,» 5 Mai 2015. [Internett]. Available:
] <https://www.grc.nasa.gov/www/k-12/airplane/compth.html>. [Funnet 02 03 2016].
- [22 C. Soares, Gas Turbines: A handbook of land, air and sea applications 2.nd edition,
] Oxford: Butterworth Heinemann, 2008.
- [23 G. Brown, «bae.okstate.edu,» Oklahoma State University, 22 06 2000. [Internett].
] Available: <https://bae.okstate.edu/faculty-sites/Darcy/DarcyWeisbach/Darcy-WeisbachEq.htm>. [Funnet 15 03 2016].
- [24 N. Petchers, Combined Heating, Cooling & Power Handbook: Technologies and
] Applications, Lilburn, Ga: The Fairmount Press, 2002.
- [25 Burbank~enwiki, «wikipedia.com,» 24 Juli 2006. [Internett]. Available:
] <https://en.wikipedia.org/w/index.php?curid=6095994>. [Funnet 1 Mai 2016].
- [26 P. Janshon, Modern Gas Turbine Systems, Cambridge: Woodhead publishing, 2013.
]
- [27 A. H. Lefebvre, Gas Turbine Combustion 2nd Edition, Philadelphia: Taylor and Francis,
] 1998.
- [28 P. F. Philip Walsh, Gas Turbine Performance 2nd. Edition, Oxford: Blackwell Science
] Ltd, 2004.
- [29 D. P. S. Ramesh K Shah, Fundamentals of Heat Exchanger Design, Hoboken: John Wiley
] & Sons, 2003.
- [30 O. B. kleven, «Gas turbine with external combustion, applying a rotating regenerating heat
] exchanger». Norge Patentnr. WO 2010064921 A1, 24 Nov 2008.

- [31 O. B. Kleven, *Bilde*, Hol, 2009.
]
- [32 E. Thirunavukarasu, «Modeling and Simulation Study of A Dynamic GasTurbine System
] In A Virtual Test Bed Environment (Master's thesis),» University of South Carolina, 2013.
- [33 N. A. Centre, «<http://www.gspteam.com/>,» Netherlands Aerospace Centre, 2016.
] [Internett]. Available: <http://www.gspteam.com/>. [Funnet 15 04 2016].
- [34 ASME, «<http://www.allstar.fiu.edu/>,» ALLSTAR Network, 12 03 2004. [Internett].
] Available: <http://www.allstar.fiu.edu/aero/turbine3.html>. [Funnet 15 03 2016].
- [35 M. D. S. H. C. EATOCK, «Design Study of an Advanced Concept Simple Gas Turbine
] for Possible Use in Low Emission Automobiles,» *ASME*, p. 3, 1972.
- [36 Microturbine Marine Energy, «Microturbine Marine Energy,» ASAV Projektentwicklung,
] Munich, [Internett]. Available: <http://www.mme-generators.com/general-information>.
[Funnet Mai 2 2016].
- [37 Nasa, «[Everystockphoto.com](http://www.everystockphoto.com/),» Nasa, [Internett]. Available:
] <http://www.everystockphoto.com/photo.php?imageId=2022514>. [Funnet 2 Mai 2016].
- [38 C. T. Corperation, «Technical Reference Capstone model c30 Performance,» Capstone
] Turbine Corporation, Chatsworth , 2006.
- [39 Wilson Solar Power, «<http://www.wilsonsolarpower.com/>,» Wilson Solar Power, 2016.
] [Internett]. Available:
<http://www.wilsonsolarpower.com/nonsolarapps/heatexchanger.html>. [Funnet 12 Mai
2016].
- [40 U. H. Martin Kautz, «The externally-fired gas-turbine (EFGT-Cycle) for decentralized use
] of biomass,» *Applied Energy*, nr. Volume 84, pp. 795-805, 2007.
- [41 A. F. M. R. S. Alberto Traversoa, «Externally Fired micro-Gas Turbine: Modelling and
] experimental performance,» *Applied Thermal Engineering*, vol. Volume 26, pp. 1935-
1941, 2006.
- [42 Duk, «[WikiMedia](https://commons.wikimedia.org/wiki/File:Brayton_cycle.svg),» 2015. [Internett]. Available:
] https://commons.wikimedia.org/wiki/File:Brayton_cycle.svg.
- [43 RingAlm Romerike AS, «<http://renflis.no/>,» RingAlm Romerike AS, 2016. [Internett].
] Available: <http://renflis.no/norsk/priser/>. [Funnet 29 April 2016].

Vedleggs-liste

Vedlegg A:

Beregninger for reostat

Vedlegg B:

Oppsett av simulering

Vedlegg C:

Eksempler på simulering

Vedlegg D:

Beregninger av energipriser

Vedlegg E:

Beregninger av energipriser

Vedlegg A – Beregninger Reostat

Motstand stål 6,897E-07 Ohm*m²/m
Diameter 1,2 mm
Areal 4,52E-06 m²
0,152 ohm/m

lengde ståltråd	2	3	4	5	6	7	8	9	10
Tot motstand	0,30	0,46	0,61	0,76	0,91	1,07	1,22	1,37	1,52

volt	15	15	15	15	15	15	15	15	15
ampere	49,19	32,80	24,60	19,68	16,40	14,06	12,30	10,93	9,84
Watt	737,91	491,94	368,96	295,16	245,97	210,83	184,48	163,98	147,58

Vedlegg B - Oppsett av simulering

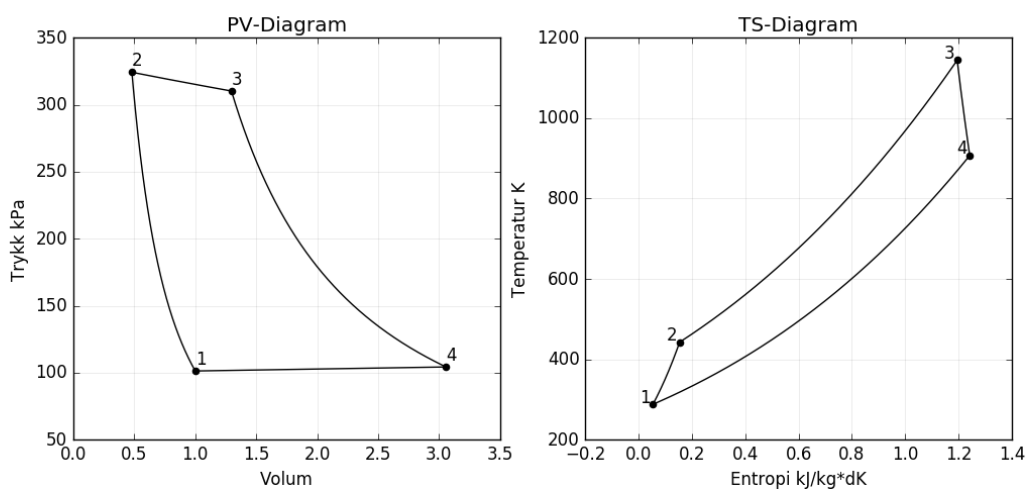
Hvordan enkel-syklus turbin settes opp i programmet

```
1. def NormalCycle():
2.     places = [0,1,2,3]
3.
4.     TVSP = [[[T_amb], [V_0], [s_0], [p_0]]]
5.
6.     TVSP = InitialiseTurb()
7.
8.     TVSP,workIn = Compressor(TVSP,Pr,n_c)
9.
10.    PipeLoss(TVSP, n_pr)
11.
12.    TVSP,heatIn = CombChamb(TVSP,Tit, n_pc)
13.
14.    TVSP,workOut = Turbine(TVSP,p_0,n_ext,n_t)
15.
16.    TVSP,heatOut = Exhaust(TVSP,T_amb,n_ext)
17.
18.    n_tot, P_out, P_eff = Eff2(workIn, workOut, heatIn)
19.
20.    return TVSP[1:], P_out, n_tot, P_eff,places
```

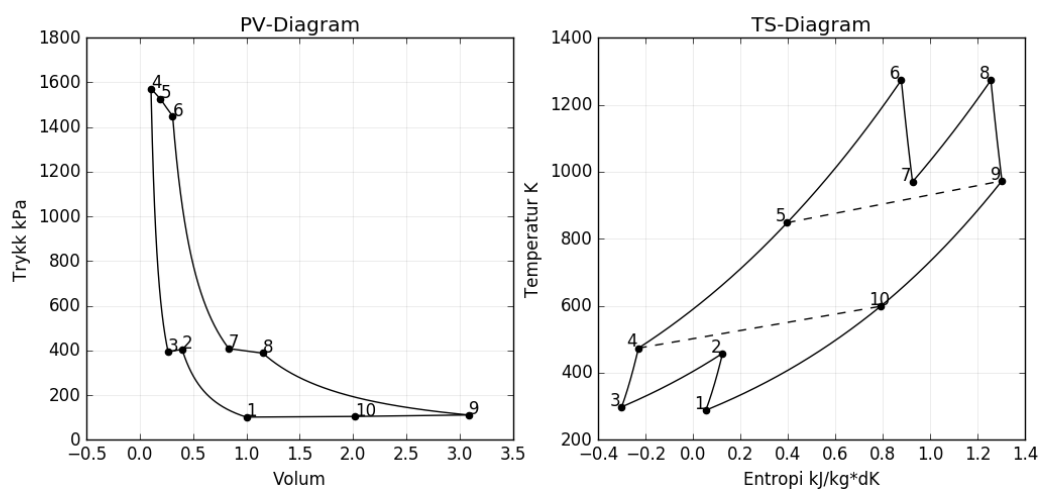

Vedlegg C - Eksempler på simulering

Avhengig av konfigurasjonene kan programmet simulere flere forskjellige gassturbin typer, alt fra enkelsyklus lavtrykksturbin, til mer avanserte flertrinns gassturbiner med nedkjøling og varmeveksling mellom kompressor og turbintrinn.

Simulering av enkelsyklus lavtrykksturbin:



Simulering av tottrinns gassturbin med varmeveksler og intercooler.



Vedlegg D – Beregning av energipris

Type Brennstoff	Brennverdi	Egenvekt	pris/enhet	Pris / kwh	Pris eks mva	Ares Turbin		Besparelse
						20,5% EI	Capstone Turbin 26% EI	
Gass (LNG)*	10kw/sm3	-	1239 Øre/sm3	123,9 Øre/kwh	99,1 Øre/kwh	488 Øre/kwh	381 Øre/kwh	
Biofyringsolje	11,5 kw/kg	0,870 kg/l	908 Øre/l	90,8 Øre/kwh	72,6 Øre/kwh	358 Øre/kwh	279 Øre/kwh	
Parafin	12,0 kw/kg	0,803 kg/l	767 Øre/l	79,6 Øre/kwh	63,7 Øre/kwh	314 Øre/kwh	245 Øre/kwh	
Diesel (farget)	11,6 kw/kg	0,851 kg/l	710 Øre/l	71,9 Øre/kwh	57,6 Øre/kwh	284 Øre/kwh	221 Øre/kwh	
Fyringsolje	11,9 kw/kg	0,845 kg/l	706 Øre/l	70,2 Øre/kwh	56,2 Øre/kwh	277 Øre/kwh	216 Øre/kwh	
Gass(propan)				60,0 Øre/kwh	48,0 Øre/kwh	236 Øre/kwh	185 Øre/kwh	
Pellets	4,9 kw/kg	0,650 kg/l	190 Øre/kg	38,8 Øre/kwh	31,0 Øre/kwh	153 Øre/kwh		17,23 %
Flis (Gran/furu)	1,9 kw/kg		47 Øre/kg	24,5 Øre/kwh	19,6 Øre/kwh	97 Øre/kwh		47,64 %
Kull*	8,1 kw/kg		48 Øre/kg	5,9 Øre/kwh	4,7 Øre/kwh	23 Øre/kwh		87,35 %
Restavfall	2,1 kw/kg		-20 Øre/kg	-9,5 Øre/kwh	-7,6 Øre/kwh	-38 Øre/kwh		120,33 %

<http://energi1olje.no/>

<http://www.shell.no/products-services/solutions-for-businesses/services/fuel-pricing.html>

<http://www.statoil.com/no/InvestorCentre/AnalyticalInformation/InternalGasPrice/Pages/default.aspx>

<http://www.ngfenergi.no/sites/default/files/files/Temahefte%20Biobrenslar.pdf>

<http://renflis.no/norsk/priser/>

Vedlegg E - Kildekode

```

# -*- coding: utf-8 -*-
"""
Created on Wed Jan 24 15:36:52 2016
@author: FredrikNerol

Program for simulering av gassturbiner og PV-TS diagrammer
"""

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import pylab
import math
import pandas as pd
from scipy.integrate import simps
from pylab import rcParams

rcParams['figure.figsize'] = 12, 5

global k, Cp, R , Cpc, Cx, g, Kel

Kel = 273.15 # 0 deg C in kelvin
k = 1.40#2#1.402 #Compression factor for air
kt = 1.333 # compression factor combustion. prod
Cp = 1.005#1.004646 #kj/kg*k
Cpc = 1.1462 #constant pressure combustion products
Cx = 0.995
R = 0.287052 # kj/kg*k
g = 100*9.81

def Head(T,P1,P2,M):
    #NewCon(T)
    T2 = T_Isen(T, P1, P2,1.)
    p = DensityPT(T,P1/1000)
    p2 = DensityPT(T2,P2/1000)
    Had = Energy(range(int(T),int(T2)))*1000
    print Had
    return Had

def NsDs(rpm, D, T, P1,P2,M):
    P1 = P1*1.e3
    P2 = P2*1.e3
    rps = rpm/60.
    G = g/100.

    rho = DensityPT(T,P1/1000)
    Q = M/rho

    Had = Head(T,P1,P2,M)
    #ns = (rps*math.sqrt(Q))/((G*Had)**(3./4.))
    ns = 2*(math.pi**0.5)*rps*(Q**0.5)/((2*Had)**(3/2.))
    ds = D*((G*Had)**(1./4.))/(math.sqrt(Q))
    ds = ((math.pi**0.5)*D/2.)*(((2*Had)**1/4.)/(Q**2/4.))
    return ns, ds

def Rey(rho,v,D,mju):
    Re = (rho*v*D)/mju
    return Re

def Fd(e,D,Re):

```

```

    """Returns dimensionless friction factor"""
    if Re > 2000:
        fd = 0.25/(np.log((e/(3.7*D))+(5.74/(Re**0.9))))**2)
    if Re <= 2000:
        fd = 64/Re
    return fd

def DensityPT(tK,P_d):
    dens = P_d/(tK*R)
    return dens

def Veloc(D,spesGrav):
    V_d = (M_t/spesGrav)/(math.pi*D**2/4)
    return V_d

def DarcyPress(e,D,L,spesGrav,visc,Press):
    """Returns pressure drop in Pa"""
    v = Veloc(D,spesGrav)
    Re = Rey(spesGrav,v,D,visc)
    fd = Fd(e,D,Re)
    dp = fd*(L*spesGrav*v**2)/(D*2*9.81)
    return dp

def PressDrop(e,D,L,temp,press):
    """Calculates pressure drop from roughness(mm), diameter(mm), length(mm), temperature
    e,D,L = e/1000., D/1000., L/1000.
    visc = AbsVisc(temp)
    spesGrav = DensityPT(temp,press)
    dp = DarcyPress(e,D,L,spesGrav,visc,press)
    return dp/1000

def AbsVisc(tempK):
    """Returns absolute viscosity in (10-5 kg/m s) from kelvin"""
    x = tempK
    visc = -4.58087364E-13*x**4 + 2.37098193E-09*x**3 - 4.96519558E-06*x**2 + 7.1096
    return visc*10e-5

def kK(tempK):
    """ Set ratio of specific heats based on temp kelvin R^2 = R^2 = 0,9993"""
    global k
    x = tempK
    #k = -6*(10**-14)*(x^4) + 3*(10**-10)*(x**3) - 4*(10**-07)*(x**2) + 0.0001*x + 1
    if 2050 >= tempK >= 250:
        #k=(4e-17)*(x**5) - (3e-13)*(x**4) + (7e-10)*(x**3) - (7e-07)*(x**2) + 0.0001
        k = 4.0547928867E-17*x**5 - 2.6808572524E-13*x**4 + 6.6496566017E-10*x**3 -
        # 4E-17 x^5 - 3E-13 x^4 + 7E-10 x^3 - 7E-07 x^2 + 0,0002*x
    if tempK < 250:
        k = 1.4016
    if tempK > 2050:
        k = 1.298
    k = (k + (CpK(tempK)/CvK(tempK)))/2
    return k
def NewCon(T):
    kK(T)
    CpK(T)
    CvK(T)

```

```

def CpK(tempK, cv = True):
    """Set dynamic viscosity (10-5 kg/m s)  $\nu^2 = 0,9,9976$ """
    if cv:
        global Cp
        x = tempK
        if 1750 >= tempK >= 200:
            Cp = 1.12294654E-13*x**4 - 5.35621210E-10*x**3 + 8.27168720E-07*x**2 - 2.954
        elif tempK < 200:
            Cp = 1.00251
        elif tempK > 1750:
            Cp = 1.2299
        return Cp

def CvK(tempK):
    """Sets Cv from temp kelvin  $R^2 = 0,9967$ """
    global Cv
    x = tempK
    #Cv = (-10**10)*(x**3) + 3*(10**7)*(x**2) - (10**5)*x + 0.7022
    if 1750 >= tempK >= 200:
        #Cv = -9.6504899122E-11*x**3 + 2.5278981749E-07*x**2 - 9.7182850821E-06*x +
        Cv = 1.2656931367E-13*x**4 - 5.8680092436E-10*x**3 + 8.8827123283E-07*x**2 -
    elif tempK < 200:
        Cv = 0.7152
    elif tempK > 1750:
        Cv = 0.946
    return Cv

def T_Poly(T_0, p_1, p_2, N_is=1.):# Calc new temp isentropic
    if varcon:
        N_pol = N_Poly(T_0,p_1,p_2,N_is)
    else:
        N_pol = N_is
    return T_0*((p_2/p_1)**((k-1)/(k*N_pol)))

def N_Poly(T_1, p_1, p_2, N_is):
    T_2 = T_Isen(T_1, p_1, p_2, N_is)
    n_pol = ((k-1)/k)*math.log(p_2/p_1)/(math.log(T_2/T_1))
    return n_pol

def T_Isen(T_0, p_1, p_2, N=1.):# Calc new temp isentropic
    return T_0 + T_0*((p_2/p_1)**((k-1)/k)-1)/N#T_0*abs(((p_2/p_1)**((k-1)/k)))

def P_Isen(p_0,T_0,T_1, n ,k ):
    return p_0/((T_0/T_1)**(k/((k-1)*n)))

def V_Isen(V_0, p_1, p_2):#Calc new volume isentropic
    return V_0*((p_1/p_2)**(1/k))

def V_IsenT(V_0, t_1, t_2):#Calc new volume isentropic from temp
    return V_0*((t_1/t_2)**(1/(k-1)))

#def V_IsenTP(V_0, t_1, t_3, p_1, p_2):#Calc new volume isentropic from temp
#    V_1 = V_Isen(V_0,p_1,p_2)
#    t_2 = T_Isen2(t_1,p_1,p_2,1)
#    V_3 = V_Iso(V_1,t_2,t_3)
#    return V_3

def V_IsenTP2(V_0, t_1, t_3, p_1, p_2):#Calc new volume isentropic from temp
    V_1 = V_Isen(V_0,p_1,p_2)

```



```

t_2 = T_Isen(t_1,p_1,p_2,1)
V_3 = V_Isen(V_1,t_3,t_2)
return V_3

def V_Iso(V_0,T_1,T_2): #Calc new volum Isobaric
return V_0*(T_2/T_1)**(1/k)

def V_isoPT(v_0,t_0,t_1,p_0,p_1):
return p_0*v_0*t_1/(p_1*t_0)

def V_Iso2(V_0,T_1,T_2): #Calc new volum Isobaric
return V_0 + V_0*((T_2-T_1)/273.15-1)**(1/k)

def T_Stream(T_0,M): # Freestream Turbojet
return T_0*(1+(k-1)*(M**2)/2)

def TurbPow(T_2,T_3,T_4,M_t1,M_t2):
return T_4-M_t1*Cp*(T_3-T_2)/(M_t2*Cpc)

def ForceCalc(p_8,p_0,T_8,M_t):
choke = Is_Choked(p_8,p_0)
if choke:
A, F, V= Choked(p_0,p_8,T_8, M_t)
else:
A, F, V= UnChoked(p_0,p_8,T_8, M_t)
return A,F,V

def Is_Choked(p_8,p_0): #Calculate if nozzle is choked
p_crit = (((kt+1)/2)**(kt/(kt-1)))
if (p_8/p_0) >= p_crit:
return True
else:
return False

def Choked(p_0,p_8,T_8,M_t): #Calc speed for choked nozzle
T_8s = T_8/((kt+1)/2)
p_8s = P_Isen(p_8,T_8,T_8s,1,kt)
d_8s = p_8s/(R*T_8s)
V_8 = Speed(T_8,T_8s)
A_8 = M_t/(d_8s*V_8)
F_g = Cx*((M_t*V_8/g)+A_8*(p_8s-p_0))
return A_8,F_g,V_8

def UnChoked(p_8s, p_8,T_8,M_t): #Calc for unchoked
T_8s = T_8/(p_8/p_8s)**((kt-1)/kt)
V_8 = Speed(T_8,T_8s)
F_g = Cx*(M_t*V_8/g)
A= 1
return A , F_g, V_8

def Speed(T_8,T_8s):
return math.sqrt(2*g*Cpc*(T_8-T_8s))

def Drag(V_0,M_t):
return M_t*V_0/g

def S_Delta(S_0, T_1, T_2, p_1, p_2): #Calculate new entropy
return S_0 + Cp*np.log(T_2/T_1)-R*np.log(p_2/p_1)

```

```

def Isentr( TVSP, p_3,pts=2,N=1.): #Isentropic expansion
    T_0, v_0, s_0, p_0, = TVSP[0][-1], TVSP[1][-1], TVSP[2][-1], TVSP[3][-1]
    T = []
    P = []
    V = []
    S = []

    T.append(T_0) #Initiate first temp
    P.append(p_0) #initiate first pressure
    V.append(v_0) #initiate first volume
    S.append(s_0) #initiate first entalpy

    if pts > 2:
        Temp = T_Poly
        if varcon == False:
            N = N_Poly(T_0,p_0,p_3,N)

    else:
        Temp = T_Isen

    for p_2 in np.linspace(p_0,p_3,pts)[1:]:
        if varcon == True:#Variable Cv, Cp, k values based on temp
            NewCon(T[-1])
            T.append(Temp(T[-1],P[-1],p_2,N)) #append isentropic temp change

            P.append(p_2) #append new pressure
            V.append(V_isoPT(V[-1],T[-2],T[-1],P[-2],P[-1])) #append isentropic volun
            S.append(S_Delta( S[-1],T[-2],T[-1],P[-2],P[-1])) #append entalpy change

    return T[:], V[:], S[:], P[:] #return values

def Isobar( TVSP, T_3,pts=1,N=[1.,"%"]):
    T_0, v_0, s_0, p_0, = TVSP[0][-1], TVSP[1][-1], TVSP[2][-1], TVSP[3][-1]
    T = []
    P = []
    V = []
    S = []

    T.append(T_0) #Initiate first temp
    P.append(p_0) #initiate first pressure
    V.append(v_0) #initiate first volume
    S.append(s_0) #initiate first entalpy

    dp = 0.

    if pts == 1:
        T_0 = T_3

    if len(np.shape(N)) <= 1:
        if N[1] == "%":
            dp = p_0*(1-N[0])/(pts-1)
        elif N[1] == "kPa":
            dp = N[0]/(pts-1)

        elif N[1] == "mmW":
            dp = N[0]*0.009806648601/(pts-1)
    else:
        for loss in N:

```

```

        if loss[1] == "%":
            dp += p_0*(1-loss[0])

    for loss in N:
        if loss[1] == "kPa":
            dp += loss[0]
        elif loss[1] == "mmW":
            dp += loss[0]*0.009806648601
    dp = dp/(pts-1)
for T_2 in np.linspace(T_0,T_3,pts)[1:]: #divide temp into pts segments, but ski
    T.append(T_2) #increment temperature
    if varcon == True:#Variable Cv, Cp, k values based on temp
        NewCon(T[-1])
    P.append(P[-1]-dp) #incremente pressure with pressure loss
    V.append( V_isoPT(V[-1],T[-2],T[-1],P[-2],P[-1])) #append isobaric volume
    S.append( S_Delta(S[-1],T[-2],T[-1],P[-2],P[-1])) #append entalpy change

return T[:], V[:], S[:], P[:] #return values

def Isoterm( TVSP, p_3,pts=1,N=0):
    T_0, v_0, s_0, p_0, = TVSP[0][-1], TVSP[1][-1], TVSP[2][-1], TVSP[3][-1]
    T = []
    P = []
    V = []
    S = []

    T.append(T_0) #Initiate first temp
    P.append(p_0) #initiate first pressure
    V.append(v_0) #initiate first volume
    S.append(s_0) #initiate first entalpy

    if pts == 1:
        p_0 = p_3

    for p_2 in np.linspace(p_0,p_3,pts)[1:]: #divide temp into pts segments, but ski
        P.append(p_2) #incremente pressure with pressure loss
        T.append(T_0) #increment temperature
        if varcon == True: #Variable Cv, Cp, k values based on temp
            NewCon(T[-1])
        V.append( V_isoPT(V[-1],T[-2],T[-1],P[-2],P[-1])) #append isobaric volume
        S.append( S_Delta(S[-1],T[-2],T[-1],P[-2],P[-1])) #append entalpy change

    return T[:], V[:], S[:], P[:] #return values

def Earr(TVSP):
    dEarr = []
    for i in TVSP[1:]:
        dEarr.append([0])
        i= i[0]
        #dEarr[-1] = abs(1.006*(i[-1]-i[0]))
        for j in range(1,len(i)):
            #dEarr[-1] += abs(1.005*(i[j]-i[j-1]))
            if varcon:
                dEarr[-1] += abs(CpK(i[j])*(i[j]-i[j-1]))
            else:
                dEarr[-1] = abs(1.005*(i[-1]-i[0]))
    return dEarr

def newton(par, val, val_1, step, find,res = 1.,dire = "pos"):
    Found = False

```

```

if abs(val - find) < res:
    Found = True
    return par, step, Found
elif val > find > val_l or val < find < val_l:
    step = (step/2)

if dire == "pos":
    if val > find:
        par += step
    elif val < find:
        par -= step
if dire == "neg":
    if val > find:
        par -= step
    elif val < find:
        par += step
return par, step, Found

def Plotit2(T,V,S,P,c):
    plt.figure(1)
    #pltType = [plt.plot ,plt.scatter] #0 = reg plot, 1= scatterplot

    plt.subplot(121)

    plt.plot(V,np.array(P),color=c)
    plt.scatter(V,np.array(P),color=c)
    plt.ylabel('Trykk kPa')
    plt.xlabel('Volum')

    plt.subplot(122)
    plt.plot(S,np.array(T)-k,color=c)
    plt.scatter(S,np.array(T)-k,color=c)

    plt.ylabel('Temperatur K')
    plt.xlabel('Entropi kJ/kg*dK')

    plt.show()

def Garret(pr):
    M_t = 0.31*((-3.9952*pr**2+46.371*pr-24.695)/(60*2.2))/(0.73)
    n_c = -0.0029*pr**4+0.0296*pr**3-0.1237*pr**2+0.2531*pr+0.577
    return M_t, n_c

def makeline(x,y,zx):
    #x=x.transpose()
    #y=y.transpose()
    #zx=zx.transpose()
    xmax = []
    ymax = []
    zmax = []

    for x1,y1,z1 in zip(x,y,zx.tolist()):
        i = z1.index(max(z1))
        xmax.append(x1[i])
        ymax.append(y1[i])

```

```

        zmax.append(z1[i])
    return xmax,ymax,zmax

def plotptv(Tmp,Prs,Effs,Pvirk,Virk,Kel):
    y = Tmp-Kel
    x = Prs
    z1 = np.array(Effs)
    z1p = z1/z1.max(axis=1).max()
    z2 = np.array(Virk)
    z2p = z2/z2.max(axis=1).max()
    #z1 = z1p*z2.max(axis=1).max()
    #z3 = np.array(Pvirk)
    z4 = z1p*z2p
    z5 = z4

    x, y = np.meshgrid(x, y)

    lines = 15
    #lines = np.arange(0.08, 0.088, 0.02)
    marks = []

    plt.subplot(121)
    plt.title("Total El. Effekt ut")
    plt.grid(b=True, which='major', color='k', linestyle='-',alpha = 0.1)
    # plt.ylabel('Temperatur C')
    plt.ylabel('Virkningsgrad Varmeveksler %')
    plt.xlabel('Trykkforhold Pr')
    CS1 =plt.contour(x, y, z1,lines,colors='k')
    global v
    v = CS1.allsegs
    for vi in v:
        try:
            marks.append(vi[0][-6].tolist())
        except:
            try:
                marks.append(vi[0][-5].tolist())
            except:
                pass
    print marks

    #plt.clabel(CS1, fontsize=9,manual = marks, inline=1,fmt='%1.0f kw')
    plt.clabel(CS1, fontsize=9, inline=1,fmt='%1.0f kw')
    xmp,ymp,zmp = makeline(x,y,z1)
    #plt.plot(xmp,ymp, "k--")

    #plt.colorbar()
    #####
    plt.subplot(232)
    #for i, val in enumerate(z4):
    #    z4[i] = z4[i]/max(val)
    #CS2 = plt.contour(x, y, z4,lines,cmap="Blues")
    #plt.clabel(CS2, fontsize=9, inline=1)
    xma,yma,zma = makeline(x,y,z4)
    #plt.plot(xma,yma,c = "k")
    #plt.colorbar() #need a colorbar to show the intensity scale
    plt.subplot(122)
    plt.title("Total El. Virkningsgrad")
    plt.grid(b=True, which='major', color='k', linestyle='-',alpha = 0.1)
    #plt.ylabel('Temperatur C')
    plt.ylabel('Virkningsgrad Varmeveksler %')

```

```

plt.xlabel('Trykkforhold Pr')
CS2 = plt.contour(x, y, z2*100,lines,colors='r')

marks = []
v = CS2.allsegs
for vi in v:
    try:
        marks.append(vi[0][-6].tolist())
    except:
        try:
            marks.append(vi[0][-5].tolist())
        except:
            pass
plt.clabel(CS2, fontsize=9,manual = marks, inline=1,fmt='%1.0f' + '%')
xmv,ymv,zmv = makeline(x,y,z2)
#plt.plot(xmp,ymp,"k--",label = "Optimal effekt ut")
#plt.plot(xma,yma,c = "k")
plt.plot(xmv,ymv,"r--", label = "Optimal virkningsgrad")
# plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)

plt.subplot(121)
#plt.plot(xma,yma,c = "k")
plt.plot(xmv,ymv, "r--")

yv = x[0]
zv = z1
# plt.subplot(234)
# for zvr in zv:
#     plt.plot(yv,zvr, c= "g")
#     xm,ym,zm = makeline(x,y,z1)
#     plt.plot(xm,zm,c = "g")
#
#     zv = z5
#     plt.subplot(235)
#     for zvr in zv:
#         plt.plot(yv,zvr,c= "b")
#         xm,ym,zm = makeline(x,y,z5)
#         plt.plot(xm,zm,c = "b")
#
#     zv = z2
#     plt.subplot(236)
#     for zvr in zv:
#         plt.plot(yv,zvr,c= "r")
#         xm,ym,zm = makeline(x,y,z2)
#         plt.plot(xm,zm,c = "r")

plt.show()

def plotptv2(Tmp,Prs,Effs,Pvirk,Virk,Kel):
    y = Tmp-Kel
    x = Prs
    z1 = np.array(Effs)
    z1p = z1/z1.max(axis=1).max()
    z2 = np.array(Virk)
    z2p = z2/z2.max(axis=1).max()
    #z1 = z1p*z2.max(axis=1).max()
    #z3 = np.array(Pvirk)
    z4 = z1p*z2p
    z5 = z4

```

```

x, y = np.meshgrid(x, y)

lines = 30
lines = [2]

plt.subplot(111)
CS1 = plt.contour(x, y, z1, lines)
v = CS1.allsegs[0][0]
print v
plt.clabel(CS1, fontsize=9, inline=1)
lines = [0.088]
plt.subplot(111)
CS2 = plt.contour(x, y, z2, lines, color = "k")
v = CS2.allsegs[0][0]

print v
plt.clabel(CS2, fontsize=9, inline=1)
xmv, ymv, zmv = makeline(x, y, z2)

plt.show()

def Sintef():
    nmb = 8
    T = [0]*nmb
    P = [0]*nmb
    V = [0]*nmb
    S = [0]*nmb
    V[0] = 1.
    T[0] = T_amb
    P[0] = p_0
    S[0] = s_0

    #Compressor
    P[1] = P[0]*Pr
    T[1] = T_Isen(T[0], P[0], P[1], n_c)

    #Loss in pipes
    P[2] = P[1]*n_pr
    T[2] = T[1]

    #Heat exchanger as combustor
    T[3] = Tit
    P[3] = P[2]*n_ph

    # Turbine
    P[4] = P[0]/(n_ph2*n_pc)
    T[4] = T_Isen(T[3], P[3], P[4], n_t)

    #Combustion
    P[5] = P[4]*n_pc
    T[5] = Tit+D_t

    #Heat Exchanger
    P[6] = P[5]*n_ph2
    T[6] = T[1]+D_t

    #To environment

```

```

P[7] = P[0]
T[7] = T[0]

for i in range(len(T))[1:]:
    V[i] = V_isoPT(V[i-1],T[i-1],T[i],P[i-1],P[i])
    S[i] = S_Delta(S[i-1], T[i-1],T[i], P[i-1],P[i])
print pd.DataFrame([(np.array(T)-Kel).tolist(),P,V,S),["t","p","v","S"]).transpose()

Plotit2(T,V,S,P,"r")

def HeatExchCold(TVSP):
    TVSP.append(TVSP[-1])
    T_h = TVSP[-1][0][-1]#[[TVSP[-1][0][-2:-1]], [TVSP[-1][1][-2:-1]], [TVSP[-1][2][-2:-1]]]
    num = len(TVSP) - 1
    return TVSP,num, T_h

def HCore(TVSP,Temp,Loss):
    Tmp = Isobar(TVSP, Temp, pts,Loss) #calculate combustion values
    eng = Energy(Tmp[0])
    return Tmp, eng

def heat(Temp, eng, Dir = "up"):
    otemp = Temp
    div = 200
    dJ = eng/float(div)
    if Dir == "up":
        for i in range(0,div):
            Temp += dJ/CpK(Temp,False)
    elif Dir == "down":
        for i in range(0,div):
            Temp -= dJ/CpK(Temp,False)
    return Temp

def HeatExchWarm(TVSP, Temp, num, n_ph, n_ph2, n_pc, D_T):
    if (D_T[1] == "%"):
        Heateng = (TVSP[-1][0][-1] - Temp)
        D_t = Heateng - Heateng*D_T[0]
        T2 = TVSP[-1][0][-1] -D_t
        T1 = Temp+D_t

    elif (D_T[1] == "C"):
        D_t = D_T[0]
        T2 = TVSP[-1][0][-1] -D_t
        T1 = Temp + D_t

    TVSP[num],engHe1 = HCore(TVSP[num-1],T2,n_ph)
    TVSP[num+1]= Isobar(TVSP[num], TVSP[num+1][0][-1], pts, n_pc)
    engC = Energy(TVSP[num+1][0])
    heEng = Energy(TVSP[num][0])

    Tmp = Isobar(TVSP[-1], T1, pts, n_ph2) #calculate cooldown values
    print heEng*M_t
    print engC*M_t
    TVSP.append(Tmp)

    return TVSP,engC

```



```

def InitialiseTurb():
    TVSP = [[[T_amb], [V_0], [s_0], [p_0]]]
    return TVSP

def nP(init, lossArr, dire = "up"): #New pressure
    if len(np.shape(lossArr)) <= 1:
        lossArr = [lossArr]
        """Returns new pressure values based on losses"""

    if dire == "up":
        for i in lossArr:
            if i[1] == "kPa":
                init += i[0]
            elif [1] == "mmW":
                init += i[0]*0.009806648601

        for i in lossArr:
            if i[1] == "%":
                init = init/i[0]
    elif dire == "down":
        for i in lossArr:
            if i[1] == "kPa":
                init -= i[0]
            elif [1] == "mmW":
                init -= i[0]*0.009806648601

        for i in lossArr:
            if i[1] == "%":
                init = init*i[0]

    return init

def Compressor(TVSP,Press,Eff,App = True):
    """Adds a compressor to the cycle based on pressure ratio Press and compressor e
    Tmp = Isentr(TVSP[-1], TVSP[-1][3][-1]*Press,pts,Eff)
    eng = Energy(Tmp[0])
    if App:
        TVSP.append(Tmp)
        return TVSP,eng
    else:
        return Tmp,eng

def PipeLoss(TVSP,Loss,typ = "abs", e = 0.01,D = 50.,L = 150.):
    """Adds a pipeloss to the cycle \n
    if typ = "abs"\n
    static pressure drop based on pressure value or percentage\n\n
    if typ = "calc"\n
    Pressure will be based on pipe roughness im mm = e\n
    pipe diameter in mm = D\n
    pipe length in mm = L\n"""

    if len(np.shape(Loss)) <= 1:
        Loss = [Loss]
    if typ == "abs":
        Tmp = Isoterm(TVSP[-1], nP(TVSP[-1][3][-1],Loss,"down"), pts,) #calc values
    elif typ == "calc":
        Loss = [PressDrop(e,D,L,TVSP[-1][0][-1],TVSP[-1][3][-1]),"kPa"]
        Tmp = Isoterm(TVSP[-1], nP(TVSP[-1][3][-1],Loss,"down"), pts,) #calc values

```

```

eng = Energy(Tmp[0])
TVSP.append(Tmp)
return TVSP,eng

def FauxHE(D_t,Tit,t_comp):
    DT = Tit - t_comp
    if D_t[1] == "%":
        D_T = DT - DT*D_t[0]

    elif D_t[1] == "C":
        D_T = D_t[0]

    return D_T

def Intercooler(TVSP,Temp,Loss, D_T = [1.,"%"]):

    if (D_T[1] == "%"):
        Heateng = (TVSP[-1][0][-1] - Temp)
        D_t = Heateng - Heateng*D_T[0]
    elif (D_T[1] == "C"):
        D_t = temp + D_T[0]

    """Adds intercooler, lowers gas to specified temp, returns energy absorbed from
    Tmp = Isobar(TVSP[-1], Temp+D_t, pts, Loss) #calculate cooldown values
    eng = Energy(Tmp[0])
    TVSP.append(Tmp)
    return TVSP,eng

def CombChamb(TVSP,Temp,Loss):
    """Heats up gas to specified level"""
    Tmp = Isobar(TVSP[-1], Temp, pts, Loss) #calculate combustion val
    eng = Energy(Tmp[0])
    TVSP.append(Tmp)
    return TVSP,eng

def Turbine(TVSP,Press,Loss=[0,"kpa"],Eff=1.,App = True):
    Tmp = Isentr(TVSP[-1], nP(Press,Loss), pts, Eff) #calc values for turbine
    eng = Energy(Tmp[0])
    if App:
        TVSP.append(Tmp)
        return TVSP,eng
    else:
        return Tmp,eng

def Exhaust(TVSP,Temp,Loss):
    Tmp = Isobar(TVSP[-1], Temp, pts, Loss) #values for heat out
    eng = Energy(Tmp[0])
    TVSP.append(Tmp)
    return TVSP,eng

def Energy(tmp):
    dEarr = 0

    for j in range(1,len(tmp)):

        #dEarr[-1] += abs(1.005*(i[j]-i[j-1]))

```

```

        if varcon:
            if varEng:
                dEarr += abs(((CpK(tmp[j])+CpK(tmp[j-1])))/2)*(tmp[j]-tmp[j-1]))
            else:
                dEarr += abs(CpK(tmp[0])*(tmp[j]-tmp[j-1]))
        else:
            dEarr = abs(1.005*(tmp[-1]-tmp[0]))
    return dEarr

def Turbojet():
    #Freestream
    T = [0]*9
    P = [0]*9
    v = [0]*9
    T[0] = T_amb
    P[0] = p_0
    v[0] = v_0
    #Freestream
    T[1] = T_Stream(T_amb,M)
    P[1] = P_Isen(P[0],T[0],T[1],1,k)
    #Intake
    T[2] = T[1]
    P[2] = P[1]*prf
    #Compressor
    P[3] = P[2]*Pr
    T[3] = T_Isen(T[2],P[2],P[3],n_c)
    #Combustor
    T[4] = Tit
    P[4] = P[3]*(1-n_pc)
    # Turbine
    T[5] = TurbPow(T[2],T[3],T[4],M_t,M_t)
    P[5] = P_Isen(P[4],T[4],T[5],n_t,kt)
    #Jetpipe
    T[8] = T[5]
    P[8] = P[5]*(1-n_ext)

    #choked?
    A,F_g,v[8] = ForceCalc(P[8],P[0],T[8],M_t)
    F_r = Drag(v[0],M_t)
    F_n = F_g-F_r
    i = 0
    #for t,p,v in zip(T,P,V):
    #    print "T: {0:.2f} P:{1:.2f} V:{2:.2f}".format(t,p,v)

    n_tot,Pe = NormalEff(T[0],T[3],T[4],T[8],M_t,n_co)
    n_tot = ((T[4]-T[5])-(T[4]-T[8]))/(T[4]-T[3])
    print pd.DataFrame([T,P,v],[ "t", "p", "v"]).transpose()
    print "\n", "Thrust: %5.1f kN " %F_g
    print "Effekt: %5.0f kW" %Pe
    Cfuel = 34.7*10**6 #mj / L
    fc = Pe*1000/Cfuel
    print "Forbruk: %5.2f l/sek" %fc

def VirkChart3d(gasturb,varturb,pMin=1.5,pMax=6., tMin=600,tMax=1100):
    Res = 10
    Tmp = np.linspace(tMin+Kel,tMax+Kel,10)
    Prs = np.linspace(pMin,pMax,10)

    #    Tmp = np.linspace(0.7,1.,10)
    #    Prs = np.linspace(1.5,6,100)

```

```

Virk = []
Pvirk = []
Effs = []
plot = []
global M_t,n_c,n_t,Pr, D_t, Tit
for Tit in Tmp:
    #D_t = [Tit,"%"]
    Effs.append([])
    Virk.append([])
    Pvirk.append([])

    for Pr in Prs:
        # if varturb:
        #m_t1,n_c = Garret(Pr)
        #n_t = 1/(n_c+0.05)

        TVSP, P_out, n_tot, P_virk, places = gasturb()
        n_tot = n_tot*n_el*n_co
        P_out = P_out*n_el
        Effs[-1].append(P_out)
        Pvirk[-1].append(P_virk)
        Virk[-1].append(n_tot)

plotptv(Tmp,Prs,Effs,Pvirk,Virk,Kel)

def VirkChart2d(gasturb,varturb,pMin=1.5,pMax=10, Vari = "Pr"):
    Vrs = np.linspace(pMin,pMax,30)
    Virk = []
    Pvirk = []
    Effs = []
    plot = []
    tmpvar = globals()[Vari]
    global m_t, n_c, n_t
    for Vr in Vrs:
        globals()[Vari] = Vr
        #if varturb:
        #m_t,n_c = Garret(Pr)
        #n_t = 1/(n_c+0.05)
        TVSP, P_out, n_tot, P_virk, places = gasturb()
        n_tot = n_tot*n_el*n_co
        P_out = P_out*n_el
        Effs.append(P_out)
        Pvirk.append(P_virk)
        Virk.append(n_tot)

    globals()[Vari] = tmpvar
    plt.subplot(121)
    plt.grid(b=True, which='major', color='k', linestyle='-',alpha = 0.1)
    plt.title("Total El. Effekt ut")
    plt.ylabel('Effekt ut Kw')
    plt.xlabel('Trykkforhold Pr')
    plt.plot(Vrs,Effs,"k")
    plt.subplot(122)
    plt.grid(b=True, which='major', color='k', linestyle='-',alpha = 0.1)
    plt.title("Total El. Virkningsgrad")
    plt.ylabel('Virkningsgrad %')
    plt.xlabel('Trykkforhold Pr')
    plt.plot(Vrs,np.array(Virk)*100,"k")

```

```

def PVplot(gassturb,c,typ,h,ann == "",Tempr = "C",prin = "List"):

    TVSP, P_out,n_tot,p_n,places = gassturb()

    TVSPlist = []
    integrals = []
    for i in TVSP:
        TVSPlist.append([i[0][-1],i[1][-1],i[2][-1],i[3][-1]])
    if prin == "List":
        Printit(TVSP,Tempr,places)
    Plotit(TVSP,c,typ,h,ann,places,Tempr)
    print "Kw\tKw_el\ttn_shaft\ttn_el"
    print "%5.2f" %P_out + "\t%5.2f" %(P_out*n_el) + "\t%5.3f" %(n_tot*n_co) + "\t%5.2f" %n_co
    print "\n"

def Plotit(TVSP, c = "k", typ = 0,h = [[0,0],[0,0]],ann = "",places =[0],Tempr = "K")
    plt.figure(1)
    pltType = [plt.plot ,plt.scatter] #0 = reg plot, 1= scatterplot
    k = 0
    plt.subplot(121)#.set_xticklabels([])
    #plt.subplot(121).set_yticklabels([])
    plt.title('PV-Diagram')
    plt.grid(b=True, which='major', color='k', linestyle='-',alpha = 0.1)
    V,P = 0,0
    for i in TVSP[:]:
        plt.plot(i[1],np.array(i[3]),c)

        plt.scatter(i[1][-1],np.array(i[3][-1]),color = "k")

    if ann == "a":
        for indx, pl in enumerate(places):
            indx+=1
            V,P = TVSP[pl][1][0],TVSP[pl][3][0]
            plt.annotate(str(indx),(V,P-k+5))

    plt.ylabel('Trykk kPa')
    plt.xlabel('Volum')

    plt.subplot(122)#.set_xticklabels([]) #PLOT Ts diagram
    #plt.subplot(122).set_yticklabels([])
    plt.title('TS-Diagram')
    plt.grid(b=True, which='major', color='k', linestyle='-',alpha = 0.1)

    indx =0
    T,S = 0,0
    if Tempr == "C":
        k = Kel
        plt.ylabel('Temperatur C')
    else:
        k = 0
        plt.ylabel('Temperatur K')
    for i in TVSP[:]:
        plt.plot(i[2],np.array(i[0])-k,c)
        plt.scatter(i[2][-1],np.array(i[0][-1])-k,color="k")

    if ann == "a":
        for indx, pl in enumerate(places):
            indx+=1

```

```

S,T = TVSP[p1][2][0],TVSP[p1][0][0]
plt.annotate(str(indx),(S-0.05,T-k+5))

if typ == 1:
    li = []
    for i in TVSP:
        li.append([i[0][-1]-k,i[1][-1],i[2][-1],i[3][-1]])
    for i in h:
        plt.plot([li[i[0]][2],li[i[1]][2]],[li[i[0]][0],li[i[1]][0]], 'k--')

plt.xlabel('Entropi kJ/kg*dK')

#plt.subplot(133)
#for i in TVSP[:]:
#    pltType[typ](i[3],np.array(i[0]),color=c)
plt.show()

def Printit(TVSP,Tempr,places = [0]):
    if Tempr == "K":
        Comp = 0
    elif Tempr == "C":
        Comp = Kel
    print "komp\tPress1(kPa)\tPress2(kPa)\tTemp1(°+Tempr+)\tTemp2(°+Tempr+)\tVol"
    jl = "1"
    Shit = 0
    for p,i in enumerate(TVSP):
        j = p+1
        if j in places:
            pos = str(j+1-Shit)
            ind = jl+"-"+pos
            jl = pos
        elif p == len(TVSP)-1:
            ind = pos+"-1"
        else:
            ind = jl+"-"+jl+""
            jl = jl+""
            Shit += 1

        print( ind+"\t"+"%5.2f" % i[3][0]+"\t"+"%5.2f" % i[3][-1]
            +" \t%5.2f" %((i[0][0])-Comp)+ " \t%5.2f" % ((i[0][-1])-Comp)
            +" \t%5.2f" % ( i[1][0])+ " \t%5.2f" % (i[1][-1]))
    print "\n"

def Eff2(WorkIn,WorkOut,HeatIn,loss = 0):
    """n_tot = ((T_3-T_4)-(T_2-T_1))/(T_5-T_6)\n
    Pe = Cp*((T_3-T_4)-(T_2-T_1))*M_t/n_co\n
    Pe2 = ((T_3-T_4)-(T_2-T_1))/(T_3-T_4)\n
    T_3-T_6 = Energy in turbine\n
    T_2-T_1 = Energy in compressor\n
    T_5-T_4 = Energy in combustion\n
    """
    if r_air[1] == "%":
        Pe = (WorkOut*(M_t+M_t*r_air[0])-WorkIn*M_t-loss)

```

```

n_tot = (WorkOut*(M_t+M_t*r_air[0])-WorkIn*M_t-loss)/(HeatIn*(M_t+M_t*r_air[
elif r_air[1] == "kgs":
    Pe = (WorkOut*(M_t+r_air[0])-WorkIn*M_t-loss)
    n_tot = (WorkOut*(M_t+r_air[0])-WorkIn*M_t-loss)/(HeatIn*(M_t+r_air[0]))

#n_tot = (WorkOut-WorkIn)/HeatIn
Pe2 = (WorkOut-WorkIn)/WorkOut
return n_tot, Pe ,Pe2

def MultiCycleHE(pts = 100):
    places = [0,1,2,3,4,5,6,7,8,9]

    TVSP = InitialiseTurb()

    TVSP,workIn = Compressor(TVSP,Pr,n_c)
    Intercooler(TVSP,T_amb+10,n_ph)

    TVSP,workIn2 = Compressor(TVSP,Pr2,n_c)
    workIn += workIn2

    TVSP, num,T_h = HeatExchCold(TVSP)
    TVSP, _ = CombChamb(TVSP,Tit, [n_pc,n_ph])

    TVSP,workOut = Turbine(TVSP,p_0*Pr,[n_ext],n_t)
    TVSP, heatIn = CombChamb(TVSP,Tit, n_pc)

    TVSP,workOut2 = Turbine(TVSP,p_0,[n_ext,n_ph2],n_t)
    workOut += workOut2

    TVSP, heatIn2 = HeatExchWarm(TVSP, T_h, num, n_ph, n_ph2, n_pc, D_t)
    heatIn += heatIn2
    TVSP,heatOut = Exhaust(TVSP,T_amb,n_ext)

    n_tot, P_out, P_eff = Eff2(workIn,workOut,heatIn)
    return TVSP[1:], P_out, n_tot, P_eff, places

def MultiCycle():
    places = [0,1,2,3]

    TVSP = InitialiseTurb()

    Pr = 3.
    TVSP,workIn = Compressor(TVSP,Pr,n_c)
    TVSP,s0 = Intercooler(TVSP,T_amb+10,n_ph)

    Pr2 = 8.
    TVSP,workIn2 = Compressor(TVSP,Pr2,n_c)
    workIn += workIn2

    TVSP, heatIn = CombChamb(TVSP,Tit, n_pc)
    TVSP,workOut1 = Turbine(TVSP,p_0*Pr,[n_ext],n_t)
    TVSP, num,T_h = HeatExchCold(TVSP)
    TVSP, h2 = CombChamb(TVSP,Tit, [n_pc,n_ph])

```

```

heatIn += h2
TVSP,workOut2 = Turbine(TVSP,p_0,[n_ext],n_t)
TVSP,s2 = Intercooler(TVSP,T_amb,n_ext)

workOut = workOut2 + workOut1
n_tot, P_out, P_eff = Eff2(workIn,workOut,heatIn)
return TVSP[1:], P_out, n_tot, P_eff, places

def RealRegen():
    places = [0,1,2,3,4,5]

    TVSP = InitialiseTurb()

    TVSP,workIn = Compressor(TVSP,Pr,n_c)

    TVSP, num,T_h = HeatExchCold(TVSP)

    TVSP, _ = CombChamb(TVSP,Tit, [n_pc,n_ph])

    TVSP,workOut = Turbine(TVSP,p_0,[n_ext,n_ph2],n_t)

    TVSP, heatIn = HeatExchWarm(TVSP, T_h, num, n_ph, n_ph2, n_pc, D_t)

    TVSP,heatOut = Exhaust(TVSP,T_amb,n_ext)
    print heatOut*M_t

    n_tot, P_out, P_eff = Eff2(workIn,workOut,heatIn)
    return TVSP[1:], P_out, n_tot, P_eff, places

def NormalCycle():
    places = [0,1,2,3]

    TVSP = [[[T_amb], [V_0], [s_0], [p_0]]]

    TVSP = InitialiseTurb()

    TVSP,workIn = Compressor(TVSP,Pr,n_c)

    PipeLoss(TVSP, n_pr)

    TVSP,heatIn = CombChamb(TVSP,Tit, n_pc)

    TVSP,workOut = Turbine(TVSP,p_0,n_ext,n_t)

    TVSP,heatOut = Exhaust(TVSP,T_amb,n_ext)

    n_tot, P_out, P_eff = Eff2(workIn, workOut, heatIn)

    return TVSP[1:], P_out, n_tot, P_eff,places

def NormalCyclePower(): #Case Study 1
    global n_c, n_t, n_ph,n_pc, n_pr ,n_ext, M, prf, v_0, D_t, n_co, n_h, n_ph2, n_r

    n_t = 1/0.80 #efficiency turbin 1

    n_co = 1. #Efficiency combustion chamber
    n_h = 1. #efficiency heat exchanger

    n_pr = [0., "kPa"] #Pressure loss pipes
    n_pc = [0.97, "%"]#pressure loss combustor

```



```

n_ext = [1., "kPa"] #exhaust pressure Loss
n_in = [0., "kPa"]

global varcon, Pr, Tit, T_amb, M_t, p_0, s_0, r_air, n_el

Pr = 13. # Pressure Ratio
Tit = 1450. #Turbine inlet temperature in K
M_t = 3.5 #Mass Transfer kg/s
r_air = [0.08, "kgs"] #Added mass from fuel input "kgs" or "%"
n_el = 0.99

T_amb = 288.15 #ambient temp C
p_0 = 101.325 #atm pressure kPa

Tit = Tit
T_amb = T_amb
s_0 = 0 #S_Delta(0, Kel, T_amb, 1, 1) #start entropy
V_0 = 1.
places = [0, 1, 3, 4]
#global varEng
varcon = True
#varEng = False

TVSP = [[[T_amb], [V_0], [s_0], [p_0]]]
TVSP = InitialiseTurb()

T_x = 0.
Nc2 = 1.
stp = 0.1
Found = False
while Found == False:
    TVSPval = TVSP
    T_x1 = T_x
    tmp, TurbEnn = Compressor(TVSP, Pr, Nc2, False) #Turbine(TVSPval, p_0*Pt, n_ext, n_
    T_x = tmp[0][-1]
    Nc2, stp, Found = newton(Nc2, T_x, T_x1, stp, 658., 0.01)
print Nc2, tmp[3][-1]
TVSP, workIn = Compressor(TVSP, Pr, Nc2)

PipeLoss(TVSP, n_pr)

TVSP, heatIn = CombChamb(TVSP, Tit, n_pc)

Pt = 1.1
stp = 1.
Found = False
TurbEnn = 0.
workIn2 = workIn * M_t

while Found == False:
    TVSPval = TVSP
    TurbEnn_1 = TurbEnn
    tmp, TurbEnn = Turbine(TVSPval, p_0 * Pt, [0, "kpa"], n_t, False)
    TurbEnn = TurbEnn * (M_t + r_air[0])
    Pt, stp, Found = newton(Pt, TurbEnn, TurbEnn_1, stp, workIn2, 0.01)
    #print Pt, stp, Found, workIn, TurbEnn

print tmp[3][0] / tmp[3][-1]
TVSP, _ = Turbine(TVSP, p_0 * Pt, [0, "kpa"], n_t)

```

```

T_x = 0.
Nc2= 1/0.9
stp = 0.1
Found = False
while Found == False:
    tmp, TurbEnn = Turbine(TVSP,p_0 ,n_ext, Nc2,False)#Turbine(TVSPval,p_0*Pt,n_
    T_x = tmp[0][-1]
    Nc2, stp, Found = newton(Nc2, T_x, T_x1, stp, 868., 0.1, "neg")
print 1/Nc2
print tmp[3][0]/tmp[3][-1]

TVSP,workOut2 = Turbine(TVSP,p_0,n_ext,Nc2)

TVSP,heatOut = Exhaust(TVSP,T_amb,n_ext)

n_tot, P_out, P_eff = Eff2(0, workOut2, heatIn)
#n_tot, P_out, P_eff = Eff(T_1,T_2,T_4,T_5,T_3,T_4,M_t,n_co)
return TVSP[1:], P_out, n_tot, P_eff,places

def case2Set1():
    global n_c, n_t, n_ph,n_pc, n_pr ,n_ext, M, prf, v_0, D_t, n_co, n_h, n_ph2, n_r

    n_c = 0.85#Efficiency compressor
    n_t = 1/0.9 #efficiency turbin
    n_co = 1. #Efficiency combustion chamber

    n_pr = [0., "kPa"] #Pressure loss pipes
    n_ph = [0., "kPa"]#[50., "kPa"]#pressure loss heat exchanger side1
    n_ph2 = [0., "kPa"]##pressure loss heat exchanger side2
    n_pc = [1.-0.0, "%"]#pressure loss combustor
    n_ext = [0., "kPa"]#exhaust pressure loss
    n_in = [0., "kPa"]
    prf = 1. #Inlet pressure loss
    n_n = 1. #nozzle pressure loss

    global varcon,varEng,pts,Pr,Tit,T_amb, M_t, p_0, T_amb, s_0, r_air,n_el

    Pr = 5.6 # Pressure Ratio
    Tit = 1563.-Kel #Turbine inlet temperature in C
    M_t = 1.#Mass Transfer kg/s
    r_air = [0.0, "%"] #Added mass from fuel input "kgs" or "%"
    n_el = 1.

    T_amb = 300.-Kel #ambient temp C
    p_0 = 101.325 #atm pressure kPa
    V_0 = 1.
    Tit = Tit+Kel
    T_amb = T_amb+Kel
    s_0 = 0#S_Delta(0,Kel,T_amb,1,1)#start entropy
    pts = 2

    #varcon = False
    varcon = True
    #varEng = True

def CaseStudy2(): #Case Study 2
# case2Set1()
# Pr = 4.7
# Tit = 1389.

```

```

# print Pr, Tit
# PVplot(CaseStudy2, "k", 0, [[1,5],[2,4]], "a", "K", "no")
#
# Pr = 5.6
# Tit = 1563.
# print Pr, Tit
# PVplot(CaseStudy2, "k", 0, [[1,5],[2,4]], "a", "K", "no")
#
# Pr = 8.
# Tit = 1480.
# print Pr, Tit
# PVplot(CaseStudy2, "k", 0, [[1,5],[2,4]], "a", "K", "no")
#
# Pr = 6
# Tit = 1751.
# print Pr, Tit
# PVplot(CaseStudy2, "k", 0, [[1,5],[2,4]], "a", "K", "no")
#
# Pr = 3.8
# Tit = 1666.
# print Pr, Tit
# PVplot(CaseStudy2, "k", 0, [[1,5],[2,4]], "a", "K", "no")

```

```

places = [0,1,3,4]
TVSP = [[[T_amb], [V_0], [s_0], [p_0]]]

TVSP = InitialiseTurb()

TVSP,workIn = Compressor(TVSP,Pr,n_c)

PipeLoss(TVSP, n_pr)

TVSP,heatIn = CombChamb(TVSP,Tit, n_pc)

TVSP,workOut = Turbine(TVSP,p_0,n_ext,n_t)

TVSP,heatOut = Exhaust(TVSP,T_amb,n_ext)
n_tot, P_out, P_eff = Eff2(workIn, workOut, heatIn)
#n_tot, P_out, P_eff = Eff(T_1,T_2,T_4,T_5,T_3,T_4,M_t,n_co)
return TVSP[1:], P_out, n_tot, P_eff,places

```

```

def NormalCyclePower2(pts = 10):
    places = [0,1,3,4]
    TVSP = [[[T_amb], [V_0], [s_0], [p_0]]]

    TVSP = InitialiseTurb()

    TVSP,workIn = Compressor(TVSP,Pr,n_c)

    PipeLoss(TVSP, n_pr)

    TVSP,heatIn = CombChamb(TVSP,Tit, n_pc)

    TVSP, _ = Turbine(TVSP,p_0*2.8,[0,"kpa"],n_t)

    TVSP,workOut2 = Turbine(TVSP,p_0,n_ext,n_t2)

    TVSP,heatOut = Exhaust(TVSP,T_amb,n_ext)

```

```

n_tot, P_out, P_eff = Eff2(0, workOut2, heatIn)
#n_tot, P_out, P_eff = Eff(T_1,T_2,T_4,T_5,T_3,T_4,M_t,n_co)
return TVSP[1:], P_out, n_tot, P_eff,places

def RealAres(pts = 10):
    places = [0,1,2,3,4,5]

    TVSP = InitialiseTurb()

    TVSP,workIn = Compressor(TVSP,Pr,n_c)
    t_comp = TVSP[-1][0][-1]

    TVSP, hin = CombChamb(TVSP,Tit,n_ph)
    TVSP,workOut = Turbine(TVSP,p_0,[n_ext,n_pc,n_ph2],n_t)

    D_T = FauxHE(D_t,Tit,t_comp)

    TVSP,heatIn = CombChamb(TVSP,Tit+D_T,n_pc)

    TVSP, heeng = Exhaust(TVSP,t_comp + D_T,n_ph2)
    TVSP, hout = Exhaust(TVSP,T_amb,n_ext)

    print "input Heat ",heatIn*M_t
    print "Exchanged Heat ",heeng*M_t
    print "output heat ",hout*M_t

    n_tot, P_out, P_eff = Eff2(workIn,workOut,heatIn)
    return TVSP[1:], P_out, n_tot, P_eff, places

def Case3():
    "SSS-12"
    global n_c, n_t, n_ph,n_pc, n_pr ,n_ext, M, prf, v_0, D_t, n_co, n_h, n_ph2, n_r

    n_c = 0.785#Efficiency compressor
    n_t = 1/0.883 #efficiency turbine
    n_co = 0.98 #Efficiency combustion chamber

    n_pc = [1.-0.02,"%"]#press loss combustor
    n_ext = [1-0.063,"%"]#exhaust pressure loss
    n_in = [1.-0.008,"%"]
    n_g = 3.500 #w
    n_par = 2.833 #w

    global varcon, varEng,pts,Pr,Tit,T_amb, M_t, p_0, T_amb, s_0, r_air, n_el

    Pr = 11.1 # Pressure Ratio
    Pt = 10.1 # Press ratio Turbine
    Tit = 1344-Kel #Turbine inlet temperature in C
    r_air = [0.01,"kgs"] #Added mass from fuel input"kgs" or "%"
    M_t =0.481#Mass Transfer kg/s
    n_el = 1.
    T_amb = 15. #ambient temp C
    p_0 = 101.325 #atm pressure

    #varcon = True
    Tit = Tit+Kel

```

```

T_amb = T_amb+Kel
s_0 = 0#S_Delta(0,Kel,T_amb,1,1)#start entropy
V_0 = 1.

places = [0]
TVSP = [[[T_amb], [V_0], [s_0], [p_0]]]

TVSP = InitialiseTurb()
PipeLoss(TVSP, n_in)

TVSP,workIn = Compressor(TVSP,Pr,n_c)

TVSP,heatIn = CombChamb(TVSP,Tit, n_pc)

TVSP,workOut = Turbine(TVSP,TVSP[-1][3][-1]/Pt,n_ext,n_t)

Exhaust(TVSP,TVSP[-1][0][-1],n_ext)

rest = [TVSP[-1][3][-1]-p_0,"kPa"]
Exhaust(TVSP,T_amb,rest)

n_tot, P_out, P_eff = Eff2(workIn, workOut, heatIn,(n_g+n_par))

#n_tot, P_out, P_eff = Eff(T_1,T_2,T_4,T_5,T_3,T_4,M_t,n_co)
return TVSP[1:], P_out, n_tot, P_eff,places

#Variables
def sSet():
    global n_c, n_t, n_ph,n_pc, n_pr ,n_ext, M, prf, v_0, D_t, n_co, n_h, n_ph2, n_r

    D_t = 70.0#delta t heat exchanger
    n_c = 0.80#Efficiency compressor
    n_t = 1/0.85 #efficiency turbine
    n_co = 0.98 #Efficiency combustion chamber
    n_h = 0.95 #efficiency heat exchanger
    n_pr = 0.995 #Pressure loss pipes
    n_ph = 1-0.02#pressure loss heat exchanger side1
    n_ph2 = 1-0.10##pressure loss heat exchanger side2
    n_pc = 1-0.1#pressure loss combustor
    n_ext = 1-0.01#exhaust pressure loss
    prf = 1. #Inlet pressure loss
    n_n = 1. #nozzle pressure loss

    v_0 = 0. #Start Speed
    V_0 = 1. #Start Volume
    M = 0. # Mach Number

def iSet():
    global n_c, n_t, n_ph,n_pc, n_pr ,n_ext, M, prf, v_0, D_t, n_co, n_h, n_ph2, n_r

    D_t = 0.#delta t heat exchanger
    n_c = 1.#Efficiency compressor
    n_t = 1/1. #efficiency turbine
    n_co = 1. #Efficiency combustion chamber
    n_h = 1. #efficiency heat exchanger

    n_pr = [0., "kPa"] #Pressure loss pipes
    n_ph = [0., "kPa"]#[50., "kPa"]#pressure loss heat exchanger side1

```

```

n_ph2 = [0., "kPa"] ##pressure loss heat exchanger side2
n_pc = [0., "kPa"] #pressure loss combustor
n_ext = [0., "kPa"] #exhaust pressure Loss
n_in = [0., "kPa"]

prf = 1. #Inlet pressure Loss
n_n = 1. #nozzle pressure Loss

global varcon,pts,Pr,Tit,T_amb, M_t, p_0, T_amb, s_0

Pr = 4.2 # Pressure Ratio
Tit = 901. #Turbine inlet temperature in C
T_amb = 20. #ambient temp C
M_t = 0.3 #Mass Transfer kg/s
p_0 = 101.325 #atm pressure
Tit = Tit+Kel
T_amb = T_amb+Kel
s_0 = 0 #S_Delta(0,Kel,T_amb,1,1)#start entropy

def iSet2():
    global n_c, n_t, n_ph,n_pc, n_pr ,n_ext, M, prf, v_0, D_t, n_co, n_h, n_ph2, n_r
    D_t = [0.75, "%"] #delta t heat exchanger
    #D_t = [102, "C"] #delta t heat exchanger
    n_c = 0.85 #Efficiency compressor
    n_t = 1/0.87 #efficiency turbin
    n_co = 1. #Efficiency combustion chamber

    n_pr = [1., "kPa"] #Pressure Loss pipes
    n_ph = [0.97, "%"] #[50., "kPa"] #pressure loss heat exchanger side1
    n_ph2 = [0.94, "%"] ##pressure loss heat exchanger side2
    n_pc = [0.95, "%"] #pressure loss combustor
    n_ext = [3., "kPa"] #exhaust pressure Loss
    n_in = [0., "kPa"]
    prf = 1. #Inlet pressure Loss
    n_n = 1. #nozzle pressure Loss

    global varcon,pts,Pr,Pr2,Tit,T_amb, M_t, p_0, T_amb, s_0, r_air, n_el

    Pr = 4. # Pressure Ratio
    Pr2 = 4.
    Tit = 1000. #Turbine inlet temperature in C
    r_air = [0.01, "%"] #Added mass from fuel input "kgs" or "%"
    M_t = 2.1 #Mass Transfer kg/s

    n_el = 0.95
    T_amb = 15. #ambient temp C
    p_0 = 101.325 #atm pressure kPa
    fuel = 0
    Tit = Tit+Kel
    T_amb = T_amb+Kel
    s_0 = 0 #S_Delta(0,Kel,T_amb,1,1)#start entropy
    V_0 = 1.

def baseSet3():
    global n_c, n_t, n_ph,n_pc, n_pr ,n_ext, M, prf, v_0, D_t, n_co, n_h, n_ph2, n_r

    D_t = [1., "%"] #delta t heat exchanger
    n_c = 1. #Efficiency compressor

```

```

n_t = 1/1. #efficiency turbin
n_co = 1. #Efficiency combustion chamber

n_pr = [0., "kPa"] #Pressure Loss pipes
n_ph = [0., "kPa"]#[50., "kPa"]#pressure Loss heat exchanger side1
n_ph2 = [0., "kPa"]##pressure Loss heat exchanger side2
n_pc = [1., "%"]#pressure Loss combustor
n_ext = [0., "kPa"]#exhaust pressure Loss
n_in = [0., "kPa"]
prf = 1. #Inlet pressure Loss
n_n = 1. #nozzle pressure Loss

global varcon,pts,Pr,Tit,T_amb, M_t, p_0, T_amb, s_0, r_air, n_el

Pr = 13. # Pressure Ratio
Tit = 1450.-Kel #Turbine inlet temperature in C
M_t = 3.5#Mass Transfer kg/s
r_air = [0.0, "kgs"] #Added mass from fuel input "kgs" or "%"
n_el = 1.
T_amb = 288.15-Kel #ambient temp C
p_0 = 101.325 #atm pressure kPa

Tit = Tit+Kel
T_amb = T_amb+Kel
s_0 = 0#S_Delta(0,Kel,T_amb,1,1)#start entropy
V_0 = 1.

def CapstoneBcp():
global n_c, n_t, n_ph,n_pc, n_pr ,n_ext, M, prf, v_0, D_t, n_co, n_h, n_ph2, n_r

D_t = [0.84, "%"]#delta t heat exchanger
#D_t = [102, "C"]#delta t heat exchanger
n_c = 0.751#Efficiency compressor
n_t = 1/0.84 #efficiency turbin
n_co = 0.99 #Efficiency combustion chamber

n_pr = [1., "%"] #Pressure Loss pipes
n_ph = [0.945, "%"]#[50., "kPa"]#pressure Loss heat exchanger side1
n_ph2 = [0.945, "%"]##pressure Loss heat exchanger side2
n_pc = [0.975, "%"]#pressure Loss combustor
n_ext = [0., "kPa"]#exhaust pressure Loss
n_in = [0., "kPa"]
prf = 1. #Inlet pressure Loss
n_n = 1. #nozzle pressure Loss

global varcon,pts,Pr,Tit,T_amb, M_t, p_0, T_amb, s_0, r_air, n_el

Pr = 3.9 # Pressure Ratio
Tit = 871. #Turbine inlet temperature in C
r_air = [0.0, "kgs"] #Added mass from fuel input "kgs" or %"
M_t = 0.31#Mass Transfer kg/s

n_el = 0.96
T_amb = 15. #ambient temp C
p_0 = 101.325 #atm pressure kPa
Tit = Tit+Kel
T_amb = T_amb+Kel
s_0 = 0#S_Delta(0,Kel,T_amb,1,1)#start entropy
V_0 = 1.

```

```

global varcon,pts,Pr,Tit,T_amb, M_t, p_0, T_amb, s_0, n_el, r_air, v_0, V_0, M_0
global n_c, n_t, n_ph,n_pc, n_pr ,n_ext, M, prf, v_0, D_t, n_co, n_h, n_ph2, n_n, v_
#varEng = False
varEng = True
varcon = True
#varcon = False
pts = 500

#sSet()

#plt.clf()

#sSet()
#PVplot(Case3, "k--",0,[[1,5],[2,4]],)

def Capstone():
    global n_c, n_t, n_ph,n_pc, n_pr ,n_ext, M, prf, v_0, D_t, n_co, n_h, n_ph2, n_r

    D_t = [0.84,"%"]#delta t heat exchanger
    #D_t = [102,"C"]#delta t heat exchanger
    n_c = 0.751#Efficiency compressor
    n_t = 1/0.84 #efficiency turbin
    n_co = 0.99 #Efficiency combustion chamber

    n_pr = [1.,"%"] #Pressure loss pipes
    n_ph = [0.945,"%"]#[50.,"kPa"]#pressure loss heat exchanger side1
    n_ph2 = [0.945,"%"]##pressure loss heat exchanger side2
    n_pc = [0.975,"%"]#pressure loss combustor
    n_ext = [0.,"kPa"]#exhaust pressure loss
    n_in = [0.,"kPa"]
    prf = 1. #Inlet pressure loss
    n_n = 1. #nozzle pressure loss

    global varcon,pts,Pr,Tit,T_amb, M_t, p_0, T_amb, s_0, r_air, n_el

    Pr = 3.9 # Pressure Ratio
    Tit = 871. #Turbine inlet temperature in C
    r_air = [0.0,"kgs"] #Added mass from fuel input "kgs" or "%"
    M_t = 0.31#Mass Transfer kg/s

    n_el = 0.96
    T_amb = 15. #ambient temp C
    p_0 = 101.325 #atm pressure kPa
    Tit = Tit+Kel
    T_amb = T_amb+Kel
    s_0 = 0#S_Delta(0,Kel,T_amb,1,1)#start entropy
    V_0 = 1.

#PVplot(Case3, "k",0,[[1,5],[2,4]], "a")
Capstone()

#Tit = 100.

```



```

#Turbec()
#iSet2()
PVplot(RealRegen, "k", 1, [[0,4],[1,3]], "a", "C")

#Tit = 800+Kel
PVplot(RealAres, "k", 1, [[0,4],[1,3]], "a", "C")

#PVplot(NormalCycle, "k", 0, [[2,8],[3,7]], "a", "C")
#PVplot(MultiCycleHE, "k", 1, [[2,8],[3,7]], "a")
#PVplot(Ideal, "b")
#NsDs()

#plt.plot(x,mt)
#plt.plot(x,vr)
#VirkChart(RealRegen, False, 600.+Kel, 1600.+Kel, "Tit")
#plt.clf()
#
#VirkChart3d(RealRegen, False)
#VirkChart3d(RealAres, False)
#VirkChart2d(NormalCycle, False, 2., 10.)
#n_ph2 = [0.92, "%"]
#n_pc = [0.94, "%"]
#VirkChart2d(RealRegen, False, pmin, pmax)
#VirkChart2d(RealAres, False, 1.5, 6., "Pr")
#VirkChart2d(RealAres, False, 600.+Kel, 1600.+Kel, "Tit")
#VirkChart2d(RealAres, False, 2., 10.)
#VirkChart2d(NormalCycle, False, 2., 10.)

```




Norges miljø- og biovitenskapelig universitet
Noregs miljø- og biovitenskapelige universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway