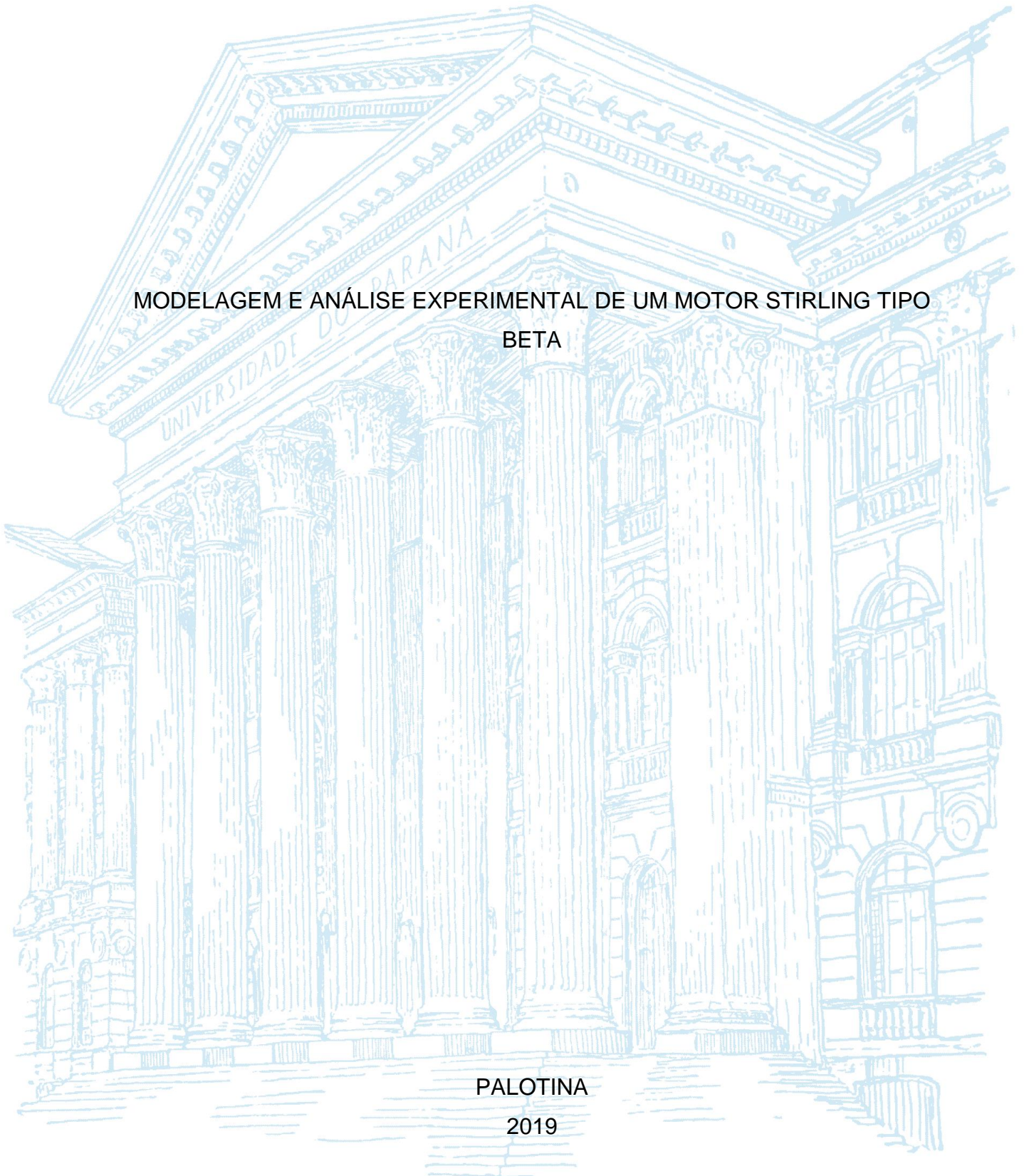


UNIVERSIDADE FEDERAL DO PARANÁ

EDUARDO MARCELO SAKAGUTI

MODELAGEM E ANÁLISE EXPERIMENTAL DE UM MOTOR STIRLING TIPO
BETA



PALOTINA

2019

EDUARDO MARCELO SAKAGUTI

MODELAGEM E ANÁLISE EXPERIMENTAL DE UM MOTOR STIRLING TIPO
BETA

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia de Energia, Setor de Palotina, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Engenheiro de Energia.

Orientador: Prof. Dr. Wilson de A. Beninca

PALOTINA

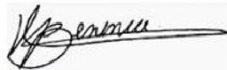
2019

TERMO DE APROVAÇÃO

EDUARDO MARCELO SAKAGUTI

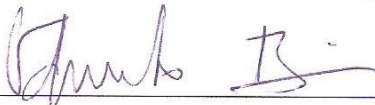
MODELAGEM E ANÁLISE EXPERIMENTAL DE UM MOTOR STIRLING TIPO BETA

Trabalho de Conclusão de Curso apresentado ao curso de Graduação em Engenharia de Energia, da Universidade Federal do Paraná, Setor Palotina, como requisito parcial à obtenção do título de Bacharel em Engenharia de Energia.



Prof. Dr. Wilson de Aguiar Beninca

Orientador – Departamento de Engenharias e Exatas, UFPR



Prof. Dr. Eduardo Lucas Konrad Burin

Departamento de Engenharias e Exatas, UFPR



Prof. Me. Maurício Romani

Departamento de Engenharias e Exatas, UFPR

Palotina, 05 de julho de 2019.

AGRADECIMENTOS

Agradeço a minha família por me apoiar desde o início até a realização deste trabalho.

Agradeço especialmente à minha esposa Simone e à minha mãe Nelly que foram cruciais para que esta conquista tenha se concretizado.

À minha filha Livia pela compreensão nos momentos em que não pude atender seus convites para brincar.

Aos meus professores e professoras, em especial ao meu orientador e amigo professor Wilson A. Beninca pela oportunidade desenvolver este trabalho desafiador e gratificante.

Aos meus colegas de classe com quem convivi e que também fazem parte desta conquista.

Agradeço especialmente à minha amiga Nathalia Dullius pela parceria nos Programas de Iniciação Científica e nos estudos sobre o motor Stirling.

Ao meu amigo e torneiro mecânico Sr. Luiz pela disposição em me ajudar na construção do protótipo de estudo deste trabalho.

RESUMO

O motor Stirling é uma máquina térmica de combustão externa. Esta característica permite a utilização de uma grande variedade de fontes de calor, inclusive as renováveis, biomassa e solar. Seu princípio de funcionamento se baseia no trabalho de expansão e compressão de um gás submetido a variações de temperatura. As principais configurações, segundo a geometria desses motores, são chamadas de Alfa, Beta e Gama. Este trabalho estudou a modelagem isotérmica, conhecida por teoria de Schmidt e a modelagem adiabática de Finkelstein para o ciclo de potência dos motores Stirling tipo Beta. Um protótipo foi construído no intuito de obter resultados experimentais e compará-los aos teóricos. Entretanto, a dificuldade na contenção do ar no protótipo, além de perdas por atrito, resultou em uma geração insuficiente de energia para o funcionamento do motor. A carga do mecanismo de acionamento também pesou negativamente para esta tarefa. Uma análise experimental utilizando um motor elétrico de indução demonstrou que seria necessário 75 W de potência para o acionar o motor a 180 rpm. Pelo modelo adiabático, a pressão média suficiente para alcançar tal potência deveria ser de 11 bar. Neste estudo pôde-se demonstrar que a variação das temperaturas nos espaços de expansão e compressão, propostas no modelo adiabático, é responsável pela diminuição do gradiente de temperatura do sistema e, conseqüentemente, pela redução da eficiência deste modelo em relação ao isotérmico.

Palavras-chave: Motor Stirling. Energia. Modelagem.

ABSTRACT

The Stirling engine is an external combustion thermal engine. This feature allows the use of a wide variety of heat sources, including renewable, biomass and solar. Its principle of operation is based on the work of expansion and compression of a gas subjected to temperature variations. The main configurations, according to the geometry of these motors, are called Alpha, Beta and Gamma. This paper studies isothermal modeling known as Schmidt's theory and Finkelstein's adiabatic model for the power cycle of Stirling type Beta engines. A prototype was constructed in order to obtain experimental results and compare them to theorists. However, the difficulty in containing air in the prototype, in addition to friction losses, resulted in insufficient power generation for the engine operation. The load of the drive mechanism also weighed negatively for this task. An experimental analysis using an induction electric motor demonstrated that it would require 75 W of power to drive the motor at 180 rpm. By the adiabatic model, the average pressure sufficient to achieve such power should be 11 bar. In this study it was possible to demonstrate that the temperature variation in the expansion and compression spaces, proposed in the adiabatic model, is responsible for the decrease of the temperature gradient of the system and, consequently, for the decrease of the efficiency of this model in relation to the isothermal.

Keywords: Stirling engine. Energy. Modeling.

LISTA DE FIGURAS

FIGURA 1:	CLASSIFICAÇÃO SEGUNDO A GEOMETRIA.....	20
FIGURA 2:	DIAGRAMAS PV E TS DO CICLO STIRLING	20
FIGURA 3:	MOTOR STIRLING HIPOTÉTICO	21
FIGURA 4:	COMPARAÇÃO ENTRE DIAGRAMAS P _x V IDEAL E ISOTÉRMICO	23
FIGURA 5:	VARIAÇÕES DOS VOLUMES DE COMPRESSÃO E EXPANSÃO PARA O TIPO BETA.....	24
FIGURA 6:	DIAGRAMA P _x V	24
FIGURA 7:	MECANISMO RÔMBICO DE ACIONAMENTO	25
FIGURA 8:	MODELO ESQUEMÁTICO MODELO ADIABÁTICO	27
FIGURA 9:	VOLUME DE CONTROLE GENERALIZADO	28
FIGURA 10:	CABEÇOTE DE COMPRESSOR DE AR.....	30
FIGURA 11:	AMORTECEDOR DE AUTOMÓVEL	31
FIGURA 12:	MONTAGEM DO CILINDRO.....	32
FIGURA 13:	MATERIAIS UTILIZADOS E MONTAGEM DO PISTÃO DE ARMAZENAMENTO TÉRMICO.....	33
FIGURA 14:	AVANÇO DO PISTÃO DE ARMAZENAMENTO TÉRMICO	34
FIGURA 15:	MONTAGEM DO MECANISMO DE ACIONAMENTO	35
FIGURA 16:	PONTO DE FIXAÇÃO DO MECANISMO DE ACIONAMENTO NA POLIA	35
FIGURA 17:	POSIÇÃO DAS HASTES NOS ACIONAMENTOS RÔMBICO E DO PROTÓTIPO	39
FIGURA 18:	VARIAÇÃO DA POTÊNCIA PELO AUMENTO DA TEMPERATURA DE EXPANSÃO	43
FIGURA 19:	TEMPERATURA DO FLUIDO NOS COMPONENTES DO MOTOR .	45
FIGURA 20:	DIAGRAMAS P _x V ISOTÉRMICO E ADIABÁTICO	46
FIGURA 21:	VARIAÇÃO DA POTÊNCIA PELO AUMENTO DE PRESSÃO.....	47

LISTA DE TABELAS

TABELA 1:	GEOMETRIA E CONDIÇÕES DE TRABALHO DO MOTOR.....	36
TABELA 2:	RESULTADOS PARA A ANÁLISE ISOTÉRMICA	43
TABELA 3:	RESULTADOS PARA A ANÁLISE ADIABÁTICA	44
TABELA 4:	RESULTADOS PARA A POTÊNCIA NECESSÁRIA PARA O ACIONAMENTO DO MOTOR.....	46

LISTA DE SÍMBOLOS

T_c – Temperatura de compressão
 T_k – Temperatura do resfriador
 T_r – Temperatura do regenerador
 T_h – Temperatura do aquecedor
 T_e – Temperatura de expansão
 V_c – Volume de compressão
 V_k – Volume do resfriador
 V_r – Volume do regenerador
 V_h – Volume do aquecedor
 V_e – Volume de expansão
 m_c – Massa de gás no volume de compressão
 m_k – Massa de gás no volume do resfriador
 m_r – Massa de gás no volume do regenerador
 m_h – Massa de gás no volume do aquecedor
 m_e – Massa de gás no volume de expansão
 m_{ck}' – Fluxo de massa entre o volume de compressão e o resfriador
 m_{kr}' - Fluxo de massa entre o resfriador e o regenerador
 m_{rh}' - Fluxo de massa entre o regenerador e o aquecedor
 m_{he}' - Fluxo de massa entre o aquecedor e o volume de expansão
 T_{ck} – Temperatura do fluxo de massa m_{ck}'
 T_{kr} - Temperatura do fluxo de massa m_{kr}'
 T_{rh} - Temperatura do fluxo de massa m_{rh}'
 T_{he} - Temperatura do fluxo de massa m_{he}'
 W_c – Trabalho de compressão
 W_e – Trabalho de expansão
 Q_k – Fluxo de calor no resfriador
 Q_h – Fluxo de calor no aquecedor
 dU – Diferencial de fluxo de energia interna
 dQ – Diferencial de fluxo de calor
 dW – Diferencial de trabalho
 m_i – Massa de gás que entra no sistema
 m_o – Massa de gás que sai do sistema

h_i – Entalpia de entrada

h_o – Entalpia de saída

Dm – Diferencial de massa

SUMÁRIO

1 INTRODUÇÃO	16
1.1 JUSTIFICATIVA	17
1.2 OBJETIVOS	17
1.2.1 Objetivo geral	17
1.2.2 Objetivos específicos.....	17
2 REVISÃO DE LITERATURA	19
2.1 CICLOS DE POTÊNCIA.....	19
2.2 MOTOR STIRLING	19
2.3 CICLO STIRLING IDEAL.....	20
2.4 TEORIA DE SCHMIDT.....	23
2.5 MECANISMOS DE ACIONAMENTO	25
2.6 CICLO ADIABÁTICO IDEAL	27
3 MATERIAL E MÉTODOS	30
3.1 MATERIAIS	30
3.1.1 Cabeçote de compressor de ar	30
3.1.2 Carcaça de amortecedor de automóvel;.....	31
3.1.3 Materiais diversos	31
3.2 MONTAGEM	32
3.2.1 CILINDRO E PISTÃO DE POTÊNCIA.....	32
3.2.2 PISTÃO DESLOCADOR	32
3.3 MECANISMO DE ACIONAMENTO.....	34
3.4 AFERIÇÃO DOS VOLUMES E DEFINIÇÃO DE PARÂMETROS	36
3.5 SIMULAÇÃO DO MODELO ISOTÉRMICO.....	36
3.5.1 Cálculo do inventário de gás do protótipo	36
3.5.2 Entrada de dados e execução do algoritmo	37
3.6 SIMULAÇÃO DO MODELO ADIABÁTICO.....	38
3.7 CÁLCULO DA POTÊNCIA	39
3.8 ANÁLISE EXPERIMENTAL.....	39
3.8.1 Diminuição de carga.....	40
3.8.2 Estimativa da potência necessária para o funcionamento do protótipo.....	40
4 RESULTADOS E DISCUSSÕES	41
4.1 INVENTÁRIO DE GÁS E PRESSÃO DO SISTEMA	41

4.2 RESULTADOS DA ANÁLISE ISOTÉRMICA.....	41
4.2.1 Redução da eficiência	41
4.2.2 Pressão média	42
4.2.3 Variação da potência com o aumento da rotação do motor	42
4.2.4 Variação da potência com o aumento da temperatura de expansão	43
4.3 RESULTADOS DA ANÁLISE ADIABÁTICA.....	44
4.3.1 Variação da potência com o aumento da rotação do motor	44
4.3.2 Variação das temperaturas de expansão e compressão.....	44
4.4 COMPARAÇÃO DOS DIAGRAMAS PRESSÃO POR VOLUME	45
4.5 RESULTADOS DA ANÁLISE EXPERIMENTAL	46
5 CONSIDERAÇÕES FINAIS	48
REFERÊNCIAS.....	49
ANEXO A	51
ANEXO B	53
APÊNDICE A	80
APÊNDICE B	81

1 INTRODUÇÃO

O motor Stirling foi inventado no século XIX e teve seu desenvolvimento praticamente paralisado por mais de cem anos devido ao avanço tecnológico dos motores a combustão interna, movidos à combustíveis fósseis. Com a crescente demanda ambiental relacionada à diminuição de gases do efeito estufa como o dióxido de carbono, o motor Stirling voltou a se tornar objeto de estudo de pesquisadores de todo o mundo pela sua disponibilidade de operação utilizando várias fontes de calor, entre elas as renováveis como biomassa, biogás, residual e solar (GHEITH et al. (2011).

O funcionamento do motor Stirling baseia-se na expansão e na compressão de um gás de trabalho que se alterna em um gradiente de temperatura. O aumento e a diminuição do volume devido ao aquecimento e ao resfriamento do gás movimentam pistões, convertendo energia térmica em energia mecânica.

As principais configurações são Alfa, Beta e Gama, que se diferenciam pela disposição dos pistões. A configuração Alfa inclui dois pistões em cilindros separados. Na configuração Beta, um pistão e um deslocador estão dispostos no mesmo cilindro. Para o tipo Gama, pistão e deslocador estão em cilindros separados (SHENDAGE et al., 2011)

A modelagem do ciclo termodinâmico do motor Stirling inicia-se pelo ciclo ideal que consiste em dois processos isotérmicos e dois processos isovolumétricos e opera entre uma temperatura mínima T_C e uma temperatura máxima T_E .

Posteriormente, na modelagem isotérmica conhecida por teoria de Schmidt, o movimento dos pistões foi considerado senoidal, permitindo uma transição gradual entre os processos. Além disso, volumes não percorridos pelos pistões, chamados de volumes mortos, são considerados.

Outra evolução importante na modelagem dos Stirling foi a adiabática ideal de Finkelstein, onde a temperatura do gás pode variar entre limites mínimos e máximos, nas câmaras de expansão e compressão, dependendo do sentido do movimento do gás. Esta análise eliminou a situação paradoxal da modelagem isotérmica em que os trocadores de calor não poderiam contribuir com qualquer transferência líquida de calor por não haver um gradiente de temperatura (THOMBARE; VERMA, 2008).

Os estudos mais recentes sobre os motores Stirling estão direcionados aos fatores que desviam o desempenho do motor real do caso ideal. Além dos problemas

comuns a qualquer máquina térmica como as irreversibilidades e a não-idealidade das transferências de calor, aspectos mecânicos e fluidodinâmicos, o motor Stirling ainda é penalizado por perdas de carga no regenerador e perdas mecânicas no mecanismo de potência mecânica.

Com base nas considerações apresentadas, este estudo buscou comparar os resultados teóricos entre as modelagens isotérmica e adiabática para um motor Stirling de configuração Beta. As dimensões do motor, utilizadas na simulação, têm origem em um protótipo construído pelo autor.

1.1 JUSTIFICATIVA

O desenvolvimento de tecnologias alternativas para a geração de energia elétrica a partir de fonte renováveis se torna relevante do ponto de vista ambiental, considerando a previsão para o aquecimento global, de 1,5 °C entre os anos de 2030 de 2052 (IPCC, 2018), provocado pelo aumento da emissão de gases do efeito estufa ocasionado pelo consumo de combustíveis fósseis.

1.2 OBJETIVOS

1.2.1 Objetivo geral

O objetivo deste trabalho é estudar os fenômenos que envolvem o funcionamento de um motor Stirling de configuração Beta através das modelagens matemáticas isotérmica e adiabática, bem como construir um protótipo e comparar os resultados teóricos de desempenho aos resultados reais.

1.2.2 Objetivos específicos

São objetivos específicos deste trabalho:

- Verificar a influência da temperatura e da pressão no desempenho do motor ideal e real em um estudo paramétrico variando estas grandezas nas modelagens propostas neste trabalho.
- Comparar os resultados das modelagens isotérmica e adiabática.

- Construir um protótipo do motor Stirling tipo Beta utilizando e adaptando partes de dispositivos diversos para reaproveitar seus mecanismos, simplificando o processo construtivo.

2 REVISÃO DE LITERATURA

Uma revisão bibliográfica é apresentada nesta seção, abordando o conceito de ciclos de potência ou motores, classificações principais de motores Stirling e seu funcionamento em um ciclo idealizado. Em seguida a teoria de Schmidt e a análise adiabática de Finkelstein são apresentadas evidenciando o ganho de qualidade na modelagem.

2.1 CICLOS DE POTÊNCIA

Dispositivos utilizados para gerar potência, chamados de motores operam segundo um ciclo termodinâmico. O ciclo pode ser aberto, quando o fluido é renovado ao final de cada ciclo, ou fechado, quando o fluido volta ao seu estado inicial e circula novamente (ÇENGEL; BOLES, 2013).

Motores como os de automóveis ou turbinas a gás são máquinas térmicas de ciclo aberto, já que seus gases de exaustão são descarregados e substituídos por uma nova mistura de ar e combustível. Máquina térmica é um dispositivo que permite transformar calor em trabalho, onde uma substância entra no volume de controle, passa por uma sequência de processos e resulta em produtos de combustão na saída. (BRUNETTI, 2016)

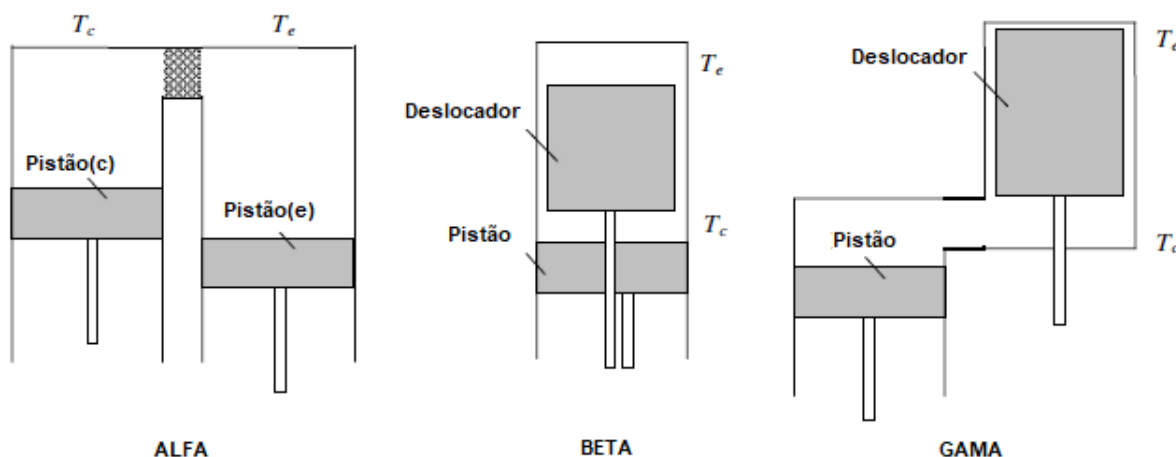
As máquinas térmicas também podem ser classificadas como máquinas de combustão interna e externa. Nos motores de automóveis, o calor fornecido para o fluido de trabalho vem da queima do combustível dentro das fronteiras do sistema, o que caracteriza uma combustão interna. Em usinas de potência a vapor, o calor é fornecido ao fluido de trabalho de uma fonte externa de combustão.

2.2 MOTOR STIRLING

O motor Stirling é uma máquina de combustão externa permitindo ser operado por uma grande variedade de fontes de calor. São comumente classificados em três tipos sob termos de configuração e nomeadas em Alfa, Beta e Gama (FIGURA 1). A configuração Alfa inclui dois pistões em cilindros separados. Na configuração Beta, um pistão e um deslocador ficam acomodados em um mesmo cilindro. Na configuração gama, deslocador e pistão ficam em cilindros separados. Embora a

configuração Beta seja mais complexa, ela tem vantagens sobre as outras por permitir uma sobreposição de volumes. Com isso pode-se ter trocadores de calor 30% maiores do que na configuração gama para a mesma taxa de compressão. (SHENDAGE et al, 2011)

FIGURA 1: CLASSIFICAÇÃO SEGUNDO A GEOMETRIA

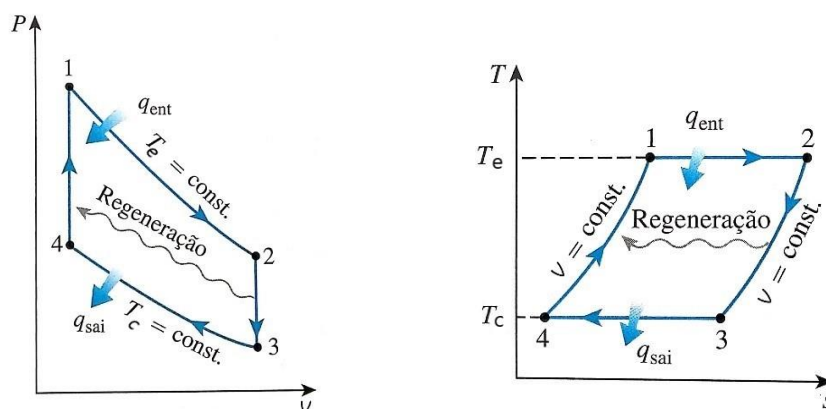


FONTE: CHENG (2012).

2.3 CICLO STIRLING IDEAL

A análise básica do ciclo termodinâmico do motor Stirling para a obtenção de seu desempenho é o ciclo Stirling ideal. Ele consiste em dois processos isotérmicos de fornecimento e rejeição de calor e dois processos de regeneração a volume constante (Figura 1) e opera entre temperaturas máxima T_E e mínima T_C .

FIGURA 2: DIAGRAMAS PV E TS DO CICLO STIRLING



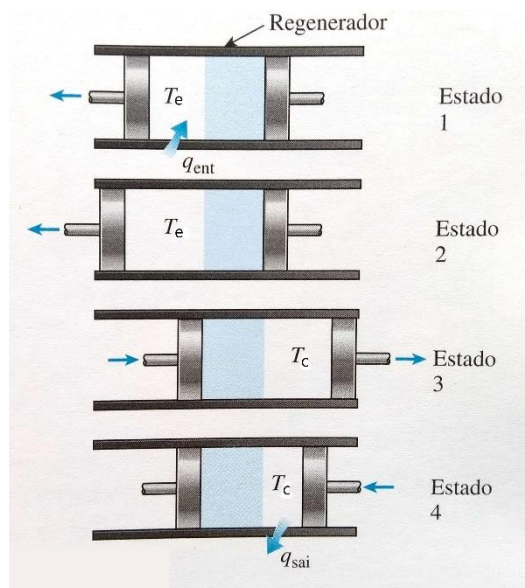
FONTE: ÇENGEL; BOLES (2013).

Para um melhor entendimento sobre os processos, um motor hipotético está representado na FIGURA 3, composto por um pistão de cada lado e um regenerador no meio.

À esquerda da figura, entre o pistão e o regenerador há uma câmara de expansão à alta temperatura T_E e analogamente à direita, uma câmara de compressão à baixa temperatura T_C .

O regenerador é um dispositivo que armazena parte do calor rejeitado pelo gás, quando este se desloca da câmara quente para a câmara fria. O calor armazenado é absorvido pelo gás no sentido inverso do deslocamento, aumentando a eficiência do ciclo. É composto, geralmente de uma malha metálica de alta capacidade térmica.

FIGURA 3: MOTOR STIRLING HIPOTÉTICO



FONTE: ÇENGEL; BOLES (2013)

Os quatro processos detalhados são:

- Expansão isotérmica 1-2: na câmara quente, o fluido recebe calor através de um trocador, se expande de forma isotérmica à temperatura T_E e movimenta o pistão realizando trabalho.
- Resfriamento isocórico 2-3: os dois pistões se movimentam simultaneamente para a direita mantendo o volume constante. O fluido

atravessa o regenerador, onde calor é armazenado, chegando à câmara de compressão com temperatura T_C .

- c) Compressão isotérmica 3-4: na câmara fria, o pistão da direita comprime o gás enquanto calor é rejeitado a um sumidouro mantendo a temperatura T_C .
- d) Aquecimento isocórico 4-1: os dois pistões se movimentam simultaneamente para a esquerda mantendo o volume constante. O fluido atravessa o regenerador, onde recebe calor, chegando à câmara de expansão à temperatura T_E .

As equações (1) e (3) se referem aos trabalhos de expansão e compressão a temperaturas constantes T_E e T_C , respectivamente, enquanto as equações (2) e (4) representam os cálculos de rejeição e fornecimento de calor a volumes constantes (MARTINI, 1983).

$$W_{1-2} = \int_{V_1}^{V_2} P dv = mRT_E \ln \frac{V_2}{V_1} \quad (1)$$

$$Q_{2-3} = mC_V(T_C - T_E) \quad (2)$$

$$W_{3-4} = \int_{V_3}^{V_4} P dv = mRT_C \ln \frac{V_1}{V_2} \quad (3)$$

$$Q_{4-1} = mC_V(T_E - T_C) \quad (4)$$

Onde:

- W_{1-2} = trabalho de expansão sobre o pistão
- W_{3-4} = trabalho de compressão sobre o pistão
- Q_{2-3} = calor rejeitado para o sumidouro
- Q_{4-1} = calor fornecido ao fluido de trabalho
- V_1 e V_2 = volumes mínimo e máximo, respectivamente
- m = massa do gás
- C_V = calor específico a volume constante
- R = constante universal dos gases ideais

A eficiência η do ciclo, equação (5), é conhecida como a fórmula para a eficiência de Carnot, ou seja, o limite para a eficiência do motor Stirling é a máxima possível.

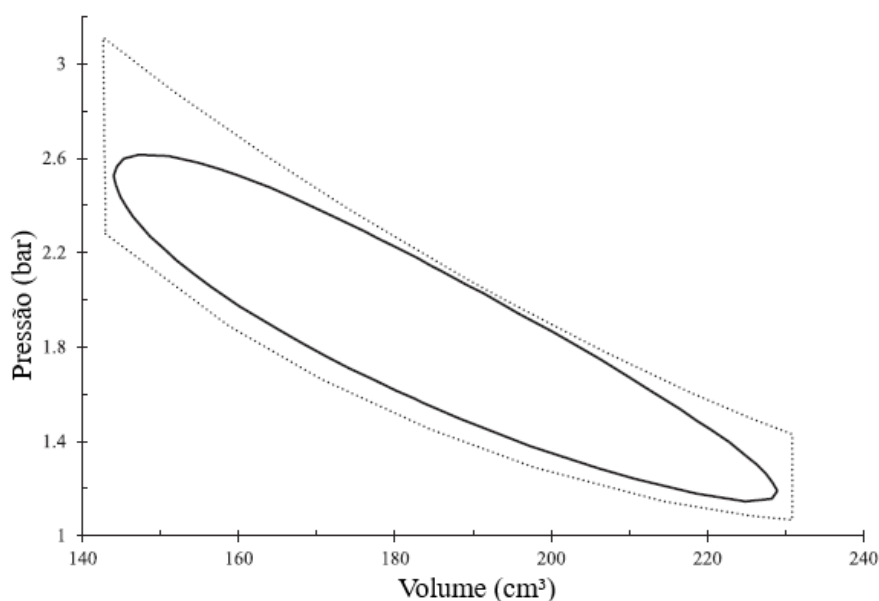
$$\eta = \frac{T_E - T_C}{T_E} \quad (5)$$

2.4 TEORIA DE SCHMIDT

O modelo ideal, impossível de ser reproduzido por causa da distinção perfeita entre os processos, evoluiu em 1871 quando Schmidt (1871) publicou uma solução analítica para o ciclo Stirling onde o deslocador e o pistão de potência (tipos Beta e Gama) ou os dois pistões (tipo Alfa) se movimentam de maneira senoidal, permitindo a análise das transições graduais entre um processo e o outro. Ranieri et al. (2018) afirma que a operação senoidal do motor resulta na presença do fluido de trabalho em ambos espaços (quente e frio) durante os cursos de expansão e compressão.

Ainda bastante idealizado, o modelo isotérmico, como é conhecido, também considera os volumes mortos do sistema, levando a modelagem dos motores Stirling a um comportamento mais próximo a de um motor real. A FIGURA 4 apresenta a diferença entre os diagramas pressão por volume da análise ideal e isotérmica.

FIGURA 4: COMPARAÇÃO ENTRE DIAGRAMAS P×V IDEAL E ISOTÉRMICO



FONTE: BATAINEH (2018)

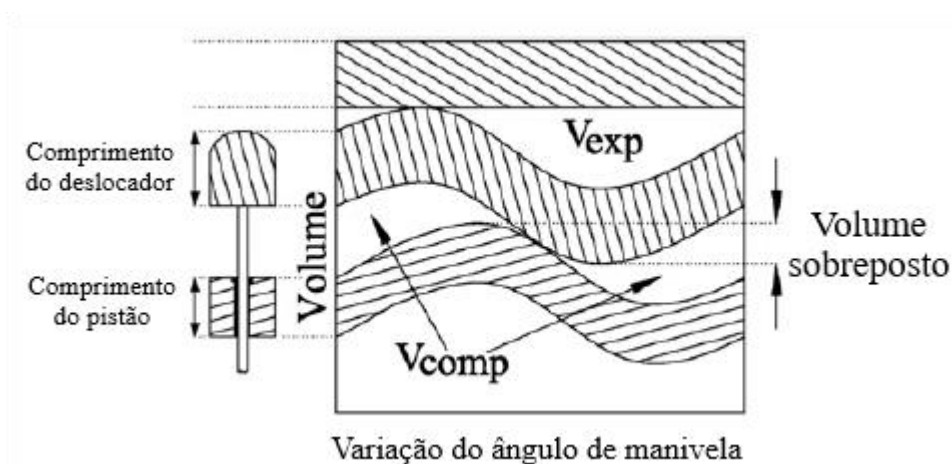
A análise de Schmidt ainda considera temperaturas conhecidas e constantes em todas as partes do motor, ausência de vazamentos e perdas de carga, além do comportamento ideal do gás de trabalho (MARTINI, 1983)

A FIGURA 5 representa o movimento senoidal do pistão e do deslocador de um motor Stirling tipo Beta. Nele é possível observar a variação dos volumes de

expansão e compressão do motor em um ciclo, bem como a diferença de fase entre pistão e deslocador.

Como mencionado anteriormente, no tipo Beta, pistão e deslocador ocupam o mesmo cilindro. Esta característica possibilita uma sobreposição de volumes, já que, para certo ângulo de fase, pistão e deslocador podem percorrer uma região do cilindro em comum, o que cria um espaço de trabalho útil (HIRATA, 1997).

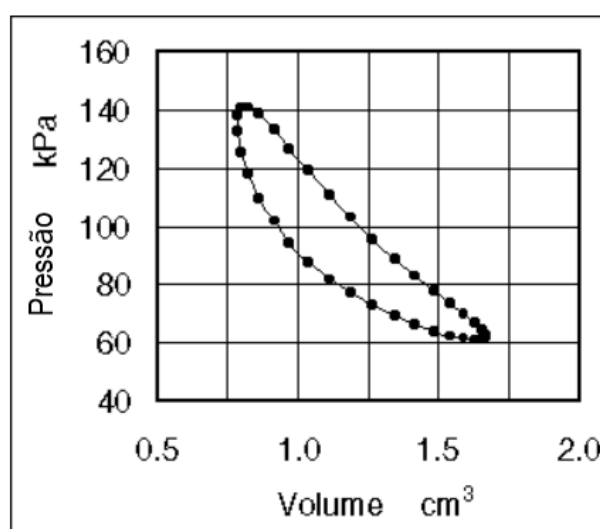
FIGURA 5: VARIAÇÕES DOS VOLUMES DE COMPRESSÃO E EXPANSÃO PARA O TIPO BETA



FONTE: Adaptado de SHENDAGE et al. (2011)

O desempenho do motor pode ser calculado através de um diagrama P-V (FIGURA 6) formado pelos valores de pressão e volume para um ciclo completo, ou seja, com o ângulo de manivela θ variando de 0 a 360 graus.

FIGURA 6: DIAGRAMA PxV



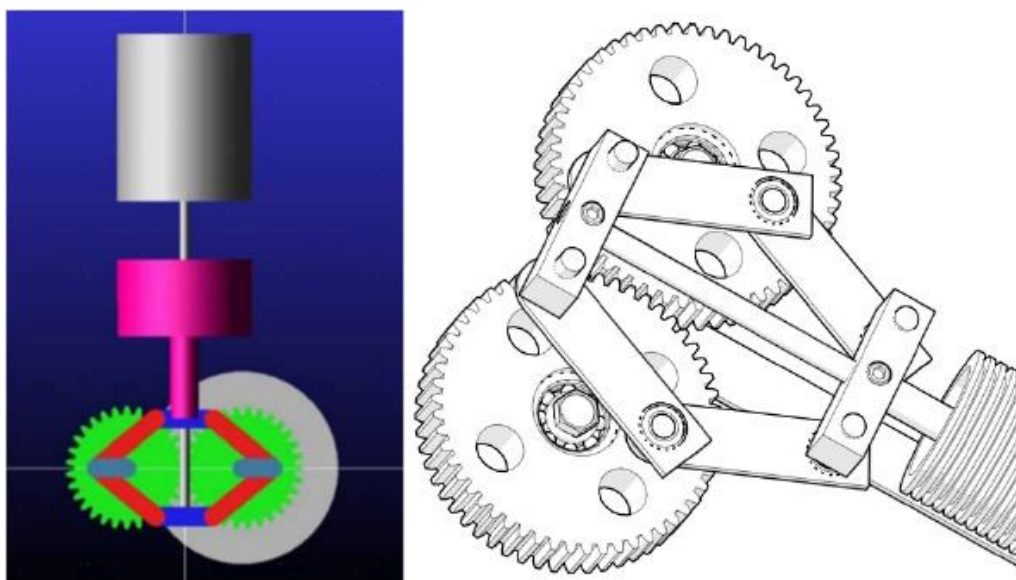
FONTE: HIRATA (1997)

Martini (1983) encontrou resultados muito próximos calculando o trabalho líquido de um motor hipotético por métodos numéricos e pelas equações de Schmidt.

2.5 MECANISMOS DE ACIONAMENTO

Mecanismos de acionamento são utilizados na conversão da energia térmica em energia mecânica nos motores Stirling. Na geração elétrica, por exemplo, o movimento linear do(s) pistão(ões) precisa ser transformado em movimento rotativo do rotor de um gerador. Uma comparação entre estes mecanismos revela que o mecanismo rômbo é o mais adequado para os motores do tipo Beta (EROL ET AL. (2017). Ele consiste em um mecanismo de hastes conectadas a duas engrenagens sincronizadas girando em direções opostas e um virabrequim. A diferença de fase entre o movimento dos dois pistões deve movimentar as hastes em uma sincronia que seja capaz de manter o movimento circular das engrenagens.

FIGURA 7: MECANISMO RÔMBICO DE ACIONAMENTO



FONTES: EROL et al. (2017); LEITE (2018)

Cinar et al. (2005), em seu estudo experimental com um protótipo do tipo Beta, concluiu que materiais mais resistentes à altas temperaturas e a utilização de fluidos com maiores condutividades térmicas como hélio e hidrogênio, poderiam elevar o desempenho.

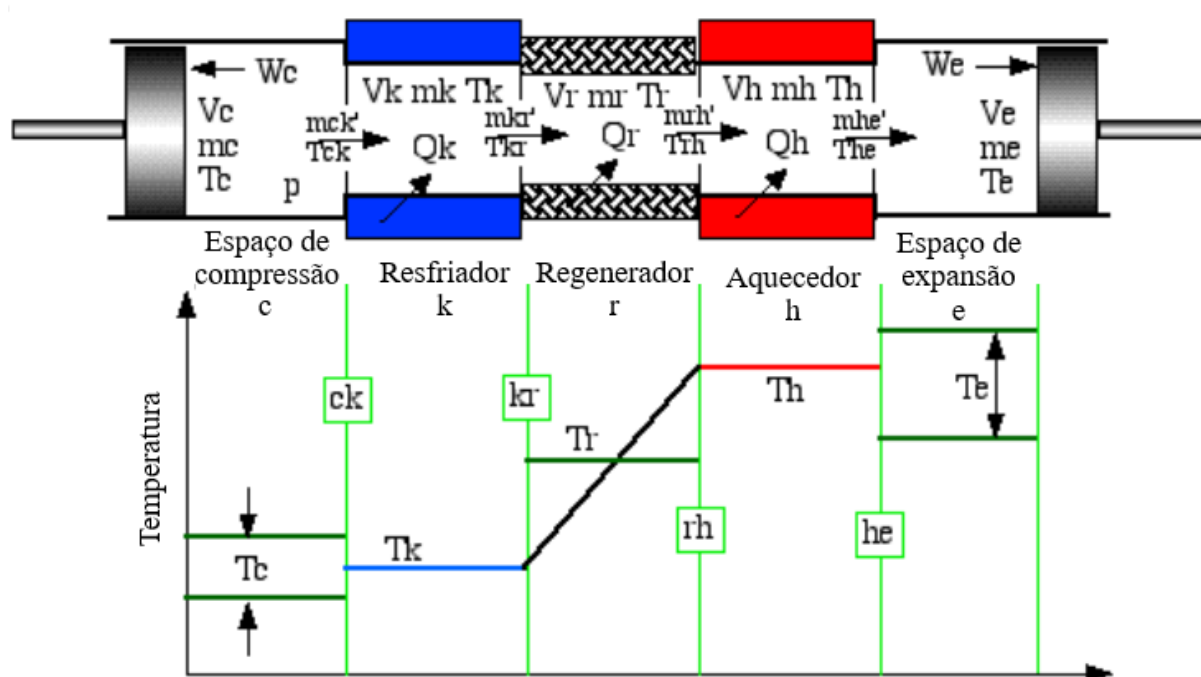
Caetano et al. (2019) combinou a análise do modelo de Schmidt com simulações em CFD, no intuito de diminuir o esforço computacional do método. A pressão média e o perfil de temperatura obtidos no modelo isotérmico foram utilizados como condição inicial e de fronteira na análise fluidodinâmica dos autores.

2.6 CICLO ADIABÁTICO IDEAL

Um próximo passo na direção de um modelo mais real é considerar que o gás de trabalho tenha variação de temperatura na câmara de expansão e na câmara de compressão, ou seja, deixa de ter comportamento isotérmico e passa ter comportamento adiabático. Esta análise foi realizada primeiramente por Finkelstein (1960). Os motores reais tendem a ser mais adiabáticos do que isotérmicos, principalmente em elevadas rotações (URIELI, 2018).

As equações desenvolvidas para este modelo baseiam-se em um motor dividido em cinco células conectadas em série e trocadores ideais de calor (FIGURA 8). Os sufixos (c, k, r, h, e) representam cada parte e os sufixos duplos (ck, kr, rh, he) representam as interfaces entre as células. Q , T , m e V representam calor, temperatura, massa e volume, respectivamente (URIELI; BERCHOWITZ, 1984).

FIGURA 8: MODELO ESQUEMÁTICO MODELO ADIABÁTICO



FONTE: URIELI; BERCHOWITZ (1984)

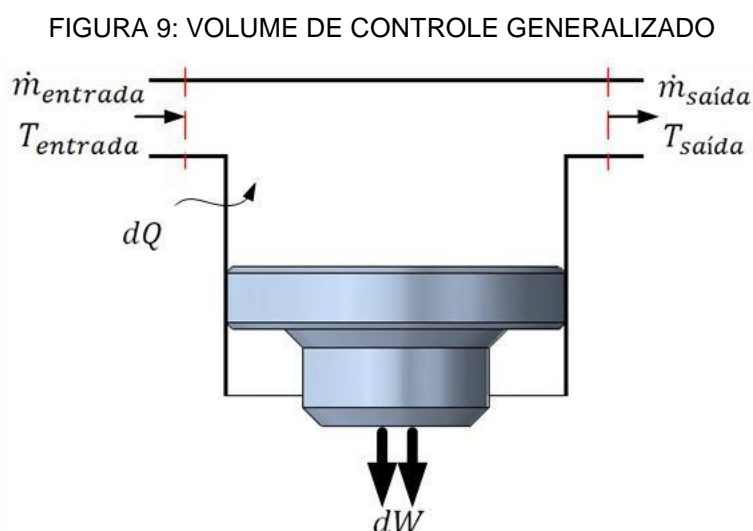
Neste caso, os espaços de compressão (c) e expansão (e) são adiabáticos e as temperaturas de cada um (Tc e Te) variam de acordo com o sentido do fluxo de massa do fluido de trabalho. Considerando o sentido da esquerda para a direita do observador como positivo, se o fluxo de massa (mck') na interface ck é positivo, então

a temperatura nesta interface (T_{ck}) é igual à temperatura do espaço de compressão (T_c), senão, T_{ck} é igual à temperatura do resfriador (T_k). Analogamente, se m_{he}' é positivo, T_{he} é igual a T_h , senão T_{he} é igual a T_e . A sentença a seguir resume estas condições.

$$\begin{aligned} \text{se } m_{ck}' > 0 \text{ então } T_{ck} &= T_c \text{ senão } T_{ck} = T_k \\ \text{se } m_{he}' > 0 \text{ então } T_{he} &= T_h \text{ senão } T_{he} = T_e \end{aligned}$$

Nos trocadores de calor, representados pelas células k para o resfriador e h para o aquecedor, o gás mantém as temperaturas constantes T_k e T_h , respectivamente. No regenerador, apesar de não ocorrer na prática, uma temperatura média e constante T_r é considerada. Essa temperatura pode ser melhor representada por uma variação linear entre as temperaturas dos trocadores quente e frio (SANT'ANNA; YANO, 2012).

Delimitadas as condições de fronteira para cada um dos volumes de controle e considerando o gás como ideal, a equação (6) da conservação da massa e da energia é aplicada em cada uma das células representadas de forma generalizada na FIGURA 9. A equação (7) da continuidade é aplicada no intuito de avaliar a direção do fluxo de massa e assim, definir as temperaturas T_{ck} e T_{he} . É simplesmente uma afirmação de que a taxa de acúmulo de massa em uma célula é igual ao fluxo líquido de massa em cada uma delas (THOMBARE; VERMA, 2008).



FONTE: SANT'ANNA; YANO (2012)

$$dU = dQ - dW + \dot{m}_i h_i - \dot{m}_o h_o \quad (6)$$

$$D_m = m_i - m_o \quad (7)$$

A aplicação das equações citadas em todos os volumes de controle resulta em equações diferenciais para a pressão do sistema, para a massa nos volumes de controle, para a temperatura nas interfaces *ck* e *he*, para o calor transferido nos trocadores e para os trabalhos envolvidos. Todas as equações envolvidas estão listadas na seção ANEXO I deste documento.

Timoumi et al. (2008) desenvolveu um modelo de simulação numérico, validado pelo modelo de Urieli e Berchowitz (1984), para otimizar o desempenho do motor levando em conta perdas por vazamento de gás, por condução térmica entre os trocadores e entre o regenerador e ambiente externo.

Nos estudos de Sripakagorn e Srikam (2011) foi utilizada a rotina *simples* de Urieli e Berchowitz (1984), baseada no modelo adiabático do mesmo autor que avalia as transferências de calor e perdas pelo fluxo do fluido nos trocadores de calor do motor. O potencial de um protótipo do motor Stirling tipo Beta de temperatura moderada (350°C e 500°C) foi considerada uma opção vantajosa para a aplicação na geração elétrica em sistemas de concentradores solares.

3 MATERIAL E MÉTODOS

A metodologia empregada consiste na escolha e preparação dos materiais (3.1), na montagem (3.2) do protótipo de um motor Stirling tipo Beta, na aferição dos volumes de trabalho e volumes mortos do protótipo (3.3), assim como na definição dos parâmetros de operação do motor (3.3); e na simulação dos modelos isotérmico (3.5) e adiabático (3.6).

3.1 MATERIAIS

Os materiais utilizados na construção do protótipo envolvem peças reutilizadas provenientes de diversos dispositivos. Nesta seção eles estão detalhados quanto as suas geometrias, bem como as adaptações que foram necessárias para a montagem e para a análise experimental.

3.1.1 Cabeçote de compressor de ar

Utilizado originalmente na produção de ar comprimido, faz o papel do pistão de potência. É onde o trabalho de expansão e compressão do gás movimenta o pistão de forma linear no qual um mecanismo de transmissão o transforma no movimento circular da polia (FIGURA 10).

FIGURA 10: CABEÇOTE DE COMPRESSOR DE AR



FONTE: O autor (2019)

3.1.2 Carcaça de amortecedor de automóvel;

Foi escolhido por ter um diâmetro interno similar ao cilindro do pistão de potência e, principalmente por dispor de um dispositivo de retenção da pressão. Os amortecedores de automóveis (FIGURA 11) trabalham sob altas pressões e não possuem vazamentos. O movimento de uma haste que desliza através de um orifício equipado com um retentor é responsável pela função de amortecimento. Para sua utilização no protótipo, foi cortado em um tamanho apropriado para desempenhar o papel do cilindro que abriga o regenerador. É o espaço onde o regenerador absorve e cede calor ao gás, promovendo sua compressão e expansão ciclicamente. A haste também foi aproveitada como detalhado na seção (3.2.2), bem como o retentor de pressão.

FIGURA 11: AMORTECEDOR DE AUTOMÓVEL



FONTE: O autor (2019)

3.1.3 Materiais diversos

Materiais metálicos para a construção de flanges, aletas, mecanismo de transmissão foram utilizados, além de elementos para vedação e parafusos. O regenerador foi construído com a carcaça de uma bobina em alumínio e palha de aço inox.

3.2 MONTAGEM

A montagem do protótipo se resumiu no acoplamento do cabeçote de compressor com o cilindro preparado através de um flange (3.2.1); na confecção do deslocador e do regenerador (3.2.2) e do mecanismo de acionamento do motor (3.3).

3.2.1 CILINDRO E PISTÃO DE POTÊNCIA

A FIGURA 12 apresenta a montagem do cabeçote de compressor com parte do cilindro do amortecedor. Um flange foi confeccionado em torno mecânico para o acoplamento das duas peças. Também foram construídas aletas e soldadas na parte inferior do cilindro com a função de facilitar a transferência de calor para o ambiente, já que esta é a região do resfriador do motor.

O conjunto pistão de potência, biela, virabrequim e polia estão integrados ao cabeçote de compressor de ar e precisaram apenas de lubrificação.

FIGURA 12: MONTAGEM DO CILINDRO



FONTE: O autor (2019)

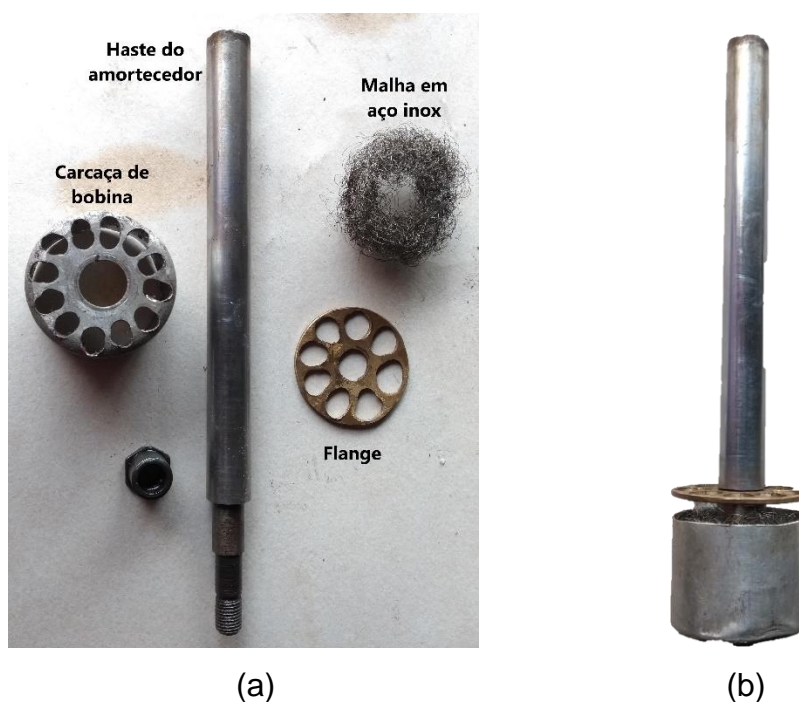
3.2.2 PISTÃO DESLOCADOR

Neste protótipo, diferentemente da maioria dos protótipos encontrados na literatura, onde o movimento do pistão deslocador empurra o gás forçando-o a passar

pelo regenerador, optou-se por construir o regenerador no espaço interno do deslocador. O motivo para esta decisão foi a maior simplicidade na construção do protótipo, já que permite a reutilização de materiais com características semelhantes às necessárias ao seu funcionamento. Desta forma, a nomenclatura ficou distorcida para este dispositivo, já que sua função não é deslocar o gás, mas passar através dele forçando a troca de calor entre o fluido e a malha metálica. Para ajustar o nome do dispositivo à sua função, este texto se referirá a ele, daqui até o final, como pistão de armazenamento térmico.

O pistão de armazenamento térmico foi construído com parte da carcaça de uma bobina, em alumínio, um flange em cobre e uma malha metálica de aço inox (FIGURA 13-a). Este foi fixado à haste do amortecedor que, por sua vez, faz a conexão com o mecanismo de acionamento, no exterior do cilindro (FIGURA 13-b).

FIGURA 13: MATERIAIS UTILIZADOS E MONTAGEM DO PISTÃO DE ARMAZENAMENTO TÉRMICO



FONTE: O autor (2019)

A parte inferior do pistão de armazenamento térmico foi confeccionado na forma de um copo. A configuração Beta do motor Stirling permite a sobreposição de volumes, o que implica no avanço do deslocador no espaço de compressão. A forma de copo do deslocador evita um possível choque dele com as partes metálicas na

interface do cabeçote com o cilindro, no caso deste ser construído na forma de flange, por exemplo. Para um melhor entendimento, uma foto do pistão deslocador é apresentada na FIGURA 14.

FIGURA 14: AVANÇO DO PISTÃO DE ARMAZENAMENTO TÉRMICO



FONTE: O autor (2019)

3.3 MECANISMO DE ACIONAMENTO

É o mecanismo que conecta o movimento do pistão de potência ao pistão de armazenamento térmico. Esta conexão deve satisfazer uma sincronia de movimento entre os dois pistões que seja capaz de expandir o gás pelo aquecimento e contraí-lo pelo resfriamento, realizando trabalhos de expansão e compressão. Esta condição é satisfeita por uma diferença de fase entre os movimentos dos dois pistões, que para a análise de Schmidt é de 90° (MARTINI, 1983).

A reutilização de materiais realizada neste trabalho dificultou a construção de um mecanismo de acionamento encontrado na literatura, como o rômbico, citado neste texto. Sua montagem (FIGURA 15) e seu funcionamento foi adaptado às condições construtivas que os materiais utilizados impuseram.

FIGURA 15: MONTAGEM DO MECANISMO DE ACIONAMENTO



FONTE: O autor (2019)

O curso do pistão de armazenamento térmico foi definido pelo ponto de fixação da haste do mecanismo de acionamento na polia do cabeçote (FIGURA 16). Esta haste movimenta um balancim que, por sua vez, movimenta o pistão de armazenamento térmico.

FIGURA 16: PONTO DE FIXAÇÃO DO MECANISMO DE ACIONAMENTO NA POLIA



FONTE: O autor (2019)

3.4 AFERIÇÃO DOS VOLUMES E DEFINIÇÃO DE PARÂMETROS

Definida a configuração mecânica do protótipo, realizou-se a aferição dos volumes de interesse para as modelagens previstas para este trabalho.

As especificações do motor e as condições de operação previstas estão apresentadas na TABELA 1.

TABELA 1: GEOMETRIA E CONDIÇÕES DE TRABALHO DO MOTOR

Comprimento do curso do pistão de potência	42,90 mm
Comprimento do curso do deslocador	48,30 mm
Diâmetro do pistão de potência	51,00 mm
Diâmetro do pistão deslocador	49,00 mm
Volume percorrido pelo pistão de potência	91,12 cm ³
Volume percorrido pelo deslocador (face quente)	91,08 cm ³
Volume percorrido pelo deslocador (face fria)	91,08 cm ³
Volume do regenerador	83,80 cm ³
Volume morto de expansão	2,12 cm ³
Volume morto de compressão	0,40 cm ³
Ângulo de fase entre os volumes	90 °
Temperatura de expansão	600 K
Temperatura de compressão	300 K

FONTE: O autor (2019)

3.5 SIMULAÇÃO DO MODELO ISOTÉRMICO

No modelo isotérmico de Martini (1983), o autor utiliza o que ele chama *inventário de gás* (mR) do sistema como dado de entrada, onde m é o número de mols do sistema e R é a constante universal dos gases. É uma medida para a quantidade de gás contido no sistema fechado.

3.5.1 Cálculo do inventário de gás do protótipo

O número de mols do sistema foi calculado com base no volume máximo possível, ou seja, quando o pistão de potência está em seu ponto morto inferior. Neste

volume, à temperatura de 25°C e pressão atmosférica de 1 atm e considerando o ar como gás ideal, a lei dos gases ideais, equação (8), dá a medida do inventário de gás do protótipo em Joules por Kelvin.

$$mR = \frac{PV}{T} \quad (8)$$

3.5.2 Entrada de dados e execução do algoritmo

Utilizando os dados de volume e condições de trabalho (TABELA 1), as equações (9) a (12) foram calculadas em uma rotina em Matlab® elaborada pelo autor, baseada no modelo de Martini (1983), para os ângulos de manivela correspondentes a um ciclo completo, ou seja, de 0° a 360°. Os resultados para a pressão e para o volume em cada ângulo foram plotados e a curva fechada resultante pode ser integrada, chegando ao resultado do trabalho líquido (13). O algoritmo pode ser encontrado no APÊNDICE B.

$$V_e(\theta) = \frac{V_{se}}{2}(1 + \cos \theta) + V_{de} \quad (9)$$

$$V_c(\theta) = \frac{V_{sc}}{2}(1 + \cos \theta) + V_{dc} + V_{sp}[1 - \cos(\theta - \alpha)] \quad (10)$$

$$V_t(\theta) = V_e + V_r + V_c \quad (11)$$

onde:

V_e = volume de expansão

V_c = volume de compressão

V_r = volume do regenerador

V_t = volume total

V_{se} = volume percorrido pelo regenerador (face quente)

V_{sc} = volume percorrido pelo regenerador (face fria)

V_{sp} = volume percorrido pelo pistão de potência

V_{de} = volume morto de expansão

V_{dc} = volume morto de compressão

θ = ângulo de manivela

α = ângulo de fase entre os volumes

$$P(\theta) = \frac{mR}{\frac{V_e(\theta)}{T_e} + \frac{V_r}{T_r} + \frac{V_c(\theta)}{T_c}} \quad (12)$$

$$W_{líq} = \oint P(\theta) dV_t(\theta) \quad (13)$$

3.6 SIMULAÇÃO DO MODELO ADIABÁTICO

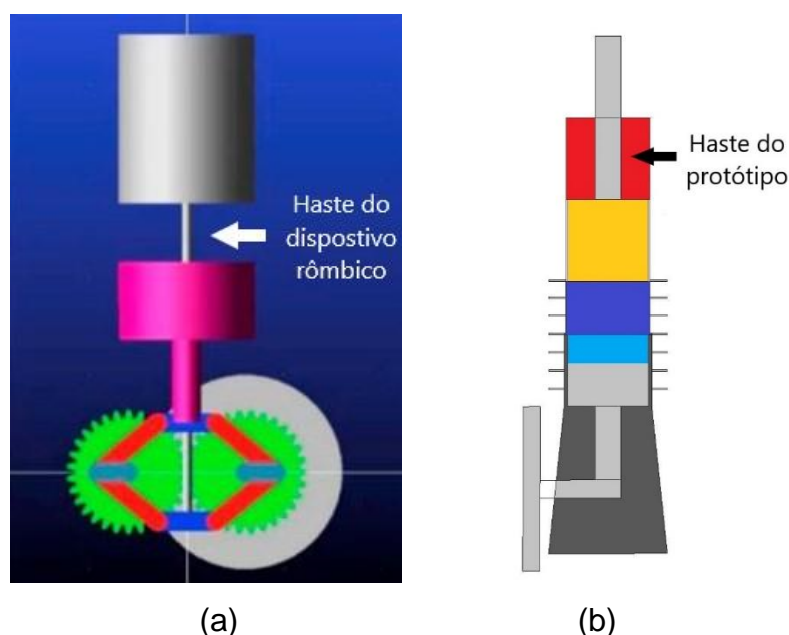
Para a simulação do modelo adiabático foi utilizado o *software* SEA (Stirling Engine Analysis) escrito em Matlab®, disponível no *site* da Universidade de Ohio, Estados Unidos (URIELI, 2018), cujo desenvolvimento foi realizado pelo engenheiro mecânico e professor desta universidade, Israel Urieli. Diferentemente do método utilizado na modelagem isotérmica, onde o inventário de gás do sistema (mR) foi utilizado como um dado de entrada, o SEA utiliza a pressão média do motor. A análise da lei dos gases ideais (8) justifica esta substituição, já que uma grandeza é diretamente proporcional à outra.

Desta forma, foi utilizado o resultado da simulação isotérmica para a P_{med} como um dado de entrada, além dos dados geométricos e das condições de trabalho do motor (TABELA 1).

O SEA contempla a simulação para o tipo Beta do motor Stirling, mas o mecanismo de acionamento considerado é o rômbo. Diante disto, algumas adaptações foram necessárias para uma melhor representatividade da modelagem do protótipo deste trabalho.

No módulo *engine* do programa, onde são especificados os parâmetros geométricos do motor, o cálculo para o volume de compressão do tipo Beta considera o volume da haste que atravessa o espaço de compressão (FIGURA 17-a), ou seja, subtrai o volume da haste. No protótipo em estudo, a haste em questão atravessa o volume de expansão (FIGURA 17-b). As alterações no algoritmo estão indicadas no ANEXO II deste documento, na página 58.

FIGURA 17: POSIÇÃO DAS HASTES NOS ACIONAMENTOS RÔMBICO E DO PROTÓTIPO



FONTE: (a) EROL et al. (2017) e (b) o autor (2019)

3.7 CÁLCULO DA POTÊNCIA

A potências teóricas L_t do motor, equação (14), foram determinadas, para as duas modelagens em estudo, nas rotações 90, 120, 150, 180, 430 e 860 rpm.

$$L_t = W_{liq} \times \frac{rpm}{60} \quad (14)$$

3.8 ANÁLISE EXPERIMENTAL

Inicialmente a parte externa da câmara de expansão do protótipo foi submetida ao aquecimento com maçarico a gás até o metal alcançar a incandescência. A temperatura estimada do aço SAE 1010, utilizado atualmente em amortecedores (PAULA, 2014), ultrapassou os 900 K (CHIAVERINI, 1979). Em seguida, foram realizadas, sem sucesso, tentativas de partida manual através do giro da polia do protótipo.

Como a temperatura de expansão utilizada na simulação foi de 600 K, valor inferior à do ensaio, e o motor não partiu, optou-se por não realizar a medição da temperatura exata alcançada. Desta forma, a análise experimental deste trabalho passou a focar na busca das razões pelas quais o motor não partiu.

3.8.1 Diminuição de carga

O protótipo foi construído com materiais metálicos, cujo peso impacta negativamente no seu funcionamento, pois o trabalho de expansão e compressão do gás sobre o pistão de potência deve ser capaz de movimentar a massa destes materiais e do próprio pistão. Outro ponto negativo para o desempenho do motor são as forças de atrito entre a haste e o retentor de pressão e nas conexões do mecanismo de acionamento. Desta forma, algumas medidas foram tomadas no intuito de diminuir a carga do motor, como a usinagem interna da haste metálica em torno mecânico e das partes móveis do mecanismo de acionamento, bem como o balanceamento deste mecanismo. Também foi realizada a lubrificação dos pontos de atrito mecânico.

3.8.2 Estimativa da potência necessária para o funcionamento do protótipo.

Uma análise foi realizada no intuito de estimar a potência que o trabalho líquido realizado pelo gás deve exercer sobre o pistão de potência para que o protótipo funcione como uma máquina térmica.

Em uma bancada, a polia do protótipo foi conectada ao mecanismo de rotação de um motor elétrico de indução através de uma correia tipo V. Um inversor de frequência foi utilizado para controlar a rotação do motor elétrico através de um *dimmer*. O motor elétrico é trifásico (220 V), potência nominal de 120 W e fator de potência de 0,46. O modelo do inversor é o VF-S7e da marca Toshiba, trifásico (220) e 400 W de potência nominal.

Com o motor elétrico acoplado ao protótipo, ajustou-se a frequência elétrica do motor de indução para que a rotação do protótipo alcançasse 90, 120, 150 e 180 rpm. Para cada rotação, foram medidos tensão e corrente através de multímetros para o cálculo da potência elétrica necessária para manter o protótipo funcionando naquela rotação através da equação (15), onde $V_{2\phi}$ é a tensão medida entre duas fases do motor de indução, i é a corrente elétrica de uma das fases e f_p é o fator de potência do motor elétrico.

$$P_{3\phi} = \sqrt{3} \cdot V_{2\phi} \cdot i \cdot f_p \quad (15)$$

4 RESULTADOS E DISCUSSÕES

Esta seção apresenta os resultados obtidos para as modelagens propostas e para a análise experimental. Inicia-se com o inventário de gás do sistema (4.1), um dado de entrada para o modelo isotérmico. Em seguida tem-se os valores obtidos para as análises isotérmica (4.2) e adiabática (4.3). Finalmente, o resultado do ensaio do protótipo em bancada.

4.1 INVENTÁRIO DE GÁS E PRESSÃO DO SISTEMA

O inventário de gás mR do sistema foi calculado considerando o volume máximo possível do protótipo de 242 cm³, à temperatura de 25°C e pressão atmosférica de 1 bar alcançando 0,08470 J/K.

4.2 RESULTADOS DA ANÁLISE ISOTÉRMICA

Os resultados da análise isotérmica está dividida na discussão sobre a eficiência do modelo (4.2.1) e na execução do algoritmo com os dados de entrada devidamente implementados gerando resultados de pressão (4.2.2) e desempenho que foram avaliados segundo as rotações do motor previstas na metodologia (4.2.3). Também foi analisada a variação da potência em relação ao aumento da temperatura de expansão para este modelo (4.2.4).

4.2.1 Redução da eficiência

A sobreposição dos diagramas PxV do ciclo ideal e do modelo isotérmico de Schmidt apresentado na FIGURA 4 revelam a diminuição do trabalho líquido do segundo em relação ao primeiro. Entretanto, na literatura, a eficiência indicada para a teoria de Schmidt é a mesma de Carnot, assim como o ciclo ideal (HIRATA, 1997; URIELI; BERCHOWITZ, 1984; FORMOSA; DESPESE, 2010). Isto ocorre porque na análise de Schmidt, não é possível determinar os fluxos de calor durante o ciclo, já que a compressão e a expansão são consideradas isotérmicas e os trocadores, ideais (GARCÍA et al., 2018).

Ranieri et al. (2018), em uma análise isotérmica, determinou os trabalhos de compressão e expansão para a operação senoidal dos pistões, considerando o fluxo de calor entre os trocadores e o gás e considerando as quantidades parciais do fluido presentes em cada câmara ao longo do ciclo. O autor concluiu que, para o tipo Beta, a eficiência sofre uma redução de 25,7 % em relação à de Carnot e afirma que a eficiência dos motores Stirling com movimento senoidal devem ser comparadas ao ciclo senoidal, ao invés da eficiência de Carnot.

4.2.2 Pressão média

A pressão média obtida na simulação foi de 1,9 bar. Este valor está relacionado ao inventário de gás do sistema que fora calculado para a condição de 25°C e 1 atm. Esta grandeza é diretamente proporcional à potência do motor Stirling segundo a equação (16) (BEALE¹, 1969 apud HACHEM et al., 2018), onde W_m é a potência, K é o fator de correção de potência, P é a pressão média e V_{sc} é o volume percorrido de compressão.

$$W_m = K \cdot P \cdot f \cdot V_{sc} \quad (16)$$

A potência de um motor Stirling pode ser controlada pela regulação da pressão através de um compressor externo. A redução da pressão implica na redução de massa do fluido de trabalho e, conseqüentemente, na diminuição da potência. Analogamente, o aumento da pressão fornece maior quantidade de gás ao sistema, aumentando também a potência do motor (JASPERS, 1975).

4.2.3 Variação da potência com o aumento da rotação do motor

ATABELA 2 apresenta os resultados obtidos na análise isotérmica, onde a potência L_t aumenta com o aumento da rotação do motor. Como verificado o valor para a eficiência ficou em 50%, portando idêntica à eficiência de Carnot.

¹ Beale, W., **Free Piston Stirling Engines - Some Model Tests and Simulations**. SAE Technical Paper 690230, 1969

TABELA 2: RESULTADOS PARA A ANÁLISE ISOTÉRMICA

Ensaio	rpm	L_t (W)
1	90	6,6
2	120	8,8
3	150	10,9
4	180	13,1
5	430	31,3
6	860	62,7

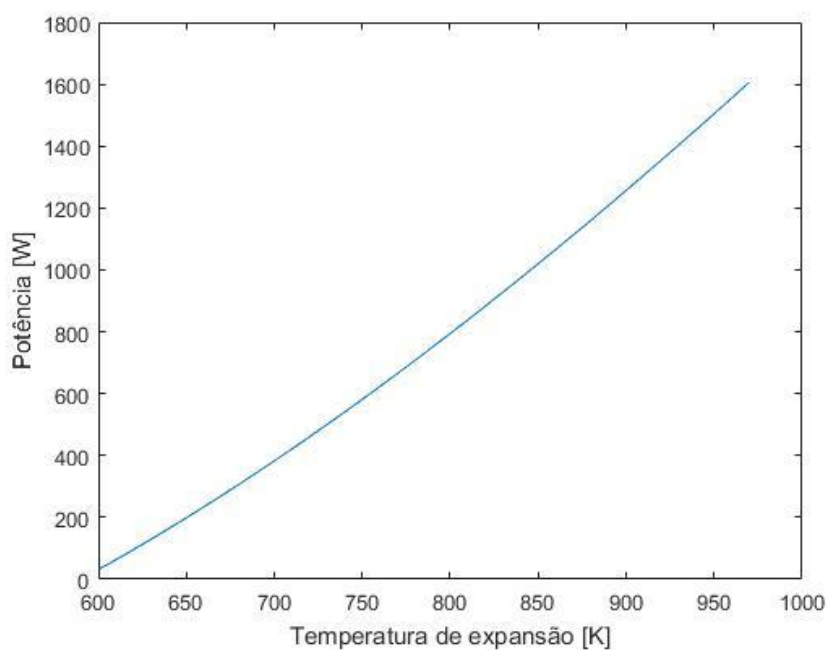
FONTE: O autor (2019)

LEGENDA: Variação da potência com o incremento da rotação entre as temperaturas 300 K e 600 K.

4.2.4 Variação da potência com o aumento da temperatura de expansão

Uma análise paramétrica foi realizada apresentando a variação da potência com o aumento da temperatura de expansão (FIGURA 18), mantendo a temperatura de compressão constante a 300 K e a rotação de 430 rpm. A análise mostra a importância do gradiente de temperatura entre a fonte fria e a fonte quente do motor Stirling.

FIGURA 18: VARIAÇÃO DA POTÊNCIA PELO AUMENTO DA TEMPERATURA DE EXPANSÃO



FONTE: O autor (2019)

4.3 RESULTADOS DA ANÁLISE ADIABÁTICA

A pressão média operacional de 1,9 bar, encontrada na simulação isotérmica foi utilizada na adiabática, como mencionado. A análise da variação da potência em relação às rotações do motor também foi realizada (4.3.1). Por fim, a variação da temperatura do fluido nas câmaras de expansão e compressão foi analisada e discutida (4.3.2).

4.3.1 Variação da potência com o aumento da rotação do motor

Na análise do incremento da rotação para o modelo adiabático percebe-se uma diminuição na potência alcançada (TABELA 3) em relação ao isotérmico (TABELA 2). A eficiência alcançada nesta análise foi de 39,1%.

TABELA 3: RESULTADOS PARA A ANÁLISE ADIABÁTICA

Ensaio	rpm	L_t
1	90	6.4
2	120	8.6
3	150	10,7
4	180	12,9
5	430	30.8
6	860	61,4

FONTE: O autor (2019)

LEGENDA: Variação da potência com o incremento da rotação entre as temperaturas 300 K e 600 K.

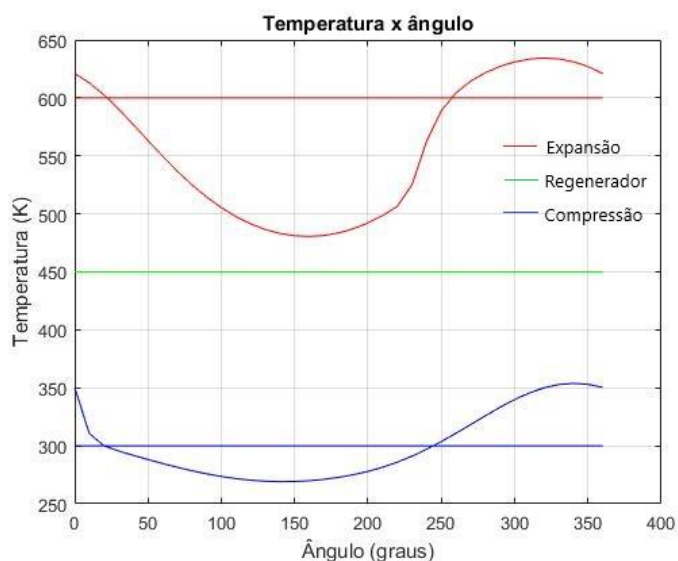
4.3.2 Variação das temperaturas de expansão e compressão

No modelo adiabático, como visto, as temperaturas do fluido nas câmaras de expansão e compressão não são constantes como no isotérmico. O *software* utilizado gera o gráfico (FIGURA 19) representativo para estas variações. Nota-se as temperaturas constantes dos trocadores de calor (300 e 600 K) no regenerador (450 K).

As curvas variando em torno das temperaturas dos trocadores quente e frio representam a variação da temperatura do fluido nas câmaras de expansão e

compressão, respectivamente. Para a variação do fluido na câmara de expansão, por exemplo, uma amplitude em torno de 160 K é observada. Isso implica numa redução da temperatura média de expansão que será refletida no desempenho térmico da máquina, o qual não mais alcançará a eficiência de Carnot (GARCÍA et al., 2018).

FIGURA 19: TEMPERATURA DO FLUIDO NOS COMPONENTES DO MOTOR

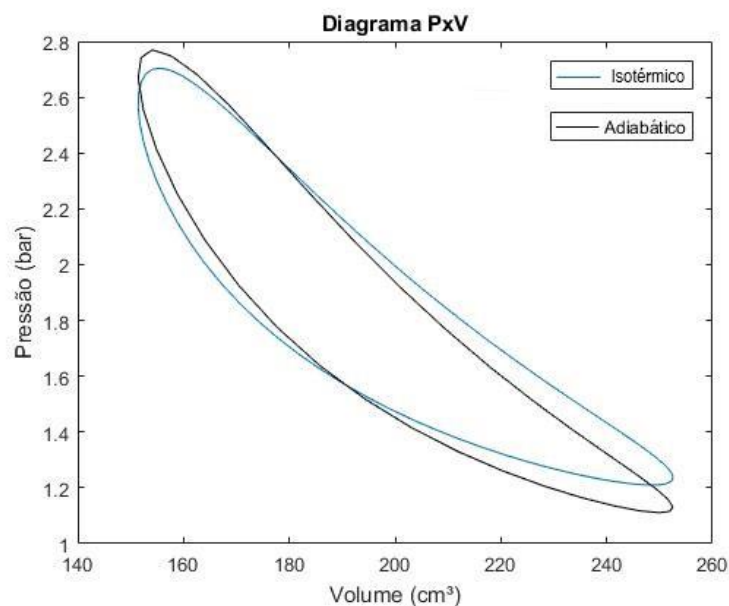


FONTE: O autor (2019)

4.4 COMPARAÇÃO DOS DIAGRAMAS PRESSÃO POR VOLUME

A FIGURA 20 compara os diagramas P_xV das simulações isotérmica e adiabática. Observa-se uma leve diminuição na espessura da figura formada pela curva fechada do diagrama adiabático em relação ao isotérmico, incidindo assim na diminuição da área do primeiro em relação ao segundo e, conseqüentemente, na diminuição do trabalho estimado pela modelagem adiabática.

FIGURA 20: DIAGRAMAS PxV ISOTÉRMICO E ADIABÁTICO



FONTE: O autor (2019)

4.5 RESULTADOS DA ANÁLISE EXPERIMENTAL

Para a análise experimental, o protótipo foi acionado à temperatura ambiente nas rotações 90, 120, 150 e 180 rpm, através do acoplamento a um motor elétrico de indução com o objetivo de mensurar a potência elétrica necessária para o seu funcionamento. Os dados para a tensão e a corrente foram utilizados para o cálculo das potências (TABELA 4). As rotações de 430 e 860 rpm, simuladas nas modelagens, não foram testadas experimentalmente em razão de limites operacionais impostos pela vibração do mecanismo de acionamento.

TABELA 4: RESULTADOS PARA A POTÊNCIA NECESSÁRIA PARA O ACIONAMENTO DO MOTOR

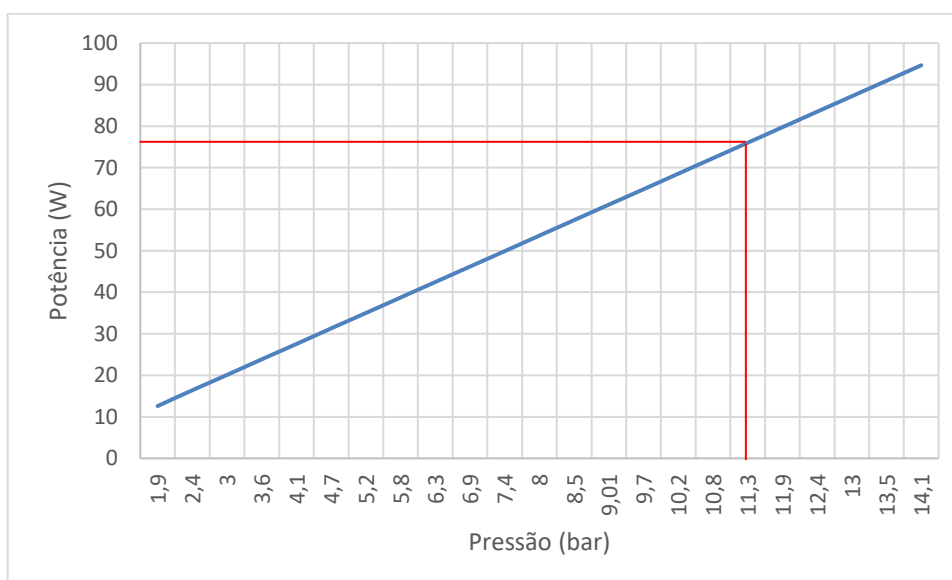
Ensaio	RPM	Tensão (V)	Corrente (A)	Potência (W)
1	90	103,9	0,48	39,7
2	120	126,4	0,51	51,4
3	150	154,2	0,56	68,8
4	180	171,4	0,55	75,1

FONTE: O autor (2019)

Os ensaios para a estimativa de potência mínima necessária para o funcionamento do motor demonstraram que o trabalho promovido pelo gás nas condições propostas é insuficiente para esta tarefa.

Numa análise no modelo adiabático verificou-se que a pressão média necessária para o motor gerar 75 W a 180 RPM é de 11 bar aproximadamente, (FIGURA 21), ou seja, 5,6 vezes maior do que a pressão considerada na modelagem e na análise experimental de 1,9 bar. Pressões médias de operação na faixa de 10 a 20 MPa são utilizadas para se obter altos níveis de potência (ERBAY; YAVUZ, 1997 apud THOMBARE; VERMA, 2008).

FIGURA 21: VARIAÇÃO DA POTÊNCIA PELO AUMENTO DE PRESSÃO



FONTE: O autor (2019)

LEGENDA: Análise adiabática da variação da potência com o incremento da pressão média entre as temperaturas 300 K e 600 K.

5 CONSIDERAÇÕES FINAIS

O protótipo de um motor Stirling de configuração Beta foi construído no intuito de se adquirir resultados experimentais para compará-los a resultados teóricos obtidos através da avaliação das modelagens isotérmica e adiabática. A primeira foi baseada na teoria de Schmidt e a segunda na análise de Finkelstein.

A construção do protótipo está dividida na escolha e na preparação dos materiais e nas montagens do mecanismo de acionamento e dos elementos que configuram o tipo Beta do motor Stirling.

O estudo verificou a importância e a dificuldade em conter o gás comprimido no cilindro, bem como a necessidade da construção de mecanismos leves, diminuindo seu impacto na carga do motor.

Apesar da eficiência atribuída aos motores Stirling, pela teoria de Schmidt, ser comparada à eficiência de Carnot, estudos demonstram que o movimento senoidal dos pistões, admitida na teoria, reduz a eficiência do motor, pois permite a localização parcial de gás nos espaços de compressão e expansão ao longo do ciclo. Desta forma, o fornecimento e rejeição de calor do gás deixa de ser integral, como no ciclo ideal, causando a redução da eficiência.

A variação das temperaturas nos espaços de expansão e compressão proposta no modelo adiabático demonstrou a menor eficiência alcançada deste em relação ao isotérmico, pela diminuição do gradiente de temperatura.

Para estudos futuros, recomenda-se a construção de um mecanismo de acionamento mais leve, diminuindo a carga do motor.

REFERÊNCIAS

- BATAINEH, K. M. **Numerical thermodynamic model of alpha-type Stirling engine. Case Studies in Thermal Engineering**, v. 12, p. 104–116, 2018.
- BRUNETTI, F. **Motores de Combustão Interna**. Blucher, 2016.
- CAETANO, B. C.; LARA, I. F.; BORGES, M. U.; SANDOVAL, O. R.; VALLE, R. M. **A novel methodology on beta-type Stirling engine simulation using CFD**. *Energy Conversion and Management*, v. 184, p. 510–520, 2019.
- ÇENGEL, Y. A.; BOLES, M. A. **Termodinâmica**. 7o ed. 2013.
- CHIAVERINI, V. **Aços e Ferros Fundidos**. Associação Brasileira de Metais, 1979.
- CINAR, C.; YUCESU, S.; TOPGUL, T.; OKUR, M. **Beta-type Stirling engine operating at atmospheric pressure**. *Applied Energy*, v. 81, n. 4, p. 351–357, 2005.
- ERBAY, L. B.; YAVUZ, H. **Analysis of the stirling heat engine at maximum power conditions**. *Energy*, v. 22, n. 7, p. 645–650, 1997.
- EROL, D.; YAMAN, H.; DOĞAN, B. **A review development of rhombic drive mechanism used in the Stirling engines**. *Renewable and Sustainable Energy Reviews*, v. 78, p. 1044–1067, 2017.
- FINKELSTEIN, T. **Generalized Thermodynamic Analysis Of Stirling Engines**. Warrendale, PA: SAE International, 1960.
- GARCÍA, M. T.; CARVAJAL TRUJILLO, E.; VÉLEZ GODIÑO, J.; SÁNCHEZ MARTÍNEZ, D. **Thermodynamic Model for Performance Analysis of a Stirling Engine Prototype**. *Energies*, v. 11, n. 10, p. 2655, 2018.
- HACHEM, H.; GHEITH, R.; ALOUI, F.; BEN NASRALLAH, S. **Technological challenges and optimization efforts of the Stirling machine: A review**. *Energy Conversion and Management*, v. 171, p. 1365–1387, 2018.
- HIRATA, K. **SCHMIDT Theory For Stirling Engines**. Disponível em: <<http://www.bekkoame.ne.jp/~khirata/academic/schmidt/schmidt.htm>>. Acesso em: 3/7/2018.
- IPCC. **Global Warming of 15°C**. Disponível em: <https://www.ipcc.ch/site/assets/uploads/sites/2/2019/05/SR15_SPM_version_report_LR.pdf>. Acesso em: 22/6/2019.
- JASPERS, H. **Power Control System for Stirling Engines**. 1975. Disponível em: <<https://patentimages.storage.googleapis.com/00/18/b1/6c37b1ddcb265a/US3886744.pdf>>. Acesso em: 8/7/2019.

LEITE, V. F. **Construção De Um Motor Stirling Tipo Beta Com Acionamento Rômbico.** , 2018.

MARTINI, W. R. **Stirling engine design manual.** 1983.

PAULA, E. B. D. **Estudo sobre a aplicação de aços dua phase na indústria automotiva com o objetivo de redução de massa em amortecedores estruturais.**, p. 102, 2014.

RANIERI, S.; PRADO, G.; MACDONALD, B. D. **Efficiency Reduction in Stirling Engines Resulting from Sinusoidal Motion.** , 2018.

SANT'ANNA, B. H. D.; YANO, D. H. S. **Projeto E Simulação De Um Motor Stirling.** , p. 127, 2012.

SCHMIDT G. **Classical analysis of operation of Stirling engine.** A report published in German Engineering Union (Original German), vol. XV; 1871. p. 1–12.

SHENDAGE, D. J.; KEDARE, S. B.; BAPAT, S. L. **An analysis of beta type Stirling engine with rhombic drive mechanism.** Renewable Energy, v. 36, n. 1, p. 289–297, 2011.

SRIPAKAGORN, A.; SRIKAM, C. **Design and performance of a moderate temperature difference Stirling engine.** Renewable Energy, v. 36, n. 6, p. 1728–1733, 2011.

THOMBARE, D. G.; VERMA, S. K. **Technological development in the Stirling cycle engines.** Renewable and Sustainable Energy Reviews, v. 12, n. 1, p. 1–38, 2008.

TIMOUMI, Y.; TLILI, I.; BEN NASRALLAH, S. **Performance optimization of Stirling engines.** Renewable Energy, v. 33, n. 9, p. 2134–2144, 2008.

URIELI, I. **Stirling Cycle Machine Analysis.** Disponível em: <<https://www.ohio.edu/mechanical/stirling/isothermal/Schmidt.html>>. Acesso em: 28/5/2018.

URIELI, I.; BERCHOWITZ, D. M. **Stirling cycle engine analysis.** Bristol: Hilger, 1984.

ANEXO A

Equações referentes à análise adiabática (URIELI, 2018).

$$P = \frac{mR}{\left(\frac{V_c}{T_c} + \frac{V_k}{T_k} + \frac{V_r}{T_r} + \frac{V_h}{T_h} + \frac{V_e}{T_e}\right)} \quad (17)$$

$$dp = \frac{-\gamma p \left(\frac{dV_c}{T_{CK}} + \frac{dV_e}{T_{HE}}\right)}{\left[\frac{V_c}{T_{CK}} + \gamma \left(\frac{V_k}{T_k} + \frac{V_r}{T_r} + \frac{V_h}{T_h}\right) + \frac{V_e}{T_{HE}}\right]} \quad (18)$$

$$mc = pV_c/(RT_c) \quad (19)$$

$$mk = pV_k/(RT_k) \quad (20)$$

$$mr = pV_r/(RT_k) \quad (21)$$

$$mh = pV_h/(RT_h) \quad (22)$$

$$me = pV_e/(RT_e) \quad (23)$$

$$dmc = (p dV_c + V_c dp/\gamma) / R T_{ck} \quad (24)$$

$$dme = (p dV_e + V_e dp/\gamma) / R T_{he} \quad (25)$$

$$dmk = mk dp/p \quad (26)$$

$$dmr = mr dp/p \quad (27)$$

$$dmh = mh dp/p \quad (28)$$

$$mck' = -dmc \quad (29)$$

$$mkr' = mck' - dmk \quad (30)$$

$$mhe' = dme \quad (31)$$

$$mrh' = mhe' - dmh \quad (32)$$

$$dT_c = T_c (dp/p + dV_c/V_c - dmc/mc) \quad (33)$$

$$dT_e = T_e (dp/p + dV_e/V_e - dme/me) \quad (34)$$

$$dQ_k = V_k dp cv/R - cp(T_{ck} mck' - T_k mkr') \quad (35)$$

$$dQ_r = V_r dp \, cv/R - cp(T_k mkr' - T_h mhr') \quad (36)$$

$$dQ_h = V_h dp \, cv/R - cp(T_h mhr' - T_{he} mhe') \quad (37)$$

$$dW_c = p \, dV_c \quad (38)$$

$$dW_e = p \, dV_e \quad (39)$$

$$dW = dW_c + dW_e \quad (40)$$

$$W = W_c + W_e \quad (41)$$

ANEXO B

Módulos do software SEA (Stirling Engine Analisis) (URIELI, 2018)

Módulo 1 - sea

```

clc;
clear all;
%define a specific engine
define;
choice = 'x';
while(~strncmp(choice, 'q', 1))
    fprintf('Choose simulation:\n');
    choice = input('a)diabatic, s)imple q)uit: ', 's');
    if(strncmp(choice, 'a', 1))
        [var, dvar] = adiabatic;
    elseif(strncmp(choice, 's', 1))
        [var, dvar] = simple;
    end
end
fprintf('quitting simulation...\n');

```

Módulo 2 – define

```

function define
% define the stirling engine geometric
% and operational parameters
% Israel Urieli 4/1/02 (April Fool's Day)
% Modified 2/12/2010 to include no-matrix regenerator awgr0
% Modified 7/10/2016 to include betadrive engine type
clc;
clear all;
% The set of global variables defined are:
% engine
global engine_type % s)inusoidal, y)oke r)ockerV, b)etadrive
global vclc vcle % compression, expansion clearance vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global b1 % Ross yoke length (1/2 yoke base) [m]
global b2 % Ross yoke height [m]
global crank % crank radius [m]
global dcomp dexp % diameter of compression/expansion pistons [m]
global acomp aexp % area of compression/expansion pistons [m^2]
global ymin % minimum yoke vertical displacement [m]
global conrodc conrode % length of comp/exp piston connecting rods [m]
global ycmay yemay % maximum comp/exp piston vertical displacement [m]
% heatex/cooler
global vk % cooler void volume [m^3]
global ak % cooler internal free flow area [m^2]
global awgk % cooler internal wetted area [m^2]
global dk % cooler hydraulic diameter [m]
global lk % cooler effective length [m]
% heatex/heater
global vh % heater void volume [m^3]
global ah % heater internal free flow area [m^2]

```

```

global awgh % heater internal wetted area [m^2]
global dh % heater hydraulic diameter [m]
global lh % heater effective length [m]
% heatex/regenerator
global lr % regenerator effective length [m]
global cqwr % regenerator housing thermal conductance [W/K]
global matrix_type % m)esh f)oil n)o matrix
global vr % regen void volume [m^3]
global ar % regen internal free flow area [m^2]
global awgr0 % no matrix regenerator wetted area [m^2]
global awgr % regen internal wetted area [m^2]
global dr % regen hydraulic diameter [m]
% gas
global rgas % gas constant [J/kg.K]
global cp % specific heat capacity at constant pressure [J/kg.K]
global cv % specific heat capacity at constant volume [J/kg.K]
global gama % ratio: cp/cv
global mu0 % dynamic viscosity at reference temp t0 [kg.m/s]
global t0 t_suth % reference temp. [K], Sutherland constant [K]
global prandtl % Prandtl number
% operat
global pmean % mean (charge) pressure [Pa]
global tk tr th % cooler, regenerator, heater temperatures [K]
global freq omega % cycle frequency [herz], [rads/s]
global mgas % total mass of gas in engine [kg]
% new data file
global new fid

new = input('Create a new data file? (y/n)','s');
if strcmp(new,'y',1)
    filename = input('enter new filename: ','s');
    fid = fopen(filename,'w');
else
    fid = 0;
    while fid < 1
        filename = input('open filename: ','s');
        [fid, message] = fopen(filename,'r');
        if fid == -1
            display(message)
            display('press ^C to exit')
        end
    end
end
engine
heatex
gas
operat
status = fclose(fid);

```

Módulo 3 – engine

```

function engine
% Define engine configuration and drive geometric parameters.
% Israel Urieli 4/14/02
% Modified 6/14/2016 to include Beta free piston engine

global engine_type % s)inusoidal, y)oke r)ocker-V, b)etadrive
global new fid % new data file

```

```

engine_type = 'u';
while(strncmp(engine_type, 'u', 1))
    if(strncmp(new, 'y', 1))
        fprintf('Available engine types are:\n');
        fprintf('  s)inusoidal drive\n');
        fprintf('  y)oke drive (Ross)\n');
        fprintf('  r)ocker-V drive (Ross)\n');
        fprintf('  b)eta drive (Free piston)\n');
        engine_type = input('enter engine type ', 's');
        fprintf(fid, '%c\n', engine_type(1));
    else
        engine_type = fscanf(fid, '%c', 1);
    end
if(strncmp(engine_type, 's', 1))
    sindrive;
elseif(strncmp(engine_type, 'y', 1))
    yokedrive;
elseif(strncmp(engine_type, 'r', 1))
    rockerVdrive;
elseif(strncmp(engine_type, 'b', 1))
    betadrive;
else
    fprintf('engine type is undefined\n')
    engine_type = 'u';
end
end
%=====
function sindrive
% Sinusoidal drive engine configuration
% Israel Urieli 4/14/02

global vclc vcle % compression, expansion clearence vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global new fid % new data file

fprintf('sinusoidal drive engine configuration\n')
if(strncmp(new, 'y', 1))
    vclc = input('enter compression space clearence volume [m^3]: ');
    vswc = input('enter compression space swept volume [m^3]: ');
    vcle = input('enter expansion space clearence volume [m^3]: ');
    vswe = input('enter expansion space swept volume [m^3]: ');
    phase = input('enter expansion phase angle advance [degrees]: ');
    fprintf(fid, '%.3e\n', vclc);
    fprintf(fid, '%.3e\n', vswc);
    fprintf(fid, '%.3e\n', vcle);
    fprintf(fid, '%.3e\n', vswe);
    fprintf(fid, '%.1f\n', phase);
else
    vclc = fscanf(fid, '%e', 1);
    vswc = fscanf(fid, '%e', 1);
    vcle = fscanf(fid, '%e', 1);
    vswe = fscanf(fid, '%e', 1);
    phase = fscanf(fid, '%f', 1);
end
fprintf('\nsinusoidal drive engine data summary:\n');
fprintf(' comp clearence, swept vols %.1f, %.1f [cm^3]\n',
vclc*1e6, vswc*1e6);
fprintf(' exp clearence, swept vols %.1f, %.1f [cm^3]\n',
vcle*1e6, vswe*1e6);

```

```

fprintf(' expansion phase angle advance %.1f[degrees]\n', phase);
alpha = phase * pi/180;
%=====
function yokedrive
% Ross yoke drive engine configuration
% Israel Urieli 4/14/02

global vclc vcle % compression,expansion clearence vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global b1 % Ross yoke length (1/2 yoke base) [m]
global b2 % Ross yoke height [m]
global crank % crank radius [m]
global dcomp dexp % diameter of compression/expansion pistons [m]
global acomp aexp % area of compression/expansion pistons [m^2]
global ymin % minimum yoke vertical displacement [m]
global new fid % new data file

fprintf('Ross yoke drive engine configuration\n');
if(strncmp(new,'y',1))
    vclc = input('enter compression space clearence volume [m^3]: ');
    vcle = input('enter expansion space clearence volume [m^3]: ');

    b1 = input('enter Ross yoke length b1 (1/2 yoke base) [m]: ');
    b2 = input('enter Ross yoke height b2 [m]: ');
    crank = input('enter crank radius [m]: ');

    dcomp = input('enter compression piston diameter [m]: ');
    dexp = input('enter expansion piston diameter [m]: ');

    fprintf(fid, '%.3e\n', vclc);
    fprintf(fid, '%.3e\n', vcle);
    fprintf(fid, '%.3e\n', b1);
    fprintf(fid, '%.3e\n', b2);
    fprintf(fid, '%.3e\n', crank);
    fprintf(fid, '%.3e\n', dcomp);
    fprintf(fid, '%.3e\n', dexp);
else
    vclc = fscanf(fid,'%e',1);
    vcle = fscanf(fid,'%e',1);
    b1 = fscanf(fid,'%e',1);
    b2 = fscanf(fid,'%e',1);
    crank = fscanf(fid, '%e',1);
    dcomp = fscanf(fid, '%e',1);
    dexp = fscanf(fid, '%e',1);
end
acomp = pi*dcomp^2/4.0;
aexp = pi*dexp^2/4.0;
yoke = sqrt(b1^2 + b2^2);
ymax = sqrt((yoke + crank)^2 - b2^2);
ymin = sqrt((yoke - crank)^2 - b2^2);

vswc = acomp*(ymax - ymin);
vswe = aexp*(ymax - ymin);
thmaxe = asin(ymax/(yoke + crank));
thmaxc = pi - thmaxe;
thmine = pi + asin(ymin/(yoke - crank));
thminc = 3*pi - thmine;
alpha = 0.5*(thmaxc - thmaxe) + 0.5*(thminc - thmine);
phase = alpha*180/pi;

```

```

fprintf('\nRoss yoke drive engine data summary:\n');
fprintf(' yoke length b1 (1/2 yoke base) %.1f [mm]\n', b1*1e3);
fprintf(' yoke height b2 %.1f [mm]\n', b2*1e3);
fprintf(' crank radius %.1f [mm]\n', crank*1e3);
fprintf(' compression piston diameter %.1f [mm]\n', dcomp*1e3);
fprintf(' expansion piston diameter %.1f [mm]\n', dexp*1e3);
fprintf(' comp clearance,swept vols %.1f, %.1f [cm^3]\n',
vclc*1e6,vswc*1e6);
fprintf(' exp clearance,swept vols %.1f, %.1f [cm^3]\n',
vcle*1e6,vswe*1e6);
fprintf(' ymin = %.1f(cm), ymax = %.1f(cm)\n',ymin*1e2,ymax*1e2)
fprintf(' alpha = %.1f(degrees)\n',phase);
%=====
function rockerVdrive
% Ross rocker-V drive engine configuration
% Israel Urieli 4/14/02 & Martine Long 2/14/05

global vclc vcle % compression,expansion clearance vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global crank % crank radius [m]
global dcomp dexp % diameter of compression/expansion pistons [m]
global acomp aexp % area of compression/expansion pistons [m^2]
global conrodc conrode % length of comp/exp piston connecting rods [m]
global ycmx yemx % maximum comp/exp piston vertical displacement [m]
global new fid % new data file

fprintf('Ross rocker-V drive engine configuration\n');
if(strncmp(new,'y',1))
    vclc = input('enter compression space clearance volume [m^3]: ');
    vcle = input('enter expansion space clearance volume [m^3]: ');
    crank = input('enter crank radius [m]: ');
    conrodc = input('enter compression piston connecting rod length [m]:
');
    conrode = input('enter expansion piston connecting rod length [m]: ');
    dcomp = input('enter compression piston diameter [m]: ');
    dexp = input('enter expansion piston diameter [m]: ');
    phase = input('enter expansion phase angle advance [degrees]: ');

    fprintf(fid, '%.3e\n', vclc);
    fprintf(fid, '%.3e\n', vcle);
    fprintf(fid, '%.3e\n', crank);
    fprintf(fid, '%.3e\n', conrodc);
    fprintf(fid, '%.3e\n', conrode);
    fprintf(fid, '%.3e\n', dcomp);
    fprintf(fid, '%.3e\n', dexp);
    fprintf(fid, '%.1f\n', phase);
else
    vclc = fscanf(fid,'%e',1);
    vcle = fscanf(fid,'%e',1);
    crank = fscanf(fid,'%e',1);
    conrodc = fscanf(fid,'%e',1);
    conrode = fscanf(fid,'%e',1);
    dcomp = fscanf(fid,'%e',1);
    dexp = fscanf(fid,'%e',1);
    phase = fscanf(fid,'%f',1);
end
acomp = pi*dcomp^2/4.0;
aexp = pi*dexp^2/4.0;
ycmx = conrodc + crank;

```

```

ymin = conrodc - crank;
ymax = conrode + crank;
ymin = conrode - crank;
vswc = acomp*(ymax - ymin);
vswe = aexp*(ymax - ymin);

fprintf('\nRoss rocker-V drive engine data summary:\n');
fprintf(' crank radius %.1f [mm]\n', crank*1e3);
fprintf(' compression piston connecting rod length %.1f [mm]\n',
conrodc*1e3);
fprintf(' expansion piston connecting rod length %.1f [mm]\n',
conrode*1e3);
fprintf(' compression piston diameter %.1f [mm]\n', dcomp*1e3);
fprintf(' expansion piston diameter %.1f [mm]\n', dexp*1e3);
fprintf(' comp clearance,swept vols %.1f, %.1f [cm^3]\n',
vclc*1e6,vswc*1e6);
fprintf(' exp clearance,swept vols %.1f, %.1f [cm^3]\n',
vcle*1e6,vswe*1e6);
fprintf(' COMPRESSION ymin = %.1f(cm), ymax =
%.1f(cm)\n',ymin*1e2,ymax*1e2)
fprintf(' EXPANSION ymin = %.1f(cm), ymax =
%.1f(cm)\n',ymin*1e2,ymax*1e2)
fprintf(' expansion phase angle advance %.1f[degrees]\n', phase);
alpha = phase * pi/180;
=====
function betadrive
% Free piston Beta drive engine configuration (assume sinusoidal motion)
% Israel Urieli 1/22/08

global vclc vcle % compression,expansion clearence vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global new fid % new data file

fprintf('beta drive engine configuration\n')
if(strncmp(new,'y',1))
    xpa = input('enter piston amplitude (m): ');
    xda = input('enter displacer amplitude (m): ');
    phid = input('enter displacer phase angle advance [degrees]: ');
    dp = input('enter piston diameter (m): ');
    dd = input('enter displacer diameter (m): ');
    dr = input('enter displacer rod diameter (m): ');
    vclc = input('enter compression space clearence volume [m^3]: ');
    vcle = input('enter expansion space clearence volume [m^3]: ');

    fprintf(fid, '%.3e\n', xpa);
    fprintf(fid, '%.3e\n', xda);
    fprintf(fid, '%.1f\n', phid);
    fprintf(fid, '%.3e\n', dp);
        fprintf(fid, '%.3e\n', dd);
        fprintf(fid, '%.3e\n', dr);
        fprintf(fid, '%.3e\n', vclc);
    fprintf(fid, '%.3e\n', vcle);
else
    xpa = fscanf(fid,'%e',1);
    xda = fscanf(fid,'%e',1);
    phid = fscanf(fid,'%f',1);
    dp = fscanf(fid,'%e',1);
        dd = fscanf(fid,'%e',1);
        dr = fscanf(fid,'%e',1);
        vclc = fscanf(fid,'%e',1);

```

```

    vcle = fscanf(fid, '%e', 1);
end

ap = pi*dp*dp/4; % piston area (m^2)
ad = pi*dd*dd/4; % displacer area (m^2)
ar = pi*dr*dr/4; % displacer rod area (m^2)
vpa = xpa*(ap - ar); % (piston - rod) volume amplitude (m^3)
vda = xda*(ad - ar); % (displacer - rod) volume amplitude (m^3)
vea = xda*(ad) % displacer volume amplitude (m^3)
phi = phid*pi/180; % radians

delta = atan2(vda*sin(phi), (vda*cos(phi) - vpa));
% compression space volume to piston amplitude phase advance
vca = sqrt(vpa*vpa - 2*vpa*vda*cos(phi) + vda*vda)
% compression space volume amplitude (m^3)

vswc = 144.61e-6; %2*vca; %compression space swept volume (m^3) (MODIFICADO)
vswe = 91.08e-6; %2*vea; % expansion space swept volume (m^3) (MODIFICADO)
alpha = pi + phi - delta; % expansion phase angle advance (radians)

fprintf('\nbeta drive engine data summary:\n');
fprintf(' comp clearance, swept vols %.1f, %.1f [cm^3]\n',
vclc*1e6, vswc*1e6);
fprintf(' exp clearance, swept vols %.1f, %.1f [cm^3]\n',
vcle*1e6, vswe*1e6);
fprintf(' expansion phase angle advance %.1f[degrees]\n\n', alpha*180/pi);
%=====

```

Módulo 4 – *heatex*

```

function heatex
% Specify heat exchanger geometric parameters
% Israel Urieli 3/31/02 (modified 12/01/03)
% Modified 2/14/2010 annulus and slots wetted area
cooler;
regen;
heater;
%=====
function cooler
% Specify cooler geometric parameters
% Israel Urieli 4/15/02

global vk % cooler void volume [m^3]
global ak % cooler internal free flow area [m^2]
global awgk % cooler internal wetted area [m^2]
global dk % cooler hydraulic diameter [m]
global lk % cooler effective length [m]
global new fid % new data file

cooler_type = 'u';
while (strcmp(cooler_type, 'u', 1))
    if (strcmp(new, 'y', 1))
        fprintf('Available cooler types are:\n')
        fprintf('  p, for smooth pipes\n')
        fprintf('  a, for smooth annulus\n')
        fprintf('  s, for slots\n')
        cooler_type = input('enter cooler type ', 's');
        fprintf(fid, '%c\n', cooler_type(1));
    end
end

```



```

else
    fscanf(fid, '%c',1); % bypass the previous newline character
    cooler_type = fscanf(fid, '%c',1);
end
if(strncmp(cooler_type,'p',1))
    [vk,ak,awgk,dk,lk] = pipes;
elseif(strncmp(cooler_type,'a',1))
    [vk,ak,awgk,dk,lk] = annulus;
elseif(strncmp(cooler_type,'s',1))
    [vk,ak,awgk,dk,lk] = slots;
else
    fprintf('cooler type is undefined\n')
    cooler_type = 'u';
end
end
fprintf('cooler data summary:\n');
fprintf(' void volume(cc) %.2f\n', vk*1e6)
fprintf(' free flow area (cm^2) %.2f\n', ak*1e4)
fprintf(' wetted area (cm^2) %.2f\n', awgk*1e4)
fprintf(' hydraulic diameter(mm) %.2f\n', dk*1e3)
fprintf(' cooler length (cm) %.2f\n', lk*1e2)

%=====
function heater
% Specify heater geometric parameters
% Israel Urieli 4/15/02

global vh % heater void volume [m^3]
global ah % heater internal free flow area [m^2]
global awgh % heater internal wetted area [m^2]
global dh % heater hydraulic diameter [m]
global lh % heater effective length [m]
global new fid % new data file

heater_type = 'u';
while(strncmp(heater_type,'u',1))
    if(strncmp(new,'y',1))
        fprintf('Available heater types are:\n')
        fprintf('  p, for smooth pipes\n')
        fprintf('  a, for smooth annulus\n')
        fprintf('  s, for slots\n')
        heater_type = input('enter heater type ','s');
        fprintf(fid, '%c\n', heater_type(1));
    else
        fscanf(fid, '%c',1); % bypass the previous newline character
        heater_type = fscanf(fid, '%c',1);
    end
    if(strncmp(heater_type,'p',1))
        [vh,ah,awgh,dh,lh] = pipes;
    elseif(strncmp(heater_type,'a',1))
        [vh,ah,awgh,dh,lh] = annulus;
    elseif(strncmp(heater_type,'s',1))
        [vh,ah,awgh,dh,lh] = slots;
    else
        fprintf('heater type is undefined\n')
        heater_type = 'u';
    end
end
end
fprintf('heater data summary:\n');
fprintf(' void volume(cc) %.2f\n', vh*1e6)
fprintf(' free flow area (cm^2) %.2f\n', ah*1e4)

```

```

fprintf(' wetted area (cm^2) %.2f\n', awgh*1e4)
fprintf(' hydraulic diameter(mm) %.2f\n', dh*1e3)
fprintf(' heater length (cm) %.2f\n', lh*1e2)

%=====
function [v,a,awg,d,len] = pipes
% homogeneous smooth pipes heat exchanger
% Israel Urieli 4/15/02
global new fid % new data file

fprintf('homogeneous bundle of smooth pipes\n')
if(strncmp(new,'y',1))
    d = input('enter pipe inside diameter [m] : ');
    len = input('enter heat exchanger length [m] : ');
    num = input('enter number of pipes in bundle : ');
    fprintf(fid, '%.3e\n', d);
    fprintf(fid, '%.3e\n', len);
    fprintf(fid, '%d\n', num);
else
    d = fscanf(fid,'%e',1);
    len = fscanf(fid,'%e',1);
    num = fscanf(fid,'%d',1);
end
a = num*pi*d*d/4;
v = a*len;
awg = num*pi*d*len;
%=====
function [v,a,awg,d,len] = annulus
% annular gap heat exchanger
% Israel Urieli 12/01/03
% Modified 2/14/2010 wetted area
global new fid % new data file

fprintf(' annular gap heat exchanger\n')
if(strncmp(new,'y',1))
    dout = input('enter annular gap outer diameter [m] : ');
    din = input('enter annular gap inner diameter [m] : ');
    len = input('enter heat exchanger length [m] : ');
    fprintf(fid, '%.3e\n', dout);
    fprintf(fid, '%.3e\n', din);
    fprintf(fid, '%.3e\n', len);
else
    dout = fscanf(fid,'%e',1);
    din = fscanf(fid,'%e',1);
    len = fscanf(fid,'%e',1);
end

a = pi*(dout*dout - din*din)/4;
v = a*len;
awg = pi*dout*len;
d = dout - din;
%=====
function [v,a,awg,d,len] = slots
% slots heat exchanger
% Israel Urieli 12/01/03
% Modified 2/14/2010 wetted area
global new fid % new data file

fprintf(' slots heat exchanger\n')
if(strncmp(new,'y',1))

```

```

w = input('enter width of slot [m] : ');
h = input('enter height of slot [m] : ');
len = input('enter heat exchanger length [m] : ');
num = input('enter number of slots : ');
fprintf(fid, '%.3e\n', w);
fprintf(fid, '%.3e\n', h);
fprintf(fid, '%.3e\n', len);
fprintf(fid, '%d\n', num);
else
w = fscanf(fid, '%e', 1);
h = fscanf(fid, '%e', 1);
len = fscanf(fid, '%e', 1);
num = fscanf(fid, '%d', 1);
end

a = num*w*h;
v = a*len;
awg = num*(w + 2*h)*len;
d = 4*v/awg;
%=====

```

Módulo 5 – regen

```

function regen
% Specifies regenerator geometric and thermal properties
% Israel Urieli 04/20/02 (modified 12/01/03)
% modified 2/12/2010 to include awgr0 (wetted area)
% modified 11/27/2010 to include 'no regenerator matrix'

global lr % regenerator effective length [m]
global awgr0 % no matrix regenerator wetted area [m^2]
global cqwr % regenerator housing thermal conductance [W/K]
global new fid % new data file

regen_type = 'u';
while(strncmp(regen_type, 'u', 1))
    if(strncmp(new, 'y', 1))
        fprintf('Available regenerator configurations are:\n')
        fprintf('    t, for tubular regenerator set\n')
        fprintf('    a, for annular regenerator\n')
        regen_type = input('enter regenerator configuration ', 's');
        fprintf(fid, '%c\n', regen_type(1));
    else
        fscanf(fid, '%c', 1); % bypass the previous newline character
        regen_type = fscanf(fid, '%c', 1);
    end
    if(strncmp(regen_type, 't', 1))
        fprintf('tubular regenerator housing\n')
        if(strncmp(new, 'y', 1))
            dout = input('enter tube housing external diameter [m] : ');
            domat = input('enter tube housing internal diameter [m] : ');
            lr = input('enter regenerator length [m] : ');
            num = input('enter number of tubes : ');
            fprintf(fid, '%.3e\n', dout);
            fprintf(fid, '%.3e\n', domat);
            fprintf(fid, '%.3e\n', lr);
            fprintf(fid, '%d\n', num);
        else

```

```

        dout = fscanf(fid, '%e',1);
        domat = fscanf(fid, '%e',1);
        lr = fscanf(fid, '%e',1);
        num = fscanf(fid, '%d',1);
    end
    dimat = 0;
    awgr0 = num*pi*domat*lr;
elseif(strncmp(regen_type,'a',1))
    fprintf('annular regenerator housing\n')
    if(strncmp(new,'y',1))
        dout = input('enter housing external diameter [m] : ');
        domat = input('enter housing internal diameter [m] : ');
        dimat = input('enter matrix internal diameter [m] : ');
        lr = input('enter regenerator length [m] : ');
        fprintf(fid, '%.3e\n', dout);
        fprintf(fid, '%.3e\n', domat);
        fprintf(fid, '%.3e\n', dimat);
        fprintf(fid, '%.3e\n', lr);
    else
        dout = fscanf(fid, '%e',1);
        domat = fscanf(fid, '%e',1);
        dimat = fscanf(fid, '%e',1);
        lr = fscanf(fid, '%e',1);
    end
    num = 1;
    awgr0 = pi*(dimat + domat)*lr;
else
    fprintf('regenerator configuration is undefined\n')
    regen_type = 'u';
end
end
end

amat = num*pi*(domat*domat - dimat*dimat)/4; % regen matrix area
awr = num*pi*(dout*dout - domat*domat)/4; % regen housing wall area
#####temporary fix (4/20/02):
kwr = 25; % thermal conductivity [W/m/K]
% note that stainless steel thermal conductivity is temp dependent
% 25 W/m/K for normal engine conditions,
% 6 W/m/K for cryogenic coolers.
cqwr = kwr*awr/lr; % regen wall thermal conductance [W/K]

matrix(amat);
=====
function matrix(amat)
% Specifies regenerator matrix geometric and thermal properties
% Israel Urieli 03/31/02
% modified 11/27/10 for no regenerator matrix

global matrix_type % m)esh, f)oil or n)o matrix
global new fid % new data file

matrix_type = 'u';
while(strncmp(matrix_type,'u',1))
    if(strncmp(new,'y',1))
        fprintf('Available matrix types are:\n')
        fprintf(' m, for mesh matrix\n')
        fprintf(' f, for foil matrix\n')
        fprintf(' n, for no matrix\n')
        matrix_type = input('enter matrix type ','s');
        fprintf(fid, '%c\n', matrix_type(1));
    else

```

```

        fscanf(fid, '%c',1); % bypass the previous newline character
        matrix_type = fscanf(fid, '%c',1);
    end
    if(strncmp(matrix_type,'m',1))
        mesh(amat);
    elseif(strncmp(matrix_type,'f',1))
        foil(amat);
    elseif(strncmp(matrix_type,'n',1))
        nomatrix(amat);
    else
        fprintf('matrix configuration is undefined\n')
        matrix_type = 'u';
    end
end
end
=====
function mesh(amat)
% Specifies mesh matrix geometric and thermal properties
% Israel Urieli 03/31/02

global vr % regen void volume [m^3]
global ar % regen internal free flow area [m^2]
global awgr % regen internal wetted area [m^2]
global awgr0 % no matrix regenerator wetted area [m^2]
global lr % regenerator effective length [m]
global dr % regen hydraulic diameter [m]
global new fid % new data file

fprintf(' stacked wire mesh matrix\n')
if(strncmp(new,'y',1))
    porosity = input('enter matrix porosity : ');
    dwire = input('enter matrix wire diameter [m] : ');
    fprintf(fid, '%.3f\n', porosity);
    fprintf(fid, '%.3e\n', dwire);
else
    porosity = fscanf(fid,'%f',1);
    dwire = fscanf(fid,'%e',1);
end

ar = amat*porosity;
vr = ar*lr;
dr = dwire*porosity/(1 - porosity);
awgr = 4*vr/dr + awgr0;

fprintf(' matrix porosity: %.3f\n', porosity)
fprintf(' matrix wire diam %.2f(mm)\n', dwire*1e3)
fprintf(' hydraulic diam %.3f(mm)\n', dr*1e3)
fprintf(' total wetted area %.3e(sq.m)\n', awgr)
fprintf(' regenerator length %.1f(mm)\n', lr*1e3)
fprintf(' void volume %.2f(cc)\n', vr*1e6)
=====
function foil(amat)
% Specifies foil matrix geometric and thermal properties
% Israel Urieli 03/31/02

global vr % regen void volume [m^3]
global ar % regen internal free flow area [m^2]
global awgr % regen internal wetted area [m^2]
global awgr0 % no matrix regenerator wetted area [m^2]
global lr % regenerator effective length [m]
global dr % regen hydraulic diameter [m]

```

```

global new fid % new data file

fprintf(' wrapped foil matrix\n')
if(strncmp(new,'y',1))
    fl = input('enter unrolled length of foil [m] : ');
    th = input('enter foil thickness [m] : ');
    fprintf(fid, '%.3f\n', fl);
    fprintf(fid, '%.3e\n', th);
else
    fl = fscanf(fid,'%f',1);
    th = fscanf(fid,'%e',1);
end

am = th*fl;
ar = amat - am;
vr = ar*lr;
awgr = 2*lr*fl + awgr0;
dr = 4*vr/awgr;
porosity = ar/amat;

fprintf(' unrolled foil length: %.3f(m)\n', fl)
fprintf(' foil thickness %.3f(mm)\n',th*1e3)
fprintf(' hydraulic diam %.3f(mm)\n', dr*1e3)
fprintf(' total wetted area %f(sq.m)\n', awgr)
fprintf(' void volume %.2f(cc)\n', vr*1e6)
fprintf(' porosity %.3f\n', porosity)
%=====
function nomatrix(amat)
% Specifies conditions for no regenerator matrix
% Israel Urieli 11/27/10

global vr % regen void volume [m^3]
global ar % regen internal free flow area [m^2]
global awgr % regen internal wetted area [m^2]
global awgr0 % no matrix regenerator wetted area [m^2]
global lr % regenerator effective length [m]
global dr % regen hydraulic diameter [m]

fprintf(' no regenerator matrix\n')

ar = amat;
vr = ar*lr;
awgr = awgr0;
dr = 4*vr/awgr;

fprintf(' hydraulic diam %.3f(mm)\n', dr*1e3)
fprintf(' total wetted area %f(sq.m)\n', awgr)
fprintf(' void volume %.2f(cc)\n', vr*1e6)
%=====

```

Módulo 6 – gas

```

function gas
% specifies the working gas properties (he, h2, air)
% Israel Urieli 4/20/02

global rgas % gas constant [J/kg.K]

```

```

global cp % specific heat capacity at constant pressure [J/kg.K]
global cv % specific heat capacity at constant volume [J/kg.K]
global gama % ratio: cp/cv
global mu0 % dynamic viscosity at reference temp t0 [kg.m/s]
global t0 t_suth % reference temperature [K], Sutherland constant [K]
global prandtl % Prandtl number
global new fid % new data file

gas_type = 'un';
while(strncmp(gas_type,'un',2))
    if(strncmp(new,'y',1))
        fprintf('Available gas types are:\n');
        fprintf('  hy)drogen\n');
        fprintf('  he)lium\n');
        fprintf('  ai)r\n');
        gas_type = input('enter gas type: ','s');
        gas_type = [gas_type(1), gas_type(2)];
        fprintf(fid, '%s\n', gas_type);
    else
        fscanf(fid, '%c',1); % bypass the previous newline character
        gas_type = fscanf(fid, '%c',2);
    end
    if(strncmp(gas_type,'hy',2))
        fprintf('gas type is hydrogen\n')
        gama = 1.4;
        rgas = 4157.2;
        mu0 = 8.35e-6;
        t_suth = 84.4;
    elseif(strncmp(gas_type,'he',2))
        fprintf('gas type is helium\n')
        gama = 1.67;
        rgas = 2078.6;
        mu0 = 18.85e-6;
        t_suth = 80.0;
    elseif(strncmp(gas_type,'ai',2))
        fprintf('gas type is air\n')
        gama = 1.4;
        rgas = 287.0;
        mu0 = 17.08e-6;
        t_suth = 112.0;
    else
        fprintf('gas type is undefined\n')
        gas_type = 'un';
    end
end
cv = rgas/(gama - 1);
cp = gama*cv;
t0 = 273;
prandtl = 0.71;

```

Módulo 7 – operat

```

function operat
% Determine operating parameters and do Schmidt analysis
% Israel Urieli 4/20/02

global pmean % mean (charge) pressure [Pa]
global tk tr th % cooler, regenerator, heater temperatures [K]

```

```

global freq omega % cycle frequency [herz], [rads/s]
global new fid % new data file

if(strncmp(new,'y',1))
    pmean = input('enter mean pressure (Pa) : ');
    tk = input('enter cold sink temperature (K) : ');
    th = input('enter hot source temperature (K) : ');
    freq = input('enter operating frequency (herz) : ');
    fprintf(fid, '%.1f\n', pmean);
    fprintf(fid, '%.1f\n', tk);
    fprintf(fid, '%.1f\n', th);
    fprintf(fid, '%.1f\n', freq);
else
    pmean = fscanf(fid,'%f',1);
    tk = fscanf(fid,'%f',1);
    th = fscanf(fid,'%f',1);
    freq = fscanf(fid,'%f',1);
end

tr = (th + tk) / 2; % (MODIFICADO)
%tr = (th - tk)/log(th/tk);
omega = 2*pi*freq;
fprintf('operating parameters:\n');
fprintf(' mean pressure (kPa): %.3f\n',pmean*1e-3);
fprintf(' cold sink temperature (K): %.1f\n',tk);
fprintf(' hot source temperature (K): %.1f\n',th);
fprintf(' effective regenerator temperature (K): %.1f\n',tr);
fprintf(' operating frequency (herz): %.1f\n',freq);

Schmidt; % Do Schmidt analysis
=====
function Schmidt
% Schmidt anlysis
% Israel Urieli 3/31/02

global mgas % total mass of gas in engine [kg]
global pmean % mean (charge) pressure [Pa]
global tk tr th % cooler, regen, heater temperatures [K]
global freq omega % cycle frequency [herz], [rads/s]
global vclc vcle % compression,expansion clearence vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global vk vr vh % cooler, regenerator, heater volumes [m^3]
global rgas % gas constant [J/kg.K]

% Schmidt analysis
c = (((vswe/th)^2 + (vswc/tk)^2 + 2*(vswe/th)*(vswc/tk)*cos(alpha))^0.5)/2;
s = (vswc/2 + vclc + vk)/tk + vr/tr + (vswe/2 + vcle + vh)/th;
b = c/s;
sqrtb = (1 - b^2)^0.5;
bf = (1 - 1/sqrtb);
beta = atan(vswe*sin(alpha)/th/(vswe*cos(alpha)/th + vswc/tk));
fprintf(' pressure phase angle beta  %.1f(degrees)\n',beta*180/pi)
% total mass of working gas in engine
mgas=pmean*s*sqrtb/rgas;
fprintf(' total mass of gas:  %.3f(gm)\n',mgas*1e3)
% work output
wc = (pi*vswc*mgas*rgas*sin(beta)*bf/c);
we = (pi*vswe*mgas*rgas*sin(beta - alpha)*bf/c);
w = (wc + we);

```



```

power = w*freq;
eff = w/we; % qe = we
% Printout Schmidt analysis results
fprintf('===== Schmidt analysis =====\n')
fprintf(' Work(joules) %.3e, Power(watts) %.3e\n', w,power);
fprintf(' Qexp(joules) %.3e, Qcom(joules) %.3e\n', we,wc);
fprintf(' indicated efficiency %.3f\n', eff);
fprintf('===== \n')
% Plot Schmidt analysis pv and p-theta diagrams
fprintf('Do you want Schmidt analysis plots\n');
choice = 'n'; %input('y)es or n)o: ', 's');
if(strncmp(choice, 'y', 1))
    plotpv
end
% Plot Alan Organ's particle mass distribution in Natural Coordinates
fprintf('Do you want particle mass distribution plot\n');
choice = 'n' %input('y)es or n)o: ', 's');
if(strncmp(choice, 'y', 1))
    plotmass
end

```

Módulo 8 – *plotpv*

```

function plotpv
% plot pv and p-theta diagrams of schmidt analysis
% Israel Urieli 1/6/03

global vclc vcle % compression, expansion clearance vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global vk % cooler void volume [m^3]
global vh % heater void volume [m^3]
global vr % regen void volume [m^3]
global mgas % total mass of gas in engine [kg]
global rgas % gas constant [J/kg.K]
global pmean % mean (charge) pressure [Pa]
global tk tr th % cooler, regenerator, heater temperatures [K]

theta = 0:5:360;
vc = vclc + 0.5*vswc*(1 + cos(theta*pi/180));
ve = vcle + 0.5*vswe*(1 + cos(theta*pi/180 + alpha));
p = mgas*rgas./(vc/tk + vk/tk + vr/tr + vh/th + ve/th)*1e-5; % [bar]
vtot = (vc + vk + vr + vh + ve)*1e6; % [cc]
figure (1)
plot(vtot,p)
hold on
grid off
xlabel('Volume total (cm³)')
ylabel('Pressão (bar)')
title('Diagrama PxV')
legend('Isotérmico')
figure
plot(theta,p)
grid off
hold on
x = [0,360];
y = [pmean*1e-5, pmean*1e-5];
plot(x,y)
xlabel('crank angle (deg)')

```

```
ylabel('pressure (bar)')
title('Schmidt p-theta diagram')
```

Módulo 9 – *plotmass*

```
function plotmass

% Kyle Wilson 10-2-02
% Stirling Cycle Machine Analysis ME 589
% Particle Trajectory Map
% Equations from Organ's "'Natural' coordinates for analysis of the
practical
% Stirling cycle" and Oegik Soegihardjo's 1993 project on the same topic
% Modified by Israel Urieli (11/27/2010) to obtain correct phase advance
% angle alpha subsequent to error determined by Zack Alexy (March 2010)

% clc;clear;home;format compact;
%% Inputs from define program
global vclc vcle % compression,expansion clearance vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global vk % cooler void volume [m^3]
global vh % heater void volume [m^3]
global vr % regen void volume [m^3]
global pmean % mean (charge) pressure [Pa]
global tk tr th % cooler, regenerator, heater temperatures [K]

NT = th/tk; % Temperature ratio
Vref = vswe; % Reference volume (m^3)

%% Fixed reduced volumes (dimensionless)
vswe_r = (vswe/Vref)/NT; % Reduced expansion swept volume
vcle_r = (vcle/Vref)/NT; % Reduced expansion clearance volume
vh_r = (vh/Vref)/NT; % Reduced heater void volume
vr_r = (vr/Vref)*log(NT)/(NT-1); % Reduced regenerator void volume
vk_r = (vk/Vref); % Reduced cooler void volume
vswc_r = (vswc/Vref); % Reduced compression swept volume
vclc_r = (vclc/Vref); % Reduced compression clearance volume

%% Phase domain
angi = 0;
angf = 2*pi;
dang = 0.1;
ang = [angi:dang:angf];
n = size(ang);

%% Reduced volume variations
for i = 1:n(2)
    deg(i) = ang(i)*180/pi;
    Ve(i) = (vswe/2)*(1-cos(ang(i))); % Expansion volume vs phase
    Vc(i) = (vswc/2)*(1+cos(ang(i) - alpha)); % Compression volume vs phase
    ve(i) = (Ve(i)/Vref)/NT; % Reduced expansion vs phase
    vc(i) = Vc(i)/Vref; % Reduced compression vs phase
    vt(i) = vswe_r + vcle_r + vh_r + vr_r + vk_r + vclc_r + vc(i); % Total
volume vs phase
end
figure
step = 30;
```

```

for m = 1:step-1
    for i = 1:n(2)
        v(i) = ve(i) + (m/step)*(vt(i)-ve(i)); % Reduced volume segments
    end

    hold on
    plot(v,deg,'k:')
end
hold on
plot(ve,deg,'k')
plot(vt,deg,'k')

%% Vertical lines
L1 = vswe_r; % Boundary of reduced expansion swept volume
L2 = L1 + vcle_r; % Boundary of reduced expansion clearance volume
L3 = L2 + vh_r; % Boundary of reduced heater void volume
L4 = L3 + vr_r; % Boundary of reduced regenerator void volume
L5 = L4 + vk_r; % Boundary of reduced cooler void volume
L6 = min(vt); % Boundary of reduced expansion swept volume

point1 = [L1;L1]; % Preparing for plot
point2 = [L2;L2];
point3 = [L3;L3];
point4 = [L4;L4];
point5 = [L5;L5];
point6 = [L6;L6];
point = [0;deg(n(2))];

plot(point1,point,'r--',point2,point,'r--',point3,point,'g--')
plot(point4,point,'g--',point5,point,'b--',point6,point,'b--')
axis([0 max(vt) 0 deg(n(2))])

xlabel('Reduced volume')
ylabel('Crank Angle (deg)')
title('Particle mass plot')

hold off

```

Módulo 10 – *adiabatic*

```

function [var,dvar] = adiabatic
% ideal adiabatic simulation and temperature/energy vs theta plots
% Israel Urieli, 7/20/2002
% Returned values:
%   var(22,37) array of variable values every 10 degrees (0 - 360)
%   dvar(16,37) array of derivatives every 10 degrees (0 - 360)

% Row indices of the var, dvar arrays:
TC = 1; % Compression space temperature (K)
TE = 2; % Expansion space temperature (K)
QK = 3; % Heat transferred to the cooler (J)
QR = 4; % Heat transferred to the regenerator (J)
QH = 5; % Heat transferred to the heater (J)
WC = 6; % Work done by the compression space (J)
WE = 7; % Work done by the expansion space (J)
W = 8; % Total work done (WC + WE) (J)
P = 9; % Pressure (Pa)

```

```

VC = 10; % Compression space volume (m^3)
VE = 11; % Expansion space volume (m^3)
MC = 12; % Mass of gas in the compression space (kg)
MK = 13; % Mass of gas in the cooler (kg)
MR = 14; % Mass of gas in the regenerator (kg)
MH = 15; % Mass of gas in the heater (kg)
ME = 16; % Mass of gas in the expansion space (kg)
TCK = 17; % Conditional temperature compression space / cooler (K)
THE = 18; % Conditional temperature heater / expansion space (K)
GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)
% Size of var (ROWV,COL), dvar (ROWD,COL)
ROWV = 22; % number of rows in the var matrix
ROWD = 16; % number of rows in the dvar matrix
COL = 37; % number of columns in the matrices (every 10 degrees)
%=====
global freq % cycle frequency [herz]
global tk tr th % cooler, regenerator, heater temperatures [K]
global vk % cooler void volume [m^3]
global vr % regen void volume [m^3]
global vh % heater void volume [m^3]

% do ideal adiabatic analysis:
[var,dvar] = adiab;

% Print out ideal adiabatic analysis results
eff = var(W,COL)/var(QH,COL); % engine thermal efficiency
Qkpower = var(QK,COL)*freq; % Heat transferred to the cooler (W)
Qrpower = var(QR,COL)*freq; % Heat transferred to the regenerator (W)
Qhpower = var(QH,COL)*freq; % Heat transferred to the heater (W)
Wpower = var(W,COL)*freq; % Total power output (W)
fprintf('===== ideal adiabatic analysis results =====\n')
fprintf(' Heat transferred to the cooler: %.2f[W]\n', Qkpower);
fprintf(' Net heat transferred to the regenerator: %.2f[W]\n', Qrpower);
fprintf(' Heat transferred to the heater: %.2f[W]\n', Qhpower);
fprintf(' Total power output: %.2f[W]\n', Wpower);
fprintf(' Thermal efficiency : %.1f[%%]\n', eff*100);
fprintf('===== \n')

% Various plots of the ideal adiabatic simulation results
plotadiab(var,dvar);

```

Módulo 11 – *adiab*

```

function [var,dvar] = adiab
% ideal adiabatic model simulation
% Israel Urieli, 7/6/2002
% Returned values:
%   var(22,37) array of variable values every 10 degrees (0 - 360)
%   dvar(16,37) array of derivatives every 10 degrees (0 - 360)

global tk th % cooler, heater temperatures [K]

% Row indices of the var, dvar matrices, and the y,dy variable vectors:
TC = 1; % Compression space temperature (K)
TE = 2; % Expansion space temperature (K)

```

```

QK = 3; % Heat transferred to the cooler (J)
QR = 4; % Heat transferred to the regenerator (J)
QH = 5; % Heat transferred to the heater (J)
WC = 6; % Work done by the compression space (J)
WE = 7; % Work done by the expansion space (J)
W = 8; % Total work done (WC + WE) (J)
P = 9; % Pressure (Pa)
VC = 10; % Compression space volume (m^3)
VE = 11; % Expansion space volume (m^3)
MC = 12; % Mass of gas in the compression space (kg)
MK = 13; % Mass of gas in the cooler (kg)
MR = 14; % Mass of gas in the regenerator (kg)
MH = 15; % Mass of gas in the heater (kg)
ME = 16; % Mass of gas in the expansion space (kg)
TCK = 17; % Conditional temperature compression space / cooler (K)
THE = 18; % Conditional temperature heater / expansion space (K)
GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)
% Size of var(ROWV,COL), y(ROWV), dvar(ROWD,COL), dy(ROWD)
ROWV = 22; % number of rows in the var matrix
ROWD = 16; % number of rows in the dvar matrix
COL = 37; % number of columns in the matrices (every 10 degrees)
%=====
fprintf('=====Ideal Adiabatic Analysis=====\n')
fprintf('Cooler Tk = %.1f[K], Heater Th = %.1f[K]\n', tk, th);
epsilon = 1; % Allowable error in temperature (K)
max_iteration = 20; % Maximum number of iterations to convergence
ninc = 360; % number of integration increments (every degree)
step = ninc/36; % for saving values in var, dvar matrices
dtheta = 2.0*pi/ninc; % integration increment (radians)
% Initial conditions:
y(THE) = th;
y(TCK) = tk;
y(TE) = th;
y(TC) = tk;
iter = 0;
error = 10*epsilon; % Initial error to enter the loop
% Iteration loop to cyclic convergence
while ((error >= epsilon)&&(iter < max_iteration))
% cyclic initial conditions
    tc0 = y(TC);
    te0 = y(TE);
    theta = 0;
    y(QK) = 0;
    y(QR) = 0;
    y(QH) = 0;
    y(WC) = 0;
    y(WE) = 0;
    y(W) = 0;
    fprintf('iteration %d: Tc = %.1f[K], Te = %.1f[K]\n',iter,y(TC),y(TE))
    for i = 1:ninc
        [theta,y,dy] = rk4('dadiab',7,theta,dtheta,y);
    end
    error = abs(tc0 - y(TC)) + abs(te0 - y(TE));
    iter = iter + 1;
end

if (iter >= max_iteration)
    fprintf('No convergence within %d iteration\n',max_iteration)
end

```

```

end

% Initial var and dvar matrix
var = zeros(22,37);
dvar = zeros(16,37);

% a final cycle, to fill the var, dvar matrices
theta=0;
y(QK)=0;
y(QR)=0;
y(QH)=0;
y(WC)=0;
y(WE)=0;
y(W)=0;
[var,dvar] = filmatrix(1,y,dy,var,dvar);
for(i = 2:1:COL)
    for(j = 1:1:step)
        [theta,y,dy] = rk4('dadiab',7,theta,dtheta,y);
    end
    [var,dvar] = filmatrix(i,y,dy,var,dvar);
end
end

```

Módulo 12 – *plotadiab*

```

function plotadiab(var,dvar)
% various plots of ideal adiabatic simulation results
% Israel Urieli, 7/21/2002 (corrected temp plots 12/3/2003)
% Arguments:
%   var(22,37) array of variable values every 10 degrees (0 - 360)
%   dvar(16,37) array of derivatives every 10 degrees (0 - 360)

% Row indices of the var, dvar arrays:
TC = 1; % Compression space temperature (K)
TE = 2; % Expansion space temperature (K)
QK = 3; % Heat transferred to the cooler (J)
QR = 4; % Heat transferred to the regenerator (J)
QH = 5; % Heat transferred to the heater (J)
WC = 6; % Work done by the compression space (J)
WE = 7; % Work done by the expansion space (J)
W = 8; % Total work done (WC + WE) (J)
P = 9; % Pressure (Pa)
VC = 10; % Compression space volume (m^3)
VE = 11; % Expansion space volume (m^3)
MC = 12; % Mass of gas in the compression space (kg)
MK = 13; % Mass of gas in the cooler (kg)
MR = 14; % Mass of gas in the regenerator (kg)
MH = 15; % Mass of gas in the heater (kg)
ME = 16; % Mass of gas in the expansion space (kg)
TCK = 17; % Conditional temperature compression space / cooler (K)
THE = 18; % Conditional temeprature heater / expansion space (K)
GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)
% Size of var (ROWV,COL), dvar (ROWD,COL)
ROWV = 22; % number of rows in the var matrix
ROWD = 16; % number of rows in the dvar matrix
COL = 37; % number of columns in the matrices (every 10 degrees)

```

```

%=====
global tk tr th % cooler, regenerator, heater temperatures [K]
global vk % cooler void volume [m^3]
global vr % regen void volume [m^3]
global vh % heater void volume [m^3]

choice = 'x';
while(~strcmp(choice,'q',1))
    fprintf('Choose plot type:\n');
    fprintf('  p - for a PV diagram\n');
    fprintf('  t - for a temperature vs crank angle plot\n');
    fprintf('  e - for an energy vs crank angle plot\n');
    fprintf('  q - to quit\n');
    choice = input('p)vdiagram, t)emperature, e)nergy, q)uit: ','s');
    if(strcmp(choice,'p',1))
        figure(1)
        vol = (var(VC,:) + vk + vr + vh + var(VE,:))*1e6; % cubic
centimeters
        pres = (var(P,:))*1e-5; % bar
        plot(vol,pres,'k')
        legend('Adiabático')
        grid off
        xlabel('Volume (cm³)')
        ylabel('Pressão (bar)')
        title('Diagrama PxV')
        hold off
    elseif(strcmp(choice,'t',1))
        figure
        x = 0:10:360;
        Tcomp = var(TC,:);
        Texp = var(TE,:);
        plot(x,Tcomp,'b-',x,Texp,'r-');
        hold on
        x = [0,360];
        y = [tk,tk];
        plot(x,y,'b-')
        y = [tr,tr];
        plot(x,y,'g-')
        y = [th,th];
        plot(x,y,'r-')
        hold off
        grid on
        xlabel('Ângulo (graus)');
        ylabel('Temperatura (K)');
        title('Temperatura x ângulo');
    elseif(strcmp(choice,'e',1))
        figure
        x = 0:10:360;
        Qkol = var(QK,:); % [J]
        Qreg = var(QR,:); % [J]
        Qhot = var(QH,:); % [J]
        Work = var(W,:); % [J]
        plot(x,Qkol,'b-',x,Qreg,'g-',x,Qhot,'r-',x,Work,'k-.');
        % Wcom = var(WC,:); % [J]
        % Wexp = var(WE,:); % [J]
        % plot(x,Qkol,'b-',x,Qreg,'g-',x,Qhot,'r-',x,Work,'k-.',x,Wcom,'b--
',x,Wexp,'r--');
        grid on
        xlabel('Crank angle (degrees)');
        ylabel('Energy [Joules]');
        title('Energy vs crank angle');

```

```

    end
end
fprintf('quitting ideal adiabatic plots...\n');

```

Módulo 13 – *dadiab*

```

function [y,dy] = dadiab(theta,y)
% Evaluate ideal adiabatic model derivatives
% Israel Urieli, 7/6/2002
% Arguments: theta - current cycle angle [radians]
%             y(22) - vector of current variable values
% Returned values:
%             y(22) - updated vector of current variables
%             dy(16) vector of current derivatives
% Function invoked : volume.m

% global variables used from "define" functions
global vk % cooler void volume [m^3]
global vr % regen void volume [m^3]
global vh % heater void volume [m^3]
global rgas % gas constant [J/kg.K]
global cp % specific heat capacity at constant pressure [J/kg.K]
global cv % specific heat capacity at constant volume [J/kg.K]
global gama % ratio: cp/cv
global mgas % total mass of gas in engine [kg]
global tk tr th % cooler, regen, heater temperatures [K]

% Indices of the y, dy vectors:
TC = 1; % Compression space temperature (K)
TE = 2; % Expansion space temperature (K)
QK = 3; % Heat transferred to the cooler (J)
QR = 4; % Heat transferred to the regenerator (J)
QH = 5; % Heat transferred to the heater (J)
WC = 6; % Work done by the compression space (J)
WE = 7; % Work done by the expansion space (J)
W = 8; % Total work done (WC + WE) (J)
P = 9; % Pressure (Pa)
VC = 10; % Compression space volume (m^3)
VE = 11; % Expansion space volume (m^3)
MC = 12; % Mass of gas in the compression space (kg)
MK = 13; % Mass of gas in the cooler (kg)
MR = 14; % Mass of gas in the regenerator (kg)
MH = 15; % Mass of gas in the heater (kg)
ME = 16; % Mass of gas in the expansion space (kg)
TCK = 17; % Conditional temperature compression space / cooler (K)
THE = 18; % Conditional temeprature heater / expansion space (K)
GACK = 19; % Conditional mass flow compression space / cooler (kg/rad)
GAKR = 20; % Conditional mass flow cooler / regenerator (kg/rad)
GARH = 21; % Conditional mass flow regenerator / heater (kg/rad)
GAHE = 22; % Conditional mass flow heater / expansion space (kg/rad)
=====

% Volume and volume derivatives:
[y(VC),y(VE),dy(VC),dy(VE)] = volume(theta);

% Pressure and pressure derivatives:
vot = vk/tk + vr/tr + vh/th;
y(P) = (mgas*rgas/(y(VC)/y(TC) + vot + y(VE)/y(TE)));

```



```

top = -y(P)*(dy(VC)/y(TCK) + dy(VE)/y(THE));
bottom = (y(VC)/(y(TCK)*gama) + vot + y(VE)/(y(THE)*gama));
dy(P) = top/bottom;

```

```
% Mass accumulations and derivatives:
```

```

y(MC) = y(P)*y(VC)/(rgas*y(TC));
y(MK) = y(P)*vk/(rgas*tk);
y(MR) = y(P)*vr/(rgas*tr);
y(MH) = y(P)*vh/(rgas*th);
y(ME) = y(P)*y(VE)/(rgas*y(TE));
dy(MC) = (y(P)*dy(VC) + y(VC)*dy(P)/gama)/(rgas*y(TCK));
dy(ME) = (y(P)*dy(VE) + y(VE)*dy(P)/gama)/(rgas*y(THE));
dpop = dy(P)/y(P);
dy(MK) = y(MK)*dpop;
dy(MR) = y(MR)*dpop;
dy(MH) = y(MH)*dpop;

```

```
% Mass flow between cells:
```

```

y(GACK) = -dy(MC);
y(GAKR) = y(GACK) - dy(MK);
y(GAHE) = dy(ME);
y(GARH) = y(GAHE) + dy(MH);

```

```
% Conditional temperatures between cells:
```

```

y(TCK) = tk;
if(y(GACK)>0)
    y(TCK) = y(TC);
end
y(THE) = y(TE);
if(y(GAHE)>0)
    y(THE) = th;
end

```

```
% 7 derivatives to be integrated by rk4:
```

```
% Working space temperatures:
```

```

dy(TC) = y(TC)*(dpop + dy(VC)/y(VC) - dy(MC)/y(MC));
dy(TE) = y(TE)*(dpop + dy(VE)/y(VE) - dy(ME)/y(ME));

```

```
% Energy:
```

```

dy(QK) = vk*dy(P)*cv/rgas - cp*(y(TCK)*y(GACK) - tk*y(GAKR));
dy(QR) = vr*dy(P)*cv/rgas - cp*(tk*y(GAKR) - th*y(GARH));
dy(QH) = vh*dy(P)*cv/rgas - cp*(th*y(GARH) - y(THE)*y(GAHE));
dy(WC) = y(P)*dy(VC);
dy(WE) = y(P)*dy(VE);

```

```
% Net work done:
```

```

dy(W) = dy(WC) + dy(WE);
y(W) = y(WC) + y(WE);

```

Módulo 14 – *filmatrix*

```

function [var,dvar]=Filmatrix(j,y,dy,var,dvar);
% Fill in the j-th column of the var, dvar matrices with values of y, dy
% Israel Urieli, 7/20/2002
% Arguments: j - column index (1 - 37, every 10 degrees of cycle angle)
%             y(ROWV) - vector of current variable values
%             dy(ROWD) vector of current derivatives
%             var(ROWV,37) - matrix of current variables vs cycle angle

```

```

%           dvar(ROWD,37) - matrix of current derivatives vs cycle angle
% Returned values:
%           var(ROWV,37) - matrix of updated variables vs cycle angle
%           dvar(ROWD,37) - matrix of updated derivatives vs cycle angle

ROWV = 22; % number of rows in the var matrix
ROWD = 16; % number of rows in the dvar matrix

for(i = 1:1:ROWV)
    var(i,j) = y(i);
end
for(i = 1:1:ROWD)
    dvar(i,j) = dy(i);
end

```

Módulo 15 – volume

```

function [vc,ve,dvc,dve] = volume(theta)
% determine working space volume variations and derivatives
% Israel Urieli, 7/6/2002
% Modified 2/14/2010 to include rockerV (rockdrive)
% Modified 6/14/2016 to include Beta free piston (sinevol)
% Argument: theta - current cycle angle [radians]
% Returned values:
%   vc, ve - compression, expansion space volumes [m^3]
%   dvc, dve - compression, expansion space volume derivatives

global engine_type % s)inusoidal, y)oke r)ockerV (all alpha engines)

if(strncmp(engine_type,'s',1))
    [vc,ve,dvc,dve] = sinevol(theta);
elseif(strncmp(engine_type,'y',1))
    [vc,ve,dvc,dve] = yokevol(theta);
elseif(strncmp(engine_type,'r',1))
    [vc,ve,dvc,dve] = rockvol(theta);
elseif(strncmp(engine_type,'b',1))
    [vc,ve,dvc,dve] = sinevol(theta);
end
%=====

function [vc,ve,dvc,dve] = sinevol(theta)
% sinusoidal drive volume variations and derivatives
% Israel Urieli, 7/6/2002
% Argument: theta - current cycle angle [radians]
% Returned values:
%   vc, ve - compression, expansion space volumes [m^3]
%   dvc, dve - compression, expansion space volume derivatives

global vclc vcle % compression,expansion clearence vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]

vc = vclc + 0.5*vswc*(1 + cos(theta+pi));
ve = vcle + 0.5*vswe*(1 + cos(theta + alpha+pi));
dvc = -0.5*vswc*sin(theta+pi);
dve = -0.5*vswe*sin(theta + alpha+pi);
%=====

```

```

function [vc,ve,dvc,dve] = yokevol(theta)
% Ross yoke drive volume variations and derivatives
% Israel Urieli, 7/6/2002
% Argument: theta - current cycle angle [radians]
% Returned values:
%   vc, ve - compression, expansion space volumes [m^3]
%   dvc, dve - compression, expansion space volume derivatives

global vclc vcle % compression,expansion clearance vols [m^3]
global vswc vswe % compression, expansion swept volumes [m^3]
global alpha % phase angle advance of expansion space [radians]
global b1 % Ross yoke length (1/2 yoke base) [m]
global b2 % Ross yoke height [m]
global crank % crank radius [m]
global dcomp dexcomp % diameter of compression/expansion pistons [m]
global acomp aexp % area of compression/expansion pistons [m^2]
global ymin % minimum yoke vertical displacement [m]

sinth = sin(theta);
costh = cos(theta);
bth = (b1^2 - (crank*costh)^2)^0.5;
ye = crank*(sinth + (b2/b1)*costh) + bth;
yc = crank*(sinth - (b2/b1)*costh) + bth;

ve = vcle + aexp*(ye - ymin);
vc = vclc + acomp*(yc - ymin);
dvc = acomp*crank*(costh + (b2/b1)*sinth + crank*sinth*costh/bth);
dve = aexp*crank*(costh - (b2/b1)*sinth + crank*sinth*costh/bth);
%=====

function [vc,ve,dvc,dve] = rockvol(theta)
% Ross Rocker-V drive volume variations and derivatives
% Israel Urieli, 7/6/2002 & Martine Long 2/25/2005
% Argument: theta - current cycle angle [radians]
% Returned values:
%   vc, ve - compression, expansion space volumes [m^3]
%   dvc, dve - compression, expansion space volume derivatives

global vclc vcle % compression,expansion clearance vols [m^3]
global crank % crank radius [m]
global acomp aexp % area of compression/expansion pistons [m^2]
global conrodc conrode % length of comp/exp piston connecting rods [m]
global ycmax yemax % maximum comp/exp piston vertical displacement [m]

sinth = sin(theta);
costh = cos(theta);
beth = (conrode^2 - (crank*costh)^2)^0.5;
bcth = (conrodc^2 - (crank*sinth)^2)^0.5;
ye = beth - crank*sinth;
yc = bcth + crank*costh;

ve = vcle + aexp*(yemax - ye);
vc = vclc + acomp*(ycmax - yc);
dvc = acomp*crank*sinth*(crank*costh/bcth + 1);
dve = -aexp*crank*costh*(crank*sinth/beth - 1);

```

Módulo 16 – rk4

```
function [x, y, dy] = rk4(deriv,n,x,dx,y)
%Classical fourth order Runge-Kutta method
%Integrates n first order differential equations
%dy(x,y) over interval x to x+dx
%Israel Urieli - Jan 21, 2002
x0 = x;
y0 = y;

[y,dy1] = feval(deriv,x0,y);
for i = 1:n
    y(i) = y0(i) + 0.5*dx*dy1(i);
end
xm = x0 + 0.5*dx;
[y,dy2] = feval(deriv,xm,y);
for i = 1:n
    y(i) = y0(i) + 0.5*dx*dy2(i);
end
[y,dy3] = feval(deriv,xm,y);
for i = 1:n
    y(i) = y0(i) + dx*dy3(i);
end
x = x0 + dx;
[y,dy] = feval(deriv,x,y);
for i = 1:n
    dy(i) = (dy1(i) + 2*(dy2(i) + dy3(i)) + dy(i))/6;
    y(i) = y0(i) + dx*dy(i);
end
```

APÊNDICE A

Arquivo texto *protótipo*. (este arquivo contém os dados de entrada e as condições de operação do protótipo estudado. Deve ser utilizado quando o programa principal *sea* for executado e o usuário optar por não criar um novo arquivo.

Prototipo

```
b
4.290e-02
4.830e-02
90.0
5.100e-02
4.900e-02
1.580e-02
4.000e-07
2.120e-06
a
4.900e-02
4.700e-02
4.830e-02
t
4.700e-02
4.700e-02
4.830e-02
1
m
0.80
4.000e-04
a
4.900e-02
4.700e-02
2.990e-02
ai
14.0e+05
300.0
600.0
3
```

APÊNDICE B

Algoritmo da análise isotérmica baseado no modelo de Martini (1983).

```

% Modelo isotérmico de Martini (1984)

clc
clear variables

mr= 0.08470;      %[J/K]  inventário de gas
Th=600;          %[K]    temperatura expansao
Tc=300;          %[K]    temperatura compressao
Tr=(Th+Tc)/2;    %[K]    temperatura regenerador
f = 3;           %[rps]  rotacao do motor
Vh1=91.08e-6;    %[m³]   volume percorrido pistao deslocador (face quente)
Vc1=91.08e-6;    %[m³]   volume percorrido pistao deslocador (face fria)
Vp1=91.12e-6;    %[m³]   volume percorrido pistao de potencia
Vrd=83.80e-6;    %[m³]   volume regenerador
Vhd=2.12e-6;     %[m³]   volume morto expansao
Vcdhx=0.40e-6;   %[m³]   volume morto compressao
Vcd=Vcdhx - (Vp1) * (1-sqrt(2)/2); %   volume morto compressao para o Beta
                                           %   quando alpha=90 e Vc1=Vp1

alpha=90;
ind=1;
W_v=0;

for phi=0:1:360
    Vh = (Vh1/2)*(1 - cosd(phi)) + Vhd;
    Vc = (Vc1/2)*(1 + cosd(phi)) + Vcd + (Vp1/2) * (1 - cosd(phi - alpha));
    Vt = Vh + Vc + Vrd;
    P = mr/(Vh/Th + Vc/Tc + Vrd/Tr);
    phi_v(ind)=phi;
    Vh_v(ind)=Vh;
    Vc_v(ind)=Vc;
    Vt_v(ind)=Vt;
    P_v(ind)=P;
    W_v(ind) = P * Vt;
    ind=ind+1;
end

figure(1)
plot(Vt_v*100^3,P_v/100000)
xlabel('Volume [cm³]')
ylabel('Pressão Instantânea [bar]')

W = trapz(Vt_v,P_v);
L = W * f;
P_max = max(P_v);
P_min = min(P_v);
P_med = mean(P_v);

```