

AVALIAÇÃO DE MODELOS DE COMBATE ENTRE MÍSSIL E AERONAVE

Wilson José Vieira

IEAv/CTA,

CP 6044, São José dos Campos, SP, 12231-970o

wjvieira@ieav.cta.br

Airton Prati

IEAv/CTA,

CP 6044, São José dos Campos, SP, 12231-970o

prati@ieav.cta.br

José Paulo Breda Destro

IEAv/CTA,

CP 6044, São José dos Campos, SP, 12231-970o

destrojp@ieav.cta.br

Resumo

Este trabalho tem como objetivo contribuir na modelagem do combate entre uma aeronave defensiva, dotada de míssil ar-ar guiado pela emissão infravermelha da aeronave invasora. São utilizadas técnicas de Monte Carlo na simulação de situações de engajamento considerando as velocidades, ângulo de rastreamento, passos de controle, etc. A análise dos resultados obtidos mostra a importância de Monte Carlo na simulação de combate aéreo.

Palavras-Chaves: Simulação Monte Carlo, Combate aéreo, Míssil Ar-Ar.

Abstract

The objective of this work is to contribute to air combat modeling between a defensive aircraft, armed with an infrared guided missile, and an attacking aircraft. Monte Carlo techniques are used to simulate several situations considering velocities, tracking angles, control steps, etc. The analysis of the results shows the importance of Monte Carlo in air combat simulation.

Keywords: Monte Carlo Simulation, Air Combat, Air-Air missile;

1. INTRODUÇÃO

A importância do entendimento e modelagem dos elementos de combate aéreo vem ganhando relevância, à medida que este recurso tem sido intensivamente utilizado nas últimas décadas. Por exemplo, na Guerra do Golfo, em que os requisitos de “intervenção cirúrgica”, precisão e coordenação para diminuição do número de baixas, integrados à guerra eletrônica, caracterizam o combate aéreo como método eficiente e eficaz.

Neste trabalho foi aplicada uma modelagem bidimensional diferentemente da abordagem unidimensional [Vieira, 1997] e do combate entre múltiplas aeronaves [Vieira, 1998]. Entretanto, ele representa um estudo mais elaborado visando à otimização das trajetórias do míssil do que a modelagem tri-dimensional utilizada em [Vieira, 2000].

Em princípio, os parâmetros adotados são característicos de equipamentos nacionais, na medida em que o acesso foi possível. O míssil ar-ar é disparado e passa a perseguir a aeronave invasora até destruí-la, perdê-la de foco, ou acabar seu combustível. Os inimigos são gerados aleatoriamente e a mira do míssil é feita considerando o envelope infravermelho. São

simuladas, através de técnicas Monte Carlo, 10000 histórias, em que se obtêm a média e o desvio padrão das variáveis p_k (probabilidade de destruição da aeronave), t_k (tempo de destruição da aeronave) e d_k (distância de destruição).

A simulação mostra sua importância nas verificações preliminares de erros ligados às especificações de projeto e à coordenação dos equipamentos, evitando custos operacionais e prevenindo falhas que poderiam comprometer as missões [Shannon, 1968]. No entanto, o objetivo principal do trabalho é demonstrar o potencial da simulação Monte Carlo. Os dados utilizados e os resultados obtidos são apenas ilustrativos.

2. MODELAGEM DO COMBATE AR-AR

Conforme comentado na introdução, é desenvolvida abordagem em duas dimensões, suficiente para, em linhas gerais, suprir as hipóteses simplificadoras da modelagem. As variáveis referentes ao míssil são indexadas por m e as referentes a aeronave alvo, por a . O intervalo de tempo usado para a simulação é representado por t .

2.1. AMOSTRAGEM MONTE CARLO

O envelope infravermelho da aeronave inimiga que pode ser visualizado pelo míssil é modelado como o setor circular, com abertura $[\alpha_{\min}, \alpha_{\max}]$, centrado na origem do sensor S .

Assim, a região em que serão gerados os números aleatórios θ e r que permitirão amostrar (x, y) , que definem a posição da aeronave são dados pela seguinte expressão:

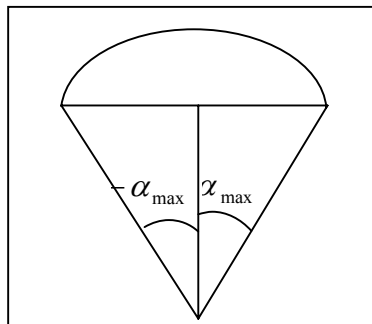


Figura 1. Representação do envelope infravermelho.

(1)

Então, o valor de θ , determina α . E de r , obtém-se ρ .

Logo, retornando-se ao referencial cartesiano (x, y) tem-se:

(2)

A direção da aeronave invasora é escolhida uniformemente no círculo $(0, 2\pi)$. Sabendo que , pode-se obter a função de distribuição de probabilidade (fdp), , que fornece . Portanto,

(3)

Para a análise das estratégias de perseguição da aeronave pelo míssil considere a Figura 2.

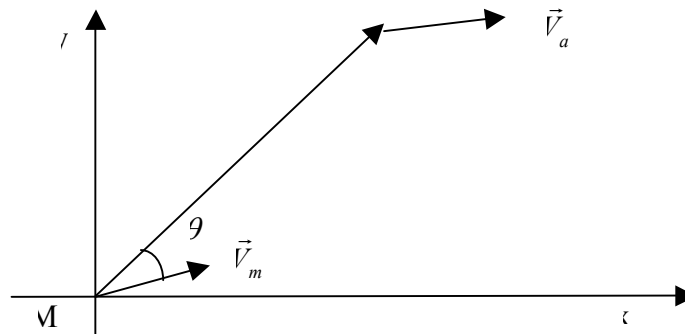


Figura 2. Posições relativas míssil-aeronave.

Neste trabalho são estudadas as seguintes estratégias do míssil perseguir a aeronave:

1. Direto à aeronave, em que . Sendo o ângulo do vetor posição míssil-aeronave com o eixo-x e , o ângulo da direção do míssil com esse eixo. Portanto, o ângulo entre os vetores e direção do míssil, , é igual a zero.
2. Em direção a uma posição futura da aeronave. Este procedimento leva o míssil ao local de interceptação com a aeronave. O míssil inicia a perseguição com direção (valor característico do sensor infravermelho). A continuação da perseguição é feita pelo ajuste da direção do míssil, sempre que (valor no Δt anterior), pela média dos ângulos dos vetores direção do míssil e do vetor posição míssil-aeronave em relação ao eixo-x. Isto é, . Esta correção considera um tempo necessário para manobra do míssil.

Na realização da simulação são introduzidas algumas hipóteses simplificadoras:

- Na geração da posição da aeronave alvo é considerada uma aproximação máxima de , é a velocidade do míssil;
- A verificação se o míssil perdeu o foco e o ajuste de sua direção, caso negativo, será feita a cada Δt , que será dado como valor de entrada.

3. RESULTADOS

Foram simuladas 1000 histórias para cada caso de combate míssil versus aeronave.

Os parâmetros de simulação estão na Tabela 1.

Tabela 1. Parâmetros de aeronave (A) e do míssil (M).

	M1 (Jararaca)	M2 (Piranha)	A1 (AMX)	A2 (Mirage)
Vc^1 [Mach]	2,0	3,0	0,7	2,0
¹ Vc = velocidade de regime em combate (duração máxima 10 s).				
R = alcance do sistema de infravermelho do míssil (mínimo de detecção). $R^2=5000$ m				
ϕ_{MAX} = ângulo de abertura do sensor do míssil, cuja espoleta é acionada quando adentra esfera de raio 10 m centrada na aeronave. $\phi^3_{MAX}=60^\circ$				

As tabelas 2 e 3 mostram os resultados obtidos em termos de percentagens considerando tempos de manobra do míssil de 0,1, 0,5 e 1 segundo. A Figura 3 mostra uma opção de saída do programa que mostra os resultados para os algoritmos de visada direta à aeronave (1) e de visada ao ponto de interceptação (2). O programa escrito em MatLab, sem opção de saída gráfica, encontra-se no Apêndice.

Tabela 2. Algoritmo I (visada direta no aeronave) - resultados das probabilidades: P_K , P_P e P_C .

T_{MAN}	[%]	A1 x M1	A1 x M2	A2 x M2
0,1 s	P_K^1	76,10	69,90	73,10
	P_P^2	18,70	30,10	10,50
	P_C^3	5,20	0,00	16,40
0,5 s	P_K^1	38,40	25,70	47,50
	P_P^2	56,10	74,30	36,00
	P_C^3	5,50	0,00	16,50
1 s	P_K^1	19,70	16,10	23,70
	P_P^2	74,80	83,90	58,70
	P_C^3	5,50	0,00	17,60

- (1) P_K – percentagem de aeronaves interceptadas pelo míssil.
- (2) P_P – percentagem de mísseis que perderam o alvo.
- (3) P_C – percentagem de mísseis que esgotaram o combustível.

Tabela 3. Algoritmo II (visada no ponto futuro)- resultados das probabilidades: P_K , P_P e P_C .

T_{MAN}	[%]	A1 x M1	A1 x M2	A2 x M2
0,1 s	P_K^1	67,70	75,70	19,80
	P_P^2	27,20	24,30	71,40
	P_C^3	5,10	0,00	8,80
0,5 s	P_K^1	19,20	16,20	3,50
	P_P^2	71,50	83,80	85,40
	P_C^3	5,30	0,00	11,10
1 s	P_K^1	7,10	6,80	0,90
	P_P^2	87,10	93,20	87,50
	P_C^3	5,80	0,00	11,60

- (1) P_K – percentagem de aeronaves interceptadas pelo míssil.
- (2) P_P – percentagem de mísseis que perderam o alvo.
- (3) P_C – percentagem de mísseis que esgotaram o combustível.

Saída do Programa MissilArAr_2004 - DATA: 6/4/2004 - HORA: 9:21:49

DADOS DE ENTRADA:

Numero de historias = 1000
 Delta tempo (timestep) = 0.020000
 Velocidade do míssil = 400.000000
 Velocidade da aeronave = 250.000000
 Tempo de vôo do míssil(s) = 15.000000
 Tempo entre manobras(s) = 0.100000
 Angulo máximo de visão = 1.047198 (rad) / 60 (°)
 Distancia p/ disparar(m) = 5000.000000
 Distancia p/ detonação(m) = 10.000000

Resultados:

nk1 = 542 (54.20 %) (Número de aeronaves abatidas pelo míssil tipo 1)
 np1 = 73 (7.30 %) (Número de mísseis tipo 1 que perderam o alvo)
 nc1 = 385 (38.50 %) (Número de mísseis tipo 1 que esgotaram o combustível)

 nk2 = 334 (33.40 %) (Número de aeronaves abatidas pelo míssil tipo 2)
 np2 = 362 (36.20 %) (Número de mísseis tipo 2 que perderam o alvo)
 nc2 = 304 (30.40 %) (Número de tipo 2 que esgotaram o combustível)

Figura 3. Saída do programa

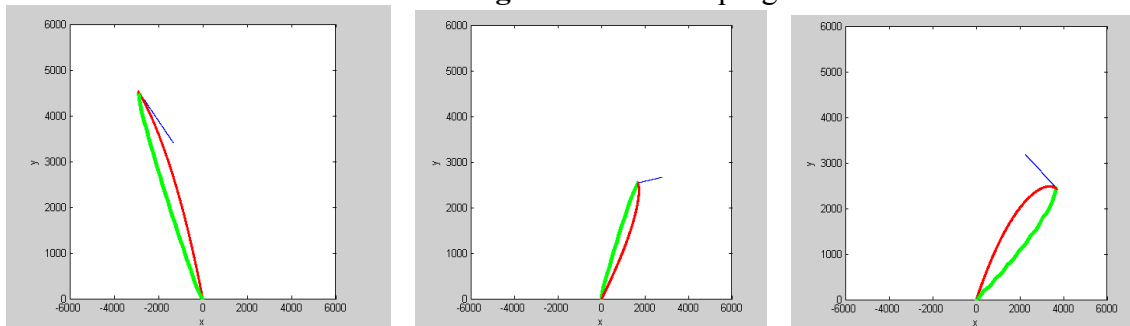


Figura 4. Combate A2 x M2, tempo de manobra 1,0 s.

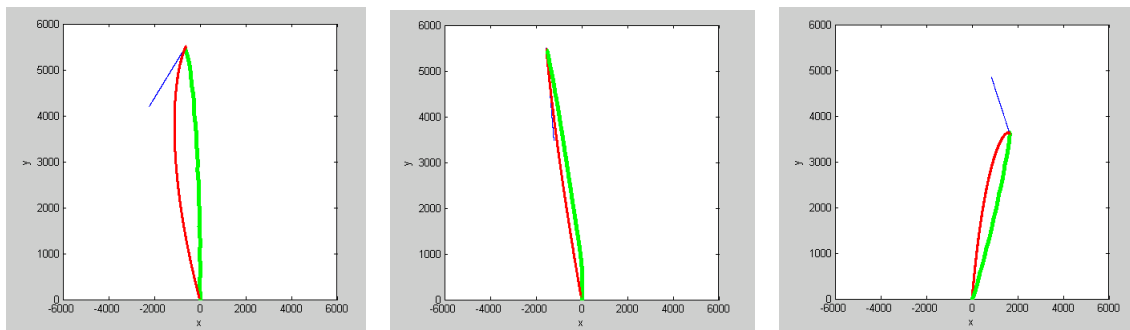


Figura 5. Combate A2 x M2, tempo de manobra 0,1 s.

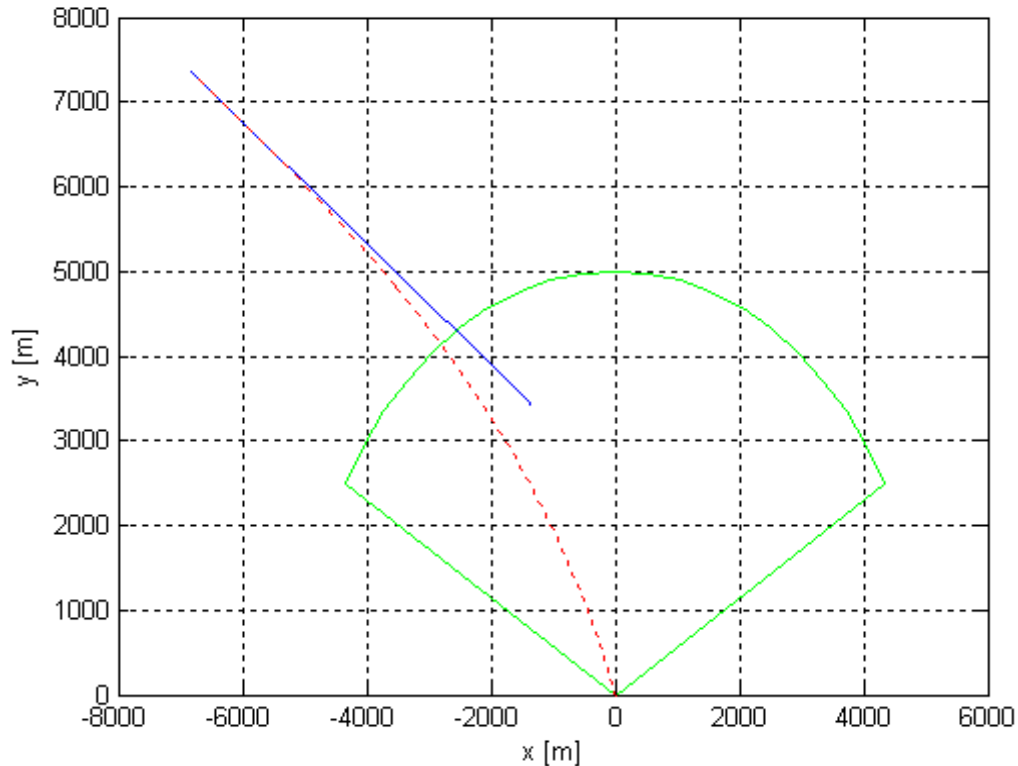


Figura 6. Cone do infravermelho do míssil em combate A2 x M2, tempo de manobra 0,1 s

4. CONCLUSÕES

A manobrabilidade do míssil é fundamental para eficiência de interceptação. Para maiores tempos de manobra, aumenta a probabilidade de perda do alvo. Com relação à autonomia, pode-se inferir que o Algoritmo I é mais eficiente.

Para a razão de velocidades entre míssil e aeronave de 1,5 (A2 x M2) é significativa a superioridade do Algoritmo I para todos os casos de combate simulados.

De maneira geral, o Algoritmo I foi mais eficiente. A exceção foi para o caso A1 x M2 para 0,1 s de tempo de manobra. Uma explicação para isto é que o Algoritmo II não consegue manter a trajetória de interceptação com a velocidade necessária.

A simulação Monte Carlo para o problema do combate míssil x aeronave pode ser feita para modelos bastante detalhados. No entanto, neste trabalho foi mostrada a potencialidade do método apenas para um modelo de demonstração.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] VIEIRA, W.J. et alii, “Estatística da Simulação Monte Carlo do Combate Míssil-Aeronave”, Nota Técnica nº 01/00, IEAv-CTA, S. J. dos Campos, 2000.

- [2] VIEIRA, W.J. e Prati, A., “Simulação Monte Carlo de combate entre múltiplas aeronaves e defesa antiaérea”, II Simpósio de Pesquisa Operacional e III Simpósio de Logística da Marinha (SPOLM-98), Rio de Janeiro, 1998.

- [3] VIEIRA, W.J. e Prati, A., “*Simulação Monte Carlo de Combate Aeronave-Defesa Antiaérea*”, I Simpósio de Pesquisa Operacional da Marinha e II Simpósio de Logística da Marinha, 16 e 17 de dezembro de 1997, Rio de Janeiro, RJ.

- [4] SHANNON, R. E., “*Systems Simulation: The Art and Science*”. p. 287-295, Prentice-Hall, Englewood Cliffs, New Jersey, 1975. Condensed from: R. E. Shannon, “*Air Defense Engagement Duel-Model I*”. UARI Report No. 53, University of Alabama Research Institute, Huntsville, Alabama, September, 1968.

- [5] MatLab - <http://www.mathworks.com/products/matlab/>.

APÊNDICE: PROGRAMA DEMONSTRAÇÃO

```

% Variáveis:
%
% distokil   dist. mín. p/ detonação da carga explosiva do
míssil
%           e destruição da aeronave;
% dk         distância média destruição da aeronave;
% phimax    metade do angulo do envelope infravermelho;
% nhist     número total de histórias;
% pk        probabilidade de destruição da aeronave;
% rei       raio do setor esferico do sistema de detecção
infravermelho;
% sm        distância percorrida pelo míssil até a
interseção de rotas;
% timestep  incremento de tempo, corresponde a atitude do
míssil;
% tvmil     tempo de vôo do míssil;
% uvaer     cosenos diretores da aeronave;
% uvmil     cosenos diretores do míssil;
% vela     velocidade média da aeronave;
% vmil     velocidade do míssil;
% saer     posição da aeronave;
% smil     posição do míssil;
%
function MissilArAr_2004()
rand('state',0); out = fopen('ala2.out', 'w');

    % DADOS DE ENTRADA
    nhist = input(' Entre o numero desejado de historias -
nhist: ');
    rei = 5000.;    phimax = pi/3.;    timestep =
0.01;
    timeMan = 0.10;    vmil = 700.0;    vela
= 250.;
    tvmil = 10;    reimin = vmil;    distokil =
10.;
    phimaxg = phimax*180/pi;
    nh = 0 ;    tvmil = tvmil/timestep;

% Inicio do loop das historias
while nh < nhist
    nh = nh + 1;

```



```

% Calculo da trajetoria do missil pelo algoritmo I.
    smil(1)      = 0.0;      smil(2)      = 0.0;

%Sorteio da distancia da aeronave alvo no lançamento do
missil.
while l==1
    rnd1 = rand(1);      rnd2 = rand(1);
    saer1 = rei*sqrt(rnd1)*sin(phimax*(2.*rnd2-1));
    saer2 = rei*sqrt(rnd1)*cos(phimax*(2.*rnd2-1));
    d=sqrt((saer1-smil(1))^2+(saer2-smil(2))^2);
    if d > reimin; break; end; end
%
%Sorteio da direção da aeronave alvo.
    rnd = rand(1); temp = 2.*pi*rnd; uvaer1 = cos(temp);
uvaer2 = sin(temp);

%Cos-senos diretores do vetor posição MA.
a1 = (saer1-smil(1))/d;      a2 = (saer2-smil(2))/d;

% Algoritmo_I - Missil lançado na direção da aeronave.
    saer(1) = saer1; saer(2) = saer2; uvaer(1) = uvaer1;
uvaer(2) = uvaer2;
    alfa = acos(a1);      beta = acos(a2);
    uvmil(1) = cos(alfa); uvmil(2) = sin(alfa);
    angaer=0.0;      time = 0.0;      timeC = 0.0;      np =0;
    for i=1:tvmil
np = np+1;      time=time+timestep;
saerold(1)=saer(1);      saerold(2)=saer(2);
    smilold(1)=smil(1); smilold(2)=smil(2);
saer(1)=saerold(1)+vela*uvaer(1)*timestep;
saer(2)=saerold(2)+vela*uvaer(2)*timestep;
smil(1)=smilold(1)+vmil*uvmil(1)*timestep;
smil(2)=smilold(2)+vmil*uvmil(2)*timestep;
    a1old = a1;      a2old = a2;      angaerold=angaer;
    d=sqrt((saer(1)-smil(1))^2+(saer(2)-smil(2))^2);
    if d < distokil
        fprintf(out,'\n Alg. I - Aeronave destruída, history
# %d TempoTotal= %5.4f distancia= %9.4f\n',nh,time,d);
        break;      end
    if d > rei
        fprintf(out,'\n Alg. I - Míssil perdido ou não
disparado, history # %d ',nh);
        break;      end

% Cálculo do vetor unitário na direção Missil/aeronave

```

```

a1 = (saer(1)-smil(1))/d;          a2 = (saer(2)-smil(2))/d;

    % Cálculo do ângulo entre o vetor direção do missil e o
    vetor direção M/A.
    modaerxm = a1*uvmil(1)+a2*uvmil(2);    angaer =
acos(modaerxm);

    %Teste para verificar se a aeronave deixou o envelope
    infravermelho do missil.
    if angaer > phimax;
        fprintf(out,'\n Alg. I - Missil perdeu contato,
history # %d - angaer = %9.4f',nh,angaer);
        break;    end

    % redirecionamento do missil na direção M/A.
    timeC = timeC + timestep;
    if timeMan < timeC
        uvmil(1)=a1;    uvmil(2)=a2;    timeC = 0.0;
    end
end % do for
if i >= tvmil
    fprintf(out,'\n Alg. I - O missil terminou tempo de voo
e nao destruiu a aeronave, history # %d ',nh);
    end
% Fim do Algoritmo_I

% Algoritmo_II - Missil lançado na direção de uma posição
futura da aeronave.
saer(1) = saer1; saer(2) = saer2; smil(1) = 0.0;
smil(2) = 0.0;
uvaer(1) = uvaer1;    uvaer(2) = uvaer2;
alfa = acos(a1);    beta = acos(a2);    angaer =
0.5*phimax;
if (saer(1) >= 0.0 & saer(2)>= 0.0)
    % Aeronave posicionada no primeiro quadrante
    if (uvaer(1) > 0) & (uvaer(2) > 0)
        if (uvaer(1) - a1) < -0.0001;    alfa = alfa +
angaer;
        else alfa = alfa - angaer;
        end
    else
        if (uvaer(1) < 0) & (uvaer(2) < 0)
            if (abs(uvaer(1)) - a1) > 0.0001;    alfa = alfa +
angaer;
            else alfa = alfa - angaer;
        end
    end
end

```

```

        end
    else
        if (uvaer(1) < 0) & (uvaer(2) > 0); alfa = alfa
+angaer;
        else alfa = alfa - angaer;
    end; end; end
else
    if (saer(1) < 0.0 & saer(2) >= 0.0)
        % Aeronave posicionada no quarto quadrante
        if (uvaer(1) > 0) & (uvaer(2) > 0)
            alfa = alfa - angaer;
        else
            if (uvaer(1) < 0) & (uvaer(2) < 0)
                alfa = alfa + angaer;
            else
                if (uvaer(1) > 0) & (uvaer(2) < 0)
                    if (uvaer(1) - a1) > 0.0001; alfa = alfa -
+angaer;
                    else alfa = alfa + angaer;
                    end
                else
                    if (abs(uvaer(1)) + a1) > 0.0001 ; alfa
=alfa + angaer;
                    else alfa = alfa - angaer;
                end; end; end; end; end; end

        uvmil(1) = cos(alfa);        uvmil(2) = sin(alfa);
        time=0.0;        timeC = 0.0;        np1 =0;

        for i=1:tvmil        %tempo de perseguição
            np1 = np1+1;        time=time+timestep;
            saerold(1)=saer(1);        saerold(2)=saer(2);
            smilold(1)=smil(1);        smilold(2)=smil(2);
            saer(1)=saerold(1)+vela*uvaer(1)*timestep;
            saer(2)=saerold(2)+vela*uvaer(2)*timestep;
            smil(1)=smilold(1)+vmil*uvmil(1)*timestep;
            smil(2)=smilold(2)+vmil*uvmil(2)*timestep;
            a1old = a1;        a2old = a2;        angaerold=angaer;
            d=sqrt((saer(1)-smil(1))^2+(saer(2)-smil(2))^2);

            if d < distokil
                fprintf(out, '\n Alg. II - Aeronave destruída, history
# %d        TempoTotal= %5.4f        distancia= %9.4f\n', nh, time, d);
                break;
            end
        end
    end
end

```

```

    if d > rei
        fprintf(out,'\n Alg. II - Aeronave nao-destruída,
history # %d TempoTotal= %5.4f distancia=
%9.4f\n',nh,time,d);
        break;
    end
    a1 = (saer(1)-smil(1))/d;      a2 = (saer(2)-smil(2))/d;
    modaerxm = a1*uvmil(1)+a2*uvmil(2);      angaer =
acos(modaerxm);
    if angaer > phimax;
        fprintf(out,'\n Alg. II - Aeronave nao-destruída,
history # %d TempoTotal= %5.4f distancia=
%9.4f\n',nh,time,d);
        break;
    end
    timeC = timeC + timestep;
    if timeMan < timeC
        if angaer > angaerold; angaer = 0.5*angaer; end
        if acos(alold) < acos(a1); alfa = alfa+angaer;
        else alfa =alfa-angaer; end
        uvmil(1)=cos(alfa);      uvmil(2)=sin(alfa);
timeC = 0.0;
    end
end % do for
if i >= tvmil
    fprintf(out,'\n Alg. II - O missil terminou tempo de
voo e nao destruiu a aeronave, history # %d ',nh);
    end ; end; fclose(out);
% Fim do Algoritmo_II % Fim do Programa

```