



SPOLM 2008

ISSN 2175-6295

Rio de Janeiro- Brasil, 05 e 06 de agosto de 2008.

SOBRE O ARTIGO “SIMULAÇÃO MONTE CARLO DE COMBATE AERONAVE-DEFESA ANTIAÉREA”

Carlos Henrique Machado S. Esteves

Instituto de Aeronáutica e Espaço, CTA

Praça Marechal Eduardo Gomes, 50 — 12228-904 São José dos Campos, SP

ch.machado.esteves@gmail.com

CC(EN) Fernando Antonio Almeida Coelho

Diretoria de Sistema de Armas da Marinha

Praça Barão de Ladário, s/n. Edifício Barão de Ladário. Rio de Janeiro - RJ

feraac@hotmail.com

Filipe Rodrigues de S. Moreira

Instituto de Aeronáutica e Espaço, CTA

Praça Marechal Eduardo Gomes, 50 — 12228-904 São José dos Campos, SP

filipersmoreira@yahoo.com.br

Guilherme Felipe Reis Duarte

Instituto de Aeronáutica e Espaço, CTA

Praça Marechal Eduardo Gomes, 50 — 12228-904 São José dos Campos, SP

flpduarte@gmail.com

Rodrigo de Andrade

EMBRAER - Empresa Brasileira de Aeronáutica S.A.

Avenida Brigadeiro Faria Lima, 2170 — 12227-901 São José dos Campos, SP

rodrigo.andrade@embraer.com.br

Wilson José Vieira

Institutos de Estudos Avançados, CTA

Rodovia dos Tamoios, km 5,5 — 12228-001 São José dos Campos, SP

wjvieira@ieav.cta.br

RESUMO

Este trabalho teve como objetivo refazer a simulação Monte Carlo de um combate aéreo simplificado entre uma aeronave e uma defesa antiaérea publicada na literatura. Como houve divergências entre outros resultados publicados, este trabalho implementou um novo programa que permitiu a percepção das causas destas divergências. A principal causa foi na consideração do tempo de percurso dos mísseis da base para o cálculo das probabilidades. Uma nova comparação forneceu resultados próximos aos da publicação original.

Palavras-Chaves: Simulação de Defesa Antiaérea; Simulação Monte Carlo; Armamento

Abstract

This study aimed to rebuild the Monte Carlo simulation of a simplified air combat between an aircraft and an air defense base published in the literature. As there were differences between the other published results, this work implemented a new program that allowed the perception of the causes of these differences. The main cause was in the consideration of the time of the base missile's path for the calculation of probabilities. The correction provided results close to those of the original publication.

Keywords: Air Defense Engagement Simulation; Monte Carlo Simulation; Aerial Weapons; Computational Methods.

1. INTRODUÇÃO

Este trabalho teve como objetivo refazer a simulação Monte Carlo do combate aéreo simplificado entre uma aeronave e uma defesa antiaérea feita por Shannon [1], em 1968, e por Vieira & Prati [2], em 1997. Como houve divergências entre os resultados dos trabalhos anteriores, este trabalho implementou um novo programa que permitiu a percepção do erro no programa em [2], que considerou erradamente o tempo de percurso dos mísseis da base para o cálculo das probabilidades. A correção forneceu resultados próximos aos da publicação original [1].

No modelo de duelo entre a aeronave e a defesa antiaérea é possível as seguintes alternativas:

- A aeronave é destruída e a defesa antiaérea sobrevive;
- A aeronave sobrevive e a defesa antiaérea é destruída;
- Ambos são destruídos; e
- Ambos sobrevivem.

O modelo proposto baseia-se nas seguintes regras:

- No caso do alvo ser atingido, ele é totalmente destruído;
- Uma vez disparados os mísseis não podem ser destruídos;
- A aeronave e a defesa antiaérea começam a disparar no instante em que o oponente entrar em seu raio de ação;
- Ambos continuam a disparar até um ou ambos serem destruídos, ou acabarem os mísseis;
- A aeronave voa a velocidade constante, direto ao alvo e, após o último disparo, reverte a direção e se afasta com a mesma velocidade.

Os parâmetros da aeronave são:

V = Velocidade;

D_1 = Intervalo entre disparos da aeronave;

M = Número de mísseis;

R_1 = Raio de ação dos mísseis da aeronave,

$U_1(R)$ = Velocidade média do míssil como função da distância do disparo;

$P_1(R)$ = Probabilidade de acerto do míssil como função da distância de disparo.

Os parâmetros da defesa são:

D_2 = Intervalo entre disparos da defesa;

N = Número de mísseis da defesa;

R_2 = Raio de ação dos mísseis da defesa,

$U_2(R)$ = Velocidade média do míssil como função da distância do disparo;

$P_2(R)$ = Probabilidade de acerto do míssil como função da distância de disparo.

As equações do modelo estão descritas em [1, 2].

2. DESENVOLVIMENTO

A simulação feita nesse trabalho foi desenvolvida em MatLab, utilizando uma discretização do tempo de simulação em pequenos passos, verificando e tratando a ocorrência dos eventos dentro do intervalo de cada passo. Este método difere dos métodos utilizados em [1,2], que resolviam analiticamente as equações que definem o tempo de ocorrência de cada evento.

Outra diferença entre os métodos anteriores e este é a contagem da quantidade média de mísseis lançados para atingir o alvo. Nos trabalhos anteriores o número total de mísseis lançados quando o alvo é destruído é dividido pelo número de histórias para obter a quantidade média de mísseis. Já neste trabalho, o número total de mísseis lançados quando o alvo é destruído é dividido apenas pelo número de histórias em que o alvo foi destruído. Desta forma obtém-se a média de mísseis lançados para destruir o alvo, apenas quando o alvo foi destruído.

Uma interface gráfica animada foi implementada, o que facilita a verificação dos resultados do programa. Um exemplo da interface gráfica implementada é mostrado na Fig. 1.

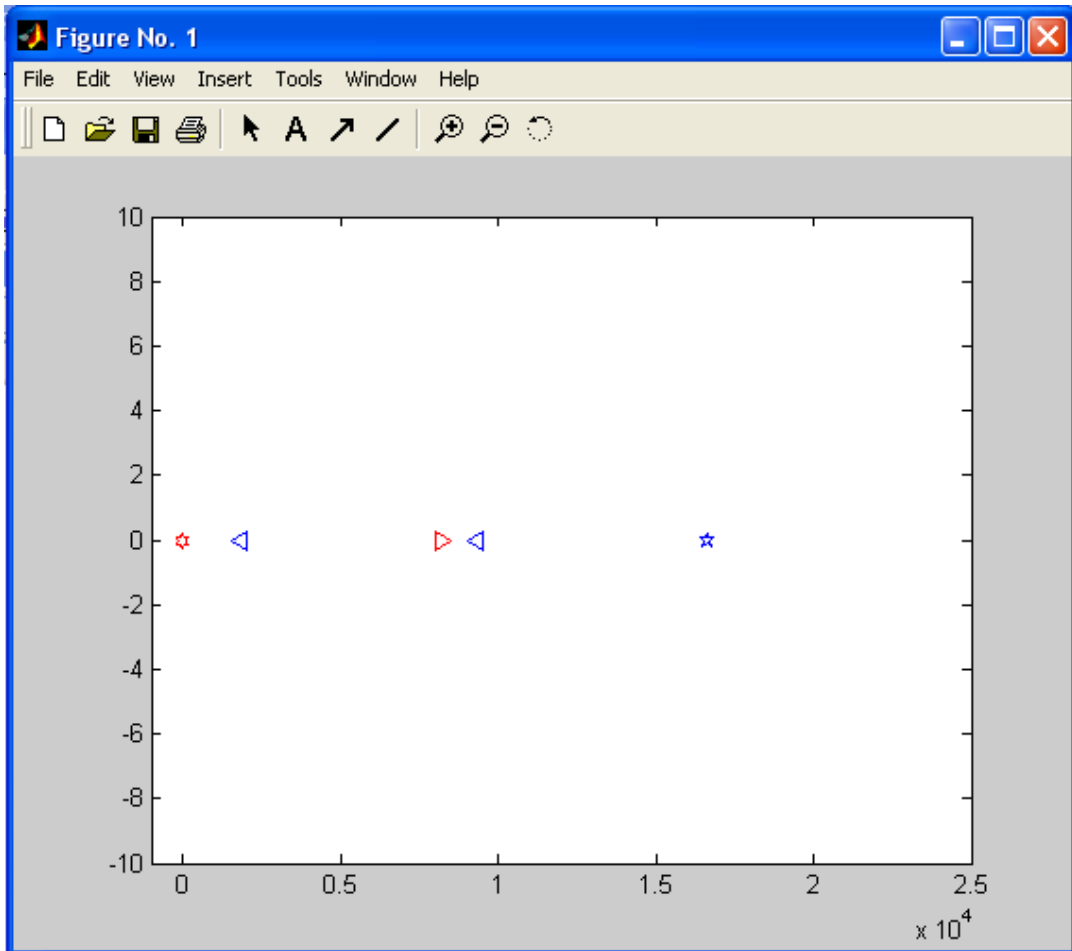


Figura 1. Interface gráfica.

Na animação, a base é exibida como uma estrela vermelha e seus mísseis são vermelhos, já a aeronave é exibida como uma estrela azul e seus mísseis são azuis.

Para rodar a simulação (ver Anexo A), a função combate deve ser chamada com os seguintes argumentos, na ordem:

- D2: Tempo entre disparos da base
- R2: Raio de ação da base
- nhist: Número de histórias simuladas
- passo: Passo do tempo da simulação

Para visualizar o modo gráfico, deve-se utilizar o $nhist=1$ e um passo pequeno, como por exemplo, 0.02. Para $nhist>1$ o modo gráfico é suprimido e as estatísticas são obtidas.

3. RESULTADOS

Este trabalho implementou o problema de duas maneiras, envolvendo diferentes suposições sobre os cálculos das velocidades e probabilidades de acerto dos mísseis da base.

Tipo 1: As velocidades obtidas pelas fórmulas U1 e U2 foram consideradas relativas, somando-se a velocidade da aeronave para os mísseis da aeronave, e somando-se ou subtraindo-se a velocidade da aeronave para os mísseis da base (dependendo se a aeronave está de aproximando ou se afastando). A distância que entra na fórmula foi considerada a distância no instante do lançamento para os mísseis da aeronave, e a

distância em que o míssil atinge o alvo para os mísseis da base. A probabilidade de acerto foi calculada sempre em relação à distância no momento do lançamento.

Tipo 2: Igual ao Tipo 1, porém a probabilidade de acerto para os mísseis da base foi calculada em relação à distância em que o míssil atinge o alvo.

As Tabelas 1 e 2 mostram os tempos de chegada dos mísseis da base e da aeronave. Não há informações sobre esses tempos em [1].

Tabela 1. Comparação entre os tempos de chegada (t_r) dos mísseis da aeronave.

Disparo Nr.	t_r [s] [Vieira&Prati]	t_r [s] [Este trabalho Tipos 1 e 2]
1	6,1200	6,1200
2	8,2287	8,2300
3	10,5060	10,5100
4	12,8992	12,9000
5	15,3727	15,3800
6	17,9008	17,9100
7	20,4632	20,4700
8	23,0420	23,0500
9	25,6191	25,6200
10	28,1729	28,1800

Esses resultados mostram que o trabalho em [2] faz as mesmas considerações sobre as velocidades que os Tipos 1 e 2 deste trabalho. A Tabela 3 mostra os resultados dos cálculos das probabilidades.

Tabela 2. Comparação entre os tempos de chegada dos mísseis da base.

Disparo Nr.	t_r [s] [Vieira&Prati]	t_r [s] [Este trabalho Tipos 1 e 2]
1	7,3412	7,5300
2	15,9550	15,9500
3	24,8135	24,7400
4	36,8593	37,0700
5	50,4333	49,5900

Tabela 3. Comparação entre as probabilidades P2 para o Tipo 1 e Tipo 2.

Disparo Nr.	P2 Tipo 1	P2 Tipo 2
1	0.20	0.32
2	0.50	0.58
3	0.70	0.70
4	0.70	0.61
5	0.42	

Nota-se que a única diferença entre as implementações dos Tipos 1 e 2 deste trabalho está no cálculo da probabilidade dos mísseis da base acertar o alvo. No Tipo 1, a probabilidade é calculada através da fórmula P2, utilizando como parâmetro a distância entre a base e a aeronave no momento do lançamento do míssil, enquanto no tipo 2, o parâmetro é a distância entre a base e a aeronave no momento em que o míssil chega ao alvo.

A Tabela 4 mostra a comparação entre os resultados dos 5 casos testados nos trabalhos anteriores e os deste trabalho. As estatísticas deste trabalho foram obtidas para 5000 histórias utilizando um passo de 0.05s. Em [2] foram utilizadas 10000 histórias. Os programas estão nos apêndices.

4. CONCLUSÕES

Percebe-se que todos os resultados obtidos para a implementação do Tipo 1 estão próximos dos resultados obtidos na implementação em [2], enquanto os resultados da implementação do Tipo 2 estão muito próximos dos resultados obtidos pela implementação em [1]. Obviamente há diferenças devido à aleatoriedade dos testes. Portanto, podemos considerar que a discrepância dos resultados observada em [2] foi solucionada.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] R. E. Shannon *Systems Simulation: The Art and Science*. pág. 287-295, Prentice-Hall, Englewood Cliffs, New Jersey, 1975. Condensado de: R. E. Shannon, *Air Defense Engagement Duel-Model I*. UARI Report No. 53, University of Alabama Research Institute, Huntsville, Alabama, setembro, 1968.
- [2] W. J. Vieira; A. Prati, “Simulação Monte Carlo de Combate Aeronave-Defesa Antiaérea”. Anais do I Simpósio de Pesquisa Operacional e II Simpósio de Logística da Marinha, EGN/CASNAV, Rio de Janeiro, RJ, 1997.
- [3] W. J. Vieira; A. Prati, “Simulação Monte Carlo de Combate entre Múltiplas Aeronaves e Defesa Antiaérea”. Anais do II Simpósio de Pesquisa Operacional e III Simpósio de Logística da Marinha, EGN/CASNAV, Rio de Janeiro, RJ, 1998.

Tabela 4. Comparação de Resultados.

	Referências	I	II	III	IV	V
Base destruída, aeronave sobrevive	[Shannon, 1968]	51,5	49,5	42,3	62,8	46,2
	Este Trabalho (Tipo 2)	51,06	50,28	41,02	59,28	47,48
	[Vieira&Prati, 1997]	62,2	60,6	53,7	66,1	63,7
	Este Trabalho (Tipo 1)	60,56	60,36	54,88	66,36	63,20
	[Vieira&Prati]*	50,57	49,35	43,53	60,12	47,86
Aeronave destruída, base sobrevive	[Shannon, 1968]	26,5	27,1	29,0	20,8	35,2
	Este Trabalho (Tipo 2)	25,42	25,42	30,64	23,08	32,92
	[Vieira&Prati, 1997]	18,6	19,3	23,4	19,1	21,1
	Este Trabalho (Tipo 1)	19,66	19,64	22,38	18,80	21,48
	[Vieira&Prati]*	25,64	25,98	29,02	22,11	32,28
Ambos destruídos	[Shannon, 1968]	22,0	23,4	28,7	16,4	18,5
	Este Trabalho (Tipo 2)	23,48	24,3	28,34	17,56	19,60
	[Vieira&Prati, 1997]	19,3	20,1	22,9	14,8	15,2
	Este Trabalho (Tipo 1)	19,76	19,96	22,74	14,84	15,3
	[Vieira&Prati]*	23,75	24,56	27,43	17,72	19,82
Ambos sobrevivem	[Shannon, 1968]	0,0	0,0	0,0	0,0	0,1
	Este Trabalho (Tipo 2)	0,04	0	0	0,08	0
	[Vieira&Prati, 1997]	0,0	0,0	0,0	0,0	0,0
	Este Trabalho (Tipo 1)	0,02	0,04	0	0	0,02
	[Vieira&Prati]*	0,04	0,11	0,02	0,05	0,04

Número médio de mísseis para destruir a base	[Shannon, 1968]	3,47	3,50	3,45	3,50	3,39
	Este Trabalho (Tipo 2)	4,4644	4,4106	4,2731	4,3811	42,2010
	[Vieira&Prati, 1997]	3,67	3,64	3,54	3,62	3,58
	Este Trabalho (Tipo 1)	4,5817	4,6320	4,3834	4,5246	4,4825
GA	[Vieira&Prati]*	3,54	3,47	3,33	3,54	3,38
Número médio de mísseis para destruir a aeronave	[Shannon, 1968]	1,37	1,38	1,51	1,17	1,73
	Este Trabalho (Tipo 2)	1,3579	1,3673	1,5042	1,1575	1,7026
	[Vieira&Prati, 1997]	1,49	1,49	1,67	1,21	2,04
	Este Trabalho (Tipo 1)	1,5038	1,5298	1,6392	1,2212	2,0430
GM	[Vieira&Prati]*	1,39	1,37	1,50	1,18	1,74

*Programa em [2] considerando a distância entre a base e a aeronave no momento em que o míssil da base chega ao alvo (P2 Tipo 2).

APÊNDICE A: PROGRAMA MATLAB

```
function combate(D2,R2,nhist,passo)
% Simulacao Monte Carlo de Combate Aereo Simplificado
% Uso: combate(D2,R2,nhist,passo)
% D2: Tempo entre disparos da base
% R2: Raio de acao da base
% nhist: Numero de historias simuladas
% passo: passo do tempo da simulacao

% 1a Avaliacao de AA-101
% lo Ten Eng Carlos Henrique MACHADO S Esteves
% Simulacao de Combate Aereo Simplificado
% Para ver o modo grafico, utilize nhist=1 e um passo pequeno, como 0.02.
% Com nhist>1 o modo grafico e suprimido e as estatisticas sao impressas.

% Comentarios
% Tempo discretizado com um passo definido. A cada iteracao o tempo e
incrementado e
% os eventos sao analisados.

% Calculo do numero medio de misseis para destruir o alvo foi feito de
maneira
%diferente: soma-se as quantidades de misseis lancados somente quando o
alvo
%foi destruido e no final divide-se pelo numero de vezes em que o alvo foi
destruido.

% No calculo da velocidade do missil da aeronave, o trabalho base soma a
% velocidade da aeronave ao valor retornado pela funcao U.

% O momento em que a aeronave solta o ultimo missil e volta tambem difere
% do trabalho base

%constantes
V = 600; %velocidade da aeronave
Dl = 2.9999; %intervalo entre disparos da aeronave. %nao compara direito
com o tempo se utilizar Dl=3. bug do matlab?
M = 10; %numero de misseis da aeronave
R1 = 20000; %raio de acao da aeronave
%D2 = 10; %intervalo entre disparos da defesa
N = 6; %numero de misseis da defesa
```

```

%R2 = 18000; %raio de acao da defesa
%passo = 0.01; %passo da simulacao
%nhist=1; %numero de historias

%estatisticas
gkas = 0; %probabilidade da base ser destruida e a aeronave sobreviver
akgs = 0; %probabilidade da aeronave ser destruida e a base sobreviver
boki = 0; %probabilidade de ambos serem destruidos
bosu = 0; %probabilidade de ambos sobreviverem
ga = 0; %numero medio de misseis para destruir a defesa (quando destroi)
gm = 0; %numero medio de misseis para destruir a aeronave (quando destroi)

rand('state',sum(100*clock));

for(j=1:nhist)
    %inicia variaveis
    t=0.0; %tempo da simulacao
    xaero = max([R1 R2]); %posicao da aeronave no eixo aeronave-base. (base
na origem)
    na = 0; %numero de misseis lancados pela aeronave
    nb = 0; %numero de misseis lancados pela base
    tprox=0.0; %tempo em que a aeronave lancara o proximo missil
    tproxb=0.0; %tempo em que a base lancara o proximo missil
    base=1; %base esta viva
    aeronave=1; %aeronave esta viva

    for(i=1:M) %misseis da aeronave
        Ma(i).condicao=0; %condicao do missil: 0 - nao lancado; 1 -
lancado; -1 - detonado;
        Ma(i).x = xaero; %posicao do missil no eixo aeronave-base;
        Ma(i).v = 0; %velocidade media do missil
        Ma(i).p = 0; %probabilidade do missil atingir o alvo
    end
    for(i=1:N) %misseis da base
        Mb(i).condicao=0; %condicao do missil: 0 - nao lancado; 1 -
lancado; -1 - detonado;
        Mb(i).x = 0; %posicao do missil no eixo aeronave-base;
        Mb(i).v = 0; %velocidade media do missil
        Mb(i).p = 0; %probabilidade do missil atingir o alvo
    end
    terminou = 0;
    while(terminou==0)
        %atualiza misseis da aeronave
        if(na>0)
            for(i=1:na)
                if(Ma(i).condicao==1) %se o missil foi lancado
                    Ma(i).x=Ma(i).x-Ma(i).v*passo;
                    if(Ma(i).x<=0) %chegou na base
                        Ma(i).condicao=-1; %missil detonado
                        if(nhist==1) disp('Missil da aeronave chegou');
                    end
                    if(rand<Ma(i).p) %destruiu base
                        base=0;
                        ga=ga+na; %conta a media do numero de misseis
lancados ate destruir a base
                    end
                    if(nhist==1) disp('Base destruida'); end
                end
            end
        end
    end
end
end
end

```



```

end
%atualiza misseis da base
if(nb>0)
    for(i=1:nb)
        if(Mb(i).condicao==1) %se o missil foi lancado
            Mb(i).x=Mb(i).x+Mb(i).v*passo;
            if(Mb(i).x>=xaero) %chegou na aeronave
                Mb(i).condicao=-1; %missil detonado
                if(nhist==1) disp('Missil da base chegou');
disp(t); end
                %Mb(i).p=P2(xaero); % !!! para obter os resultados
de Shannon !!!
                if(rand<Mb(i).p) %destruiu aeronave
                    aeronave=0;
                    %cometendo um erro: esquecendo os misseis da
                    %aeronave quando ela e destruida
                    %for(k=1:M)
                    %     if(Ma(k).condicao==1) pmissil=k; break;
end
                    %end
                    %for(k=pmissil+1:M) Ma(k).condicao=-1; end
                    gm=gm+nb; %conta a media numero de misseis
lancados ate destruir a aeronave
                    if(nhist==1) disp('Aeronave destruida'); end
                end
            end
        end
    end
end

%verifica se a aeronave lanca missil
if(aeronave==1&&base==1&&xaero<=R1&&na<M&&t>=tprox)
    na=na+1;
    Ma(na).condicao=1;
    Ma(na).x=xaero;
    Ma(na).v=U1(xaero,V);
    Ma(na).p=P1(xaero);
    tprox=t+D1; %atualiza tempo do proximo disparo
    if(nhist==1) disp('Aeronave disparou missil'); disp(t); end

    %ga=ga+1; %contador de misseis da aeronave
end
%verifica se a base lanca missil
if(base==1&&aeronave==1&&xaero<=R2&&nb<N&&tprox<=t)
    nb=nb+1;
    Mb(nb).condicao=1;
    Mb(nb).x=0;
    Mb(nb).v=U2(xaero,V,na);
    Mb(nb).p=P2(xaero);
    tprox=t+D2; %atualiza tempo do proximo disparo
    if(nhist==1) disp('Base disparou missil'); disp(t); end
    %gm=gm+1; %contador de misseis da base
end

%atualiza posicao da aeronave
if (na<M) xaero=xaero-V*passo; %aeronave se aproximando da base
else xaero=xaero+V*passo; %aeronave se afastando da base
end

%testa se a simulacao terminou

```

```

ativoa = 0;
for(i=1:M)
    if(Ma(i).condicao==1) ativoa=1; end
end
ativob=0;
for(i=1:N)
    if(Mb(i).condicao==1) ativob=1; end
end
if(base==0&&ativob==0) terminou=1; end %base foi destruida e nao
existem mais misseis da base lancados
if(aeronave==0&&ativoa==0) terminou=1; end %aeronave foi destruida
e nao existem mais misseis da aeronave lancados
if(na==M&&xaero>R2&&ativoa==0) terminou=1; end %aeronave lancou
todos os misseis e escapou do raio de acao da base
t=t+passo;

%mostra animacao na tela (somente se nhist = 1)
if(nhist==1)
    figure(1);
    for(i=1:M)
        if(Ma(i).condicao==1) misseisa(i)=Ma(i).x; else
misseisa(i)=-10000; end
    end
    for(i=1:N)
        if(Mb(i).condicao==1) misseisb(i)=Mb(i).x; else
misseisb(i)=-10000; end
    end
    if (base==0) strbase='rx'; else strbase='rh'; end
    if (aeronave==0) straero='bx'; else straero='bp'; end

plot(0,0,strbase,xaero,0,straero,misseisa,0,'b<',misseisb,0,'r>', 'EraseMode
','none');
    axis([-1000 25000 -10 10]);
    drawnow;
end
end
if(aeronave==1&&base==0) gkas=gkas+1; end
if(aeronave==1&&base==1) bosu=bosu+1; end
if(aeronave==0&&base==1) akgs=akgs+1; end
if(aeronave==0&&base==0) boki=boki+1; end
end

if(nhist>1)
    disp('Base destruida, aeronave sobrevive (GKAS)');
disp(100*gkas/nhist);
    disp('Aeronave destruida, base sobrevive (AKGS)');
disp(100*akgs/nhist);
    disp('Ambos destruidos (BOKI)'); disp(100*boki/nhist);
    disp('Ambos sobrevivem (BOSU)'); disp(100*bosu/nhist);
    disp('Numero medio de misseis para destruir a base (GA)');
disp(ga/(gkas+boki));
    disp('Numero medio de misseis para destruir a aeronave (GM)');
disp(gm/(akgs+boki));
end

function u = U1(r,V)
%velocidade media do missil da aeronave
u=2000+0.2*r-0.00000833*r*r+V; %trabalho de Vieira&Prati e de Shannon soma
V do aviao
function u = U2(r,V,na)

```

```

%velocidade media do missil da base
%u=2200+0.2*r-0.00000833*r*r %considerando a velocidade absoluta
%considerando velocidade relativa e distancia em relacao ao ponto que
%atinge a aeronave:
fl=0;
if(na~=10)
    for(k=1:10000)
        rc=r-V*k/100;
        u=2200+0.2*rc-0.00000833*rc*rc+V;
        if(abs(((r-rc)/u)-(k/100))<0.01) fl=1; break; end
    end
else
    for(k=1:10000)
        rc=r+V*k/100;
        u=2200+0.2*rc-0.00000833*rc*rc-V;
        if(abs(((rc-r)/u)-(k/100))<0.01) fl=1; break; end
    end
if(fl==0) disp('ERRO! Nao encontrou velocidade para o missil da base'); end
end
function p = P1(r)
%probabilidade do missil da aeronave acertar o alvo
if (r<=5000) p=0.6;
else p=0.764-0.000033*r;
end
function p = P2(r)
%probabilidade do missil da base acertar o alvo
if (r<=8000) p=0.7;
else p=1.1-0.00005*r;
end
end

```

APÊNDICE B: Programa SciLab [2] Corrigido

```

//***  variáveis significativas:
//***
//***      nhist - número de histórias a serem executadas(rodadas)
//***      tc -      tempo relativo no qual a aeronave dispara seu
último míssil
//***      r(t) -      distância da aeronave a partir da bateria antiaérea
a qualquer tempo relativo t (função)
//***      e(i) -      tempo no qual o i-ésimo míssil-ar é disparado
//***      f(i) -      tempo no qual o i-ésimo míssil-ar atinge o alvo
//***      g(j) -      tempo no qual o j-ésimo míssil-terra é disparado
//***      h(j) -      tempo no qual o j-ésimo míssil-terra atinge o alvo
//***      tr -      tempo durante o qual a aeronave está dentro for
alcance limiar da bateria antiaérea
//***      l -      número máximo de mísseis disparafors pela bateria
antiaérea.
//***      gkas - porcentagem de sobrevivência da aeronave com a
destruição da bateria antiaérea.
//***      akgs - porcentagem de sobrevivência da bateria antiaérea com a
destruição da aeronave.
//***      boki - ambos destruídos
//***      bosu - ambos sobreviventes
//***      ga -      mísseis da aeronave por destruição
//***      gm -      mísseis da bateria anti-aérea por destruição.
//***
global r1 r2 v tc
function yul=u1(x)

```

```

        yu1=2000.+0.2*x-0.00000833*x**2;
    endfunction
    function yu2=u2(x)
        yu2=2200.+0.2*x-0.00000833*x**2;
    endfunction
    function yp1=p1(x)
        if(x <= 5000.) then
            yp1=0.6;
        else
            yp1=0.764-0.000033*x;
        end
    endfunction
    function yp2=p2(x)
        if(x <= 8000.) then
            yp2=0.7;
        else
            yp2=1.1-0.00005*x;
        end
    endfunction
    function yr=r(t)
//distância entre a aeronave e a bateria anti-aérea
        global r1 r2 v tc
        if(r1 > r2) then
            yr=r1-v*t;
            if(t > tc) then yr=r1-2.*v*tc+v*t; end
        else
            yr=r2-v*t;
        if(t > tc) then yr=r2-2.*v*tc+v*t; end
        end
    endfunction
// Início
//rand('seed',0)
v=600.,m=10,d1=3.,r1=20000.,n=6,d2=10.,r2=18000.
nhist=10000;
// inicializa a solução do problema
// número máximo de mísseis disparados pela bateria anti-aérea
for k=1:12
    e(k)=1000.;
    f(k)=1000.;
    g(k)=1000.;
    h(k)=1000.;
end
tr=2.*((r2-r1)/v+(m-1)*d1);
a=tr/d2;
l=int(a)+1;
if(l > n) l=n; end
//
// tempo relativo no qual a aeronave dispara seu último míssil
if(r1 > r2) tc=(m-1)*d1; end
if(r2 >= r1) tc=(r2-r1)/v+(m-1)*d1; end
//
// quando cada um dos mísseis-ar é disparado e(i)
// quando cada um dos mísseis-ar atinge o alvo f(i)
if(r1 > r2) then
    for i=1:m
        e(i)=(i-1)*d1;
        yr=r(e(i));
        yul=ul(yr);
        f(i)=e(i)+yr/(yul+v);
    end
end

```

```

else
for i=1:m
e(i)=(r2-r1)/v+(i-1)*d1;
yr=r(e(i));
yul=u1(yr);
f(i)=e(i)+yr/(yul+v);
end
end
//
// quando cada um dos mísseis-terra é disparado g(j)
if(r1 > r2) then
for j=1:l
g(j)=(r1-r2)/v+(j-1)*d2;
end
else
for j=1:l
g(j)=(j-1)*d2;
end
end
//
// quando cada um dos mísseis-terra atingem a aeronave h(j)
// g(j) > tc
// técnica de secante para resolver h(j) caso ii
for j=1:l
if(g(j) > tc) then
p0=tc;
pn=3*tc;
for i=1:1000
q0=p0-g(j)-r(p0)/(u2(r(p0))-v);
q1=pn-g(j)-r(pn)/(u2(r(pn))-v);
p=pn-q1*(pn-p0)/(q1-q0);
tol=p-pn;
if(abs(tol)<0.0001) break
end
p0=pn;
pn=p;
end
h(j)=p;
else
// g(j)<tc
// técnica da secante para resolver h(j) caso i
p0=0.;
pn=tc;
for i=1:100
q0=p0-g(j)-r(p0)/(u2(r(p0))+v);
q1=pn-g(j)-r(pn)/(u2(r(pn))+v);
p=pn-q1*(pn-p0)/(q1-q0);
tol=p-pn;
if(abs(tol)<0.0001) break end
p0=pn;
pn=p;
end
hcase1=p;
// g(j)<tc
// técnica da secante para resolver h(j) caso iii
p0=tc;
pn=3*tc;
for i=1:100
q0=p0-tc-(r(p0)-(r1-(m-1)*v*d1))/(u2(r(p0))-v);
q1=pn-tc-(r(pn)-(r1-(m-1)*v*d1))/(u2(r(pn))-v);

```

```

    p=pn-q1*(pn-p0)/(q1-q0);
    tol=p-pn;
    if(abs(tol)<0.0001) break end
    p0=pn;
    pn=p;
end
hcase3=p;
    if(hcase1 < hcase3) then
        h(j)=hcase1;
    else
        h(j)=hcase3;
    end
end
end
for k=1:12
    range1(k)=r(e(k));
    pp1(k)=p1(range1(k));
//    range2(k)=r(g(k)); // erro encontrado
    range2(k)=r(h(k));
    pp2(k)=p2(range2(k));
end
nhist
//    e,f,g,h,pp1, pp2
//*** a subrotina gotcha realiza a simulação de monte carlo
    nam=0;ngkas=0;nakgs=0;ii=0;
    ngm=0;nboki=0;nbosu=0;jj=0;
for nh=1:nhist
    tamax=2*tr;
    tgmax=2*tr;
    time=-0.2;
    nak=0;
    ngk=0;
        while (time < h(1))
            time=time+0.1;
            for i=1:m
                xa=e(i)-time;
                if(xa >= 0.) break end
            end
            for j=1:l
                xg=g(j)-time;
                if(xg >= 0.) break end
            end
            prox=e(i)-g(j);
            if(i==m & j==1) then time = h(1); continue end
        if(i==m) then prox= 1; end
        if(j==1) then prox=-1; end
        if(prox < 0) then
            time=e(i);
        if(e(i) >= tamax | e(i) > tgmax) continue end
            if(ngk > 0) continue end
                rnd=rand();
                if(rnd > pp1(i)) continue end
                ii=i;
                ngk=1;
                tgmax=f(i);
        else
            time=g(j);
        if(g(j) >= tamax | g(j) > tgmax) continue end
            if(nak > 0) continue end
                rnd=rand();

```

```
        if(rnd > pp2(j)) continue end
        jj=j;
            nak=1;
            tamax=h(j);
        end
    end //for while
    ngm=ngm+jj;
    nam=nam+ii;
    if(nak == 0 & ngk == 1) ngkas=ngkas+1; end
    if(nak == 1 & ngk == 0) nakgs=nakgs+1; end
    if(nak == 1 & ngk == 1) nboki=nboki+1; end
    if(nak == 0 & ngk == 0) nbosu=nbosu+1; end
end
gm= ngm/nhist
ga= nam/nhist
gkas= ngkas/nhist*100.
akgs= nakgs/nhist*100.
boki= nboki/nhist*100.
bosu= nbosu/nhist*100.
```