《系统仿真与 matlab》综合试题

题 目：战争模型

编 号：30

姓 名 罗荣胜

班 级 自动化校际交流 1802 班

学 号 U201812856

联系方式 18272051305

成 绩

系统建模与仿真研究报告

罗荣胜　　自动化校际交流 1802 班　U201812856

2020.12.15

一．战争问题模型介绍

影响一个军队战斗力的因素是多方面的，比如士兵人数、单个士兵的作战素质以及部队的军事装备，而具体到一次战争的胜负，部队采取的作战方式同样至关重要，此时作战空间同样成为讨论一个作战部队整体战斗力的一个不可忽略的因素。通过建立战争模型，以预测一场战争的大致结局。

模型假设

1. $x(t)$、$y(t)$ 表示甲乙交战双方在时刻 t 的兵力，视为双方的士兵人数。

2. 战斗减员率，它取决于双方的兵力，以 $f(x,y)$、$g(x,y)$ 分别表示甲乙双方的战斗减员率；

3. 非战斗减员率，比如由于疾病或逃跑等因素导致一个部队减员，它通常可被设与本方的兵力成正比，比例系数 $\alpha,\beta>0$ 分别对应甲乙双方；

4. 增援率，通常取决于一个已投入战争部队以外的因素，甲乙双方的增援率函数分别以 $u(t),v(t)$ 表示。

5. r_x、r_y 为甲乙双方单个士兵的射击率，通常主要取决于部队的武器装备。

6. p_x、p_y 表示甲乙双方士兵一次射击的（平均）命中率

7. S_x,S_y 分别表示甲乙双方的有效活动区域的面积

8. 正规战情形：甲乙双方均以正规部队作战，每一方士兵的活动均公开，处于对方士兵的监视与杀伤范围之内，一旦一方的某个士兵被杀伤，对方的火力立即转移到其他士兵身上。

9. 游击战情形：甲乙双方均以游击作战方式，每一方士兵的活动均具有隐蔽性，对方的射击行为局限在某个范围考虑可以被认为是盲目的。

10. 混合战情形：正规战和游击战两种情形的混合。

仿真要求

根据上述假设进行系统建模与仿真，分别模拟正规战情形，游击战情形和混合战情形，输入参数为假设 1-7，输出为交战双方人数的变化曲线，给出双方获胜的条件。要求有输入、输出界面，仿真过程。

二．试题建模过程

1.题目理解及准备工作

根据平时所学的数学建模知识与课上所讲的建模方法，很容易理解题目的要求。即分别建立三种不同的模型：正规战，游击战，混合战，以时间为线，查看不同参数下，甲乙双方人数的变化情况，并在每一种模型下，给出双方获胜的条件。当我们从界面输入相应的参数后，输出为双方人数的变化情况。

由于我参加过全国大学生数学建模比赛，所以在之前就对 matlab 的使用有一定的了解，对于老师上课讲的实例，我课下也都去跑过一遍，并且自己写了些代码，所以，对 matlab 和 GUI 的使用比较熟悉。

2.模型建立

1.模型中的参数含义

仿真时间	T
甲方初始人数	x_0
甲方t时刻的人数	$x(t)$
甲方战斗减员率	$f(x,y)$
甲方非战斗减员率	α
甲方增援率	$u(t)$
甲方单个士兵的射击率	r_x
甲方士兵一次射击的平均命中率	p_x
甲方有效活动区域的面积	s_x
甲方一次射击的有效面积	s_{rx}
乙方初始人数	y_0
乙方t时刻的人数	$y(t)$

乙方战斗减员率	$g(x,y)$
乙方非战斗减员率	β
乙方增援率	$v(t)$
乙方单个士兵的射击率	r_y
乙方士兵一次射击的平均命中率	p_y
乙方有效活动区域的面积	s_y
乙方一次射击的有效面积	s_{ry}

2.三种不同战争情形模型的数学推导

正规战争模型：

模型假设：

（i）双方士兵公开活动。甲方士兵的战斗减员仅与乙方士兵人数有关。记双方士兵人数分别为$x(t),y(t)$ ，则甲方士兵战斗减员率 $f(x,y)$可以表示为$ay(t)$，a表示乙方每个士兵的杀伤率。可知$a=r_y p_y$，r_y为乙方士兵的射击率（每个士兵单位时间的射击次数），p_y为每次射击的命中率。同理，用b表示甲方士兵对乙方士兵的杀伤率，即$b=r_x p_x$

（ii）双方的非战斗减员率仅与本方兵力成正比。减员率系数分别为α,β。

（iii）设双方的兵力增援率为$u(t),v(t)$。

模型与求解：由假设可知：

$$\begin{cases}\dfrac{dx(t)}{dt}=-f(x,y)-\alpha x(t)+u(t)=-r_y p_y y(t)-\alpha x(t)+u(t)\\ \dfrac{dy(t)}{dt}=-g(x,y)-\beta y(t)+v(t)=-r_x p_x x(t)-\beta y(t)+v(t)\\ x(0)=x_0,y(0)=y_0\end{cases}$$

游击战争模型：

（i） 乙方士兵看不见甲方士兵，甲方士兵在某个面积为s_x的区域内活

动。乙方士兵不是向甲方士兵射击，而是向该区域射击。此时，甲方士兵的战斗减员不仅与乙方兵力有关，而且随着甲方兵力增加而增加。因为在一个有限区域内，士兵人数越多，被杀伤的可能性越大。可设，甲方的战斗减员率为cxy，其中c为乙方战斗效果系数，$c=r_y\frac{s_{ry}}{s_x}$，其中r_y仍为射击率，命中率为乙方一次射击的有效面积s_{ry}与甲方活动面积s_x之比。

（ii） 双方的非战斗减员率仅与本方兵力成正比。减员率系数分别为α,β。

（iii） 设双方的兵力增援率为$u(t),v(t)$。

模型与求解：由假设可知：

$$\begin{cases}\dfrac{dx(t)}{dt}=-cx(t)y(t)-\alpha x(t)+u(t)=-r_y\dfrac{s_{ry}}{s_x}x(t)y(t)-\alpha x(t)+u(t)\\ \dfrac{dy(t)}{dt}=-dx(t)y(t)-\beta y(t)+v(t)=-r_x\dfrac{s_{rx}}{s_y}x(t)y(t)-\beta y(t)+v(t)\\ x(0)=x_0,y(0)=y_0\end{cases}$$

混合站模型：

模型假设

（i） 甲方为游击队，乙方为正规部队

（ii） 甲方的战斗减员率$f(x,y)$可以表示为cxy，其中c为乙方战斗效果系数，$c=r_y\frac{s_{ry}}{s_x}$，其中r_y仍为射击率，命中率为乙方一次射击的有效面积s_{ry}与甲方活动面积s_x之比。

（iii） 乙方士兵战斗减员率可以表示为$bx(t)$，b表示甲方每个士兵的杀伤率。可知$b=r_xp_x$，r_x为甲方士兵的射击率（每个士兵单位时间的射击次数），p_x为每次射击的命中率。

（iv） 双方的非战斗减员率仅与本方兵力成正比。减员率系数分别为α,β。

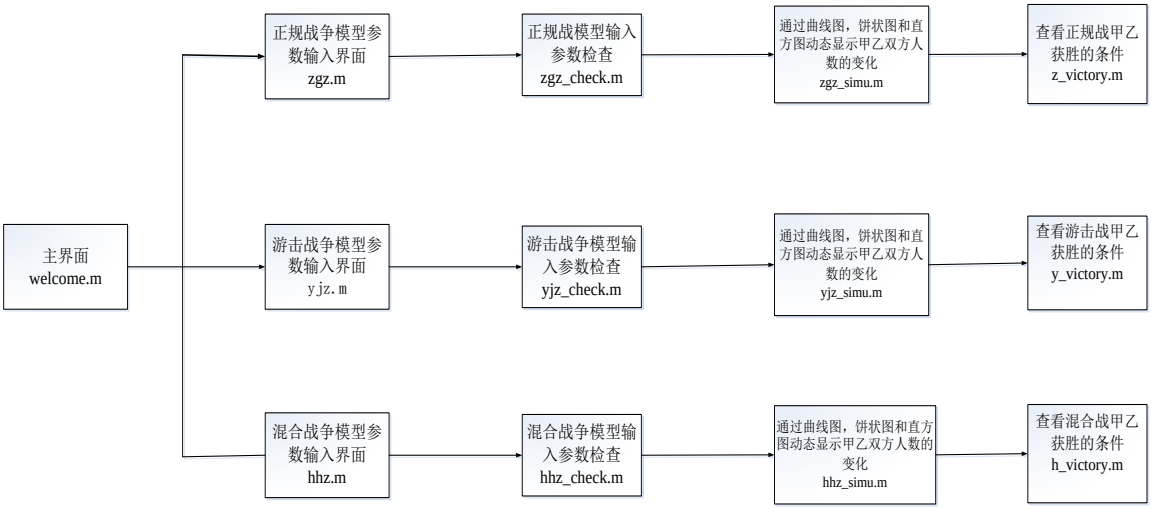
（v） 设双方的兵力增援率为$u(t),v(t)$。

模型与求解：由假设可知：

$$\begin{cases} \dfrac{dx(t)}{dt} = -cx(t)y(t) - \alpha x(t) + u(t) \\ \dfrac{dy(t)}{dt} = -bx(t) - \beta y(t) + v(t) \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

求解方式 以上三个模型中的方程组，对于第一个模型可以使用常微分方程组的的数学理论求出理论解，也可以采用数值分析中的迭代法求解微分方程的数值解，但对于后两个模型，由于含有 $x(t)y(t)$ 交叉项，用常微分方程组的的数学理论很难得出解析解，所以统一采用龙格库塔法求数值解。

三．系统仿真流程



四．系统仿真关键点

1.Matlab 仿真界面的设计

用 matlab 进行界面设计由两种方法：第一种是使用 unicontorl 手动用代码添加各种需要的界面操作工具，第二种是使用 GUIDE 工具进行界面设计，通过拖动需要的控件，并修改对应的 Tag 值，matlab 会自动生成对应的 m 文件，我们需要对各个控件的回调函数进行补充，以达到参数检查，界面跳转等目的。同时使用 get 函数对 handles 结构体中的属性进行读操作，用 set 函数对 handles 结构体中的属性进行修改操作。

2.微分方程组的处理

根据模型建立起微分方程组，采用 matlab 里面的 ode45 函数求得数值解，代码如下：

正规战争模型：（游击战和混合战类似，只是 odefunc 的函数不同）

```
tspan=[0,time];
x0=[jia_x0,yi_x0];
[t,y]=ode45('zgz_odefunc',tspan,x0);
```

```
function dx = zgz_odefunc(t, x)
global jia_r jia_p jia_alpha jia_yuan  yi_r yi_p yi_alpha yi_yuan
dx = zeros(2, 1);
try
    dx(1,1)=-yi_r*yi_p*x(2)-jia_alpha*x(1)+jia_yuan;
catch
     dx(1,1)=-yi_r*yi_p*x(2)-jia_alpha*x(1)+jia_yuan(t);
end

try
    dx(2,1)=-jia_r*jia_p*x(1)-yi_alpha*x(2)+yi_yuan;

catch
    dx(2,1)=-jia_r*jia_p*x(1)-yi_alpha*x(2)+yi_yuan(t);
end
end
```

3.仿真参数的输入

正规战争：需要输入的参数包括仿真时间T，甲方的初始人数x_0，甲方单个士兵的射击率r_x，甲方士兵一次射击的平均命中率p_x，甲方的增援率$u(t)$，甲方的非战斗减员α，乙方的初始人数y_0，乙方单个士兵的射击率r_y，乙方士兵一次射击的平均命中率p_y，乙方的增援率$v(t)$，乙方的非战斗减员β。

游击战争：需要输入的参数包括仿真时间T，甲方的初始人数x_0，甲方单个士兵的射击率r_x，甲方士兵一次射击的有效面积s_{rx}，甲方的有效活动区域的面积s_x，甲方的增援率$u(t)$，甲方的非战斗减员α，乙方的初始人数y_0，乙方单个士兵的射击率r_y，乙方士兵一次射击的有效面积s_{ry}，乙方的有效活动区域的面积s_y，

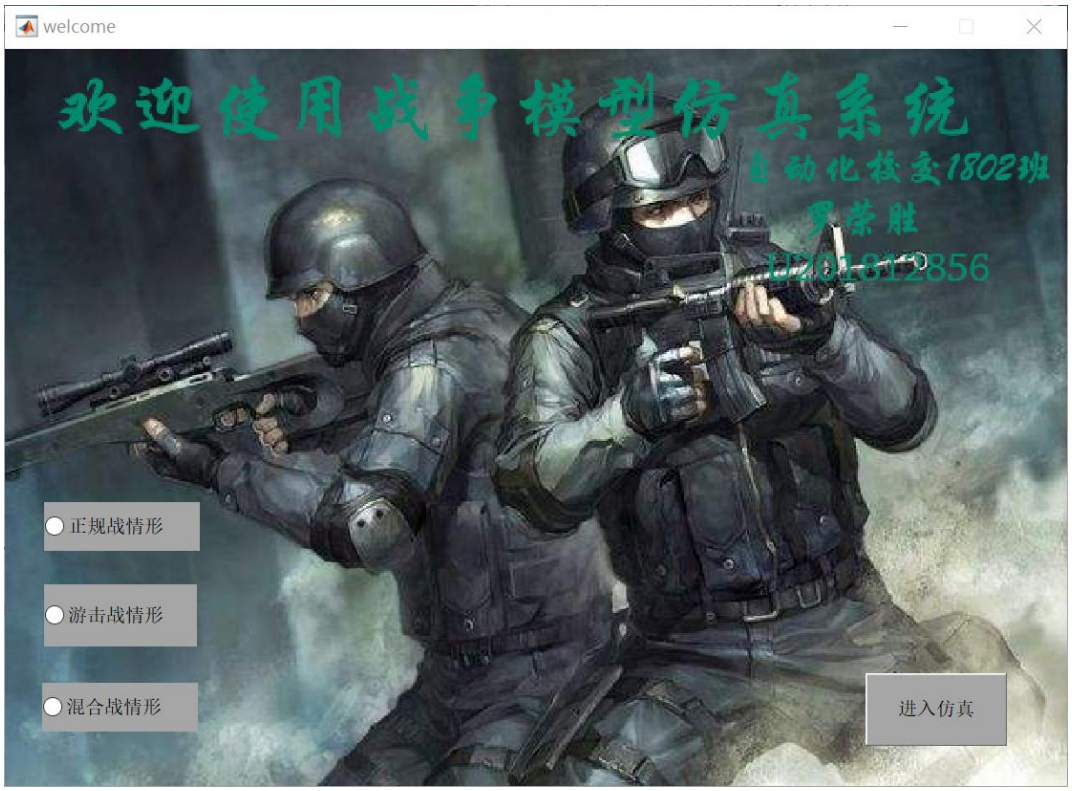
乙方的增援率$v(t)$，乙方的非战斗减员β。

混合战争：需要输入的参数包括仿真时间T，甲方的初始人数x_0，甲方单个士兵的射击率r_x，甲方士兵一次射击的平均命中率p_x，甲方的有效区域活动的面积s_x，甲方的增援率$u(t)$，甲方的非战斗减员α，乙方的初始人数y_0，乙方单个士兵的射击率r_y，乙方士兵一次射击的有效面积s_{ry}，乙方的增援率$v(t)$，乙方的非战斗减员β。

通过 GUIDE 界面设计的控件——可编辑文本框，用户把参数输入文本框中，可以用 handles.Tag 的形式，对可编辑文本框中的参数读出来，从而实现了通过键盘输入完成参数的输入。

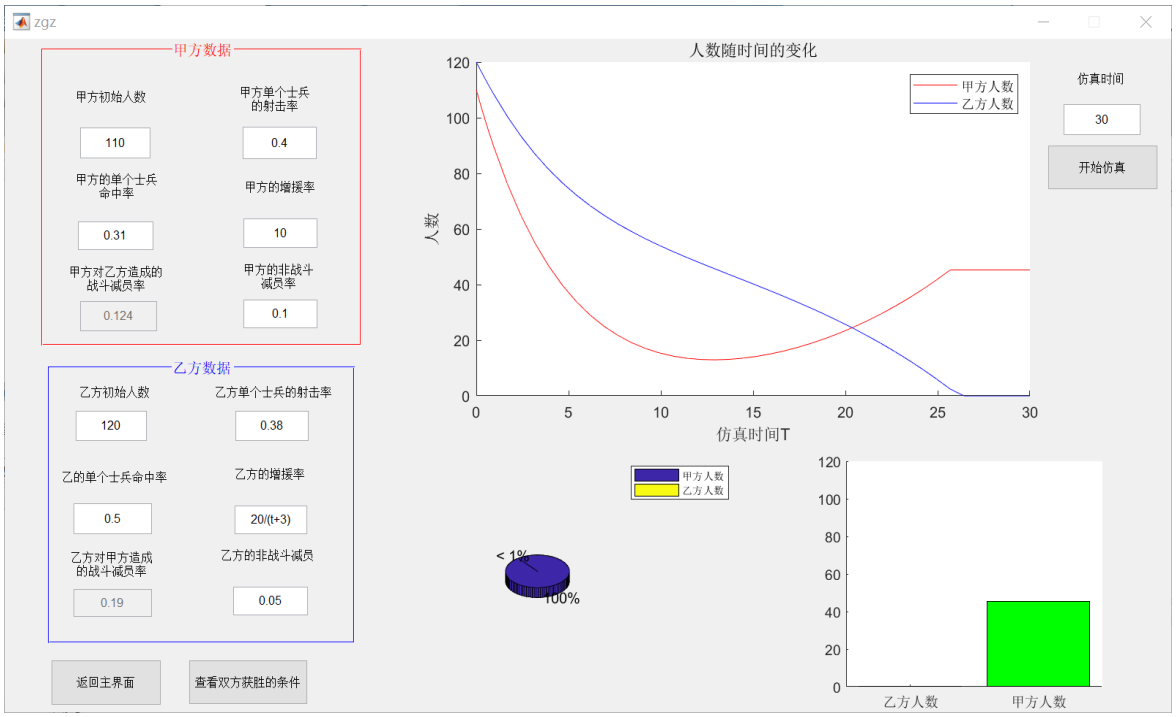
4.战争模型的选择

当用户进入仿真系统后，可以选择进入正规战争仿真，游击战争仿真，混合战争仿真，三个其中的一个，然后用户会进入对应的界面，输入需要的参数，就可以得出相应的人数变化情况。



5.仿真结果的表示

当用户输入符合要求的参数后，点击开始仿真，甲乙双方的人数变化情况通过曲线图，饼状图和直方图分别表示出来，使得结果显而易见，而且整个过程是动态显示的，使得仿真更具有真实性。



五 . 程序运行指南

1.关于文件及内容

该战争模型仿真系统一共包含 18 个 m 文件，4 个 fig 文件，7 张图片。
其中 18 个 m 文件及功能简述如下：
welcome.m:系统仿真的入口，运行后进入欢迎界面，同时，用户需要在三种模型中选出想要仿真的模型。
zgz.m：正规战争模型仿真的主界面，包括系统参数的输入，返回主界面，查看正规战争双方获胜的条件。
zgz_check.m：检查正规战争模型中各个参数输入是否合法，如输入为空，或者小于 0，都会给出 error 界面提示用户。
zgz_simu.m：开始正规战争的仿真，以曲线图，直方图，饼状图三种形式动态显示甲乙双方人数的变化。
zgz_odefunc.m：正规战争模型中建立的微分方程组，动态仿真的时候需要求其数值解。
z_victory.m：显示正规战争双方获胜的条件与输入参数之间的关系。

yjz.m：游击战争模型仿真的主界面，包括系统参数的输入，返回主界面，查看游击战争双方获胜的条件。
yjz_check.m：检查游击战争模型中各个参数输入是否合法，如输入为空，或者小于 0，都会给出 error 界面提示用户。
yjz _simu.m：开始游击战争的仿真，以曲线图，直方图，饼状图三种形式动态显示甲乙双方人数的变化。
yjz _odefunc.m：游击战争模型中建立的微分方程组，动态仿真的时候需要求其数值解。
y_victory.m：显示游击战争双方获胜的条件与输入参数之间的关系。
hhz.m：混合战争模型仿真的主界面，包括系统参数的输入，返回主界面，查看混合战争双方获胜的条件。
hhz _check.m：检查混合战争模型中各个参数输入是否合法，如输入为空，或者小于 0，都会给出 error 界面提示用户。
hhz _simu.m：开始混合战争的仿真，以曲线图，直方图，饼状图三种形式动态显示甲乙双方人数的变化。
hhz _odefunc.m：混合战争模型中建立的微分方程组，动态仿真的时候需要求其数值解。
h_victory.m：显示混合战争双方获胜的条件与输入参数之间的关系。

2.系统简介

在命令行窗口输入 welcome 或者运行 welcome.m 文件开始运行程序，进入欢迎界面,在欢迎界面可以看到作者信息。在左下角三种仿真模型中选择一个后，点击右下角的进入仿真，可以进入对应模型的仿真界面。

进入对应模型的仿真主界面后，需要输入甲乙方的参数信息，参数必须要合理，不然会报错，而且参数不能为空，然后点击开始仿真，则开始仿真过程。

六．系统模型的亮点

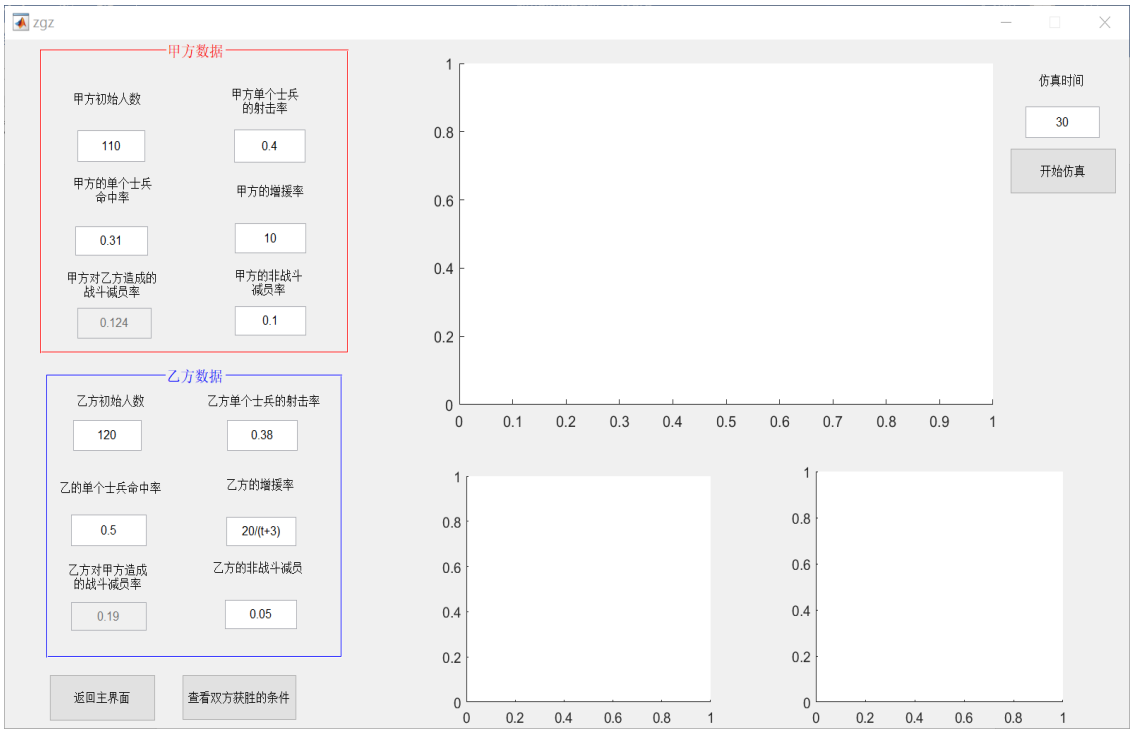
1.题目中在对游击战模型进行仿真要求时，没有考虑甲乙双方射击的有效面积，但在游击战实际过程中，双方都是朝着一个有效面积范围内瞄准射击，所以我认为需要考虑射击的有效面积这个参数，在建模和仿真的时候，我也考虑了这个因素，使得整个模型更完整。
2.为了能够更清楚,更生动地展现甲乙双方人数的变化,我通过曲线图,饼状图,直方图，三种方式，并用动态显示的方式显示人数实时变化。
3.在甲乙双方增援率的文本框中，用户可以输入关于时间 t 的函数，使得仿真情况更符合真实的战争情形。
4.在各个模型进行仿真之前，都会对输入的参数进行检查，只有当所有的参数都正确且符合要求后，仿真才会正式进行，否则系统会报错，并弹出对话框给出提示。
5.这个仿真系统设计多个界面之间的来回切换,包括从模型仿真的界面返回主界面，选择下一个想要仿真的模型，在仿真界面查看该模型对应的甲乙双方获胜的条件，需要考虑参数传递和界面之间来回切换的稳定性。

6. 所有代码和界面的设计都是自己独立完成。

七．程序运行实例分析

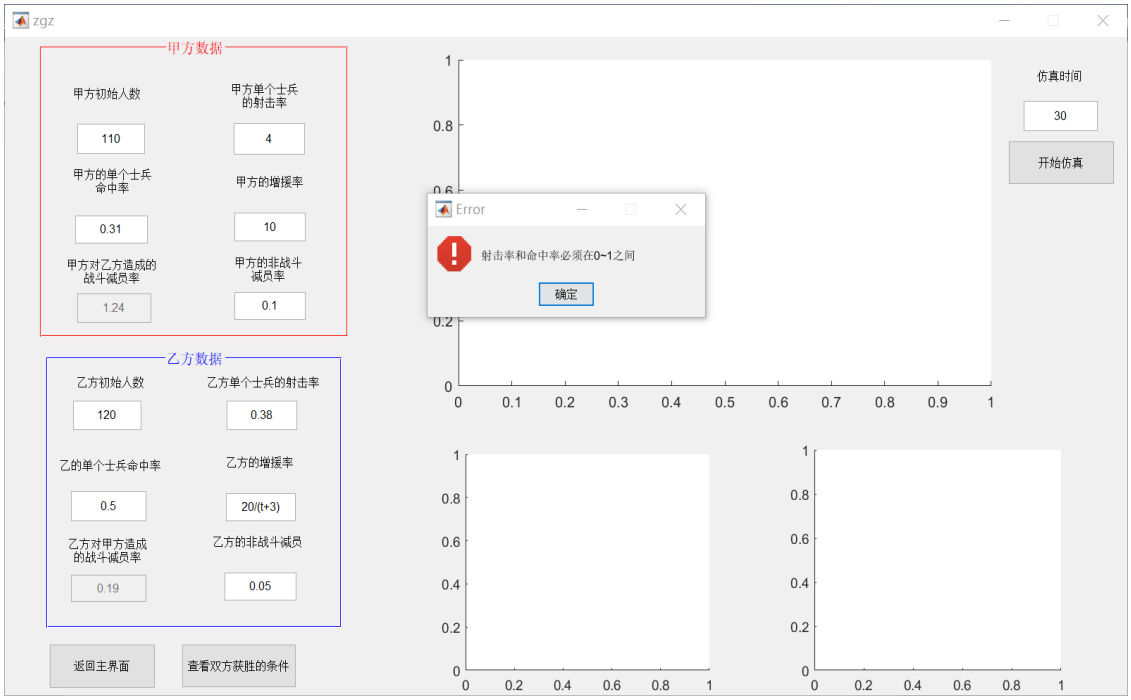
下面介绍仿真程序运行过程，并截取仿真过程中的部分图片。
在命令行窗口输入 welcome 或运行 welcome.m 文件开始运行程序，进入欢迎界面。

如用户选择正规战争情形的仿真，点击右下角的进入仿真，可以进入正规战争的仿真界面，并自动关闭欢迎的主界面。

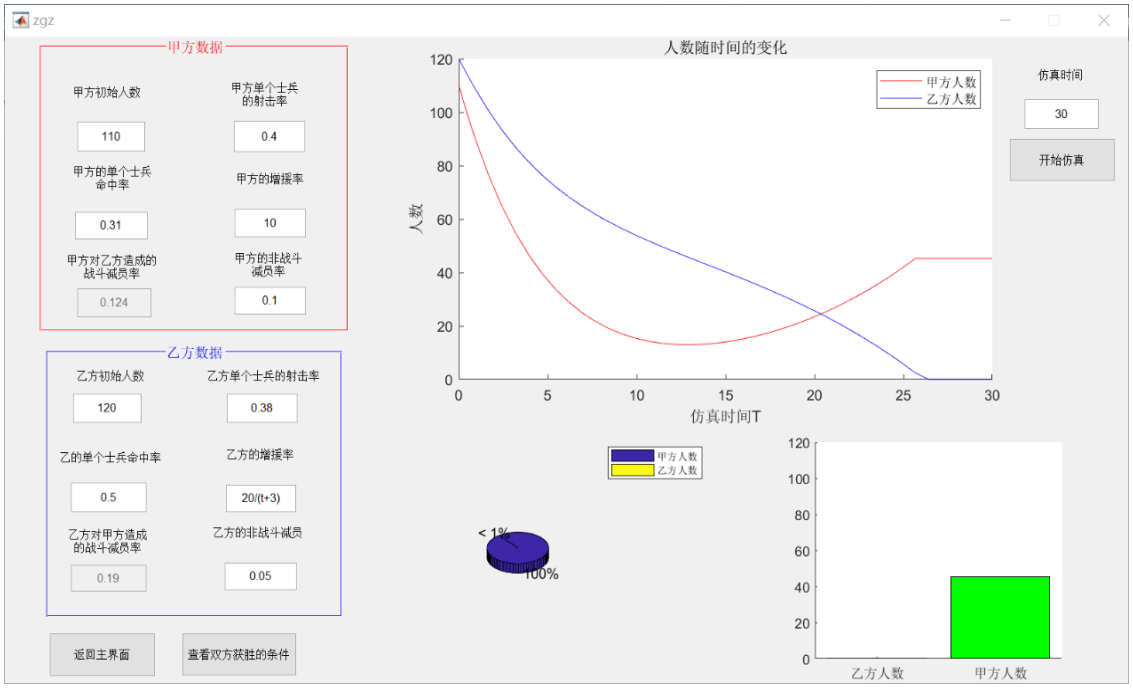


输入合理的参数，如果参数不合理，则会弹出提醒的对话框（以下举例说明）。当输入的某个参数为负数时，例如输入甲方的增援率为负数，则会弹出对话框“参数不可为负数”。

当输入的射击率或者命中率大于 1 时，会弹出对话框，提醒输入的射击率或者命中率需要在 0~1 之间。

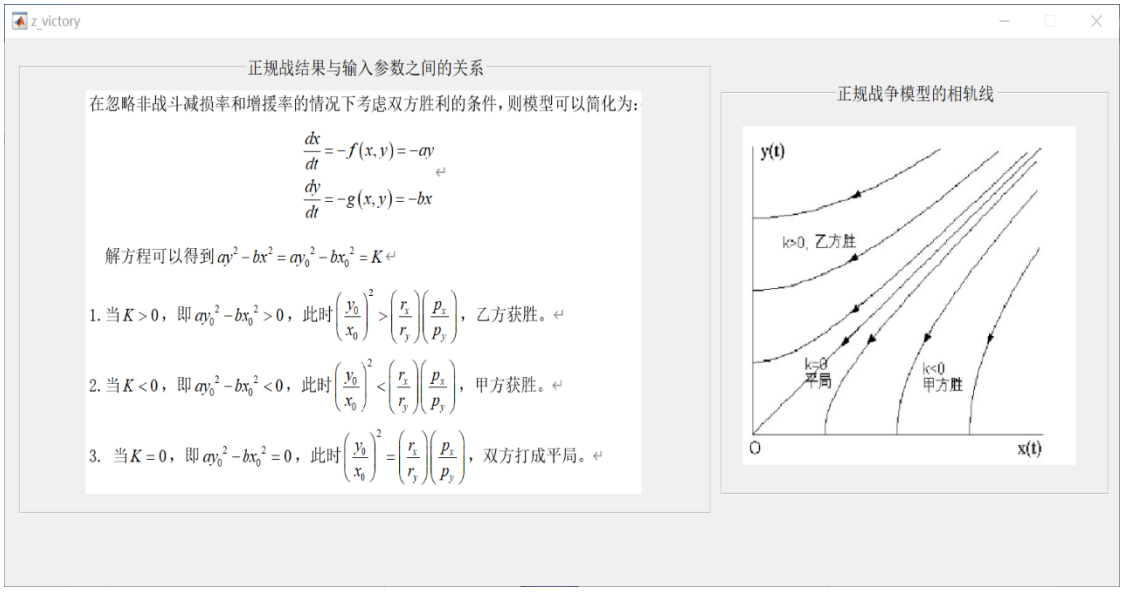


在所有的参数输入均符合要求的条件下，点击开始仿真按钮。

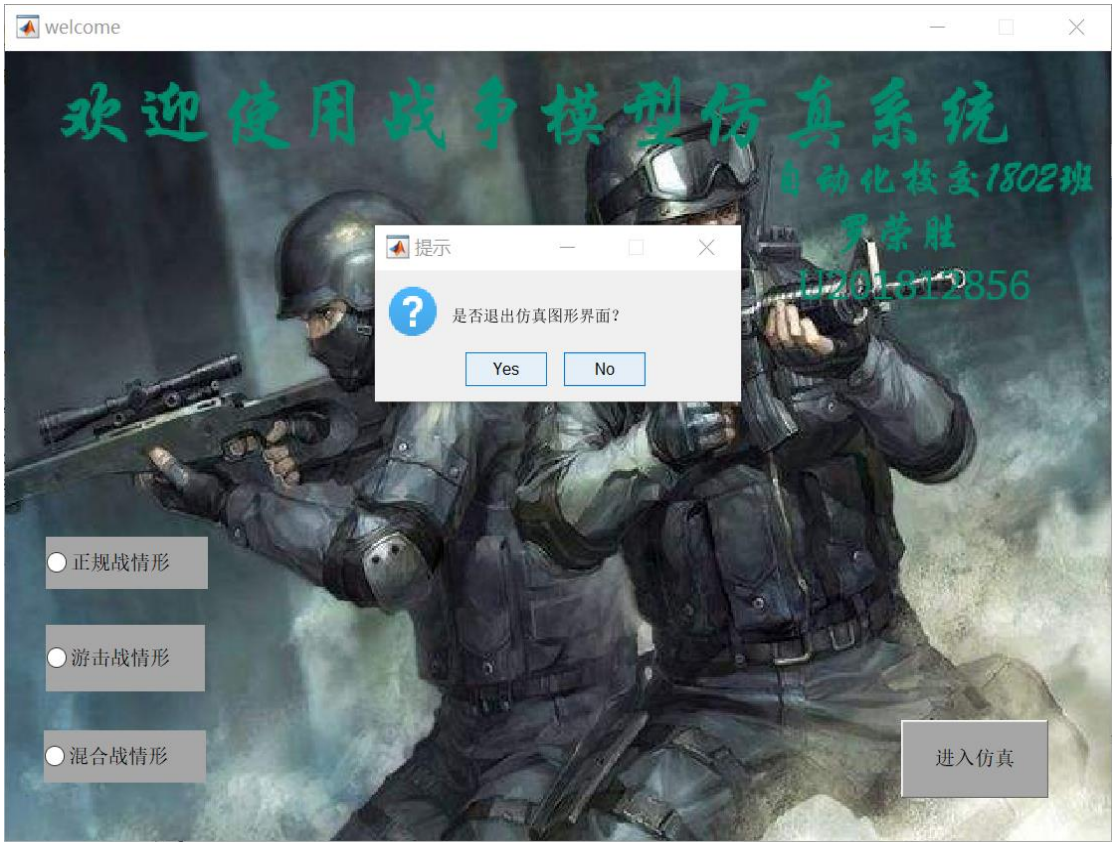


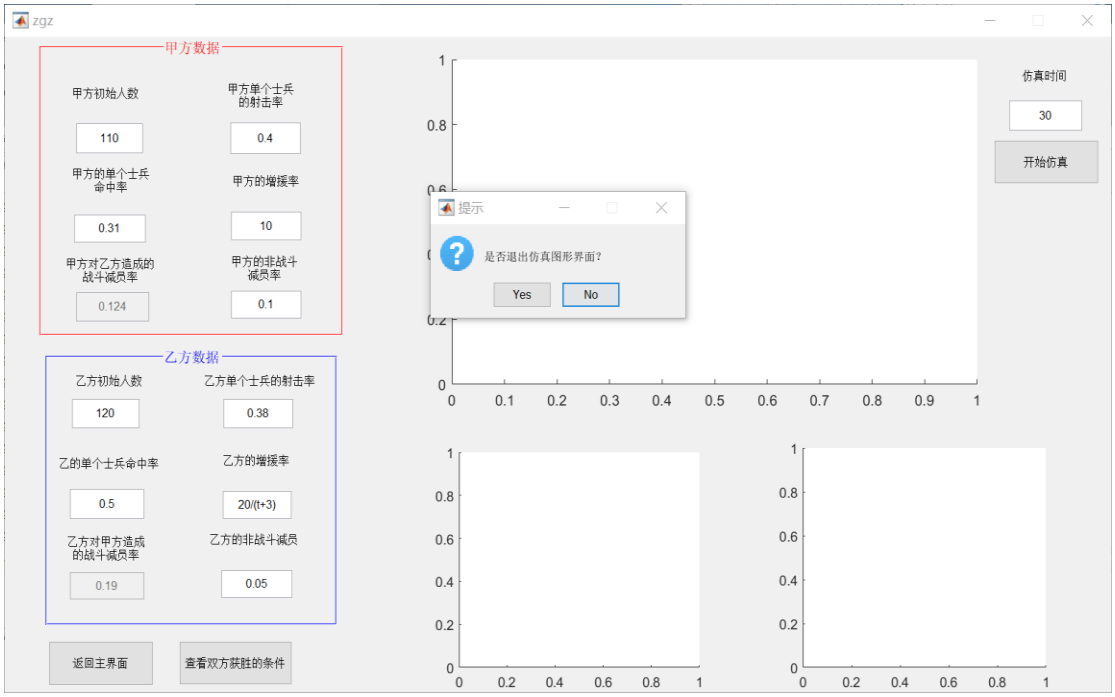
点击返回主界面，可以返回进入时的欢迎界面，重新选择想要仿真的模型。

点击查看双方获胜的条件，可以看到在正规战争模型中，甲乙双方获胜与输入参数之间的关系和相轨线。



在关闭界面的时候，会提示，确定退出仿真图形界面？





其他两个模型的仿真过程与正规战争模型的仿真一样。

八．心得体会

由于以前上过数学建模的课程，而且参加过全国大学生数学建模比赛，所以对于我做的这个战争模型比较熟悉，在参加比赛之前，我也对 matlab 有过系统

的学习，会用 matlab 去编写一些简单的，如遗传算法，模拟退火算法等等，也会用 ode45 去求得微分方程的数值解。再加上上课认真听讲，老师讲的模型课下都有看过，对于老师上传的一些实例，我也都跑过一遍，所以整个过程操作起来上手也比较快。

在开始之前，我也在 B 站上找了一系列 GUIDE 编程的学习视频，学会了如何对 GUIDE 创建的控件和 Unicontrol 创建的控件之间进行数据交互，需要用 handles.Tag 的形式，对创建的控件的各个属性值进行读取和访问。因此，在使用 GUIDE 创建控件的时候，需要对控件的 Tag 值赋一个容易记住的标识。

这个课程的本质是让我们建立合适的模型去解决一个实际问题，比如我选择的是战争模型，需要对三种不同的情形分别进行仿真。我觉得使用 Matlab 制作一个仿真系统是一个既简单又复杂的事。简单在于 Matlab 强大的功能，可以直接拖动 GUIDE 界面的控件到指定的位置。而难是因为首先要进行模型的建立，从一个理想化的模型，到越来越接近实际的一个真实的模型。

除了界面设计以外，主要就是需要写回调函数，来实现对应的功能和界面之间的来回跳转，这就需要界面在跳转的过程中保持稳定性，不能经过几次跳转后就崩掉了，或者出现跳转出错的情况，所以在写回调函数的时候要看清楚是哪个控件的回调函数。

学习完了这门课程并独立完成了这个 matlab 课程设计后，我对 GUIDE 设计图形界面有了更进一步的认识和了解，对于 matlab 图形化界面的设计和分析问题解决问题的能力已经有了一定的进步,也已经可以使用 matlab 的 GUI 功能搭建基于图形化界面的小系统，对于仿真也有了更深一层的理解。

但 matlab 功能的强大原不至于如此，我以前还用过 matlab 对提取图像的像素矩阵，然后对图像进行一些处理。对实际难以搭建的系统，仿真就有很大的用处，以后在这方面还应该进行更深层次的学习。

九．参考书籍及资料

1.《系统建模与仿真》 齐欢 王小平 编著 清华大学出版社
2.《MATLAB 程序设计与应用（第二版）》 刘卫国 编著 高等教育出版社
3.《MATLAB 实用教程》 苏金明 阮沈勇 编著 电子工业出版社

十．代码附录

1.welcome.m

```
function varargout = welcome(varargin)
% WELCOME MATLAB code for welcome.fig
%      WELCOME, by itself, creates a new WELCOME or raises the existing
%      singleton*.
%
%      H = WELCOME returns the handle to a new WELCOME or the handle to
%      the existing singleton*.
```

```
%
%     WELCOME('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in WELCOME.M with the given input arguments.
%
%     WELCOME('Property','Value',...) creates a new WELCOME or raises the
%     existing singleton*.  Starting from the left, property value pairs are
%     applied to the GUI before welcome_OpeningFcn gets called.  An
%     unrecognized property name or invalid value makes property application
%     stop.  All inputs are passed to welcome_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help welcome

% Last Modified by GUIDE v2.5 13-Dec-2020 23:17:26

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @welcome_OpeningFcn, ...
                   'gui_OutputFcn',  @welcome_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before welcome is made visible.
function welcome_OpeningFcn(hObject, eventdata, handles, varargin)
img=imread('welcome.jpg');
imshow(img,'Parent',handles.axes1);
option=0;
handles.option=option;
```

```
% Choose default command line output for welcome
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes welcome wait for user response (see UIRESUME)
% uiwait(handles.welcome_m);

% --- Outputs from this function are returned to the command line.
function varargout = welcome_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

% --- Executes on button press in zgz.
function zgz_Callback(hObject, eventdata, handles)
set(handles.zgz,'Value',1);
set(handles.yjz,'Value',0);
set(handles.hhz,'Value',0);
handles.option=1;
guidata(hObject, handles);

% --- Executes on button press in yjz.
function yjz_Callback(hObject, eventdata, handles)
set(handles.zgz,'Value',0);
set(handles.yjz,'Value',1);
set(handles.hhz,'Value',0);
handles.option=2;
guidata(hObject, handles);

% --- Executes on button press in hhz.
function hhz_Callback(hObject, eventdata, handles)
set(handles.zgz,'Value',0);
set(handles.yjz,'Value',0);
set(handles.hhz,'Value',1);
handles.option=3;
guidata(hObject, handles);

% --- Executes on button press in start.
function start_Callback(hObject, eventdata, handles)
switch handles.option
   case 1
      set(handles.zgz,'Value',0);
      set(handles.welcome_m,'Visible','off');
      zgz(handles);
```

```
    case 2
        set(handles.yjz,'Value',0);
       set(handles.welcome_m,'Visible','off');
       yjz(handles);
    case 3
        set(handles.hhz,'Value',0);
       set(handles.welcome_m,'Visible','off');
       hhz(handles);
end

% --- Executes when user attempts to close welcome_m.
function welcome_m_CloseRequestFcn(hObject, eventdata, handles)
selection = questdlg('是否退出仿真图形界面？', ...
                  '提示', ...
                  'Yes','No','No');
switch selection
      case 'Yes'
         delete(hObject);
      case 'No'
         return
end
```

2.zgz.m

```
function varargout = zgz(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @zgz_OpeningFcn, ...
                  'gui_OutputFcn',  @zgz_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before zgz is made visible.
function zgz_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
```

```
jia_r=str2double(get(handles.jia_r,'string'));
 jia_p=str2double(get(handles.jia_p,'string'));
 var=jia_r*jia_p;
 set(handles.jia_dec,'string',num2str(var));
 yi_r=str2double(get(handles.yi_r,'string'));
 yi_p=str2double(get(handles.yi_p,'string'));
 var=yi_r*yi_p;
 set(handles.yi_dec,'string',num2str(var));
% Update handles structure
guidata(hObject, handles);
% --- Outputs from this function are returned to the command line.
function varargout = zgz_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function jia_x0_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function jia_x0_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function jia_r_Callback(hObject, eventdata, handles)
 jia_r=str2double(get(handles.jia_r,'string'));
 jia_p=str2double(get(handles.jia_p,'string'));
 var=jia_r*jia_p;
 set(handles.jia_dec,'string',num2str(var));
% --- Executes during object creation, after setting all properties.
function jia_r_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function jia_p_Callback(hObject, eventdata, handles)
%caculate jia_dec
 jia_r=str2double(get(handles.jia_r,'string'));
 jia_p=str2double(get(handles.jia_p,'string'));
 var=jia_r*jia_p;
 set(handles.jia_dec,'string',num2str(var));
% --- Executes during object creation, after setting all properties.
function jia_p_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function jia_yuan_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function jia_yuan_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end
function jia_dec_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function jia_dec_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function jia_alpha_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function jia_alpha_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yi_x0_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function yi_x0_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yi_r_Callback(hObject, eventdata, handles)
%caculate yi_dec
 yi_r=str2double(get(handles.yi_r,'string'));
 yi_p=str2double(get(handles.yi_p,'string'));
 var=yi_r*yi_p;
 set(handles.yi_dec,'string',num2str(var));
% --- Executes during object creation, after setting all properties.
function yi_r_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yi_p_Callback(hObject, eventdata, handles)
%caculate yi_dec
 yi_r=str2double(get(handles.yi_r,'string'));
 yi_p=str2double(get(handles.yi_p,'string'));
 var=yi_r*yi_p;
 set(handles.yi_dec,'string',num2str(var));
% --- Executes during object creation, after setting all properties.
function yi_p_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yi_dec_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
```

```
function yi_dec_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yi_yuan_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function yi_yuan_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yi_alpha_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function yi_alpha_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes on button press in check.
function check_Callback(hObject, eventdata, handles)
zgz_check(handles);
function time_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function time_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes when user attempts to close zgz_m.
function zgz_m_CloseRequestFcn(hObject, eventdata, handles)
selection = questdlg('是否退出仿真图形界面？', ...
                '提示', ...
                'Yes','No','No');
switch selection
      case 'Yes'
         delete(hObject);
      case 'No'
         return
end
% --- Executes on button press in z_back.
function z_back_Callback(hObject, eventdata, handles)
%返回主界面，关闭当前界面
delete(handles.zgz_m);
welcome(handles);

% --- Executes on button press in z_victory.
function z_victory_Callback(hObject, eventdata, handles)
```

```
%查看甲乙双方获胜条件
z_victory(handles);
```

3.zgz_check.m

```
function zgz_check(handles)
%声明全局变量
global jia_x0 jia_r jia_p jia_alpha jia_yuan yi_x0 yi_r yi_p yi_alpha yi_yuan time
syms t
%读取甲方参数
jia_x0=str2double(get(handles.jia_x0,'string'));
jia_r=str2double(get(handles.jia_r,'string'));
jia_p=str2double(get(handles.jia_p,'string'));
jia_alpha=str2double(get(handles.jia_alpha,'string'));
time=str2double(get(handles.time,'string'));
flag1=0;
flag2=0;
%判断输入的增援率是常数还是函数
flag_jia=contains(get(handles.jia_yuan,'string'),'t');
if flag_jia==1
   jia_yuan=str2func(['@(t)',get(handles.jia_yuan,'string')]);
else
   jia_yuan=str2double(get(handles.jia_yuan,'string'));
    if(isempty(get(handles.jia_yuan,'string')))
       flag1=1;
    elseif (isnan(jia_yuan))
        flag1=1;
    elseif (jia_yuan< 0)
        flag1=2;
    end
end
%读取乙方数据
yi_x0=str2double(get(handles.yi_x0,'string'));
yi_r=str2double(get(handles.yi_r,'string'));
yi_p=str2double(get(handles.yi_p,'string'));
yi_alpha=str2double(get(handles.yi_alpha,'string'));
flag_yi=contains(get(handles.yi_yuan,'string'),'t');
%判断输入的增援率是常数还是函数
if flag_yi==1
   yi_yuan=str2func(['@(t)',get(handles.yi_yuan,'string')]);
else
   yi_yuan=str2double(get(handles.yi_yuan,'string'));
   if(isempty(get(handles.yi_yuan,'string')))
       flag2=1;
```

```
   elseif (isnan(yi_yuan))
       flag2=1;
   elseif (yi_yuan< 0)
       flag2=2;
    end
end
%判断输入的参数是否合理
    if(isempty(handles.jia_x0)||isempty(handles.jia_r)||isempty(handles.jia_p)||isempty(handles.jia_
alpha)||isempty(handles.yi_x0)||isempty(handles.yi_r)||isempty(handles.yi_p)||isempty(handles.yi_
alpha)||isempty(handles.time))
        msgbox('输入参数不能为空', 'Error','error');
    elseif(isnan(jia_x0)||isnan(jia_r)||isnan(jia_p)||isnan(jia_alpha)||isnan(yi_x0)||isnan(yi_r)||isnan(
yi_p)||isnan(yi_alpha)||isnan(time)||flag1==1||flag2==1)
        msgbox('输入参数不能为空', 'Error','error');
    elseif (jia_x0 < 0 || jia_r < 0 || jia_p < 0 || jia_alpha < 0 || yi_x0 < 0 || yi_r < 0 || yi_p <0||yi_alph
a<0||time<0||flag1==2||flag2==2)
       msgbox('输入参数不能小于 0', 'Error','error');
    elseif(jia_r>=1||jia_p>=1||yi_r>=1||yi_p>=1)
        msgbox('射击率和命中率应该在 0~1 之间', 'Error','error');
   else
      zgz_simu(handles);
   end
end
```

4.zgz_odefunc.m

```
%正规战争模型的微分方程组
function dx = zgz_odefunc(t, x)
global jia_r jia_p jia_alpha jia_yuan  yi_r yi_p yi_alpha yi_yuan
dx = zeros(2, 1);
try
  dx(1,1)=-yi_r*yi_p*x(2)-jia_alpha*x(1)+jia_yuan;
catch
   dx(1,1)=-yi_r*yi_p*x(2)-jia_alpha*x(1)+jia_yuan(t);
end

try
  dx(2,1)=-jia_r*jia_p*x(1)-yi_alpha*x(2)+yi_yuan;

catch
  dx(2,1)=-jia_r*jia_p*x(1)-yi_alpha*x(2)+yi_yuan(t);
end
end
```

5.zgz_simu.m

```
function zgz_simu(handles)
global jia_x0 yi_x0 time
%求数值解
tspan=[0,time];
x0=[jia_x0,yi_x0];
[t,y]=ode45('zgz_odefunc',tspan,x0);

set(handles.check,'Enable','off');
set(handles.z_back,'Enable','off');
set(handles.z_victory,'Enable','off');

xlabel(handles.curves_diagram, '仿真时间 T');
ylabel(handles.curves_diagram, '人数');
title(handles.curves_diagram, '人数随时间的变化');
k1=0;
k2=0;
%人数大于 0 的个数
for i=1:length(t)
   if(y(i,1)>1e-6)
      k1=k1+1;
   else
      break;
   end
end
for i=1:length(t)
   if(y(i,2)>1e-6)
      k2=k2+1;
   else
      break;
   end
end
k=min(k1,k2);
%判断哪一方获胜，哪一方战败
if(k1<k2)
   y(k1+1:end,1)=0;
   y(k1+1:end,2)=y(k1,2);
else
   y(k2+1:end,2)=0;
   y(k2+1:end,1)=y(k2,1);
end
%动态仿真
n_frames = size(t, 1);
```

```
    for k = 1:n_frames
        % Compute the input medicine volume
        hold on;
        cla(handles.curves_diagram);
        plot(handles.curves_diagram, t(1:k), y(1:k, 1), 'Color', 'r');
        plot(handles.curves_diagram, t(1:k), y(1:k, 2), 'Color', 'b');
        axis(handles.curves_diagram, [0, max(t), 0, max(max([y(:, 1), y(:, 2)]))]);
        legend(handles.curves_diagram, '甲方人数', '乙方人数');

        % Plot the pie diagram
        cla(handles.pie_diagram);
        pie3(handles.pie_diagram, [max(y(k, 1), 1e-6), max(y(k, 2), 1e-6)]);
        legend(handles.pie_diagram, '甲方人数', '乙方人数');

        % Plot the histogram
        cla(handles.hist_diagram);
        b = bar(handles.hist_diagram, ...
            categorical({'甲方人数', '乙方人数'}), ...
            [max(y(k, 2), 1e-6),max(y(k, 1), 1e-6) ]);
        b.FaceColor = 'flat';
        b.CData(1, :) = [1, 0, 0];
        b.CData(2, :) = [0, 1, 0];
        ymax = max(max(y));
        set(handles.hist_diagram, 'YLim', [0, ymax]);

        pause(0.001);
    end

set(handles.check,'Enable','on');
set(handles.z_back,'Enable','on');
set(handles.z_victory,'Enable','on');
```

6.z_victory.m

```
function varargout = z_victory(varargin)
% Z_VICTORY MATLAB code for z_victory.fig
%      Z_VICTORY, by itself, creates a new Z_VICTORY or raises the existing
%      singleton*.
%
%      H = Z_VICTORY returns the handle to a new Z_VICTORY or the handle to
%      the existing singleton*.
%
%      Z_VICTORY('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in Z_VICTORY.M with the given input arguments.
```

```
%
%     Z_VICTORY('Property','Value',...) creates a new Z_VICTORY or raises the
%     existing singleton*.  Starting from the left, property value pairs are
%     applied to the GUI before z_victory_OpeningFcn gets called.  An
%     unrecognized property name or invalid value makes property application
%     stop.  All inputs are passed to z_victory_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help z_victory

% Last Modified by GUIDE v2.5 14-Dec-2020 18:57:14

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @z_victory_OpeningFcn, ...
                   'gui_OutputFcn',  @z_victory_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before z_victory is made visible.
function z_victory_OpeningFcn(hObject, eventdata, handles, varargin)

% Choose default command line output for z_victory
handles.output = hObject;
%显示甲乙双方获胜的条件和相轨线
img1=imread('zgz_1.png');
imshow(img1,'Parent',handles.zgz_1);
img2=imread('zgz_2.png');
```

```
imshow(img2,'Parent',handles.zgz_2);
% Update handles structure
guidata(hObject, handles);
% --- Outputs from this function are returned to the command line.
function varargout = z_victory_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
```

7.yjz.m

```
function varargout = yjz(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @yjz_OpeningFcn, ...
                   'gui_OutputFcn',  @yjz_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function yjz_OpeningFcn(hObject, eventdata, handles, varargin)
% Choose default command line output for yjz
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% --- Outputs from this function are returned to the command line.
function varargout = yjz_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function yjia_x0_Callback(hObject, eventdata, handles)
function yjia_x0_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yjia_r_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function yjia_r_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function yjia_sx_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function yjia_sx_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yjia_yuan_Callback(hObject, eventdata, handles)
function yjia_yuan_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yjia_alpha_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function yjia_alpha_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yyi_x0_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function yyi_x0_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yyi_r_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function yyi_r_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yyi_sx_Callback(hObject, eventdata, handles)
function yyi_sx_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yyi_yuan_Callback(hObject, eventdata, handles)
function yyi_yuan_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yyi_alpha_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function yyi_alpha_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes on button press in y_check.
function y_check_Callback(hObject, eventdata, handles)
% hObject    handle to y_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
yjz_check(handles);
function y_time_Callback(hObject, eventdata, handles)
function y_time_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yjia_s_Callback(hObject, eventdata, handles)
function yjia_s_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function yyi_s_Callback(hObject, eventdata, handles)
% hObject    handle to yyi_s (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of yyi_s as text
%        str2double(get(hObject,'String')) returns contents of yyi_s as a double
% --- Executes during object creation, after setting all properties.
function yyi_s_CreateFcn(hObject, eventdata, handles)
% hObject    handle to yyi_s (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes when user attempts to close yjz_m.
function yjz_m_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to yjz_m (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: delete(hObject) close the figure
selection = questdlg('是否退出仿真图形界面？', ...
                  '提示', ...
                  'Yes','No','No');
```

```
switch selection
    case 'Yes'
        delete(hObject);
    case 'No'
        return
end
% --- Executes on button press in y_back.
function y_back_Callback(hObject, eventdata, handles)
% hObject    handle to y_back (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%返回主界面，关闭当前界面
delete(handles.yjz_m);
welcome(handles);
% --- Executes on button press in y_victory.
function y_victory_Callback(hObject, eventdata, handles)
% hObject    handle to y_victory (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%查看甲乙双方获胜条件
y_victory(handles);
```

8.yjz_check.m

```
function yjz_check(handles)
%声明全局变量
global yjia_x0 yjia_r yjia_sx yjia_s yjia_alpha yjia_yuan yyi_x0 yyi_r yyi_sx yyi_s yyi_alpha yyi
_yuan y_time
syms t
%读取甲方参数
yjia_x0=str2double(get(handles.yjia_x0,'string'));
yjia_r=str2double(get(handles.yjia_r,'string'));
yjia_sx=str2double(get(handles.yjia_sx,'string'));
yjia_s=str2double(get(handles.yjia_s,'string'));
yjia_alpha=str2double(get(handles.yjia_alpha,'string'));
y_time=str2double(get(handles.y_time,'string'));
flag1=0;
flag2=0;
%判断输入的增援率是常数还是函数
flag_yjia=contains(get(handles.yjia_yuan,'string'),'t');
if flag_yjia==1
   yjia_yuan=str2func(['@(t)',get(handles.yjia_yuan,'string')]);
else
   yjia_yuan=str2double(get(handles.yjia_yuan,'string'));
```

```
    if(isempty(get(handles.yjia_yuan,'string')))
       flag1=1;
    elseif (isnan(yjia_yuan))
        flag1=1;
    elseif (yjia_yuan< 0)
       flag1=2;
    end
end
%读取乙方数据
yyi_x0=str2double(get(handles.yyi_x0,'string'));
yyi_r=str2double(get(handles.yyi_r,'string'));
yyi_sx=str2double(get(handles.yyi_sx,'string'));
yyi_s=str2double(get(handles.yyi_s,'string'));
yyi_alpha=str2double(get(handles.yyi_alpha,'string'));
flag_yyi=contains(get(handles.yyi_yuan,'string'),'t');
%判断输入的增援率是常数还是函数
if flag_yyi==1
   yyi_yuan=str2func(['@(t)',get(handles.yyi_yuan,'string')]);
else
   yyi_yuan=str2double(get(handles.yyi_yuan,'string'));
   if(isempty(get(handles.yyi_yuan,'string')))
      flag2=1;
   elseif (isnan(yyi_yuan))
       flag2=1;
   elseif (yyi_yuan< 0)
       flag2=2;
    end
end
%判断输入的参数是否合理
    if(isempty(handles.yjia_x0)||isempty(handles.yjia_r)||isempty(handles.yjia_sx)||isempty(handle
s.yjia_s)||isempty(handles.yjia_alpha)||isempty(handles.yyi_x0)||isempty(handles.yyi_r)||isempty(h
andles.yyi_sx)||isempty(handles.yyi_s)||isempty(handles.yyi_alpha)||isempty(handles.y_time))
        msgbox('输入参数不能为空', 'Error','error')
    elseif(isnan(yjia_x0)||isnan(yjia_r)||isnan(yjia_sx)||isnan(yjia_s)||isnan(yjia_alpha)||isnan(yyi_x
0)||isnan(yyi_r)||isnan(yyi_sx)||isnan(yyi_s)||isnan(yyi_alpha)||isnan(y_time)||flag1==1||flag2==1)
        msgbox('输入参数不能为空', 'Error','error');
   elseif (yjia_x0 < 0 || yjia_r < 0 || yjia_sx < 0 ||yjia_s < 0 || yjia_alpha < 0 || yyi_x0 < 0 || yyi_r < 0
 || yyi_sx <0||yyi_s <0||yyi_alpha<0||y_time<0||flag1==2||flag2==2)
        msgbox('输入参数不能小于 0', 'Error','error');
    elseif(yjia_r>=1||yyi_r>=1)
        msgbox('射击率和命中率应该在 0~1 之间', 'Error','error');
   else
     yjz_simu(handles);
   end
```

```
end
```

9.yjz_odefunc.m

```
%游击战争模型的微分方程组
function dx = yjz_odefunc(t, x)
global yjia_r yjia_sx yjia_s yjia_alpha yjia_yuan  yyi_r yyi_sx yyi_s yyi_alpha yyi_yuan
dx = zeros(2, 1);
try
  dx(1,1)=-yyi_r*yyi_sx*x(1)*x(2)/yjia_s-yjia_alpha*x(1)+yjia_yuan;
catch
   dx(1,1)=-yyi_r*yyi_sx*x(1)*x(2)/yjia_s-yjia_alpha*x(1)+yjia_yuan(t);
end

try
  dx(2,1)=-yjia_r*yjia_sx*x(1)*x(2)/yyi_s-yyi_alpha*x(2)+yyi_yuan;
catch
  dx(2,1)=-yjia_r*yjia_sx*x(1)*x(2)/yyi_s-yyi_alpha*x(2)+yyi_yuan(t);
end

end
```

10.yjz_simu.m

```
function yjz_simu(handles)
global yjia_x0 yyi_x0 y_time
%求数值解
tspan=[0,y_time];
x0=[yjia_x0,yyi_x0];
[t,y]=ode45('yjz_odefunc',tspan,x0);

set(handles.y_check,'Enable','off');
set(handles.y_back,'Enable','off');
set(handles.y_victory,'Enable','off');

xlabel(handles.y_curves_diagram, '仿真时间 T');
ylabel(handles.y_curves_diagram, '人数');
title(handles.y_curves_diagram, '人数随时间的变化');
k1=0;
```

```
k2=0;
%统计大于 0 的个数
for i=1:length(t)
   if(y(i,1)>=0.5)
      k1=k1+1;
   else
      break;
   end
end
for i=1:length(t)
   if(y(i,2)>=0.5)
      k2=k2+1;
   else
      break;
   end
end
k=min(k1,k2);
%判断哪一方战败，哪一方获胜
if(k1<k2)
   y(k1+1:end,1)=0;
   y(k1+1:end,2)=y(k1,2);
else
   y(k2+1:end,2)=0;
   y(k2+1:end,1)=y(k2,1);
end
%动态仿真
n_frames = size(t, 1);
   for k = 1:n_frames
      % Compute the input medicine volume
      hold on;
      cla(handles.y_curves_diagram);
      plot(handles.y_curves_diagram, t(1:k), y(1:k, 1), 'Color', 'r');
      plot(handles.y_curves_diagram, t(1:k), y(1:k, 2), 'Color', 'b');
      axis(handles.y_curves_diagram, [0, max(t), 0, max(max([y(:, 1), y(:, 2)]))]);
      legend(handles.y_curves_diagram, '甲方人数', '乙方人数');

      % Plot the pie diagram
      cla(handles.y_pie_diagram);
      pie3(handles.y_pie_diagram, [max(y(k, 1), 1e-6), max(y(k, 2), 1e-6)]);
      legend(handles.y_pie_diagram, '甲方人数', '乙方人数');

      % Plot the histogram
      cla(handles.y_hist_diagram);
      b = bar(handles.y_hist_diagram, ...
```

```
        categorical({'甲方人数', '乙方人数'}), ...
        [max(y(k, 2), 1e-6),max(y(k, 1), 1e-6) ]);
    b.FaceColor = 'flat';
    b.CData(1, :) = [1, 0, 0];
    b.CData(2, :) = [0, 1, 0];
    ymax = max(max(y));
    set(handles.y_hist_diagram, 'YLim', [0, ymax]);

    pause(0.001);
  end
set(handles.y_check,'Enable','on');
set(handles.y_back,'Enable','on');
set(handles.y_victory,'Enable','on');
```

11.y_victory.m

```
function varargout = y_victory(varargin)
% Y_VICTORY MATLAB code for y_victory.fig
%      Y_VICTORY, by itself, creates a new Y_VICTORY or raises the existing
%      singleton*.
%
%      H = Y_VICTORY returns the handle to a new Y_VICTORY or the handle to
%      the existing singleton*.
%
%      Y_VICTORY('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in Y_VICTORY.M with the given input arguments.
%
%      Y_VICTORY('Property','Value',...) creates a new Y_VICTORY or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before y_victory_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to y_victory_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help y_victory

% Last Modified by GUIDE v2.5 14-Dec-2020 20:18:08
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @y_victory_OpeningFcn, ...
                   'gui_OutputFcn',  @y_victory_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before y_victory is made visible.
function y_victory_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to y_victory (see VARARGIN)

% Choose default command line output for y_victory
handles.output = hObject;
%显示甲乙双方获胜的条件和相轨线
img1=imread('yjz_1.png');
imshow(img1,'Parent',handles.yjz_1);
img2=imread('yjz_2.png');
imshow(img2,'Parent',handles.yjz_2);
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes y_victory wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = y_victory_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
```

12.hhz.m

```
function varargout = hhz(varargin)
% HHZ MATLAB code for hhz.fig
%      HHZ, by itself, creates a new HHZ or raises the existing
%      singleton*.
%
%      H = HHZ returns the handle to a new HHZ or the handle to
%      the existing singleton*.
%
%      HHZ('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in HHZ.M with the given input arguments.
%
%      HHZ('Property','Value',...) creates a new HHZ or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before hhz_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to hhz_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help hhz

% Last Modified by GUIDE v2.5 14-Dec-2020 23:33:54

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @hhz_OpeningFcn, ...
                   'gui_OutputFcn',  @hhz_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
```

```
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before hhz is made visible.
function hhz_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to hhz (see VARARGIN)

% Choose default command line output for hhz
handles.output = hObject;
jia_r=str2double(get(handles.hjia_r,'string'));
jia_p=str2double(get(handles.hjia_p,'string'));
 var=jia_r*jia_p;
 set(handles.hyi_dec,'string',num2str(var));
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes hhz wait for user response (see UIRESUME)
% uiwait(handles.hhz_m);

% --- Outputs from this function are returned to the command line.
function varargout = hhz_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function hjia_x0_Callback(hObject, eventdata, handles)
% hObject    handle to hjia_x0 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of hjia_x0 as text
%        str2double(get(hObject,'String')) returns contents of hjia_x0 as a double

% --- Executes during object creation, after setting all properties.
function hjia_x0_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hjia_x0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function hjia_r_Callback(hObject, eventdata, handles)
% hObject    handle to hjia_r (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of hjia_r as text
%        str2double(get(hObject,'String')) returns contents of hjia_r as a double
jia_r=str2double(get(handles.hjia_r,'string'));
jia_p=str2double(get(handles.hjia_p,'string'));
 var=jia_r*jia_p;
 set(handles.hyi_dec,'string',num2str(var));

% --- Executes during object creation, after setting all properties.
function hjia_r_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hjia_r (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
```

```
function hjia_p_Callback(hObject, eventdata, handles)
% hObject    handle to hjia_p (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of hjia_p as text
%        str2double(get(hObject,'String')) returns contents of hjia_p as a double
jia_r=str2double(get(handles.hjia_r,'string'));
jia_p=str2double(get(handles.hjia_p,'string'));
 var=jia_r*jia_p;
 set(handles.hyi_dec,'string',num2str(var));

% --- Executes during object creation, after setting all properties.
function hjia_p_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hjia_p (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function hjia_yuan_Callback(hObject, eventdata, handles)
% hObject    handle to hjia_yuan (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of hjia_yuan as text
%        str2double(get(hObject,'String')) returns contents of hjia_yuan as a double

% --- Executes during object creation, after setting all properties.
function hjia_yuan_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hjia_yuan (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
```

```
function hjia_alpha_Callback(hObject, eventdata, handles)
% hObject    handle to hjia_alpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of hjia_alpha as text
%        str2double(get(hObject,'String')) returns contents of hjia_alpha as a double

% --- Executes during object creation, after setting all properties.
function hjia_alpha_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hjia_alpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function hyi_x0_Callback(hObject, eventdata, handles)
% hObject    handle to hyi_x0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of hyi_x0 as text
%        str2double(get(hObject,'String')) returns contents of hyi_x0 as a double

% --- Executes during object creation, after setting all properties.
function hyi_x0_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hyi_x0 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
```

```
function hyi_r_Callback(hObject, eventdata, handles)
% hObject    handle to hyi_r (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of hyi_r as text
%        str2double(get(hObject,'String')) returns contents of hyi_r as a double

% --- Executes during object creation, after setting all properties.
function hyi_r_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hyi_r (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function hyi_sx_Callback(hObject, eventdata, handles)
% hObject    handle to hyi_sx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of hyi_sx as text
%        str2double(get(hObject,'String')) returns contents of hyi_sx as a double

% --- Executes during object creation, after setting all properties.
function hyi_sx_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hyi_sx (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function hyi_yuan_Callback(hObject, eventdata, handles)
% hObject    handle to hyi_yuan (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of hyi_yuan as text
%        str2double(get(hObject,'String')) returns contents of hyi_yuan as a double

% --- Executes during object creation, after setting all properties.
function hyi_yuan_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hyi_yuan (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function hyi_alpha_Callback(hObject, eventdata, handles)
% hObject    handle to hyi_alpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of hyi_alpha as text
%        str2double(get(hObject,'String')) returns contents of hyi_alpha as a double

% --- Executes during object creation, after setting all properties.
function hyi_alpha_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hyi_alpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in h_check.
function h_check_Callback(hObject, eventdata, handles)
% hObject    handle to h_check (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
hhz_check(handles);

function h_time_Callback(hObject, eventdata, handles)
% hObject    handle to h_time (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of h_time as text
%        str2double(get(hObject,'String')) returns contents of h_time as a double

% --- Executes during object creation, after setting all properties.
function h_time_CreateFcn(hObject, eventdata, handles)
% hObject    handle to h_time (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function hjia_s_Callback(hObject, eventdata, handles)
% hObject    handle to hjia_s (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of hjia_s as text
%        str2double(get(hObject,'String')) returns contents of hjia_s as a double

% --- Executes during object creation, after setting all properties.
function hjia_s_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hjia_s (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
```

```
function hyi_dec_Callback(hObject, eventdata, handles)
% hObject    handle to hyi_dec (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of hyi_dec as text
%        str2double(get(hObject,'String')) returns contents of hyi_dec as a double

% --- Executes during object creation, after setting all properties.
function hyi_dec_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hyi_dec (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes when user attempts to close hhz_m.
function hhz_m_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to hhz_m (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: delete(hObject) closes the figure
selection = questdlg('是否退出仿真图形界面？', ...
                '提示', ...
                'Yes','No','No');
switch selection
      case 'Yes'
         delete(hObject);
      case 'No'
         return
end

% --- Executes on button press in h_back.
function h_back_Callback(hObject, eventdata, handles)
% hObject    handle to h_back (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%返回主界面，关闭当前界面
delete(handles.hhz_m);
```

```
welcome(handles);

% --- Executes on button press in h_victory.
function h_victory_Callback(hObject, eventdata, handles)
% hObject    handle to h_victory (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%查看甲乙双方获胜条件
h_victory(handles);
```

13.hhz_check.m

```
function hhz_check(handles)
%声明全局变量
global hjia_x0 hjia_r hjia_p hjia_alpha hjia_yuan hjia_s hyi_x0 hyi_r hyi_sx hyi_alpha hyi_yuan h
_time
syms t
%读取甲方参数
hjia_x0=str2double(get(handles.hjia_x0,'string'));
hjia_r=str2double(get(handles.hjia_r,'string'));
hjia_p=str2double(get(handles.hjia_p,'string'));
hjia_s=str2double(get(handles.hjia_s,'string'));
hjia_alpha=str2double(get(handles.hjia_alpha,'string'));
h_time=str2double(get(handles.h_time,'string'));
flag1=0;
flag2=0;
%判断输入的增援率是常数还是函数
flag_hjia=contains(get(handles.hjia_yuan,'string'),'t');
if flag_hjia==1
   hjia_yuan=str2func(['@(t)',get(handles.hjia_yuan,'string')]);
else
   hjia_yuan=str2double(get(handles.hjia_yuan,'string'));
    if(isempty(get(handles.hjia_yuan,'string')))
       flag1=1;
    elseif (isnan(hjia_yuan))
        flag1=1;
    elseif (hjia_yuan< 0)
         flag1=2;
    end
end
%读取乙方数据
hyi_x0=str2double(get(handles.hyi_x0,'string'));
```

```
hyi_r=str2double(get(handles.hyi_r,'string'));
hyi_sx=str2double(get(handles.hyi_sx,'string'));
hyi_alpha=str2double(get(handles.hyi_alpha,'string'));
flag_hyi=contains(get(handles.hyi_yuan,'string'),'t');
%判断输入的增援率是常数还是函数
if flag_hyi==1
   hyi_yuan=str2func(['@(t)',get(handles.hyi_yuan,'string')]);
else
   hyi_yuan=str2double(get(handles.hyi_yuan,'string'));
   if(isempty(get(handles.hyi_yuan,'string')))
      flag2=1;
   elseif (isnan(hyi_yuan))
      flag2=1;
   elseif (hyi_yuan< 0)
      flag2=2;
    end
end
%判断输入的参数是否合理
    if(isempty(handles.hjia_x0)||isempty(handles.hjia_r)||isempty(handles.hjia_p)||isempty(handles.
hjia_alpha)||isempty(handles.hjia_s)||isempty(handles.hyi_x0)||isempty(handles.hyi_r)||isempty(ha
ndles.hyi_sx)||isempty(handles.hyi_alpha)||isempty(handles.h_time))
       msgbox('输入参数不能为空', 'Error','error')
    elseif(isnan(hjia_x0)||isnan(hjia_r)||isnan(hjia_p)||isnan(hjia_s)||isnan(hjia_alpha)||isnan(hyi_x0
)||isnan(hyi_r)||isnan(hyi_sx)||isnan(hyi_alpha)||isnan(h_time)||flag1==1||flag2==1)
       msgbox('输入参数不能为空', 'Error','error');
   elseif (hjia_x0 < 0 || hjia_r < 0 || hjia_p < 0 ||hjia_s < 0 || hjia_alpha < 0 || hyi_x0 < 0 || hyi_r < 0
|| hyi_sx <0||hyi_alpha<0||h_time<0||flag1==2||flag2==2)
      msgbox('输入参数不能小于 0', 'Error','error');
    elseif(hjia_r>=1||hjia_p>=1||hyi_r>=1)
       msgbox('射击率和命中率应该在 0~1 之间', 'Error','error');
   else
     hhz_simu(handles);
   end
end
```

14.hhz_odefunc.m

```
%混合战争模型的微分方程组
function dx = hhz_odefunc(t, x)
global hjia_r hjia_p hjia_s hjia_alpha hjia_yuan  hyi_r hyi_sx hyi_alpha hyi_yuan
dx = zeros(2, 1);
try
```

```
  dx(1,1)=-hyi_r*hyi_sx*x(1)*x(2)/hjia_s-hjia_alpha*x(1)+hjia_yuan;
catch
   dx(1,1)=-hyi_r*hyi_sx*x(1)*x(2)/hjia_s-hjia_alpha*x(1)+hjia_yuan(t);
end

try
  dx(2,1)=-hjia_r*hjia_p*x(1)-hyi_alpha*x(2)+hyi_yuan;
catch
  dx(2,1)=-hjia_r*hjia_p*x(1)-hyi_alpha*x(2)+hyi_yuan(t);
end

end
```

15.hhz_simu.m

```
function hhz_simu(handles)
global hjia_x0 hyi_x0 h_time
%求数值解
tspan=[0,h_time];
x0=[hjia_x0,hyi_x0];
[t,y]=ode45('hhz_odefunc',tspan,x0);

set(handles.h_check,'Enable','off');
set(handles.h_back,'Enable','off');
set(handles.h_victory,'Enable','off');

xlabel(handles.h_curves_diagram, '仿真时间 T');
ylabel(handles.h_curves_diagram, '人数');
title(handles.h_curves_diagram, '人数随时间的变化');
k1=0;
k2=0;
%统计大于 0 的个数
for i=1:length(t)
   if(y(i,1)>=1)
      k1=k1+1;
   else
      break;
   end
end
for i=1:length(t)
   if(y(i,2)>=1)
      k2=k2+1;
```

```
    else
        break;
    end
end
k=min(k1,k2);
%判断哪一方战败，哪一方获胜
if(k1<k2)
    y(k1+1:end,1)=0;
    y(k1+1:end,2)=y(k1,2);
else
    y(k2+1:end,2)=0;
    y(k2+1:end,1)=y(k2,1);
end
%动态仿真
n_frames = size(t, 1);
    for k = 1:n_frames
        % Plot the curves diagram
        hold on;
        cla(handles.h_curves_diagram);
        plot(handles.h_curves_diagram, t(1:k), y(1:k, 1), 'Color', 'r');
        plot(handles.h_curves_diagram, t(1:k), y(1:k, 2), 'Color', 'b');
        axis(handles.h_curves_diagram, [0, max(t), 0, max(max([y(:, 1), y(:, 2)]))]);
        legend(handles.h_curves_diagram, '甲方人数', '乙方人数');

        % Plot the pie diagram
        cla(handles.h_pie_diagram);
        pie3(handles.h_pie_diagram, [max(y(k, 1), 1e-6), max(y(k, 2), 1e-6)]);
        legend(handles.h_pie_diagram, '甲方人数', '乙方人数');

        % Plot the histogram
        cla(handles.h_hist_diagram);
        b = bar(handles.h_hist_diagram, ...
            categorical({'甲方人数', '乙方人数'}), ...
            [max(y(k, 2), 1e-6),max(y(k, 1), 1e-6) ]);
        b.FaceColor = 'flat';
        b.CData(1, :) = [1, 0, 0];
        b.CData(2, :) = [0, 1, 0];
        ymax = max(max(y));
        set(handles.h_hist_diagram, 'YLim', [0, ymax]);

        pause(0.001);
    end
set(handles.h_check,'Enable','on');
set(handles.h_back,'Enable','on');
```

```
set(handles.h_victory,'Enable','on');
```

16.h_victory

```
function varargout = h_victory(varargin)
% H_VICTORY MATLAB code for h_victory.fig
%      H_VICTORY, by itself, creates a new H_VICTORY or raises the existing
%      singleton*.
%
%      H = H_VICTORY returns the handle to a new H_VICTORY or the handle to
%      the existing singleton*.
%
%      H_VICTORY('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in H_VICTORY.M with the given input arguments.
%
%      H_VICTORY('Property','Value',...) creates a new H_VICTORY or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before h_victory_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to h_victory_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help h_victory

% Last Modified by GUIDE v2.5 14-Dec-2020 22:56:32

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @h_victory_OpeningFcn, ...
                   'gui_OutputFcn',  @h_victory_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```
if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before h_victory is made visible.
function h_victory_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to h_victory (see VARARGIN)

% Choose default command line output for h_victory
handles.output = hObject;
%显示甲乙双方获胜的条件和相轨线
img1=imread('hhz_1.png');
imshow(img1,'Parent',handles.hhz_1);
img2=imread('hhz_2.png');
imshow(img2,'Parent',handles.hhz_2);
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes h_victory wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = h_victory_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
```