Instructions for using a Mathematical Model of the Cardiovascular System Response to Fluid Perturbation (with Software Code)

Contact: Ramin Bighamian, Ph.D. CDRH Office of Science and Engineering Laboratories, Division of Biomedical Physics (ramin.bighamian@fda.hhs.gov)

Updated: August 05, 2024

1. General Information

This software code provides a mathematical model of the cardiovascular system (CVS) response to fluid perturbation [1], developed and validated using data collected from animal (sheep and swine) subjects. The model is built in MATLAB Simulink environment and the goal is to generate a cohort of simulated subjects against a test subject. The simulated subjects lead to a prediction envelope, which is used to evaluate predictive capability performance of the mathematical model and show its ability to generate valid virtual CVS physiological responses to fluid perturbation for the future non-clinical simulated testing setups. These testing setups may be useful as part of assessing physiological closed-loop control algorithms for automated fluid resuscitation systems. The CVS mathematical model is a low-order lumped parameter model, designed for use with virtual cohort generation tools. It takes rates of hemorrhage, urine, and fluid infusion as inputs and produces outputs for hematocrit (HCT), blood volume (BV), heart rate (HR), stroke volume (SV), cardiac output (CO), and mean arterial blood pressure (BP). The mathematical model calibration is defined based on the maximum likelihood estimation of its parameters. A compartment-based virtual cohort generation tool described in [1] is used to simulate virtual subjects and generate a prediction envelope used for model predictive capability performance.

2. System Requirements

The software code was written in a MATLAB® R2023b environment and consists of a Simulink file "CVS_Model.mdl", and a script for prediction envelope generation for an example test data. The prediction envelope generator includes a core code "Cohort_Generation.m". The data from a representative test subject as well as mathematical model parameters calibrated against 26 animal datasets have been provided in mat files, entitled "Example_Data.mat" and "Calibrated_Parameters.mat", respectively. Software has been tested in MATLAB® R2023b environment, and it requires the Simulink Toolbox.

3. Data Preparation

HCT, HR, SV, CO and BP data from an individual subject are used for comparing generated prediction envelope with a test subject to see how the simulated subjects are capable in replicating test subjects. The data should be recorded from a subject under fluid perturbation. In this work, a 3-point median filter is applied to the collected physiological data. An individual sample dataset is included in the software for demonstration purposes. The dataset includes time instants and values for 3-hour median-filtered physiological data, fluid infusion, hemorrhage, and urine collected from a sheep subject undergoing a simulated hemorrhage [2].

4. Software Package Content

The software package includes 3 hours of physiological data, hemorrhage profile, fluid infusion profile, and urine from an individual subject that were used for predictive capability assessment. The package also includes calibrated parameters from 26 individual animal datasets used for leave-one-out cross validation of the example data. The mathematical model is presented in Simulink “CVS_Model.mdl”. Virtual cohort generation includes a main code “Cohort_Generation.m” that generates prediction envelopes for all physiological variables using the compartment method presented in [1]. Normalized interval score [1] for the generated prediction envelope was also included in the software package. The codes are presented in the appendix.

5. Input and Output of the Software Code

The input and output of the software code for generating prediction envelopes are as follows:
Inputs: i) time instants for measured physiological data “Hemo_Time”, ii) median-filtered measured physiological data “Hemo_HCT”, “Hemo_HR”, “Hemo_SV”, “Hemo_CO”, “Hemo_BP”, iii) fluid infusion “Infusion_inp”, iv) hemorrhage “Hemorrhage_inp”, and v) urine “Urine_inp” all provided for the test subject in “Example_Data.mat”, and vi) calibrated parameters from 26 animal subjects used for generating prediction envelope against the test subject provided by “Calibrated_Parameters.mat”. “Calibrated_Parameters.mat” is a 26x22 matrix, where each row includes calibrated model parameters for each training animal dataset. Outputs: prediction envelope plots and normalized interval score for generated envelopes, as well as the best simulated subject [1].

6. Mathematical Model Parameters

BV parameters	A1	BV state parameter for infusion
	A2	BV state parameter for blood loss
	B_Gain	Infusion distribution factor
	B_Loss	Blood loss distribution factor
	Kp	BV proportional control gain
HR parameters	G1	Constant gain for transient decrease in HR to infusion
	G2	Constant gain for transient increase in HR to blood loss
	G3	Constant gain for long term reference increase in HR to blood loss
	Pow1	Power term for transient decrease in HR to infusion
	Pow2	Power term for long term reference increase in HR to blood loss

	Kp4	HR model proportional control gain for long term increase in HR to loss
	Ki4	HR model integral control gain for long term increase in HR to loss
SV parameters	Gsv1	Proportional gain relating SV to BV
	Gsv2	Proportional gain relating SV to HR
	A3	SV model time constant
	SV_Tar	Target steady state SV
	Kp2	SV model proportional control gain
BP parameters	A4	BP model time constant
	Kp3	BP model proportional control gain
	BP_Tar	Target steady state BP
Baseline parameters	Hct0	Baseline hematocrit
	BV0	Baseline BV
	HR0	Baseline HR
	SV0	Baseline SV
	TPR0	Baseline total peripheral resistance

7.

7. Mathematical Model Use Conditions

The mathematical model is intended for use with CVS physiological data collected under hemorrhage and fluid infusion. The physiological data subject to only fluid infusion and in absence of hemorrhage could lead to different physiological state, e.g., fluid overload, which should not be used with the developed mathematical model. Data from sheep subjects were used in this study. Each animal subject underwent hemorrhage and fluid infusion, which lasted for 180 min. At the start of each animal study, a 25 ml/kg hemorrhagic shock was induced in the subjects which lasted for 15 min. At 30 min, fluid infusion was started which continued till the end of the experiment. This infusion was performed for resuscitation with a target mean arterial pressure of 90 mmHg. At 50 and 70 min marks, two small 5 ml/kg hemorrhagic shocks were induced, each lasted for 5 min. For more information about the animal study please refer to [2].

8. References

[1] Yekanth Ram Chalumuri, Ghazal Arabidarrehdor, Ali Tivay, Catherine M Sampson, Muzna Khan, Michael Kinsky, George C Kramer, Jin-Oh Hahn, Christopher G Scully, and Ramin Bighamian, A Lumped-Parameter Model of the Cardiovascular System Response for Evaluating Automated Fluid Resuscitation Systems, IEEE Access, 12, pp. 62511-62525 (2024).

[2] Abraham Rafie, Paul Rath, Michael Michell, Robert Kirschner, Donald Deyo, Donald Prough, James Grady, George Kramer, Hypotensive Resuscitation of Multiple Hemorrhages using Crystalloid and Colloids, Shock 22, 262-269 (2004).

Appendix

A1: CVS Mathematical Model Simulink:

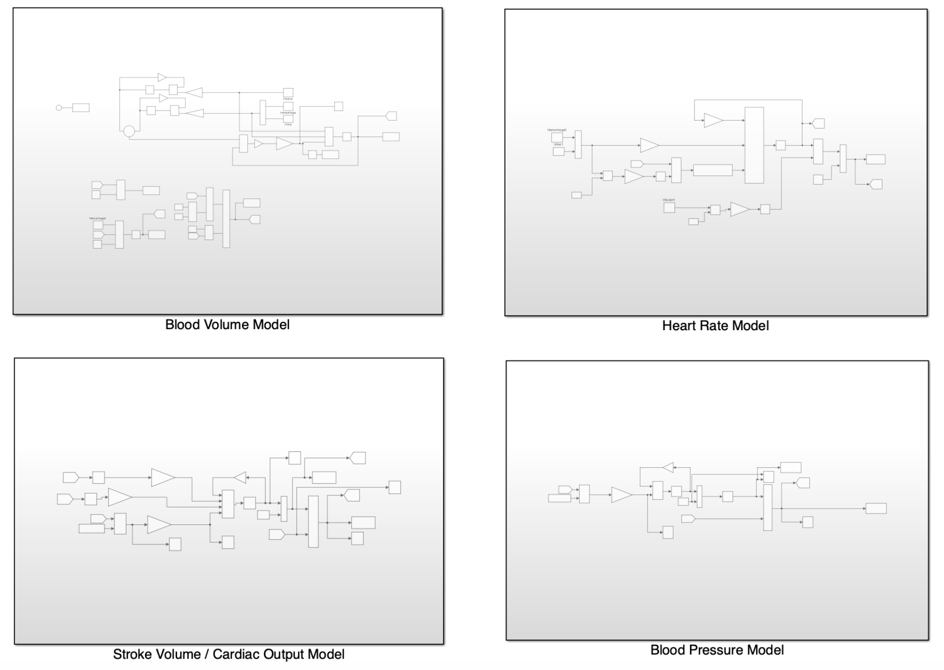

Figure 1: The entire model with 4 sub-models of BV, HR, SV (CO), and BP

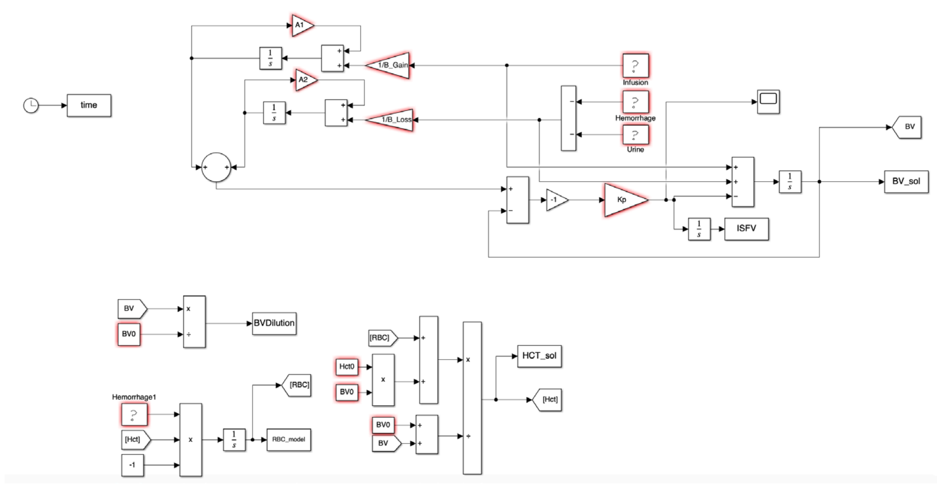

Figure 2: The BV sub-model

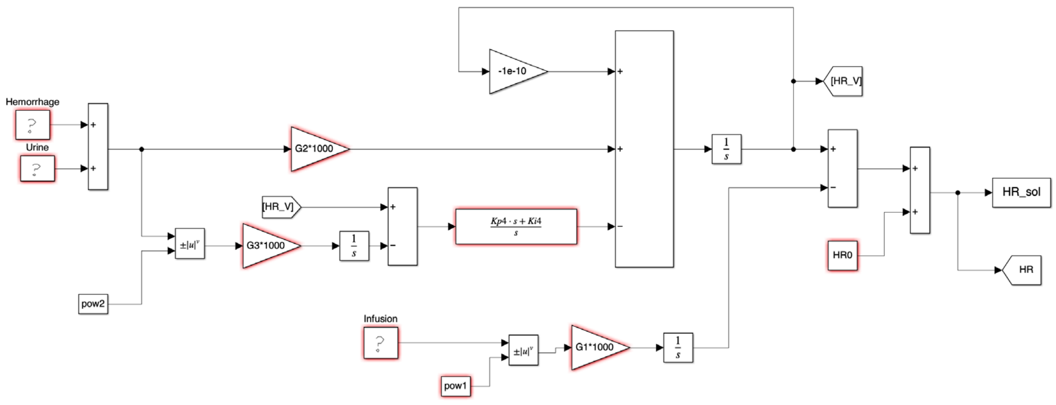

Figure 3: The HR sub-model

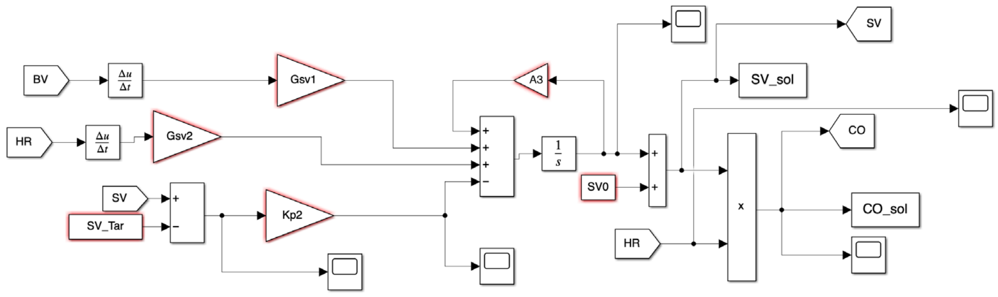

Figure 4: The SV (CO) sub-model

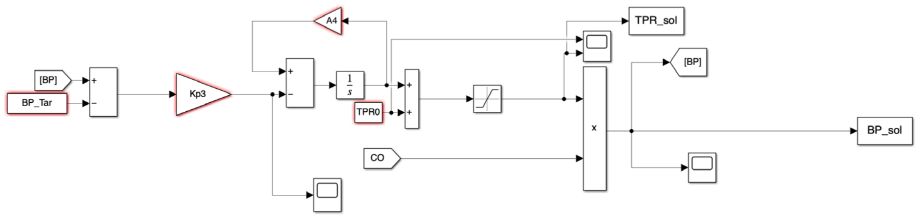

Figure 5: The BP sub-model

A2: Cohort Generation Code:

```
%{ Author Ramin Bighamian.

For questions, contact ramin.bighamian@fda.hhs.gov

BV and HR components of the model are published in the following papers:

BV model: Bighamian R, Hahn JO, Kramer G, Scully C. Accuracy assessment methods for physiological model selection toward evaluation of closed-loop controlled medical devices. PLoS One. 2021 Apr 30;16(4):e0251001. doi: 10.1371/journal.pone.0251001. PMID: 33930095; PMCID: PMC8087034.

HR model: Kanal V, Pathmanathan P, Hahn JO, Kramer G, Scully C, Bighamian R. Development and validation of a mathematical model of heart rate response to fluid perturbation. Sci Rep. 2022 Dec 12;12(1):21463. doi: 10.1038/s41598-022-25891-y. PMID: 36509856; PMCID: PMC9744837.

 If you found this software useful, please consider citing our publications above.

Public domain license FDA Software Disclaimer: This software and documentation (the "Software") were developed at the Food and Drug Administration (FDA) by employees of the Federal Government in the course of their official duties. Pursuant to Title 17, Section 105 of the United States Code, this work is not subject to copyright protection and is in the public domain. Permission is hereby granted, free of charge, to any person obtaining a copy of the Software, to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, or sell copies of the Software or derivatives, and to permit persons to whom the Software is furnished to do so. FDA assumes no responsibility whatsoever for use by other parties of the Software, its source code, documentation or compiled executables, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. Further, use of this code in no way implies endorsement by the FDA or confers any advantage in regulatory decisions. Although this software can be redistributed and/or modified freely, we ask that any derivative works bear some notice that they are derived from it, and any modified versions bear some notice that they have been modified. %}

clc
clear
close all

numsim = 10000;%number of all simulations
rand_IC = 0.1;%scaling factor for the baseline parameters normal distribution sampling

SamplingTime = 0.01; %sampling time for simulations

data = load('Example_Data'); %load one representative susjetc

Hemo_Time = data.DATASET.Measurements.HCT.Times; %Time and physioloigcal variables data
Infusion_inp = data.DATASET.Inputs.Infusion.Values;
Hemorrhage_inp = data.DATASET.Inputs.Hemorrhage.Values;
Urine_inp = data.DATASET.Inputs.UO.Values;
Hemo_HCT = 100*data.DATASET.Measurements.HCT.Values;
Hemo_CO = data.DATASET.Measurements.CO.Values;
```

```
Hemo_HR = data.DATASET.Measurements.HR.Values;
Hemo_SV = data.DATASET.Measurements.SV.Values/1000;
Hemo_MAP = data.DATASET.Measurements.MAP.Values;

%baseline values for BV,SV,Hematocrit
BV0_aux = data.DATASET.BV0;
SV0_aux = Hemo_SV(1);
Hct0_aux = Hemo_HCT(1);

simulation_time = (0:SamplingTime:180)'; %time vector for Simulink

%% load calibrated parameters from other 26 subjects
Par_aux = load('Calibrated_Parameters').Par_aux;

%% Sampling the parameters for the cohort
[A, B, C, D]=ndgrid(1:size(Par_aux,1),1:size(Par_aux,1),1:size(Par_aux,1),1:size(Par_aux,1));
A1=reshape(A,[],1);
B1=reshape(B,[],1);
C1=reshape(C,[],1);
D1=reshape(D,[],1);
d = [];
d=[A1,B1,C1,D1];
Cohort_1 = [];
for ii = 1:length(d) %making mixing virtual subjects
   Cohort_1(ii,:) = [Par_aux(d(ii,1),1:5) Par_aux(d(ii,2),6:10) Par_aux(d(ii,3),11:14) Par_aux(d(ii,4),15:22)];
end

[A, B, C]=ndgrid(1:size(Par_aux,1),1:size(Par_aux,1),1:size(Par_aux,1));
A1=reshape(A,[],1);
B1=reshape(B,[],1);
C1=reshape(C,[],1);
d = [];
d=[A1,B1,C1];
length(d)
Cohort_2 = [];
for ii = 1:length(d) %making average virtual subjects
   Cohort_2(ii,:) = (Par_aux(d(ii,1),:)+Par_aux(d(ii,2),:)+Par_aux(d(ii,3),:))/3;
end
ii;

Cohort = [Cohort_1;Cohort_2];%combining mixing and average cohorts of virtual subjects
n_size = length(Cohort);
%%
jj = 1;
jj_mat = randperm(n_size, numsim);
ccc = 1;
subject_par = [];
while jj <=numsim
   jj

   %specify model parameters from the virtual subjects
   x = Cohort(jj_mat(jj),:);
   A1=x(1);A2=x(2);B_Gain=x(3);B_Loss=x(4);Kp=x(5);
   Gsv1=x(6);Gsv2=x(7);A3=x(8);SV_Tar = x(9);Kp2=x(10);
```

```
    Kp3=x(11);A4=x(12);TPR0=x(13);BP_Tar=x(14);
    G1=x(15);G2=x(16);pow1=x(17);pow2=x(18);G3=x(19);Kp4=x(20);Ki4=x(21);HR0=x(22);

    % randomize baseline valuesaround the data baseline
    HR0 = Hemo_HR(1)+rand_IC*Hemo_HR(1)*randn;
    BV0 = BV0_aux+rand_IC*BV0_aux*randn;
    SV0 = SV0_aux+rand_IC*SV0_aux*randn;
    Hct0 = Hct0_aux+rand_IC*Hct0_aux*randn;
    TPR0 = (Hemo_MAP(1)/Hemo_CO(1))+rand_IC*(Hemo_MAP(1)/Hemo_CO(1))*randn;

    subject_par(jj,:) = [A1 A2 B_Gain B_Loss Kp  Gsv1 Gsv2 A3 SV_Tar Kp2 ...
     Kp3 A4 TPR0 BP_Tar G1 G2 pow1 pow2 G3 Kp4 Ki4 HR0 BV0 SV0 Hct0]; %subject model parameters


    sim('CVS_Model');
    response_BP(jj,:) = interp1(simulation_time,BP_sol,Hemo_Time);%down sample data to the times that physiological data
measurements are available
    response_CO(jj,:) = interp1(simulation_time,CO_sol,Hemo_Time);
    response_HCT(jj,:) = interp1(simulation_time,HCT_sol,Hemo_Time);
    response_SV(jj,:) = interp1(simulation_time,SV_sol,Hemo_Time);
    response_HR(jj,:) = interp1(simulation_time,HR_sol,Hemo_Time);

    HCT_sol = [];
    SV_sol = [];
    CO_sol = [];
    BP_sol = [];

    jj = jj+1;
end
%%
for kk = 1:numsim %compute normalized root mean squared error (NRMSE)
    HCT_rmse_nor(kk,1) = sqrt(mean((response_HCT(kk,:) - Hemo_HCT').^2))/(mean(Hemo_HCT));
    SV_rmse_nor(kk,1) = sqrt(mean((response_SV(kk,:) - Hemo_CO'./Hemo_HR').^2))/mean(Hemo_CO./Hemo_HR);
    HR_rmse_nor(kk,1) = sqrt(mean((response_HR(kk,:) - Hemo_HR').^2))/mean(Hemo_HR);
    CO_rmse_nor(kk,1) = sqrt(mean((response_CO(kk,:) - Hemo_CO').^2))/mean(Hemo_CO);
    BP_rmse_nor(kk,1) = sqrt(mean((response_BP(kk,:) - Hemo_MAP').^2))/mean(Hemo_MAP);

    %compute normalized mean absolute error (NMAE)
    HCT_MAE_nor(kk,1) = mean(abs(response_HCT(kk,:) - Hemo_HCT'))/(mean(Hemo_HCT));%Mean absolute error
    SV_MAE_nor(kk,1) = mean(abs(response_SV(kk,:) - Hemo_CO'./Hemo_HR'))/mean(Hemo_CO./Hemo_HR);
    HR_MAE_nor(kk,1) = mean(abs(response_HR(kk,:) - Hemo_HR'))/mean(Hemo_HR);
    CO_MAE_nor(kk,1) = mean(abs(response_CO(kk,:) - Hemo_CO'))/mean(Hemo_CO);
    BP_MAE_nor(kk,1) = mean(abs(response_BP(kk,:) - Hemo_MAP'))/mean(Hemo_MAP);

    %NRMSE and NAME averaged across all variables to find best simulated subjects
    mean_NRMSE (kk,1) =
(HCT_rmse_nor(kk,1)+SV_rmse_nor(kk,1)+HR_rmse_nor(kk,1)+CO_rmse_nor(kk,1)+BP_rmse_nor(kk,1))/5;
    mean_NMAE (kk,1) =
(HCT_MAE_nor(kk,1)+SV_MAE_nor(kk,1)+HR_MAE_nor(kk,1)+CO_MAE_nor(kk,1)+BP_MAE_nor(kk,1))/5;


    % max_NRMSE (kk,1) = HCT_rmse_nor(kk,1)+SV_rmse_nor(kk,1)+HR_rmse_nor(kk,1)+BP_rmse_nor(kk,1);
    max_NRMSE (kk,1) =
max([HCT_rmse_nor(kk,1);SV_rmse_nor(kk,1);HR_rmse_nor(kk,1);CO_rmse_nor(kk,1);BP_rmse_nor(kk,1)]);
```

```
    max_NMAE (kk,1) =
max([HCT_MAE_nor(kk,1);SV_MAE_nor(kk,1);HR_MAE_nor(kk,1);CO_MAE_nor(kk,1);BP_MAE_nor(kk,1)]);
end

% find best simulated subject in terms of NRMSE
min_ind = find(mean_NRMSE==min(mean_NRMSE));

figure(2) %plot the best simulated subject
subplot(231)
plot(simulation_time,1000*Infusion_inp,'LineWidth',2)
hold on
plot(simulation_time,-1000*Hemorrhage_inp,'--r','LineWidth',2)
set(gca,'XTick',0:30:180)
xlim([0 180])
ylabel('Fluid & Hemorrhage [ml/min]')
set(gca,'XTick',0:30:180)
legend('Fluid','Hemorrhage','Location','SouthEast')
grid on
box on

subplot(232)
plot(Hemo_Time,Hemo_HCT,'or','LineWidth',2)
hold on
plot(Hemo_Time,response_HCT(min_ind(1),:),'LineWidth',2)
ylabel('Hematocrit [%]')
xlim([0 180])
xticks([0 30 60 90 120 150 180])
grid on
box on

subplot(233)
plot(Hemo_Time,Hemo_HR,'or','LineWidth',2)
hold on
plot(Hemo_Time,response_HR(min_ind,:),'LineWidth',2)
ylabel('HR [bpm]')
xlim([0 180])
xticks([0 30 60 90 120 150 180])
grid on
box on

subplot(234)
plot(Hemo_Time,((Hemo_CO)./Hemo_HR),'or','LineWidth',2)
hold on
plot(Hemo_Time,response_SV(min_ind,:),'LineWidth',2)
ylabel('SV [l]')
xlim([0 180])
xticks([0 30 60 90 120 150 180])
xlabel('Time [min]')
grid on
box on

subplot(235)
plot(Hemo_Time,Hemo_CO,'or','LineWidth',2)
hold on
```

```
plot(Hemo_Time,response_CO(min_ind,:),'LineWidth',2)
ylabel('CO [lpm]')
xlim([0 180])
xticks([0 30 60 90 120 150 180])
xlabel('Time [min]')
grid on
box on

subplot(236)
plot(Hemo_Time,Hemo_MAP,'or','LineWidth',2)
hold on
plot(Hemo_Time,response_BP(min_ind,:),'LineWidth',2)
ylabel('BP [mmHg]')
xlim([0 180])
xticks([0 30 60 90 120 150 180])
xlabel('Time [min]')
grid on
box on

%%
for kk = 1:numsim %identify physiological simulations
   if min(response_HCT(kk,:)) >0 && max(response_HCT(kk,:))<50 && ...
         min(response_SV(kk,:))>0  && max(response_SV(kk,:))<0.1 &&  ...
         min(response_HR(kk,:))>0 && max(response_HR(kk,:))<300 && ...
         min(response_BP(kk,:))>0 && max(response_BP(kk,:))<150
      physiological_sims(kk) = 1;
   else
      physiological_sims(kk) = 0;
      max_NRMSE(kk,1) = 1000;
      max_NMAE(kk,1) = 1000;
   end
end

%identify relevant simulation (NRMSE<25%)
Relevant_sim = find(max_NRMSE<0.25);

figure(3) % plot prediction envelope made by relevant subjects
subplot(231)
plot(simulation_time,1000*Infusion_inp,'LineWidth',2)
hold on
plot(simulation_time,-1000*Hemorrhage_inp,'--r','LineWidth',2)
set(gca,'XTick',0:30:180)
xlim([0 180])
ylim([-100 80])
ylabel('Fluid & Hemorrhage [ml/min]')
set(gca,'XTick',0:30:180)
set(gca,'YTick',-100:20:80)
legend('Fluid','Hemorrhage','Location','SouthEast')
grid on
box on

subplot(232)
x = 1:1:length(Hemo_Time);
curve1_HCT = min(response_HCT(Relevant_sim,x));
```

```
curve2_HCT = max(response_HCT(Relevant_sim,x));
plot(Hemo_Time, curve1_HCT, 'b', 'LineWidth', 2);
hold on;
plot(Hemo_Time, curve2_HCT, 'b', 'LineWidth', 2);
x2 = [Hemo_Time', fliplr(Hemo_Time')];
inBetween = [curve1_HCT, fliplr(curve2_HCT)];
fill(x2, inBetween, 'g');
hold on
plot(Hemo_Time,Hemo_HCT,'or','LineWidth',2)
inside_interval_HCT = length(find(Hemo_HCT>=curve1_HCT' & Hemo_HCT<=curve2_HCT'))/length(Hemo_HCT);
ylabel('Hematocrit [%]')
xlim([0 180])
xticks([0 30 60 90 120 150 180])
grid on
box on

subplot(233)
curve1_HR = min(response_HR(Relevant_sim,x));
curve2_HR = max(response_HR(Relevant_sim,x));
plot(Hemo_Time, curve1_HR, 'b', 'LineWidth', 2);
hold on;
plot(Hemo_Time, curve2_HR, 'b', 'LineWidth', 2);
inBetween = [curve1_HR, fliplr(curve2_HR)];
fill(x2, inBetween, 'g');
hold on
plot(Hemo_Time,Hemo_HR,'or','LineWidth',2)
inside_interval_HR = length(find(Hemo_HR>=curve1_HR' & Hemo_HR<=curve2_HR'))/length(Hemo_HR);
ylabel('HR [bpm]')
xlim([0 180])
xticks([0 30 60 90 120 150 180])
grid on
box on

subplot(234)
curve1_SV = min(response_SV(Relevant_sim,x));
curve2_SV = max(response_SV(Relevant_sim,x));
plot(Hemo_Time, curve1_SV, 'b', 'LineWidth', 2);
hold on;
plot(Hemo_Time, curve2_SV, 'b', 'LineWidth', 2);
inBetween = [curve1_SV, fliplr(curve2_SV)];
fill(x2, inBetween, 'g');
hold on
plot(Hemo_Time,((Hemo_CO)./Hemo_HR),'or','LineWidth',2)
inside_interval_SV = length(find(Hemo_SV>=curve1_SV' & Hemo_SV<=curve2_SV'))/length(Hemo_SV);
ylabel('SV [l]')
xlim([0 180])
xticks([0 30 60 90 120 150 180])
xlabel('Time [min]')
grid on
box on

subplot(235)
curve1_CO = min(response_CO(Relevant_sim,x));
curve2_CO = max(response_CO(Relevant_sim,x));
```

```
plot(Hemo_Time, curve1_CO, 'b', 'LineWidth', 2);
hold on;
plot(Hemo_Time, curve2_CO, 'b', 'LineWidth', 2);
inBetween = [curve1_CO, fliplr(curve2_CO)];
fill(x2, inBetween, 'g');
hold on
plot(Hemo_Time,Hemo_CO,'or','LineWidth',2)
inside_interval_CO = length(find(Hemo_CO>=curve1_CO' & Hemo_CO<=curve2_CO'))/length(Hemo_CO);
ylabel('CO [lpm]')
xlim([0 180])
xticks([0 30 60 90 120 150 180])
xlabel('Time [min]')
grid on
box on

subplot(236)
curve1_BP = min(response_BP(Relevant_sim,x));
curve2_BP = max(response_BP(Relevant_sim,x));
plot(Hemo_Time, curve1_BP, 'b', 'LineWidth', 2);
hold on;
plot(Hemo_Time, curve2_BP, 'b', 'LineWidth', 2);
inBetween = [curve1_BP, fliplr(curve2_BP)];
fill(x2, inBetween, 'g');
hold on
plot(Hemo_Time,Hemo_MAP,'or','LineWidth',2)
inside_interval_BP = length(find(Hemo_MAP>=curve1_BP' & Hemo_MAP<=curve2_BP'))/length(Hemo_MAP);
ylabel('BP [mmHg]')
xlim([0 180])
xticks([0 30 60 90 120 150 180])
xlabel('Time [min]')
grid on
box on

%%
% min NRMSE and min NAME
min_max_NRMSE = min(max_NRMSE);
min_mean_NMAE = min(mean_NMAE);

%statistics of simulated subjects
physiological_sims_size = length(find(physiological_sims==1));
relevant_sims_size = length(Relevant_sim);
Average_percent_in_interval =
(inside_interval_HCT+inside_interval_HR+inside_interval_SV+inside_interval_CO+inside_interval_BP)/5;
%% compute normalized interval score for prediction envelopes
alpha = 0.05;
interval_score_HCT = [];
jj = 1;
for kk =1:length(Hemo_HCT)
    if Hemo_HCT(kk)<curve1_HCT(kk)
        interval_score_HCT(jj,1) = (2/alpha)*(curve1_HCT(kk)-Hemo_HCT(kk)) + curve2_HCT(kk)-curve1_HCT(kk);
    elseif Hemo_HCT(kk)>curve2_HCT(kk)
        interval_score_HCT(jj,1) = (2/alpha)*(Hemo_HCT(kk)-curve2_HCT(kk)) + curve2_HCT(kk)-curve1_HCT(kk);
    elseif Hemo_HCT(kk)>=curve1_HCT(kk) && Hemo_HCT(kk)<=curve2_HCT(kk)
        interval_score_HCT(jj,1) = curve2_HCT(kk)-curve1_HCT(kk);
```

```
  end
  jj = jj+1;
end
Normal_interval_score_HCT = (interval_score_HCT./Hemo_HCT)';

%HR
interval_score_HR = [];
jj = 1;
for kk =1:length(Hemo_HR)
  if Hemo_HR(kk)<curve1_HR(kk)
    interval_score_HR(jj,1) = (2/alpha)*(curve1_HR(kk)-Hemo_HR(kk)) + curve2_HR(kk)-curve1_HR(kk);
  elseif Hemo_HR(kk)>curve2_HR(kk)
    interval_score_HR(jj,1) = (2/alpha)*(Hemo_HR(kk)-curve2_HR(kk)) + curve2_HR(kk)-curve1_HR(kk);
  elseif Hemo_HR(kk)>=curve1_HR(kk) && Hemo_HR(kk)<=curve2_HR(kk)
    interval_score_HR(jj,1) = curve2_HR(kk)-curve1_HR(kk);
  end
  jj = jj+1;
end
Normal_interval_score_HR = (interval_score_HR./Hemo_HR)';

%SV
interval_score_SV = [];
jj = 1;
for kk =1:length(Hemo_SV)
  if Hemo_SV(kk)<curve1_SV(kk)
    interval_score_SV(jj,1) = (2/alpha)*(curve1_SV(kk)-Hemo_SV(kk)) + curve2_SV(kk)-curve1_SV(kk);
  elseif Hemo_SV(kk)>curve2_SV(kk)
    interval_score_SV(jj,1) = (2/alpha)*(Hemo_SV(kk)-curve2_SV(kk)) + curve2_SV(kk)-curve1_SV(kk);
  elseif Hemo_SV(kk)>=curve1_SV(kk) && Hemo_SV(kk)<=curve2_SV(kk)
    interval_score_SV(jj,1) = curve2_SV(kk)-curve1_SV(kk);
  end
  jj = jj+1;
end
Normal_interval_score_SV = (interval_score_SV./Hemo_SV)';

%CO
interval_score_CO = [];
jj = 1;
for kk =1:length(Hemo_CO)
  if Hemo_CO(kk)<curve1_CO(kk)
    interval_score_CO(jj,1) = (2/alpha)*(curve1_CO(kk)-Hemo_CO(kk)) + curve2_CO(kk)-curve1_CO(kk);
  elseif Hemo_CO(kk)>curve2_CO(kk)
    interval_score_CO(jj,1) = (2/alpha)*(Hemo_CO(kk)-curve2_CO(kk)) + curve2_CO(kk)-curve1_CO(kk);
  elseif Hemo_CO(kk)>=curve1_CO(kk) && Hemo_CO(kk)<=curve2_CO(kk)
    interval_score_CO(jj,1) = curve2_CO(kk)-curve1_CO(kk);
  end
  jj = jj+1;
end
Normal_interval_score_CO = (interval_score_CO./Hemo_CO)';

%BP
interval_score_BP = [];
jj = 1;
for kk =1:length(Hemo_MAP)
```

```
    if Hemo_MAP(kk)<curve1_BP(kk)
        interval_score_BP(jj,1) = (2/alpha)*(curve1_BP(kk)-Hemo_MAP(kk)) + curve2_BP(kk)-curve1_BP(kk);
    elseif Hemo_MAP(kk)>curve2_BP(kk)
        interval_score_BP(jj,1) = (2/alpha)*(Hemo_MAP(kk)-curve2_BP(kk)) + curve2_BP(kk)-curve1_BP(kk);
    elseif Hemo_MAP(kk)>=curve1_BP(kk) && Hemo_MAP(kk)<=curve2_BP(kk)
        interval_score_BP(jj,1) = curve2_BP(kk)-curve1_BP(kk);
    end
    jj = jj+1;
end
Normal_interval_score_BP = (interval_score_BP./Hemo_MAP)';

%normalized interval score for all physiological variables
Normal_interval_score =
[Normal_interval_score_HCT;Normal_interval_score_HR;Normal_interval_score_SV;Normal_interval_score_CO;Normal_interv
al_score_BP]';
mean_Normal_interval_score = mean(Normal_interval_score);
```

A3: Example Prediction Envelope Results for 10000 Simulation Sample Size:

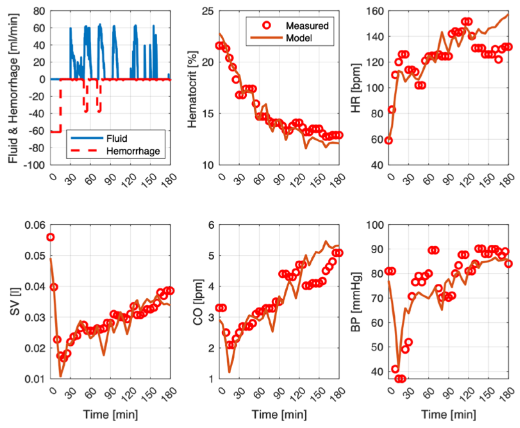

Figure 6: An example best simulated subject with minimum averaged NRMSE across all physiological variables

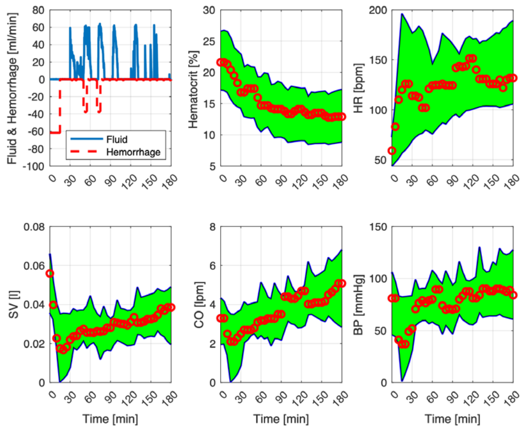

Figure 7: An example prediction envelope generated by relevant physiological simulations