Evaluating *Reducing Risks in Type 1 Diabetes Using H_∞ Control*

Ravish Kumar (14542), Snehil Verma (14700), Sahil Vartak (14789)

1. Introduction

The following results have derived from research paper titled *Reducing Risks in Type 1 Diabetes Using H_∞ Control (IEEE Transactions on Biomedical Engineering, vol. 61, no. 12, December 2014)* with the help of MATLAB R2016 application.

The class came across the paper when looking for learn application of Robust Control System in everyday life. The objective of the paper, as we have understood, is to automatically control the blood glucose level in T1DM patients which is a long standing problem and a constant threat to upcoming generation. While approaching the problem, we had two tools: Simulink and MATLAB. We chose the latter being well acquainted with the application. The following Matlab code is of the plant, *Adult#j* (as named in paper) and *gen_plant* (named in code) depicting its bode plot which turns out to be similar to the papers result ensuring the meticulousness of our code. As there are millions of people from whole over the world so we cant model the plant for each individual. Hence, for more practicality of the scheme, paper designed a 3^{rd} order plant model with only one tuning parameter known as a *priori* patient data. *TDI_r* was taken to be *180*, the average of general values found on surfing and similar goes to the other values. These values were assumed to meet the simplicity keeping care of practicality of the system. For the first part of submissions, we have derived the state space form of K for the plant, using *mixsyn* function. By using *ss2tf*, we have converted the state space form into transfer function. We used *sigma* function for getting the bode plots of sensitivity function and complementary sensitivity function. By assuming SM=0 and IFL=1, our required function will be T as given in code and its curves are presented.

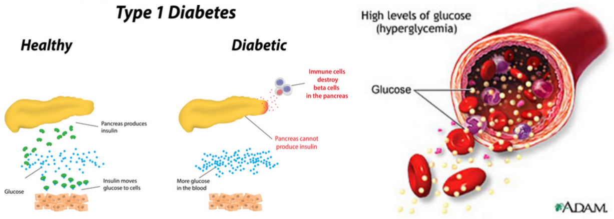
Type 1 Diabetes
Healthy
Diabetic
Pancreas produces insulin
Glucose
Insulin moves glucose to cells
Immune cells destroy beta cells in the pancreas
Pancreas cannot produce insulin
More glucose in the blood
High levels of glucose (hyperglycemia)
Glucose
ADAM.

Figure 1. (a) Difference between healthy and affected cells (b) Effect of Hyperglycemia on blood cells

2. Understanding the Biological side of the problem

TYPE1 diabetes mellitus (*T1DM*) is caused when the body's immune system turns against itself and mistakenly attacks healthy cells. Thus it is classified as autoimmune disease and the healthy cells here are the the pancreatic β-cells which in turn causes deficiency of insulin. The huge problem to tackle in this disease is the self-monitoring of blood glucose throughout individual's life. We attempt to tackle this problem by achieving high closed-loop performance . People with type 1 diabetes always need to use insulin, but treatment can lead to low BG which is known as *hypoglycemia*.

Hypoglycemia, also known as low blood sugar, is when blood sugar decreases to below normal levels. Severe cases can lead to unconsciousness and are treated with intravenous glucose. *Hyperglycemia* is a condition in which an excessive amount of glucose circulates in the blood plasma. This is generally a blood sugar level higher than 11.1 mmol/l (200 mg/dl), but symptoms may not start to become noticeable until even higher values such as 1520 mmol/l (∼250-300 mg/dl).

The UVA/Padova T1DMS is simulation software that ameliorates users to design and test treatment of in-silico subjects with Type 1 Diabetes Mellitus. The UVA/Padova T1DMS allows the user to perform experiments which examine and tune control algorithms for insulin dosing strategies and also guide and focus the emphasis of protocol designs for clinical studies.

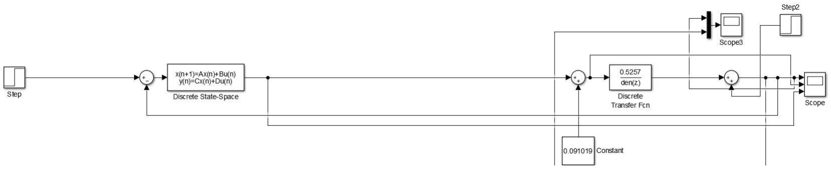

Figure 2. Block diagram without IFL

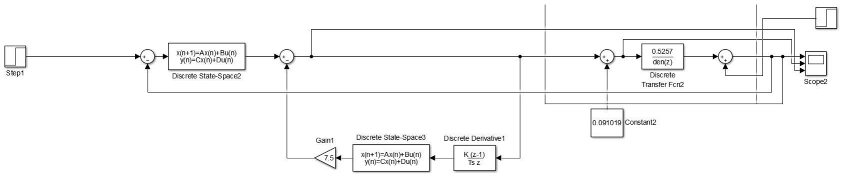

Figure 3. Block diagram with IFL

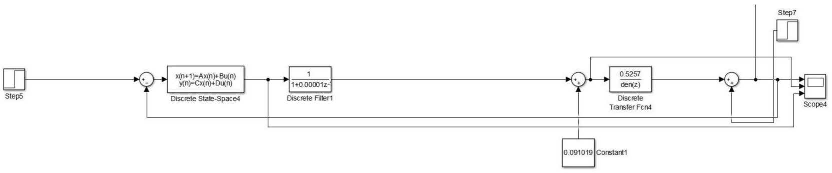

Figure 4. Block diagram with Discrete filter w/o IFL

3. MATLAB Code

```
clc;
clear all;

% from the given dataset obtaining the value of IS_i
Avg_CF = 1;
Avg_CR = 1;
TDI_r = 180;    % randomly choosed average value according to internet
Alpha = 0.5;    % given avg value in paper
Beta = 0.5;     % given avg value in paper
C_av = Alpha*Avg_CF + Beta*Avg_CR; %calculation of C_av for IS_i

TDI_i = 151;    % Total daily insulin for random person saying Id_120
c = 1;          % choosing for simplicity

IS_i = TDI_r *c/(TDI_i*C_av); % formula

Ts = 600;       % sample time used 10min == 600 seconds

z = tf('z',Ts);% used for z-transform

% poles obtained from black_box method
p1 = 0.965;
p2 = 0.95;
p3 = 0.93;

c = 60/100*(1-p1)*(1-p2)*(1-p3)*Ts;
rj = 1800/TDI_i;
% writing general plant obtained from blackbox method
gen_plant = (c*rj*z^-3)/((1 -(z^-1)*p1)*(1-(z^-1)*p2)*(1-(z^-1)*p3));

```

```
% bode(gen_plant); % uncomment for bode of gen_plant

% Hinfinity controller designing code

% weight on senstivity matrix as given in paper
Wp = (0.01434*z - 0.01365)/(z - 0.9993);
% weight on k*senstivity matrix as given n paper
Wdel_i = IS_i*1500*(0.001992*(z-1))/(z-0.992);
[Khinf,ghinf,GAM] = mixsyn(gen_plant, Wp,Wdel_i,[])
% Khinf will be in state space form and will give the desired controller

% this part of code is just to set the limits on Ms and Mt
L = gen_plant*Khinf; %  open loop transfer function.
S = inv(1+L);         % defining senstivity function
T = L*S;              % defining complementary senstivity function

% To plot the following output

sigma(S,'g',T,'r',GAM/Wp,'g-.',GAM*gen_plant/ss(Wdel_i),'r-.');
legend('S','T','GAM/Wp','GAM*gen_plant/ss(Wdel_i)','Location','Northeast');

% obtaining the controller from state space form
[b,a] = ss2tf(Khinf.A,Khinf.B,Khinf.C,Khinf.D);
% a and b will be used in determining the controller in z-transform

Zcon = ss(Khinf);             % controller in z-transformed form
zpk(Zcon);                    % factorising Z_controller
rZcon = balred(Zcon,3);       % reduced Z_controller into order 3
zpk(rZcon)                    % factorising reduced Z_controller

% obtaining step response of closed loop system (uncomment next line)
% step(gen_plant*Khinf/(1+gen_plant*Khinf));

% To get a feel of gain margin & phase margin of open loop TF from r to g

Marg = allmargin(gen_plant * Khinf);

% Khinf will be found from the mixsyn function in the discrete time domain
% but in state-space form. In z transform we have to use ss2tf function
% that will give b and a as follows:

% b = [0.0007   -0.0019    0.0011    0.0014   -0.0018    0.0006]
% a = [1.0000   -4.2934    7.3537   -6.2774    2.6677   -0.4505]

% using this we can write the controller in z-transformed form as follows:

%   0.00067949 (z+1.002) (z-0.992) (z-0.965) (z-0.95) (z-0.93)
%  -------------------------------------------------------------
%    (z-0.6034) (z-0.992) (z-0.9993) (z^2 - 1.699z + 0.7532)

% 3rd order reduced controller in z-transformed form:

%  0.00084182 (z+0.1629) (z^2 - 1.952z + 0.9527)
% -----------------------------------------------
%        (z-0.9993) (z^2 - 1.626z + 0.7324)

% results of Marg function are

% GainMargin : [5.1585 2.2600e+04]
% GMFrequency: [2.8741e-04 0.0037]
% PhaseMargin: 75.7700
% PMFrequency: 5.5464e-05
% DelayMargin: 39.7389
% DMFrequency: 5.5464e-05
% Stable: 1

% Insulin feedback loop
% Designing SIM
% Defining parameters
kd  = 0.0164/60;
ka1 = 0.0018/60;
ka2 = 0.0182/60;
m1  = 0.19/60;
m2  = 0.484/60;
m3  = 0.285/60;
m4  = 0.194/60;
Vi  = 0.05;
mu  = 7.5;

A = [-(kd+ka1),0,0,0;kd,-ka2,0,0;0,0,-(m1+m3),m2;ka1,ka2,m1,-(m2+m4)];
B = [1;0;0;0];
C = [0 0 0 1/Vi];
D = 0;

sys = ss(A,B,C,D);
% function used for convertion of continuos TF to discrete TF
opt = c2dOptions('Method','tustin','FractDelayApproxOrder',3);
sysd1 = c2d(sys,600,opt) % SIM == sysd1
% IFL = 1/(1+mu*SIM);
```

4. Project development and experience

- First of all we were unaware that Control Systems can be used in biological aspects too. So we learnt that this is also an area of research where control engineers can make their remarkable contributions.
- We learnt Simulink for simulation of our plant model, to observe the working of controllers and their behaviour on encountering disturbance in insulin levels.
- We learnt variety of new commands used in the MATLAB and Octave, especially used in robust control like ss2tf, balred, sigma, mixsyn, marg and zpk.
- We learnt black box method i.e. finding a plant model of unknown system by techniques as discussed in the class.
- We saw experimentally the effects on performance and control effort of the controller due to change in weights which supported the theory and clarified the concepts.
- We learnt discrete-time domain representation of transfer function. For this we skimmed through bilinear transformation, Z-transformation and relation between Z-plane and S-plane.
- Nyquist frequency bounds :- The choice of sampling time depends on nyquist frequency. Calculated maximum allowed sampling time was 52 mins for our plant and we assumed it to be 10 mins which is feasible.
- We learnt the reason behind the non-existing nature of bode plot after certain frequency i.e. π/Ts = 0.5e-3 rad/s.
- At first we got controller as a 5^{th} order transfer function but since it was unfeasible in terms of order, so we reduced the order of controller to 3^{rd} order.

5. Plots

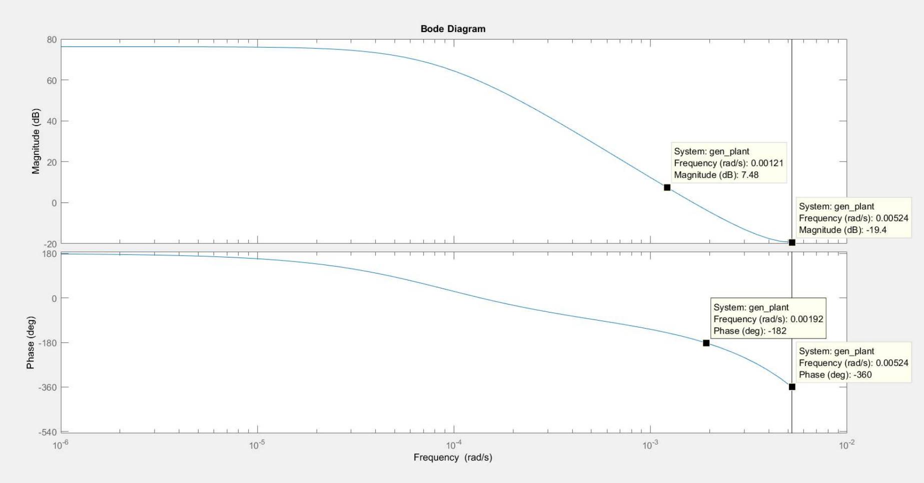

Figure 5. Bode plot of gen_plant

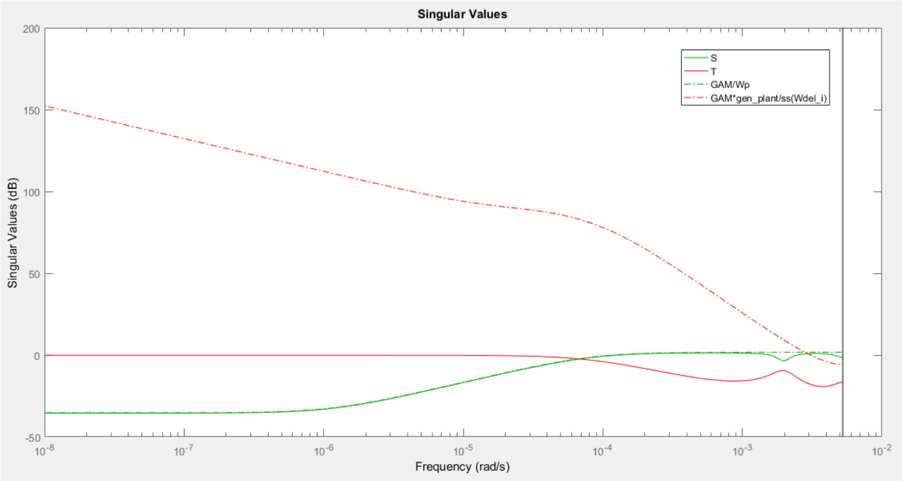

Figure 6. Bode plot of different functions via sigma()

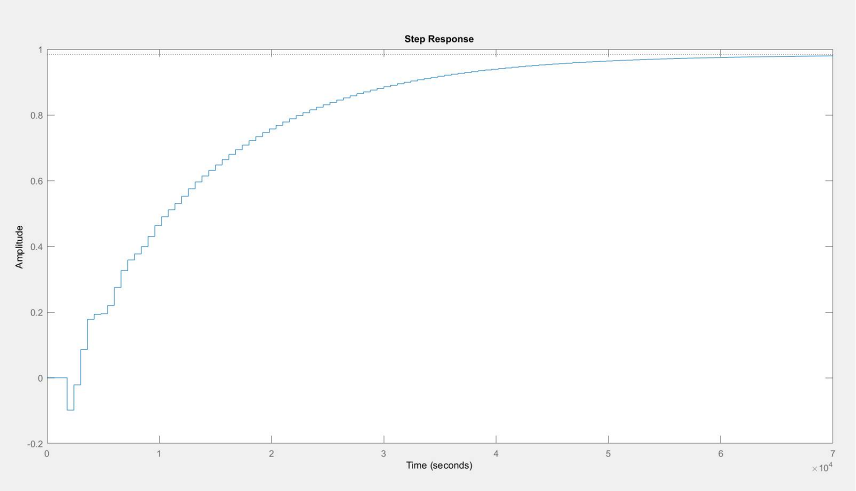

Figure 7. Step response of closed loop transfer function

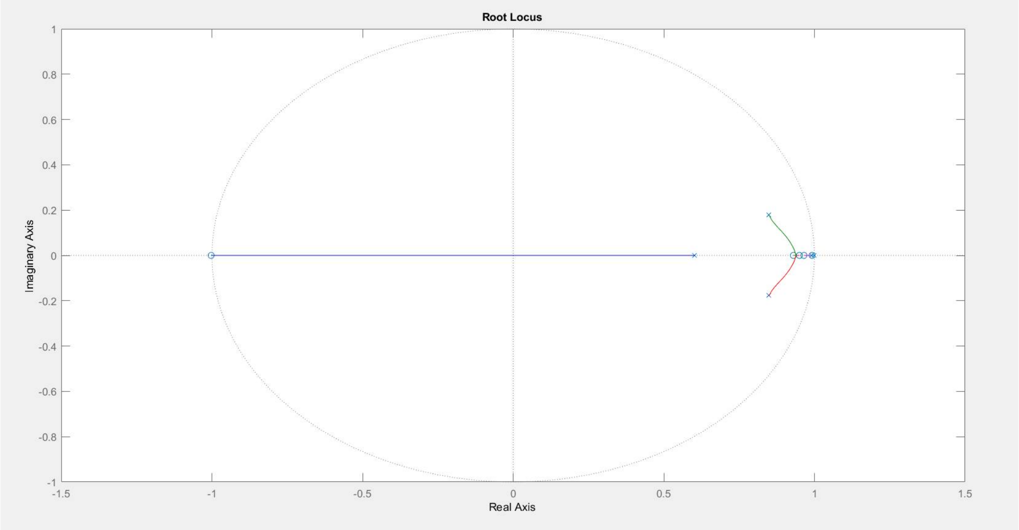

Figure 8. Root locus of controller Kh_{inf} in Z_{plane}

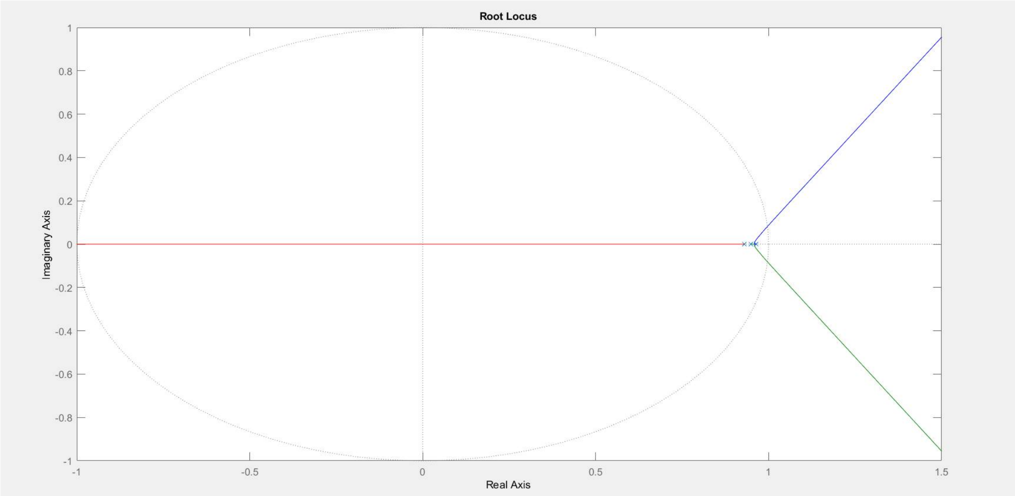

Figure 9. Root locus of gen_plant in Z_{plane}

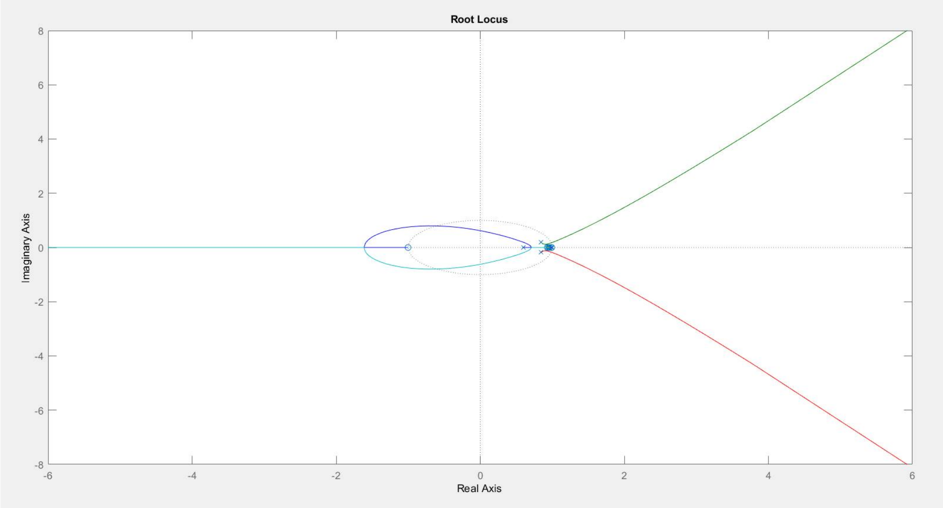

Figure 10. Root locus of gen_plant*Kh_{inf} in Z_{plane}

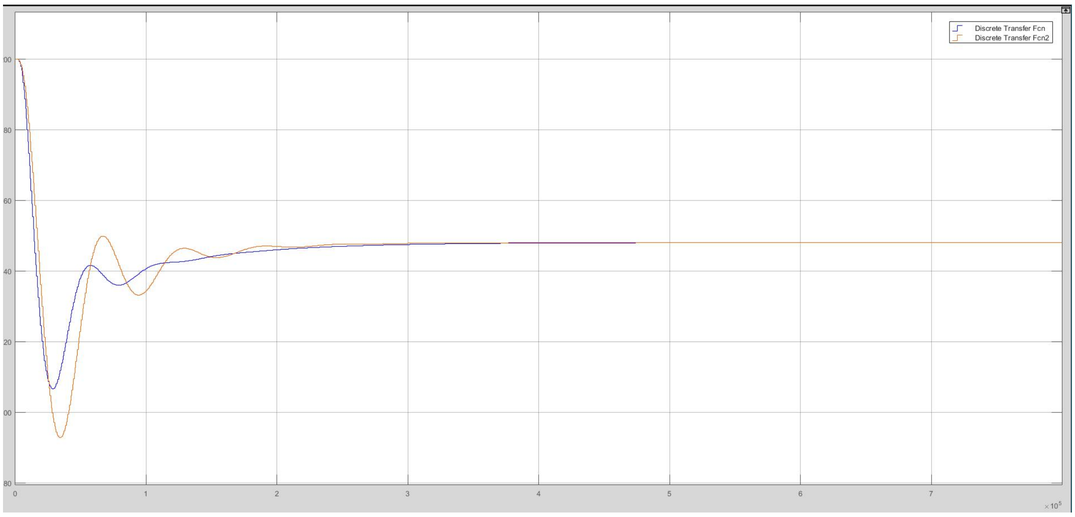

Figure 11. Comparison between close loop performances of block diagram with and without IFL

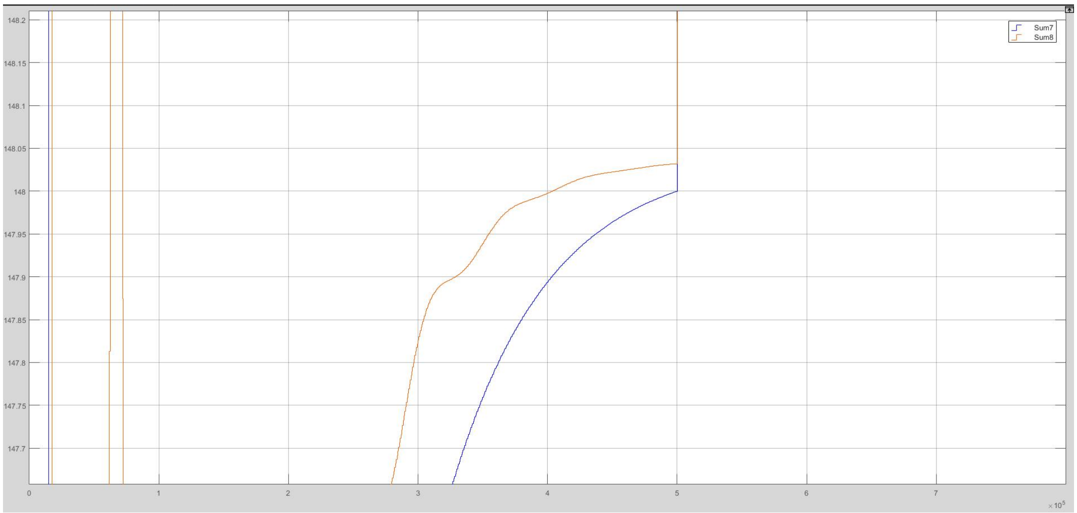

Figure 12. Showing e_{ss} in IFL is very low

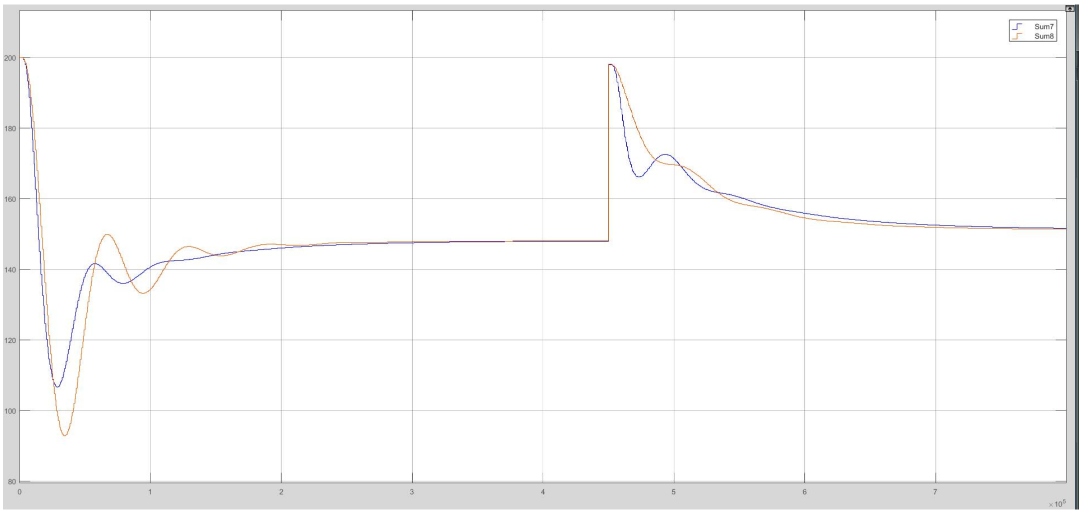

Figure 13. Disturbance with and without IFL

6. Problems faced

- Paper was not written in an efficient manner i.e. there was lack of some useful information regarding terms used in the paper such as it didnt show the presence of basal insulin which we need to add before giving input to Adult#j.
- We were first unaware of discrete time domain transfer function which cost us time.
- Safety Mechanism is still not clear to us, so we dropped safety mechanism from our simulation.
- The use of IFL is still questionable as IFL just reduced the steady state error while overshoot and settling time became worse as compared to the one without IFL block simulation.
- Settling time for disturbances is very high i.e. 4 to 5 hour, which we think is very high and inappropriate.

7. Results

- The bode plot of the open loop transfer function stopped at the frequency π/Ts which is approximately 0.5e-3.
- Basal insulin level for our plant is 0.091019 mg/dl, which we tuned from the simulation.
- Settling time of CL response was 4 to 5 hours.
- From the root locus plots of the plant model, controller and plant*controller we can decide their stabilities.
- To make sure that the input to the plant Adult#j is always non-negative we increased the weight on W$_{del}$. We increased the constant (given in paper) by a factor of 1500.
- IFL reduced the steady state error while overshoot and settling time became worse as compared to the one without IFL block simulation.
- After tuning the constants of discrete filter it seemed that the two curves (with or without filter) were overlapping each other but the filter smoothened the response.

8. Contributions in brief

Snehil Verma: Basic Programming, designing Insulin Feedback Loop (IFL), reduced order model of the controller, Simulink simulation, LaTeX presentation and documentation
Ravish Kumar: Basic Programming, brief explanation of paper to own group, weight optimization, designing Insulin Feedback Loop (IFL), Simulink block diagram designing and simulation, discrete filter design, documentation
Sahil Varthak: Basic Programming, Simulink simulation, collection of general information for documentation

Acknowledgement

We have taken efforts in this project. However, it would not had been possible without the kind support of many individuals and organisations. We would like to extend my sincere thanks to all of them. We are highly indebted of Prof. Ramprasad Potluri for his guidance, constant supervision as well as providing invaluable information about the project and supporting in completing the project. This effort would be of huge waste without thanking the developers of MATLAB which was a very necessary tool in completion of work. We would also like to express special gratitude to the writers of paper and making it available to anyone willing to work on it. Our thanks and appreciation go to our talented colleagues who were willing to help each other to best of their abilities.

References

1. IEEE Transactions on Biomedical Engineering, vol. 61, no. 12, December 2014 *Reducing Risks in Type 1 Diabetes Using H_∞ Control.*
2. ss2tf function *https://in.mathworks.com/help/matlab/ref/ss2tf.html.*
3. Bilinear transformation *https://en.wikipedia.org/wiki/Bilinear_transform.*
4. Sigma function *https://in.mathworks.com/help/robust/ref/mixsyn.html?requestedDomain=in.mathworks.com.*
5. Figure 1(a) *https://thumb9.shutterstock.com/display_pic_with_logo/636694/212467696/stock-vector-type-diabetes-labeled-diagram-212467696.jpg.*
6. Figure 1(b) *http://elbrooklyntaco.com/wp-content/uploads/2013/10/hyperglycemia.jpg.*
7. Model order reduction *https://www.mathworks.com/help/control/ref/balred.html.*
8. Control system toolbox functions *http://www.mathworks.com/help/control/functionlist.html.*