```
function [uk,ctrlstate] = PIDControl2(yk, U, ctrlPar, ctrlState)
%
% PIDControl()
%
% DESCRIPTION:
% This function implements a discretized proportional-integral-derivative
% (PID) controller for controlling the insulin flow rate. This differs from
% PIDControl by additionally having the insulin vector as input
%
% INPUT:
% yk          - Current blood glucose concentration
% U           - Insulin vector of both basal and bolus insulin
%
% ctrlPar     - vector of the following:
%                    * Ts              - Sampling time, 5 min
%                    * Kp              - Proportional gain
%                    * Ki              - Integrator gain
%                    * Kd              - Derivative gain
%                    * ybar            - The traget glucose concentration, y=108
%                    * ubar            - Nominal insulin flow rate
%                    * Ti              - Tuned parameters
%                    * Td              - Tuned parameters
%
% crtlState   - vector of the following:
%                    * Ik              - The integral term (Ik)
%                    * ykm1            - Previous glucose concentration, yk-1
%
% OUTPUT:
%   uk          - a vector of manipulated inputs
%   crtlstate   - the updated controller state
%
% PROJECT:
% Fagprojekt 2022
% A diabetes case study - Meal detection
%
% GENEREL:
% BSc                          : Mathematics and technology
% University                   : The Technical University of Denmark (DTU)
% Department                   : Applied Mathematics and Computer Science
%
% AUTHORS:
% Emma Victoria Lind
% Mariana de Sá Madsen
% Mona Saleem
%
% CONTACT INFORMATION
% s201205@student.dtu.dk
% s191159@student.dtu.dk
% s204226@student.dtu.dk
%


% Unpack control parameters
Ts      = ctrlPar(1);  %         Sampling time
Kp      = ctrlPar(2);  %         Proportional gain
%Ki      = ctrlPar(3); %         Integrator gain (not used since we calculate it in the↙
function line 68)
%Kd      = ctrlPar(4); %         Derivative gain (not used since we calculate it in the↙
```

```
function line 69)
ybar    = ctrlPar(5);  %          Target blood glucose concentration
ubar    = ctrlPar(6);  %          Nominal insulin flow rate
Ti      = ctrlPar(7);  %          Tuned parameters
Td      = ctrlPar(8);  %          Tuned parameters

% Unpack control state
Ik = ctrlState(1);    %           Value of integral at previous time step
ykm1 = ctrlState(2);  %           Previous observed glucose concentration

% Computing

ek = yk-ybar;              % Setpoint error

Ki = Kp * Ts/Ti;           % Helps controlling the steady state
Kd = Kp * Td/Ts;           % The top

Pk = Kp * ek;              % Proportional term. Controls how fast the error change

Ikp1 = Ik + Ki * ek;       % Integral term. The area of the error

Dk = Kd * (yk-ykm1);       % Derivative term. The top of the curve

uba = ubar + Pk + Ik + Dk; % Basal insulin flow rate

ubo = U;                   % Bolus insulin flow rate


% OUTPUT

% The controlled manipulated inputs at time step
uk = [uba,ubo];

% Controller state OUTPUT
ctrlstate = [Ikp1; yk];

end
```