

```

function [gaitResampledR,gaitResampledL] = resampleToPercentGait(continuousDataR,...
continuousDataL,continuosDataTime, hsR, hsL)
global F
nSamp = 100;
% resampleToPercentGaitCycle
% 1) we input data that reflects the time profile of the data (position,
% forces, dynamics, kinematics etc.)
% 2) using heel strikes we trim the continuos profile into a cell structure
% that includes the time profile of each stride (one stride per row)
% 3) identify the force time integral, peak heel strike and toe off forces

% break down into each stride hsR and hsL are in time, we want to find the
% time in continuosDataTime that is closest to the hs time, then we want to
% save these indices
for i = 1:length(hsR)
    ix = find(continuosDataTime < hsR(i),1,'last');
    if isempty(ix)
        continuosDataHsIdx(i) = NaN;
    else
        continuosDataHsIdx(i) = ix;
    end
end
cdhsi = continuosDataHsIdx;
for strd = 1:length(cdhsi)-1
    if ~isnan(cdhsi(strd)) && ~isnan(cdhsi(strd+1))
        sdR = continuousDataR(cdhsi(strd):cdhsi(strd+1));
        sdTime = continuosDataTime(cdhsi(strd):cdhsi(strd+1));
    else
        sdR = [];
        sdTime = [];
    end % nan strides will be empty
    strideDataR{strd,1} = sdR; %(~cellfun('isempty',sdR)); %remove empty cells
    strideDataTime{strd,1} = sdTime; %(~cellfun('isempty',sdR)); %remove empty cells
end
% resample each cell and writing to matrix
resampleR = NaN(size(strideDataR,1),100);
for ci = 1:size(strideDataR,1)
    a = strideDataR{ci,1};
    x = strideDataTime{ci,1};
    if ~isempty(x) && length(x) > 1
        xres = linspace(x(1),x(end), nSamp);
        y = interp1(x,a,xres);
        resampleR(ci,:) = y;
    else
        resampleR(ci,:) = NaN(1,nSamp);
    end
end
% LEFT
for i = 1:length(hsL)
    ix = find(continuosDataTime < hsL(i),1,'last');
    if isempty(ix)
        continuosDataHsIdx(i) = NaN;
    else
        continuosDataHsIdx(i) = ix;
    end
end
cdhsi = continuosDataHsIdx;
for strd = 1:length(cdhsi)-1
    if ~isnan(cdhsi(strd)) && ~isnan(cdhsi(strd+1))

```

```

        sdL = continuousDataL(cdhsi(strd):cdhsi(strd+1));
        sdTime = continuosDataTime(cdhsi(strd):cdhsi(strd+1));
    else
        sdL = [];
        sdTime = [];
    end % nan strides will be empty
    strideDataL{strd,1} = sdL; %(~cellfun('isempty',sdL)); %remove empty cells
    strideDataTime{strd,1} = sdTime; %(~cellfun('isempty',sdL)); %remove empty cells
end
% resample each cell and writing to matrix
resampleL = NaN(size(strideDataL,1),100);
for ci = 1:size(strideDataL,1)
    a = strideDataL{ci,1};
    x = strideDataTime{ci,1};
    if ~isempty(x) && length(x) > 1
        xres = linspace(x(1),x(end), nSamp);
        y = interp1(x,a,xres);
        resampleL(ci,:) = y;
    else
        resampleL(ci,:) = NaN(1,nSamp);
    end
end
gaitResampledR = resamplerR;
gaitResampledL = resampleL;
end

```

Not enough input arguments.

Error in resampleToPercentGait (line 15)
 for i = 1:length(hsR)