ENGG1811 Computing for Engineers

Ass2:

Getting Started Help for Part-2 (a, b, c, d)

Part-2(a)

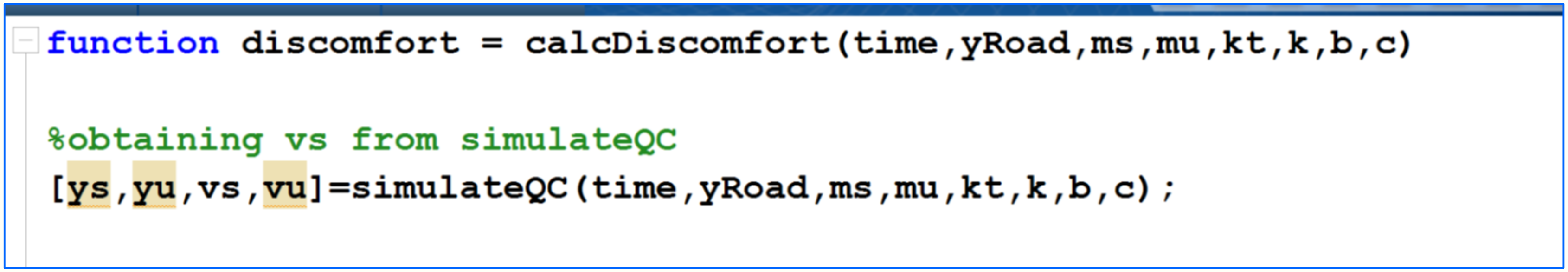
For this task, you are asked to write a Matlab function calcDiscomfort (which should be in a file called calcDiscomfort.m) which has the declaration:

function discomfort = calcDiscomfort(time, yRoad, ms, mu, kt, k, b, c)

- The 8 input parameters are **identical to** those for the function **simulateQC**.
- The function returns a scalar which measures the discomfort level for the given vehicle and suspension parameters.
- **The first step** of calcDiscomfort is to use the function simulateQC to perform a simulation, i.e., using the following line of code:

  [ys, yu, vs, vu] = **simulateQC**(time,yRoad,ms,mu,kt,k,b,c);

For example,



```
function discomfort = calcDiscomfort(time,yRoad,ms,mu,kt,k,b,c)

%obtaining vs from simulateQC
[ys,yu,vs,vu]=simulateQC(time,yRoad,ms,mu,kt,k,b,c);
```

- Once this is done, you can use the vector **vs** to determine the **acceleration**.

Part-2(a)

- Let us assume that the vector **vs** has **n** elements (how to calculate n? size(vs) ?)
- let us use **dt** to denote the time increment used in the vector time (how to calculate dt?)
- We can use **vs** to calculate the acceleration at **(n-1)** time instances:

a(i) = (vs(i+1)-vs(i))/dt

where i = 1, 2, ..., n-1

vs(i) is the *i*-th element of the vector vs and

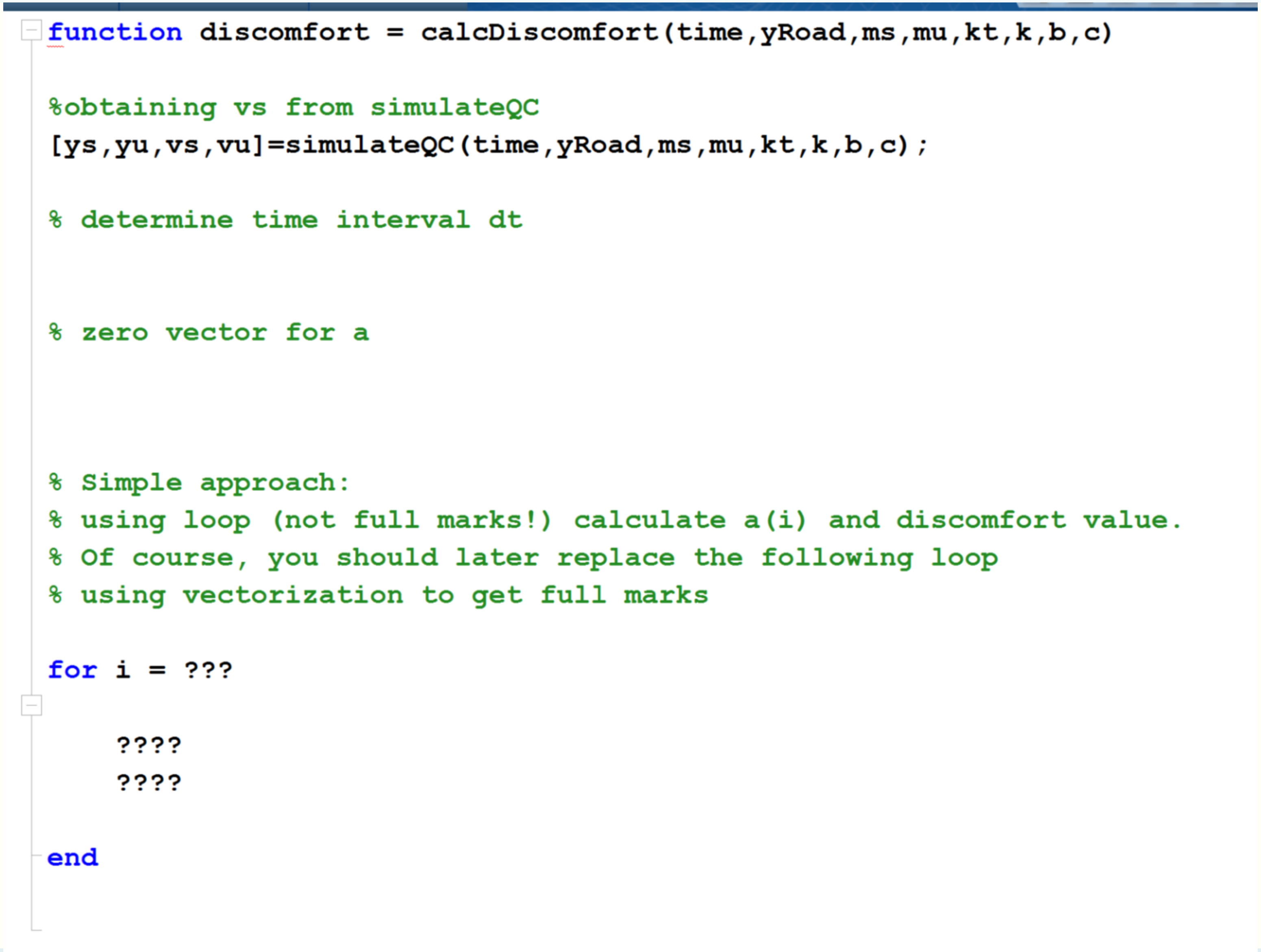
a(i) is the *i*-th element of the acceleration vector a.

The discomfort level is then given by

$$\textbf{discomfort} = a(1)^2 + a(2)^2 + \ldots + a(n-2)^2 + a(n-1)^2$$

This should be the output of the function calcDiscomfort. Intuitively, this calculation says the discomfort is higher if the acceleration is higher.

Part-2(a)



```
function discomfort = calcDiscomfort(time,yRoad,ms,mu,kt,k,b,c)

%obtaining vs from simulateQC
[ys,yu,vs,vu]=simulateQC(time,yRoad,ms,mu,kt,k,b,c);

% determine time interval dt


% zero vector for a




% Simple approach:
% using loop (not full marks!) calculate a(i) and discomfort value.
% Of course, you should later replace the following loop
% using vectorization to get full marks

for i = ???

    ????
    ????

end
```

Part-2(a)

- You can use the script **testCalcDiscomfort.m** (a file in **assign2.zip**) to test whether your calcDiscomfort function is working correctly.

- If the reported **error** is small, i.e. **less than 10^{-8}**, then it should be **fine**.

- **Important requirement on implementation:**
  - The calculation of the discomfort level from the vector vs **can be done without** using any loops.
  - However, **if you are not comfortable with vectorization**, first implement it using a loop, and later change it to vectorization, if you could.
  - You will only receive full marks for this part if the calculation is done **without** using loops, otherwise you will receive a **reduced** mark if loops are used.
  - The Matlab built-in function **sum** is useful here.

Part-2(b):

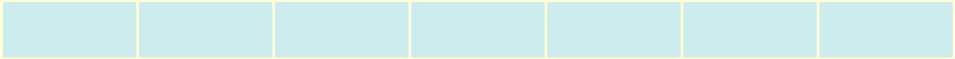
Calculating discomfort level for many pairs of (inertance b, damping coefficient c)

- The function calcDiscomfort allows you to determine the discomfort level for each set of suspension parameters: spring stiffness *k*, damping coefficient *c* and inertance *b*.
- For simplicity, we will **not change** the **value of *k***.
- We will calculate the discomfort level for many **different pairs of** (inertance,damping coefficient) or **(*b*,*c*) values**.

The steps for Task 2b are:

- Create a **vector** of equally spaced **inertances**.
  The first value is at **bLower** and the last value is **bUpper**.
  The number of inertances is given by **nInerterValues**.

b =



These three parameters are specified in Matlab script assignment2SampleParameters.m. (**Note**: The Matlab script assignment2.m reads in these parameters so you can just use them. It is a good idea for you to go through the file assignment2SampleParameters.m to see what parameters have been defined. In particular, you need to know that assignment2SampleParameters.m does not specify the values of b and c.)

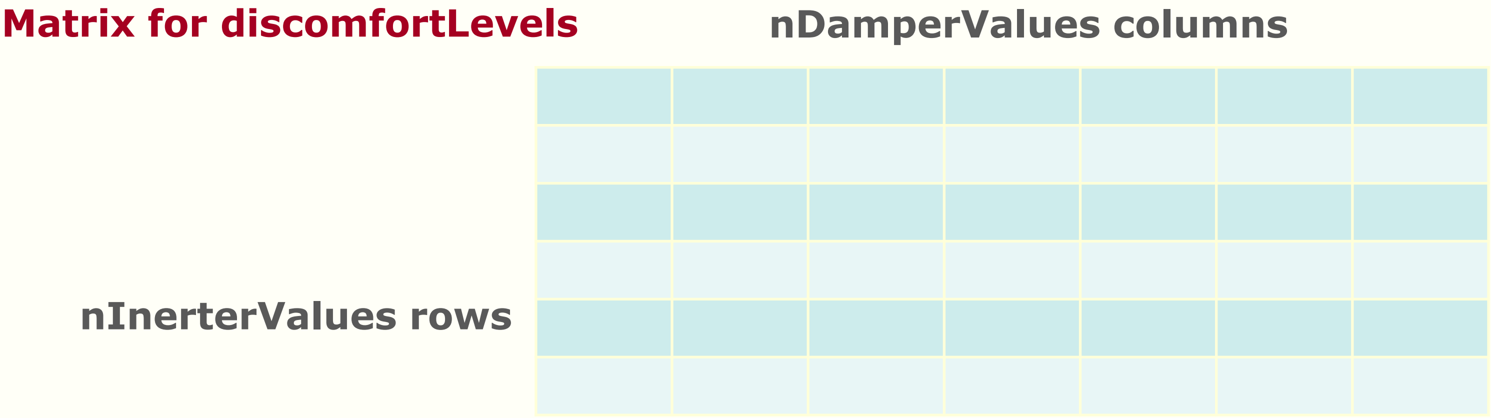
Part-2(b):

Calculating discomfort level for many pairs of (inertance b, damping coefficient c)

- Create a **vector** of equally spaced **damping coefficients**.
  The first value is at **cLower** and the last value is **cUpper**.
  The number of damping coefficients is given by **nDamperValues**.
  These three parameters are specified in Matlab script assignment2SampleParameters.m.

c =

- Create a **zero matrix discomfortLevels** with **nInerterValues rows** and **nDamperValues columns**.



Matrix for discomfortLevels

nDamperValues columns

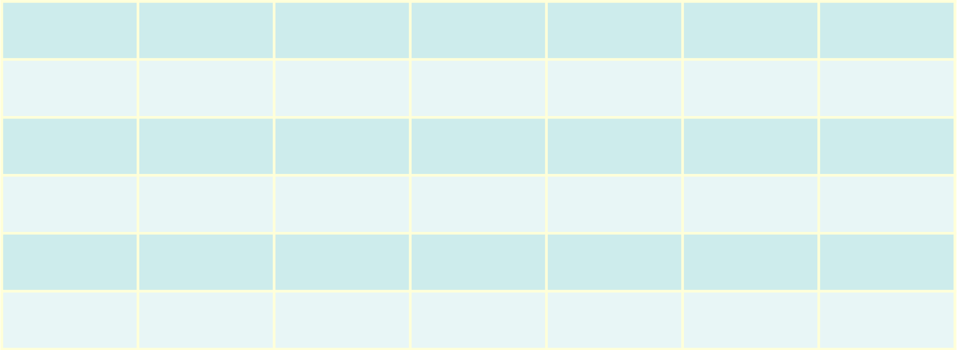
nInerterValues rows

Part-2(b):

Calculating discomfort level for many pairs of (inertance b, damping coefficient c)

Matrix for discomfortLevels **nDamperValues columns**

nInerterValues rows



- The ***(i,j)* element** of the matrix discomfortLevels, i.e. discomfortLevels(i,j), should be assigned the discomfort level of a suspension that uses the *i*-th inertance and *j*-th damping coefficient. You can assume all other parameters are as specified.

```
discomfortLevels( i, j ) = calcDiscomfort  (time, yRoad, ms, mu, kt, k, b(i), c(j) );
```

- You **can** use loops to complete this task.

Part-2(b):

Calculating discomfort level for many pairs of (inertance b, damping coefficient c)

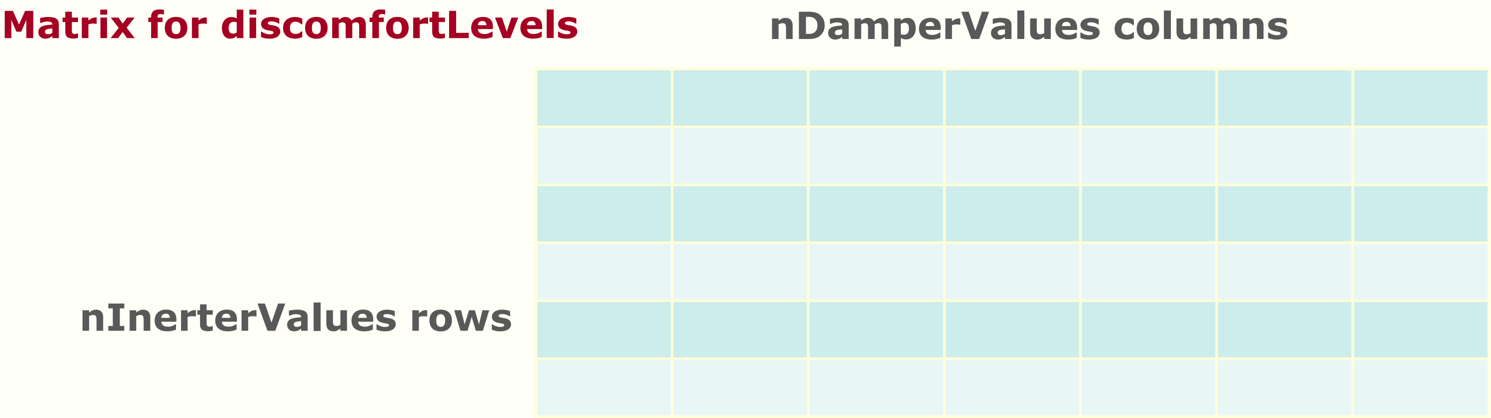
```
 % create vectors b and c, read the specs
b =  ???? ;
c =  ???? ;


% create zero matrix  for discomfortLevels, read the specs
discomfortLevels = ????  ;


% Use two nested loops  to calculate discomfortLevels for each combination of b and c values


for i = ????
    for j = ????

              ???????
              ???????
   end
end
```

Part-2(b)

- Please read the specs for information on how to test your Task-2(b).

Part-2(c):

Determining the (inertance, damping coefficient) pairs that give, respectively, the best and worst comfort



- From the matrix for **discomfortLevels,** find **min** and **max** discomfort levels. Also find the corresponding values of b and c, for example: bBest, cBest, bWorst, cWorst.

Part-2(d):

Displaying and plotting the results

Once you have obtained the best and worst (inertance, damping coefficient) pairs, do the following

1. Use a Matlab **disp** statement to display the values of the inertance, damping coefficient and discomfort level for the **most** comfortable suspension.

   For example: use **num2str** to convert a number in to string before displaying.

   display(['Most comfortable suspension: inertance = ‘, num2str(bBest) , …

2. Use a Matlab **disp** statement to display the values of the inertance, damping coefficient and discomfort level for the **least** comfortable suspension.

3. Use **simulateQC**.m to **obtain** the car body movements **ys** of the **best** suspension. Repeat that for the **worst suspension**. You should store the **ys** vectors of these two configurations in two different vectors. You need both of them later, in the next step as well as Task 3.

4. **Plot** a graph in Matlab (in figure 1) with time on the horizontal axis and displacement on the vertical axis. The graph should show **three curves**: (i) The **road surface**; (ii) The displacement of the car body **ys** for the **most** comfortable configuration; (iii) The displacement of the car body **ys** for the **least** comfortable configuration.