Design a differentially private average consensus scheme

The goal of this exercise is to implement a distributed average consensus scheme using a differentially private projected gradient descent algorithm.

*You can use the algorithms discussed in Section 4 of "**Privacy in Control and Dynamical Systems" by S. Han and G. J. Pappas** for this assignment.*

The use of MATLAB is recommended. The following tools are recommended to solve the assignment:

1) *MATLAB `spmd` function to simulate parallel communicating agents*
2) *A function to generate noise according to a Laplace distribution (provided in the appendix).*

[10 pts]

Solve the following distributed consensus problem

$$\min_{x_i \in \mathcal{X}} \sum_{i=1}^{n} \|x_i - v_i\|^2$$

Where $\mathcal{X} \coloneqq \{x \in \mathbb{R} | -1 \leq x \leq 1\}, n = 4$, and private $v_1 = 0.1, v_2 = 0.5, v_3 = 0.4, v_4 = 0.2$.

[2pts] Implement Algorithm 1 in Section 4 of the paper above using the following tuning parameters: $T = 50, q = 0.6,\ c = 1$. If you are using MATLAB use the `spmd` functionality to run code in parallel on 4 workers. Plot the trajectories of the four agents obtained using Algorithm 1. Do they reach consensus? Provide the consensus value.

[4pts] Implement Algorithm 2 in Section 4 of the paper above using the following tuning parameters: $T = 50, q = 0.6,\ c = 1, \epsilon = 0.001$. If you use MATLAB, you can use the `randlap` MATLAB function provided in the appendix to generate Laplace noise. Explain how you selected the value of the parameter b_t. Plot the trajectories of the four agents obtained using Algorithm 2. Do they reach consensus? What happens if you increase $\epsilon = 0.01$?

[2pts] Now consider that $n = 8$, and $v_1 = v_5 = 0.1, v_2 = v_6 = 0.5, v_3 = v_7 = 0.4, v_4 = v_8 = 0.2$. If you are using MATLAB you can use the `spmd` functionality to run code in parallel on 8 workers. Use Algorithms 1 to check whether the agents still reach consensus. Plot the trajectories of the agents and their consensus value.

[2pts] Use Algorithms 2 to check whether the 8 agents still reach consensus for $\epsilon = 0.001$ and for $\epsilon = 0.01$. Plot the trajectories of the agents and their consensus value. Does a larger number of agents improve the performance of algorithm under differential privacy?

Submit the assignment together with the code you used to generate the results and the plots. Add a readme file with instructions to run the code.

Appendix: MATLAB function to generate Laplace noise

```
function x = randlap(siz,lambda)
% RANDL  random numbers distributed according to the Laplace distribution
%   RANDL(N) will return an NxN matrix containing pseudo-random values
%   drawn from a Laplace distribution with zero mean and standard deviation
%   one. RAND([M,N,...,P]) will return an MxNx...xP matrix.
%   RANDL([M,N,...,P],lambda) will return an MxNx...xP matrix of
%   pseudo-random numbers with parameter lambda. CAUTION: the pdf is
%   assumed as
%           pdf = lambda/2 * exp(-lambda*abs(x))
%
% The Laplace random numbers are generated using the the RAND function to
% generate uniformly distributed numbers in (0,1) and then the probability
% integral transformation, i.e. using the fact that
%   if Fx(x) is the cdf of a random variable x, then the RV z=Fx(x) is
%   uniformly distributed in (0,1).
%
% In order to generate random numbers with mean mu and variance v, then
% generate the random numbers X with mean 0 and variance 1 and then
%       X = X*sqrt(v)+mu;
%
% C. Saragiotis, Oct 2008
if nargin==1
    lambda = sqrt(2); % this gives a std=var=1.
end
z = rand(siz);
x = zeros(siz);
in = z<=.5;
ip = z> .5;
x(in) =  1/lambda *log(2*z(in));
x(ip) = -1/lambda *log(2*(1-z(ip)));
```