EE 4770 **Homework 5** **Due: 21 April 1997**

A cross-correlation flow meter is to be designed in the following problems.

Problem 1: Choose a slurry and describe what type of sensors could be used to measure its velocity in a cross-correlation flow meter. Any slurry and any sensors can be chosen so long as they would work and would not make the next problem too difficult. Other parts of the system need not be practical and might be highly imaginative. For example, the slurry can consist of chunks of neutronium[1] and the sensors would detect the change in gravitational force on a fixed mass as the chunks flow by. (The composition of the pipe, the fluid, whether neutronium would remain in distinct chunks, and where the neutronium came from and is going to need not be addressed. The fact that there would be changing forces if such a slurry were possible, and the ability of the sensor to measure such forces are important.)

Include a diagram showing how the sensors are placed. Indicate the measurement range expected of the sensors, the distance between the sensors, and the range of flow velocities that can be measured. Specify a unit for the flow velocity. The unit need not be common, *e.g.*, furlongs per fortnight, but it cannot be chosen to make the solution simple.

Problem 2: Using the system and sensors chosen above, design a circuit to convert the output into a form suitable for the next problem. Be sure to specify all component values, voltage sources, and any other relevant details.

Problem 3: The (incomplete) program on the next page periodically reads values from two sensors and then computes and displays the flow rate. Complete the program by inserting code between the "`solution starts here`" and "`solution ends here`" comments and by choosing a value for `sample_interval`. The flow rate computed must be correct given the information provided above. The solution need not be computationally efficient. The completed program can be written in another language or in pseudocode as long as sufficient detail is given.

For those who want to run the code they write, a solution template has been linked to the course web page. The template includes the code on the next page plus stubs for the `readInterface` and other calls, so that it could be compiled and run on an ordinary system.

[1] The super high density form of matter in neutron stars.

```
#define SIZE 1000    /* Number of samples to store. */

void cross_cor()
{
 int samp_A[SIZE];   /* Hold values read from sensor A. */
 int samp_B[SIZE];   /* Hold values read from sensor B. */
 double next_sample_time;      /* Time at which sensors will be read. */
 double sample_interval = ???; /* Time between reading sensors. */

 int i;

 for(i=0;i<SIZE;i++)samp_A[i]=samp_B[i]=0.0;   /* Inititalize arrays. */

 next_sample_time = get_current_time();        /* Initialize to now. */

 while(1){
  int a = readInterfaceA();     /* Read sesnor A. */
  int b = readInterfaceB();     /* Read sesnor B. */
  double flow_velocity;

  next_sample_time += sample_interval;  /* Compute when sensors next read. */

  /* Make room for new samples. */
  for(i=SIZE-1; i>0; i--){
   samp_A[i] = samp_A[i-1]; samp_B[i] = samp_B[i-1];}

  /* Store new samples. */
  samp_A[0]=a; samp_B[0]=b;

  /* Code to compute flow velocity. */
  /* (The solution starts here.) */




  flow_velocity = ????
  /* (The solution ends here.) */

  /* Display flow velocity. */
  display(flow_velocity);

  /* Function will return at next_sample_time. */
  sleep_until( next_sample_time );

 }
}
```