Radiokommunikasjon

Proof of concept

-cheif engineer Kristoffer Loland

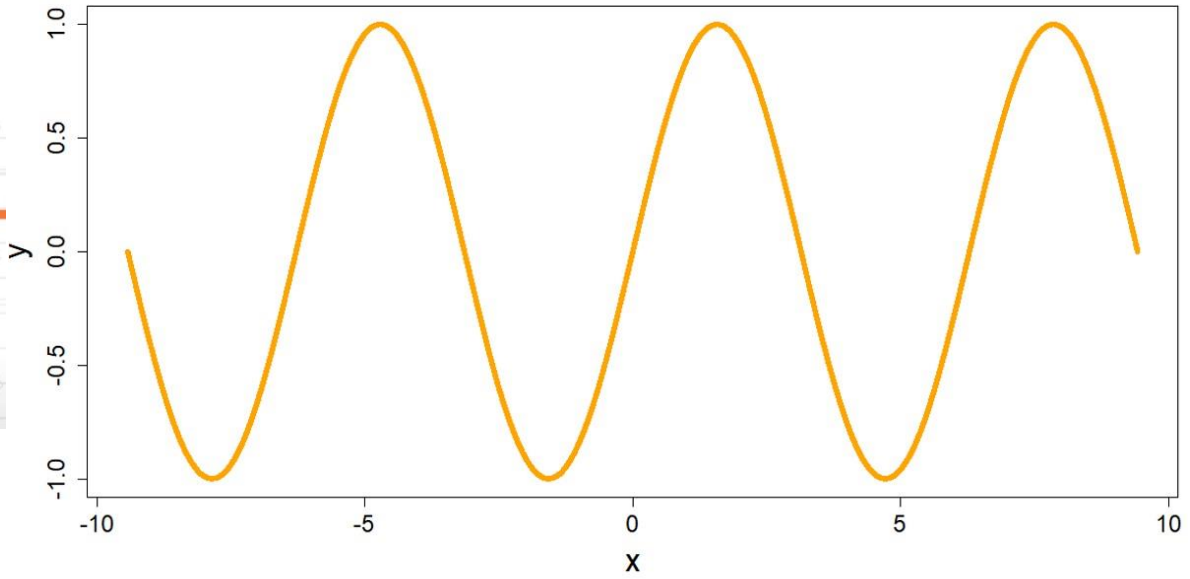
-proffesor Marius Claudey

-trainee Erlend Børli

Proof of consept

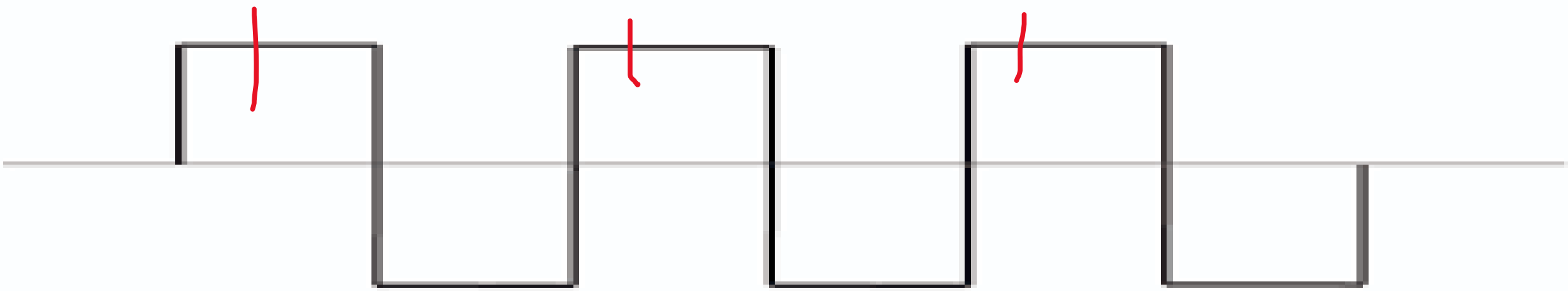
Digital Signal (Ut fra Arduino)

Analog in signal (lest av Arduino)

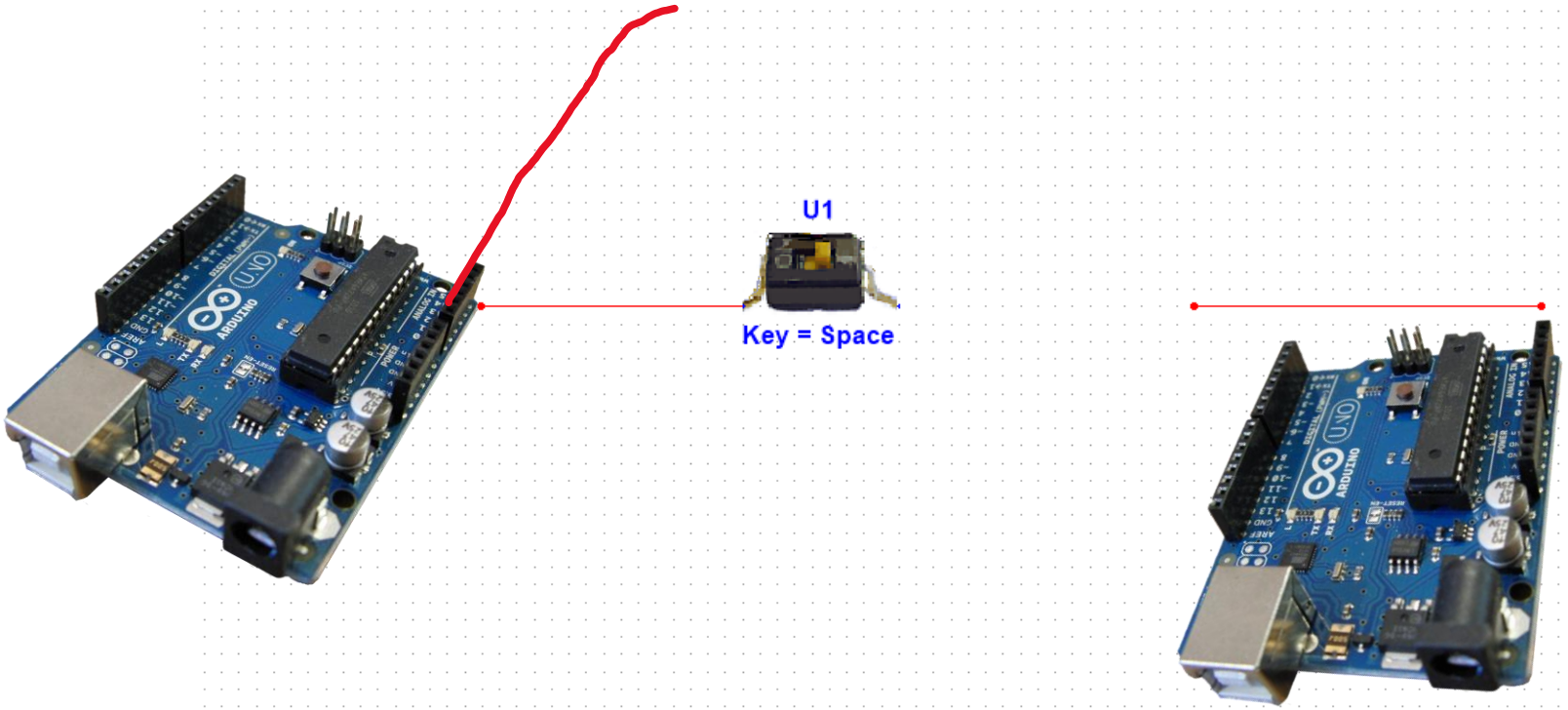


Frekvens

- Sender og mottaker må ha forskjellig frekvens
- Forskjellig klokkesignal

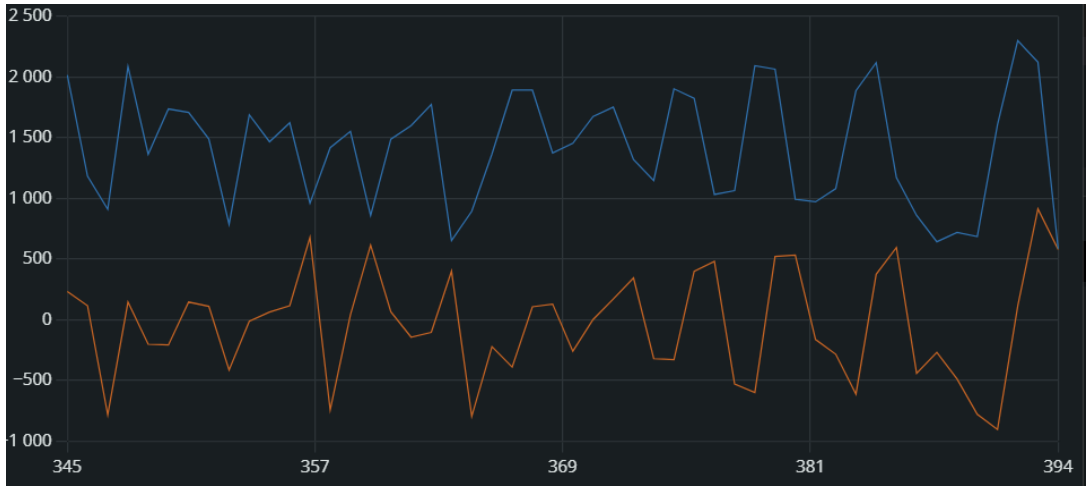


Kretstegning

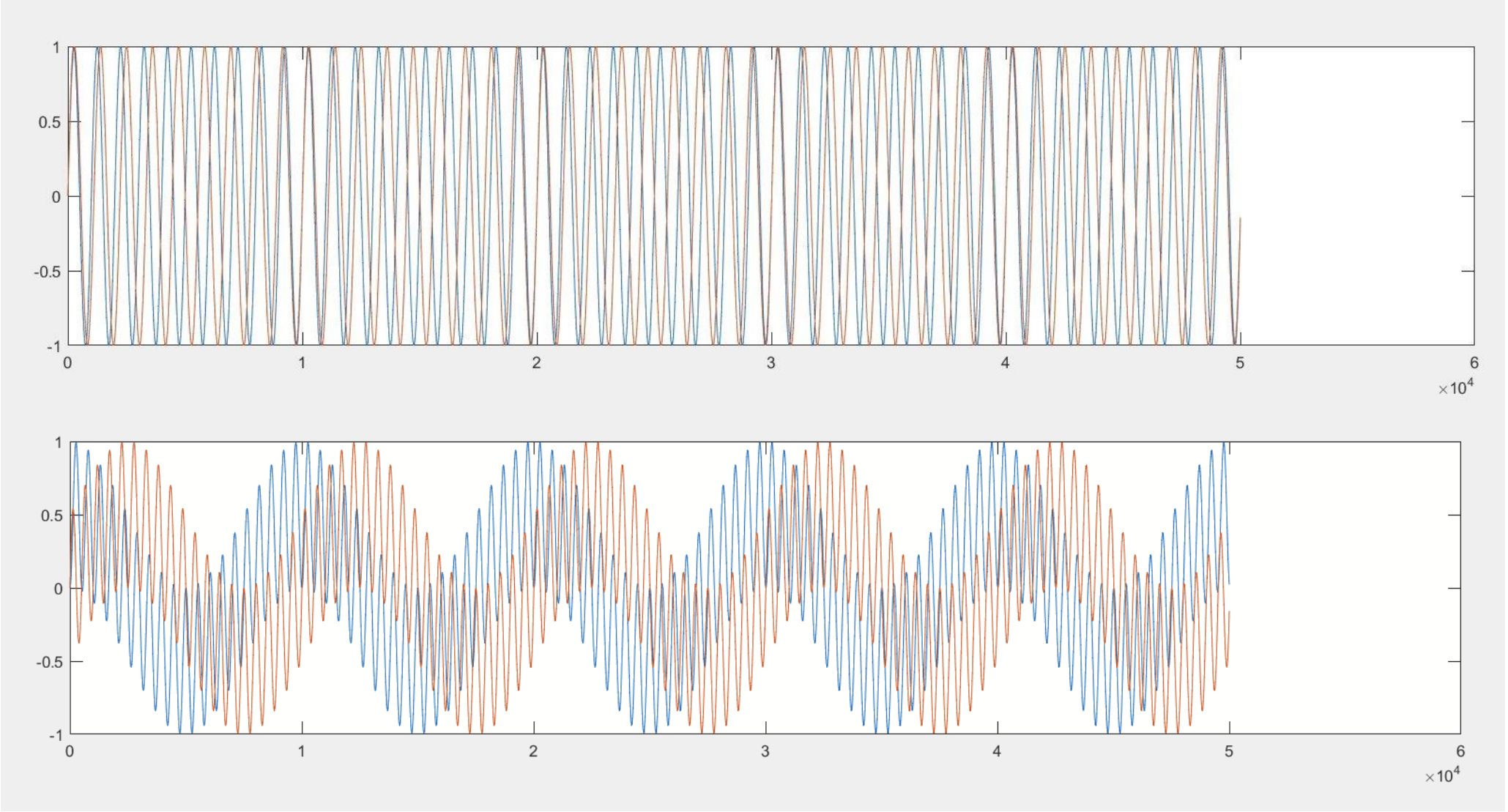


Begrensinger

- Kan ikke variere avstander
- Svært små avstander
- Sårbar for interferens

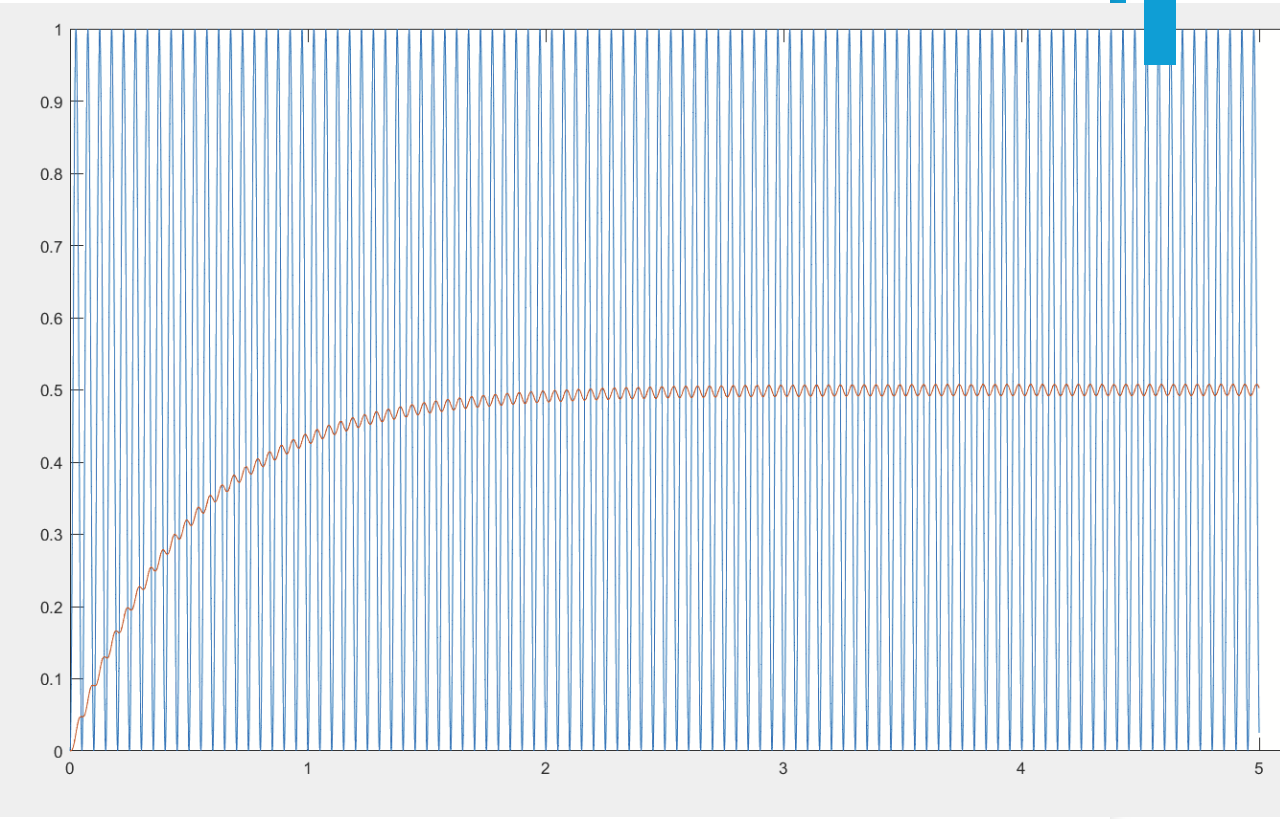


Sampling ideelt (matlab)



Filter

- Skille mellom lav og høy
- RC Filter
- Prosent forskjell
- Gjennomsnitt

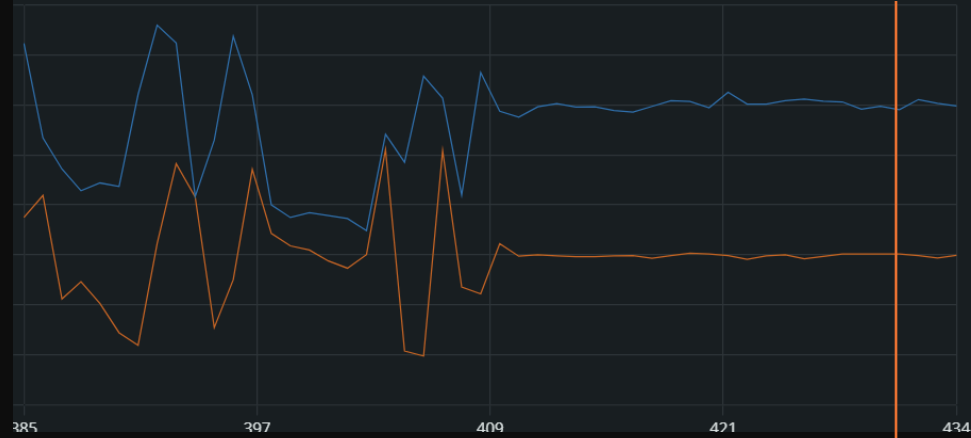


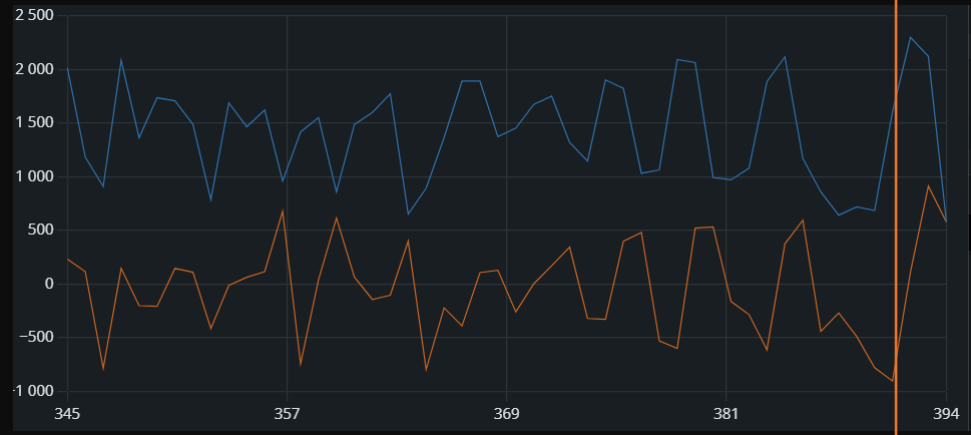
```
70    void lavpassfilter(long* x, int n, float dt, float RC) {
71      // Digital RC low pass filter
72      // Variables:
73      // input data - x
74      // time interval - dt
75      // time constant RC
76
77      float alfa = dt / (RC + dt);
78      y[0] = x[0];
79
80      for (int i = 1; i < n; i++) {
81        y[i] = alfa * x[i] + (1 - alfa) * y[i - 1];
82      }
83    }
```

Arduino serial plotter

- På-Av
- På
- På-av-på

- Stor variasjon





Kommunikasjon

- Vanskelig å definere start mtp interferens
- Bits
- ASCII

```
5   void loop() {
6     if (Serial.available() > 0) { // Check if data is available to read
7       String inputString = Serial.readString(); // Read the string from serial monitor
8       Serial.print("ASCII bytes: ");
9       Serial.print("00000010 "); // Start of text
10      for (int i = 0; i < inputString.length()-1; i++) {
11        printBinary((byte)inputString[i]); // Print ASCII byte of each character
12        Serial.print(" ");
13      }
14      Serial.print("00000011 "); // End of text
15    }
16  }
17
18  void printBinary(int num) {
19    // Print each bit of the binary representation
20    for (int i = 7; i >= 0; i--) {
21      Serial.print((num >> i) & 1); // Print each bit of the binary representation
22    }
23  }
```

```
int utsignalkort = 2;
int analogPin = A1;
uint8_t knapp = 8;

long p2;
long p3;
long produkt;
long produktSiste;
float prosentdifferanse;
unsigned char resultat;
int signTab2[] = {1,1,-1,-1};
int signTab3[] = {1,-1,-1,1};

#define MAX_SIZE 100 // Define the maximum size of your data array
long produktArray[MAX_SIZE]; // Array to store produkt values
int index = 0; // Index to keep track of where to store the next value
float y[MAX_SIZE]; // Output array

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(utsignalkort, OUTPUT);
  pinMode(knapp, INPUT_PULLUP);
}
```

```
void loop() {
  p2=0;
  p3=0;

  if (digitalRead(knapp) == HIGH) {
    for (int i = 0; i < 1000; i++) {
      for (int j = 0; j < 4; j++) {
        if (signTab2[j] == 1) {
          digitalWrite(utsignalkort, HIGH);
        } else {
          digitalWrite(utsignalkort, LOW);
        }

        int sensorValue = analogRead(analogPin); // Read analog pin
        p2 = signTab2[j] * sensorValue + p2;

        p3 = signTab3[j] * sensorValue + p3;
      }
    }
  }

  produktSiste = produkt;
  produkt = p2 * p2 + p3 * p3;

  produktArray[index] = produkt; // Store produkt in the array
  index = (index + 1) % MAX_SIZE; // Increment the index for the next round, wrap around if necessary

  lavpassfilter(produktArray, MAX_SIZE, 0.1, 10);
```

```
      Serial.print(p2);
      Serial.print(", ");
      Serial.print(p3);
      Serial.print(", ");
      Serial.print(produkt);
      Serial.print(", ");
      Serial.print(y[MAX_SIZE-1]);
      Serial.println("");
    }

```

Forbedring

- Kontrollere frekvens med interrupts
- Bedre filtrering og lesing av bits
- Check bits
- Bedre hardware
- Bedre kode