

# Commande d'un fauteuil roulant par contraction d'un muscle

## Position du problème

*Introduction*

*Objectifs*

## Traitement du signal

*Caractéristiques*

*Amplification*

*Mise en forme*

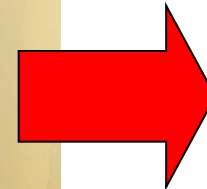
## Déplacement du fauteuil

*Conception prototype*

*Programmation*

*Asservissement*

## Conclusion



## Objectifs :

### Position du problème

*Introduction*

*Objectifs*

### Traitement du signal

*Caractéristiques*

*Amplification*

*Mise en forme*

### Déplacement du fauteuil

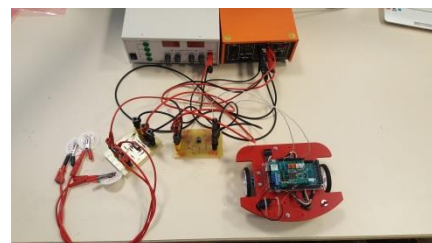
*Conception prototype*

*Programmation*

*Asservissement*

### Conclusion

- Réaliser un système adaptable à différents Handicaps.
- Permettant plusieurs types de déplacements.
- Rapidité d'exécution de moins de 3s.
- 4 commandes réalisables :
  - avancer, s'arrêter, tourner à gauche, tourner à droite
- Ne doit pas être contraignant physiquement



# Caractéristiques:

## Position du problème

Introduction

Objectifs

## Traitement du signal

Caractéristiques

Amplification

Mise en forme

## Déplacement du fauteuil

Conception prototype

Programmation

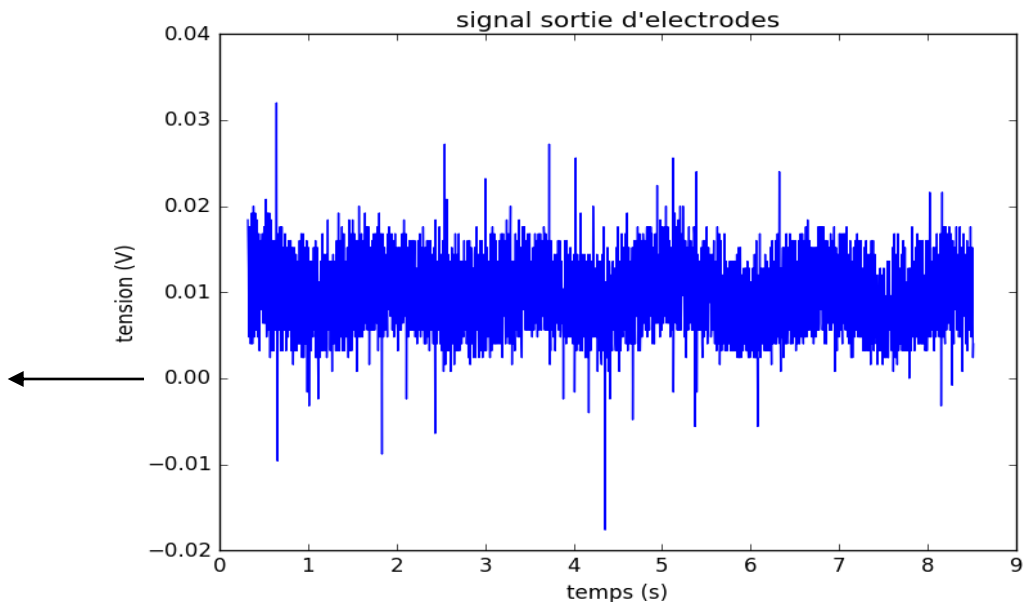
Asservissement

## Conclusion

- fréquence : entre 100 Hz et 3000HZ.
- Les signaux exploitables se trouvent vers 500 Hz
- L'amplitude en tension des signaux se trouve entre 10  $\mu$ V et 3 mV



Signal obtenu lors de 6 contractions brèves du Biceps gauche :



Impossible de distinguer des informations

## Amplification :

## Amplificateur d'instrumentation :

- Adaptation d'impédance
- Fort gain

### Position du problème

*Introduction*

*Objectifs*

### Traitement du signal

*Caractéristiques*

*Amplification*

*Mise en forme*

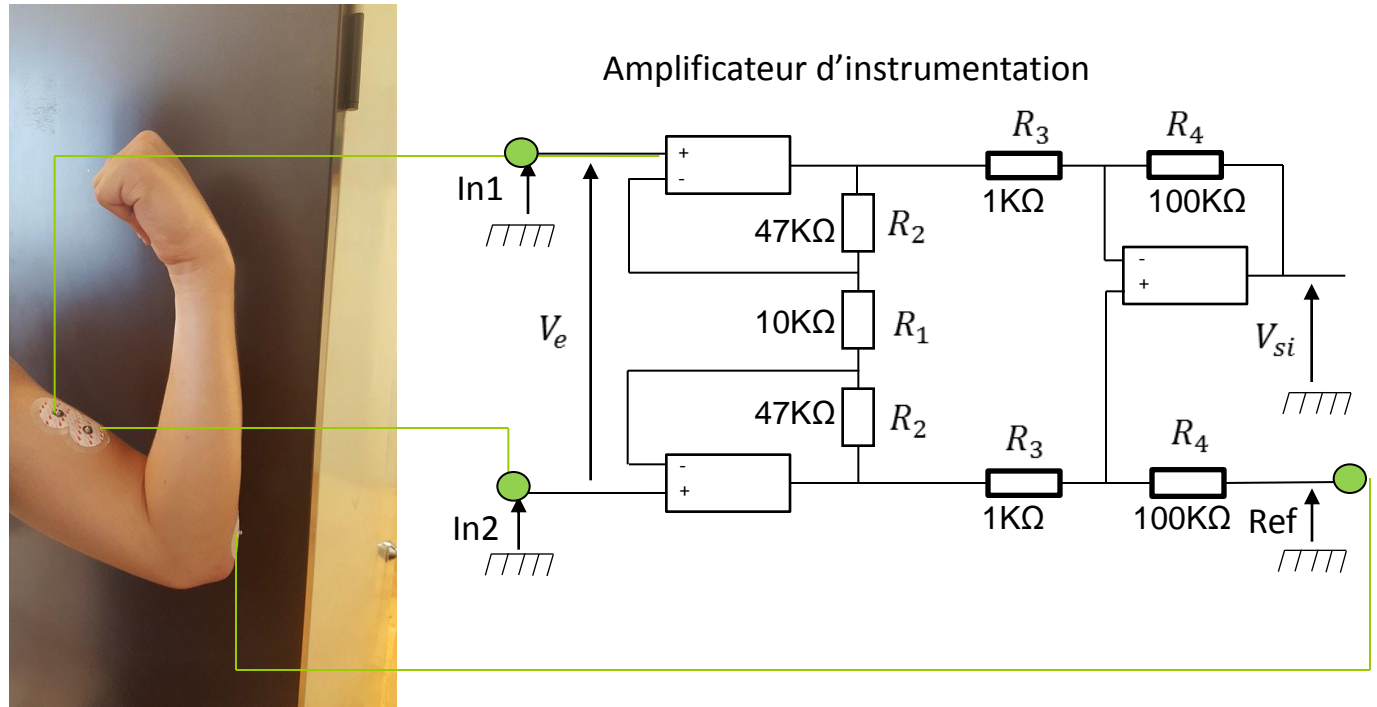
### Déplacement du fauteuil

*Conception prototype*

*Programmation*

*Asservissement*

### Conclusion



$$V_{si} = \frac{R_4}{R_3} \left( 1 + \frac{2R_2}{R_1} \right) V_e = G_1 V_e \quad G_1 = 1000$$



# Amplification :

## Position du problème

*Introduction*

*Objectifs*

## Traitement du signal

*Caractéristiques*

*Amplification*

*Mise en forme*

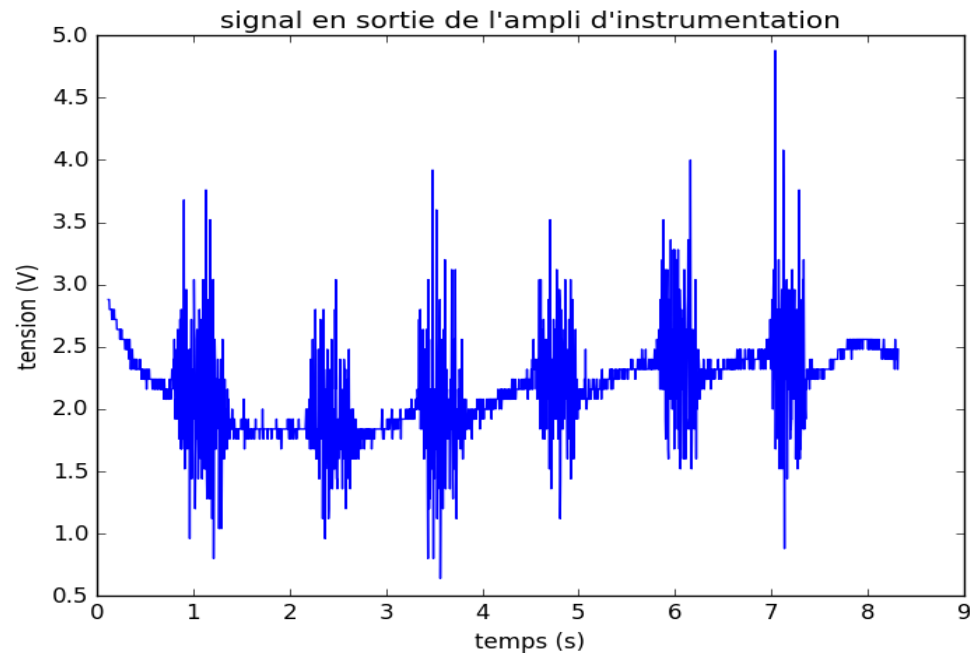
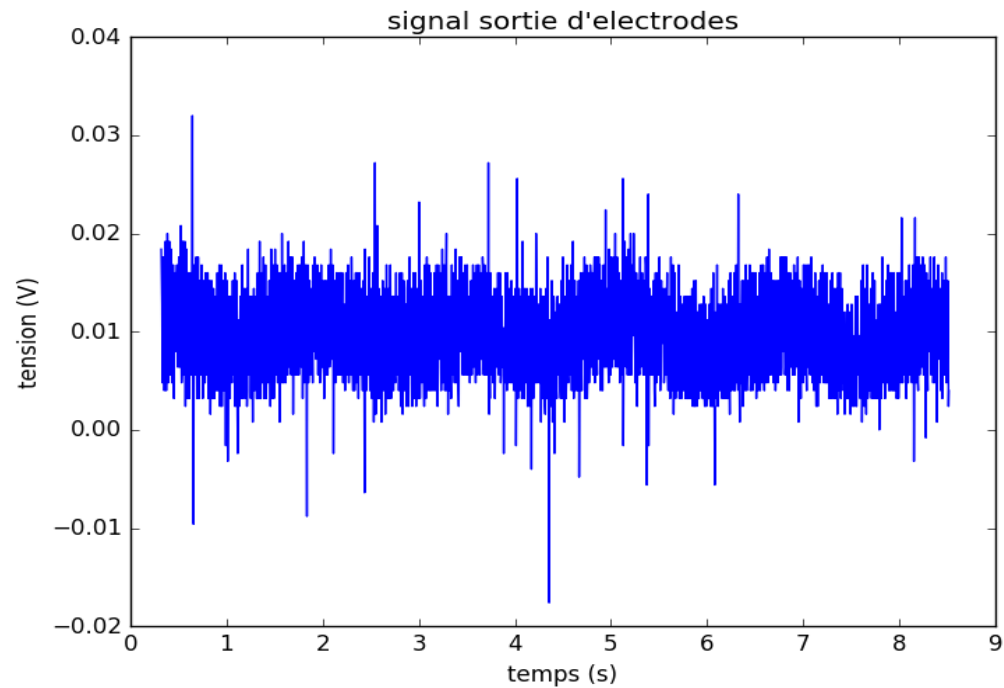
## Déplacement du fauteuil

*Conception prototype*

*Programmation*

*Asservissement*

## Conclusion



# Filtre passe-haut : Amplificateur non-inverseur :

## Amplification :

### Position du problème

Introduction

Objectifs

### Traitement du signal

Caractéristiques

Amplification

Mise en forme

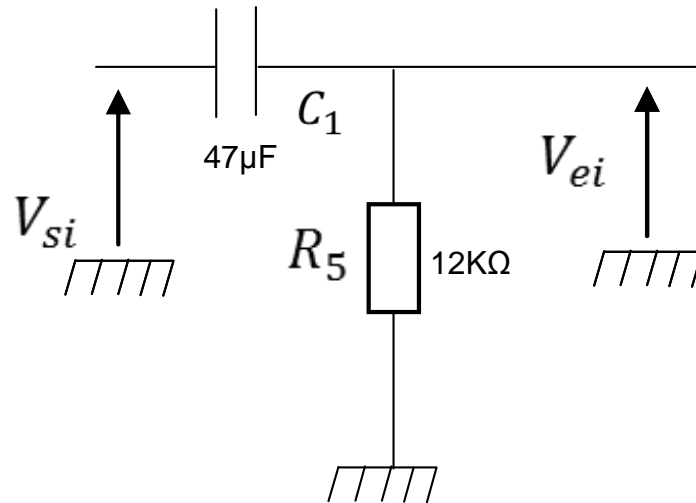
### Déplacement du fauteuil

Conception prototype

Programmation

Asservissement

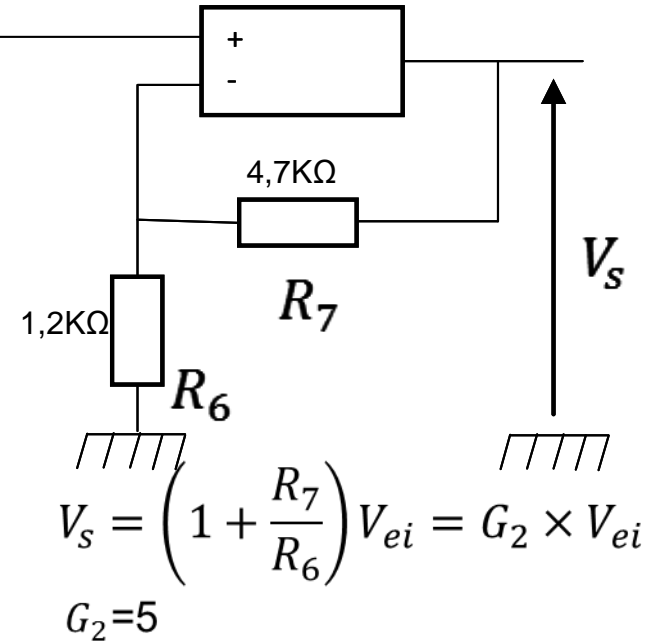
### Conclusion



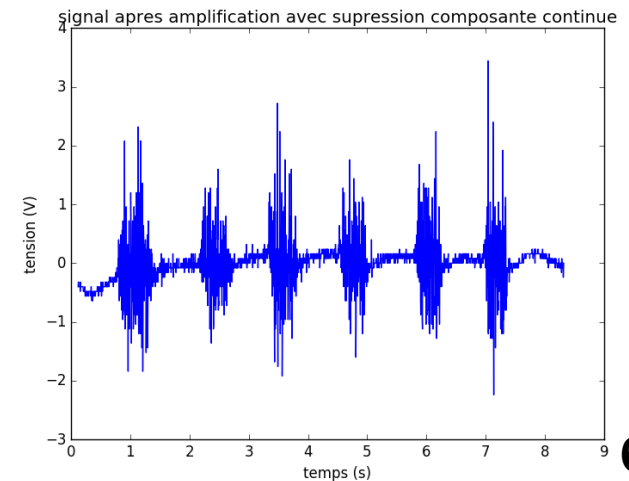
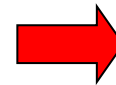
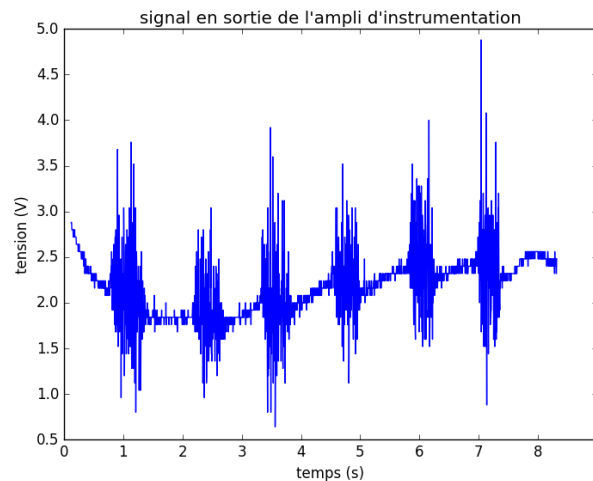
Supprime la composante continue :

Fréquence de coupure du filtre:  $f_c = \frac{1}{2\pi R_5 C_1}$

$f_c = 0,28\text{Hz}$



Amplification totale circuit :  $G = G_1 \times G_2 = 5000$



# Détecteur de crête :

## Mise en forme :

### Position du problème

Introduction

Objectifs

### Traitement du signal

Caractéristiques

Amplification

Mise en forme

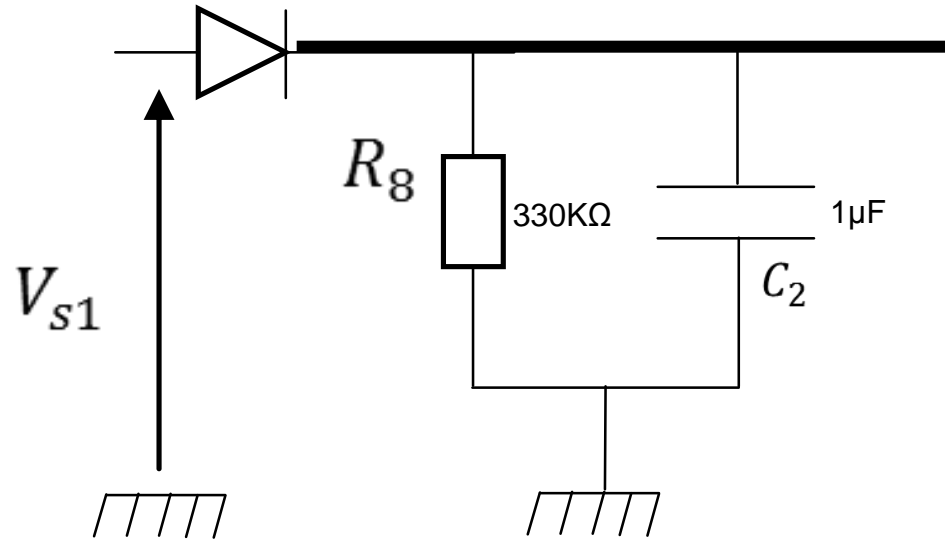
### Déplacement du fauteuil

Conception prototype

Programmation

Asservissement

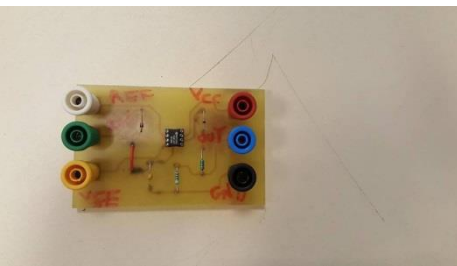
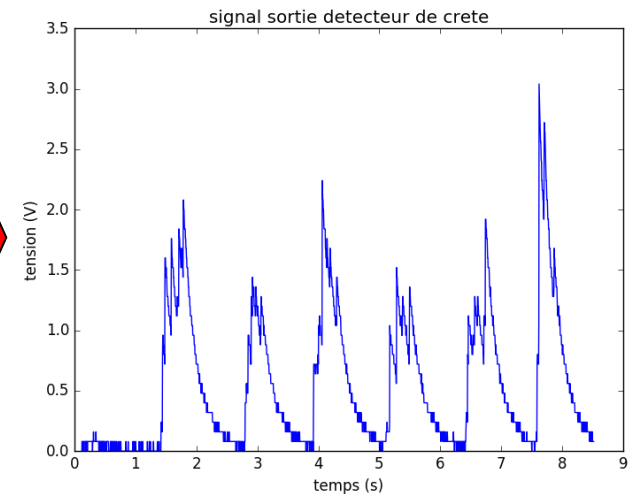
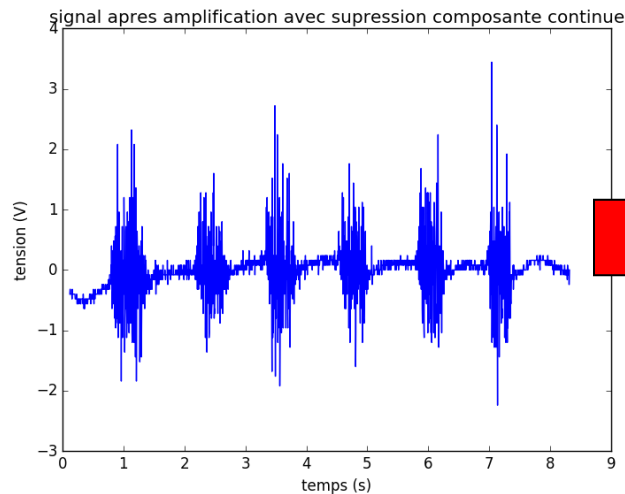
### Conclusion



Période de la porteuse:  $T_0=0,02s$

Période du modulant :  $T=1s$

Encadrement constante de temps :  $T_0 \ll \tau \ll T$      $\tau=R_8 \times C_2=0,33s$





## Mise en forme :

### Position du problème

Introduction

Objectifs

### Traitement du signal

Caractéristiques

Amplification

Mise en forme

### Déplacement du fauteuil

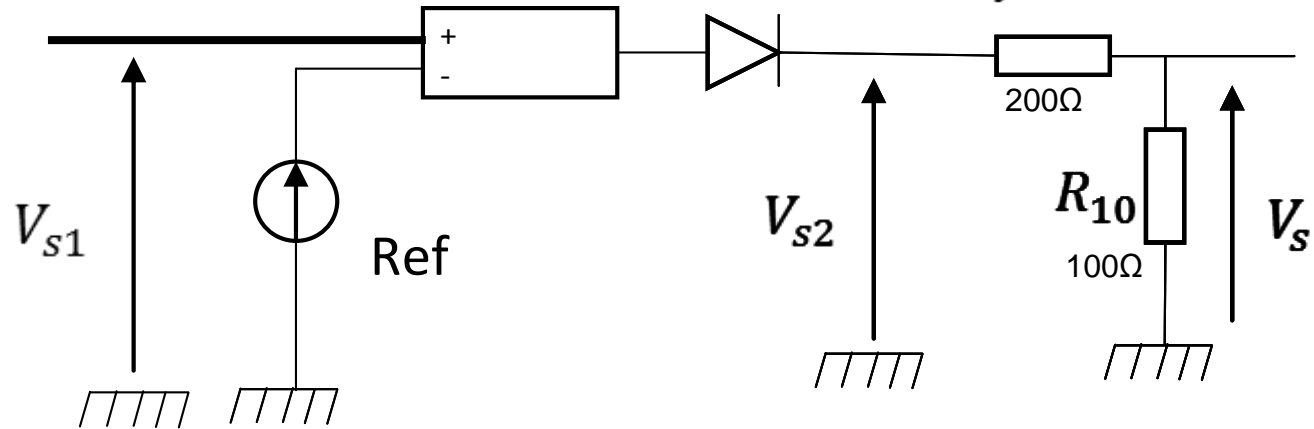
Conception prototype

Programmation

Asservissement

### Conclusion

## Comparateur avec référence :

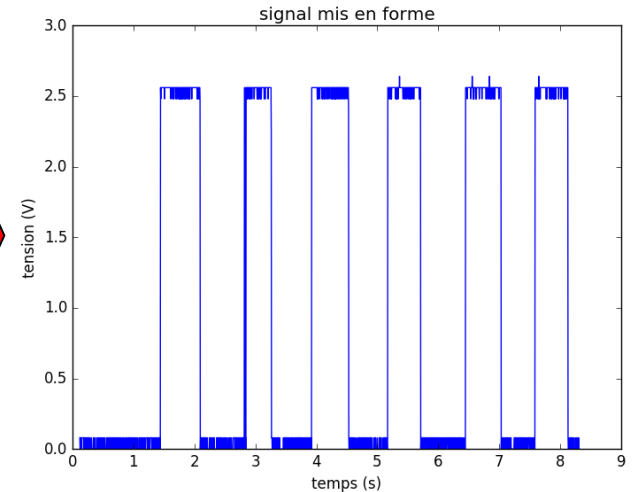
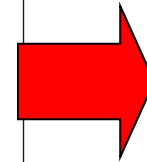
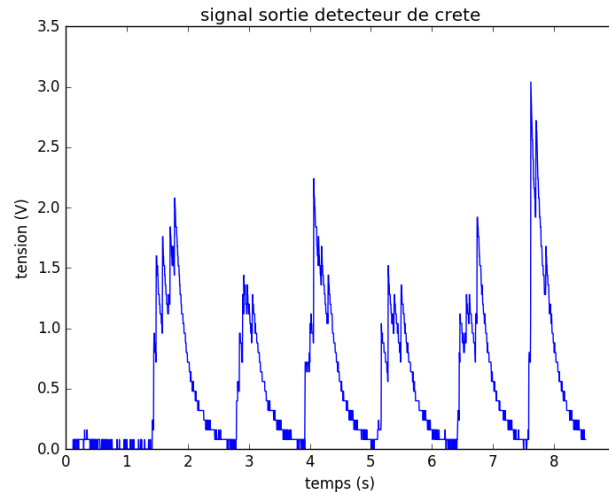


$$V_{s2} = +15 \text{ V si } V_{s1} > \text{Ref}$$

$$V_{s2} = -15 \text{ V si } V_{s1} < \text{Ref}$$

Adapte le niveau de tension pour la carte Arduino :

$$V_s = \frac{R_{10}}{R_9 + R_{10}} \times V_{s2} = \frac{1}{3} \times V_{s2}$$





# Conception :

## Position du problème

*Introduction*

*Objectifs*

## Traitement du signal

*Caractéristiques*

*Amplification*

*Mise en forme*

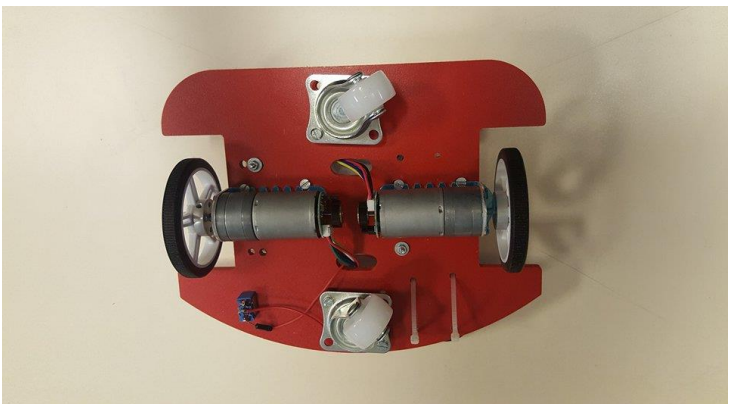
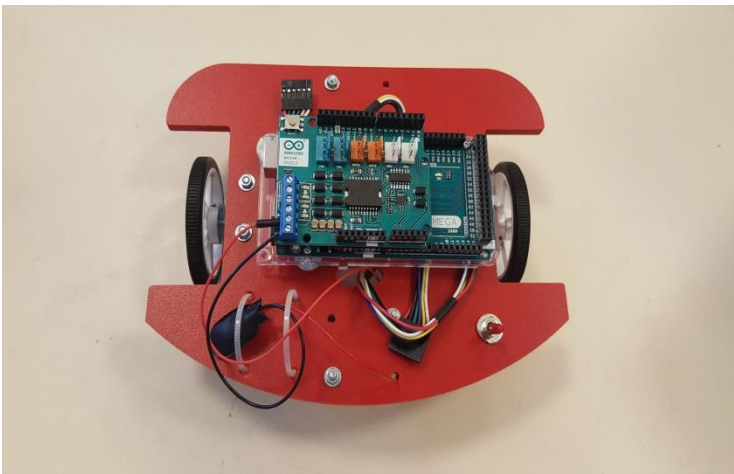
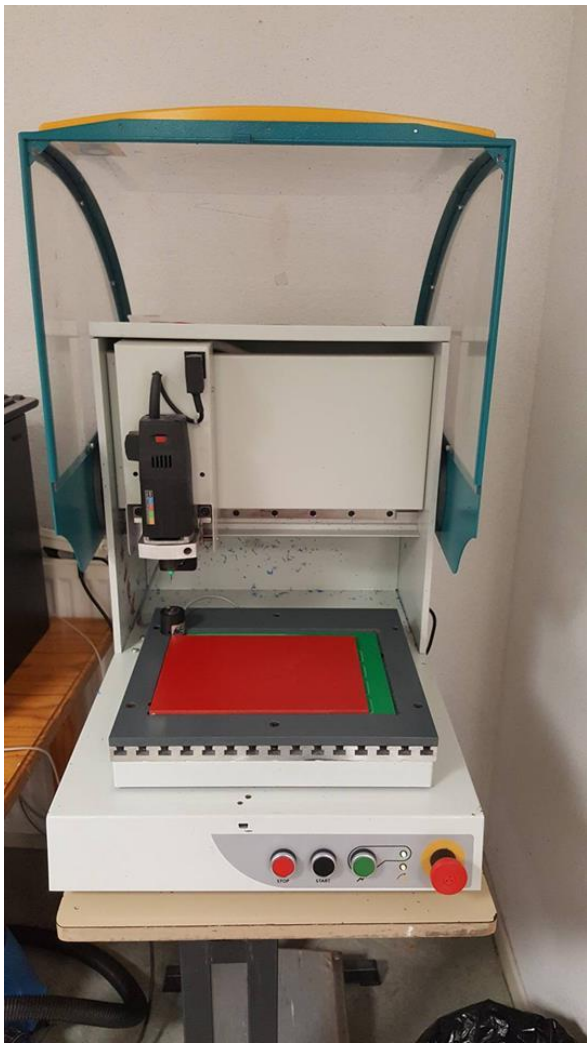
## Déplacement du fauteuil

*Conception prototype*

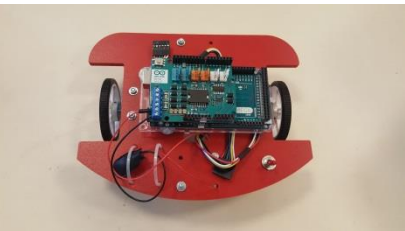
*Programmation*

*Asservissement*

## Conclusion



CAO avec le logiciel Charlygraal



# Programmation :

## Position du problème

Introduction

Objectifs

## Traitement du signal

Caractéristiques

Amplification

Mise en forme

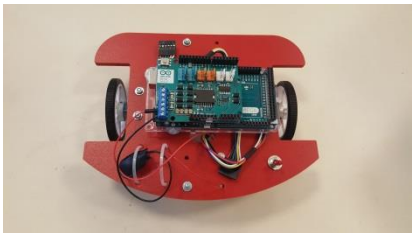
## Déplacement du fauteuil

Conception prototype

Programmation

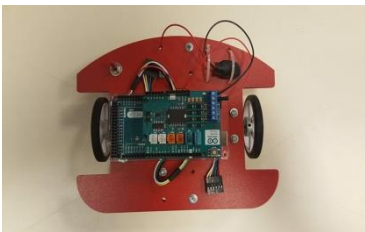
Asservissement

## Conclusion



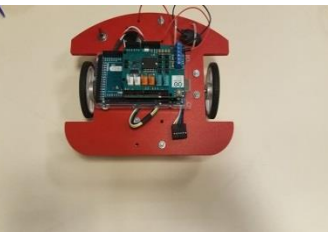
```
void CompterFrontsMontants() { ///Compte le nombre de fronts montants en 3s à compter du premier
etatCapteur=digitalRead(capteur); ///Capteur correspond à la broche d'entrée du signal
if(etatcapteur == HIGH){ ///HIGH correspond à l'état haut du signal créneau obtenu
frontsMontants=0;
instantInitial=millis(); ///Instant enregistré au premier passage à l'état haut du cycle
instantT=millis();
while(instantT-instantInitial<3000){ ///Détermine la durée d'attente de nouveaux fronts montants
etatcapteur=digitalRead(capteur);
if((etatcapteur != memoire) && (etatcapteur == HIGH)){
frontsMontants++;
}
memoire=etatCapteur;
instantT=millis();
}
}
}
```

1 front  
montant



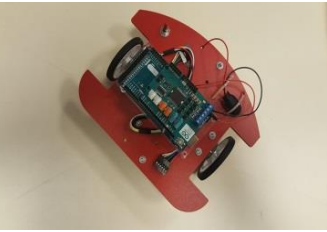
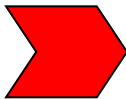
S'arrêter

2 fronts  
montants



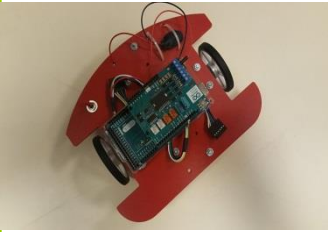
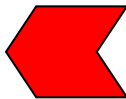
Avancer

3 fronts  
montants



Tourner à  
droite

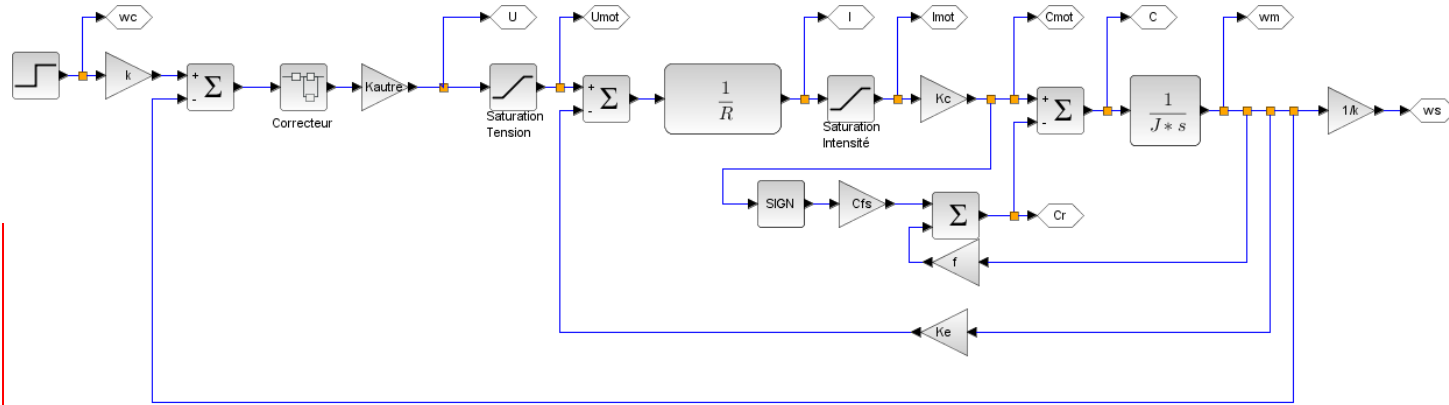
4 fronts  
montants



Tourner à  
gauche

# Boucle d'asservissement en vitesse des MCC :

## Asservissement :



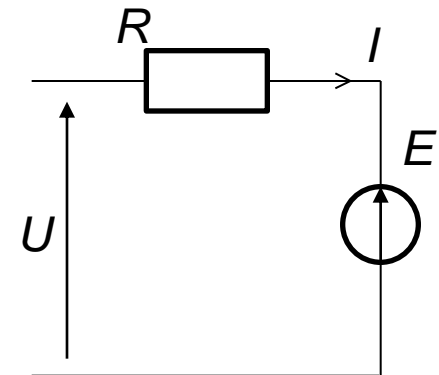
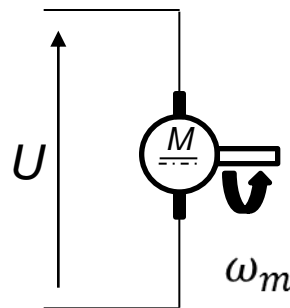
## Modèle des MCC :

$$J \frac{d\omega_m}{dt} = C_{em} - C_f$$

$$U = E - RI$$

$$C_{em} = K_c \times I$$

$$E = K_e \times \omega_m$$



## Position du problème

Introduction

Objectifs

## Traitement du signal

Caractéristiques

Amplification

Mise en forme

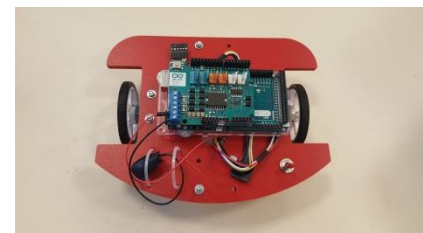
## Déplacement du fauteuil

Conception prototype

Programmation

Asservissement

## Conclusion



# Asservissement :

## Paramètres du moteur à déterminer :

$K_c$  : Constante de couple. Essai à vide  $\rightarrow K_c = 0,094 \text{ V.s}$

$K_e = K_c$  : Constante de force contre-électromotrice.  $\rightarrow K_e = 0,094 \text{ V.s}$

$R$  : Résistance de l'induit. Essai rotor bloqué  $\rightarrow R = 3,5\Omega$

$J$  : Moment d'inertie ramené au rotor du moteur.

Réponse à un échelon de tension  $\rightarrow J = 1,15 \cdot 10^{-5} \text{ kg.m}^2$

$f$  : Coefficient de frottement visqueux. Essai à vide  $\rightarrow f = 0,0024 \text{ N.m.s/rad}$

$C_r$  : Couple de frottement sec. Essai à vide  $\rightarrow C_r = 6,96 \cdot 10^{-7} \text{ N.m}$

## Position du problème

Introduction

Objectifs

## Traitement du signal

Caractéristiques

Amplification

Mise en forme

## Déplacement du fauteuil

Conception prototype

Programmation

Asservissement

## Conclusion

*Essai à vide en génératrice*



*Essai rotor bloqué*



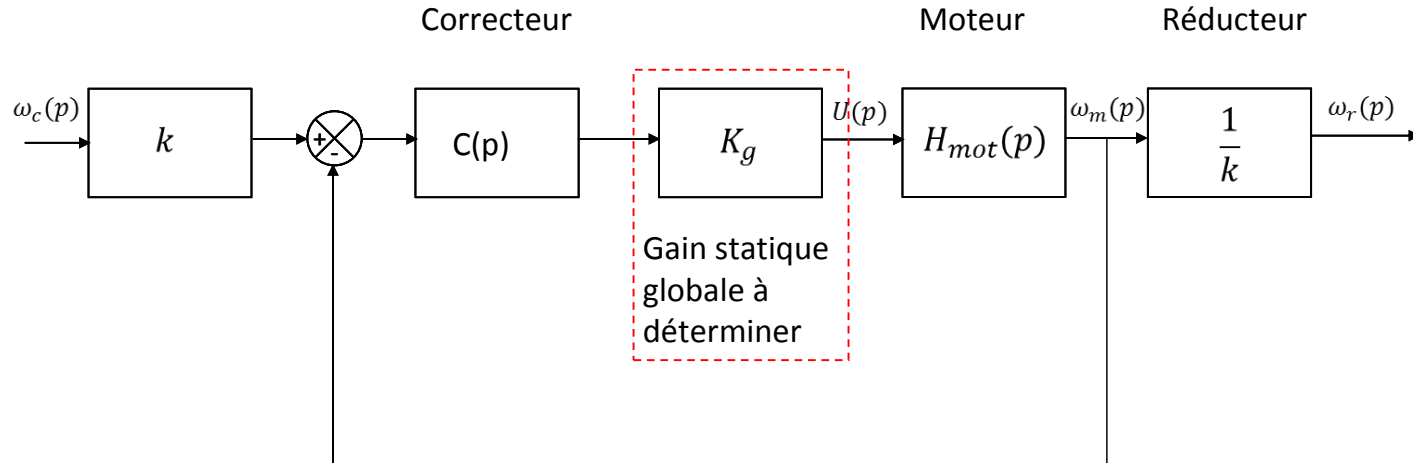
*Réponse échelon de tension*



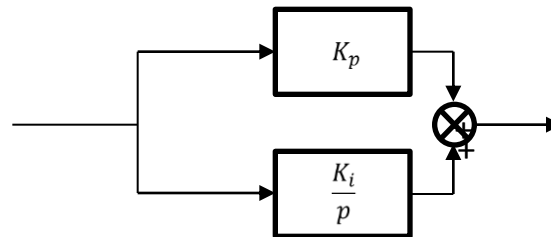


# Asservissement :

## Modèle de l'asservissement en vitesse



## Correcteur utilisé dans la boucle :



```
// Ecart entre la consigne et la mesure
ecart = vref - omega;

// Terme proportionnel
P_x = Kp * ecart;

// Calcul de la commande
commande = P_x + I_x;

// Terme intégral (sera utilisé lors du pas d'échantillonnage suivant)
I_x = I_x + Ki * dt * ecart;
```

## Position du problème

Introduction

Objectifs

## Traitement du signal

Caractéristiques

Amplification

Mise en forme

## Déplacement du fauteuil

Conception prototype

Programmation

Asservissement

## Conclusion



# Asservissement :

## Comparaison théorie-expérience (commande 19rad/s):

### Position du problème

*Introduction*

*Objectifs*

### Traitement du signal

*Caractéristiques*

*Amplification*

*Mise en forme*

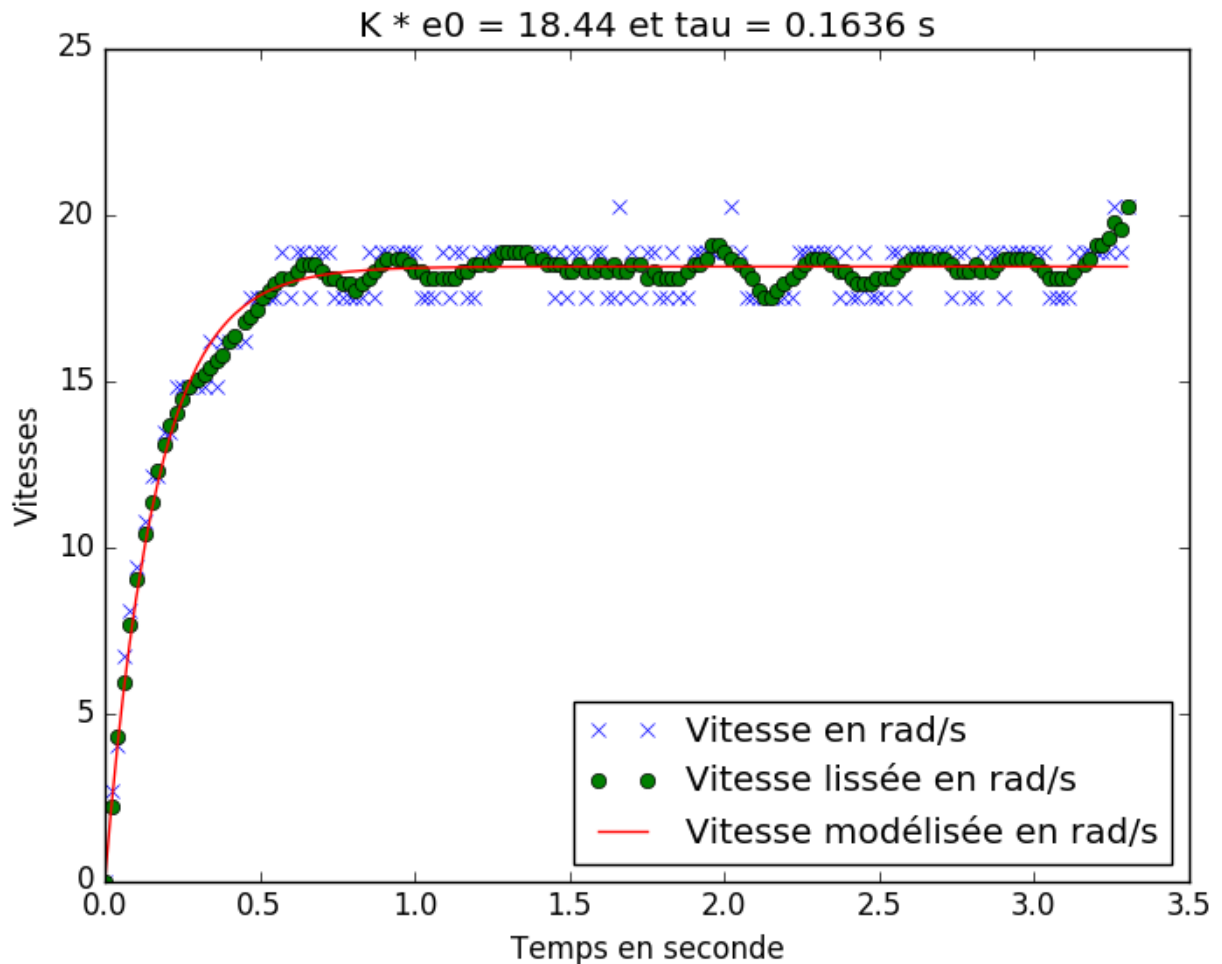
### Déplacement du fauteuil

*Conception prototype*

*Programmation*

*Asservissement*

### Conclusion



# Conclusion

## Position du problème

## Introduction

## Objectifs

## Traitement du signal

## Caractéristiques

## Amplification

### Mise en forme

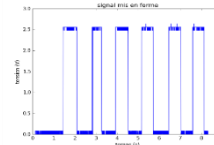
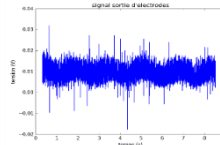
## Déplacement du fauteuil

## Conception prototype

## Programmation

## Asservissement

## Conclusion



```
void ComputerImpulsions() {
    etatsageur=digitalRead(capteur);
    if(etatsageur== HIGHER) {
        compteur++;
        chrono=millis();
        chrono=11111();
        while(chrono-chrono0<3000)
            etatsageur=digitalRead(capteur);
        delay(100);
        if(etatsageur!= memoire) && (etatsageur == HIGHER)
            compteur++;
        memoire=etatsageur;
        delay(100);
        chrono=11111();
    }
}
```

