

# Simulation d'un Contrôleur de Petite Machine à Bois

Matthieu BOYER & Sélène CORBINEAU

14 janvier 2025

# Plan

Positionnement du Problème

Implémentation

Réalisme

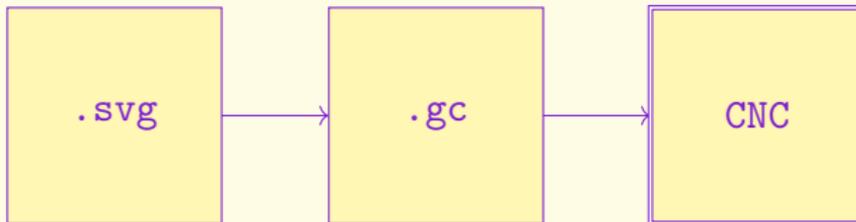
## Petite Machine à Bois



Figure – Une Petite Machine à Bois

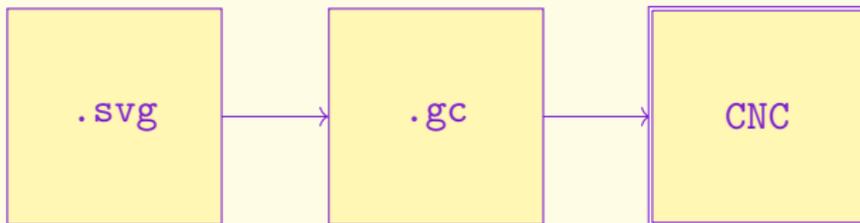
## Format d'Entrée

On convertit un fichier `.svg` en fichier `.gc` qui est une abstraction des commandes à exécuter.



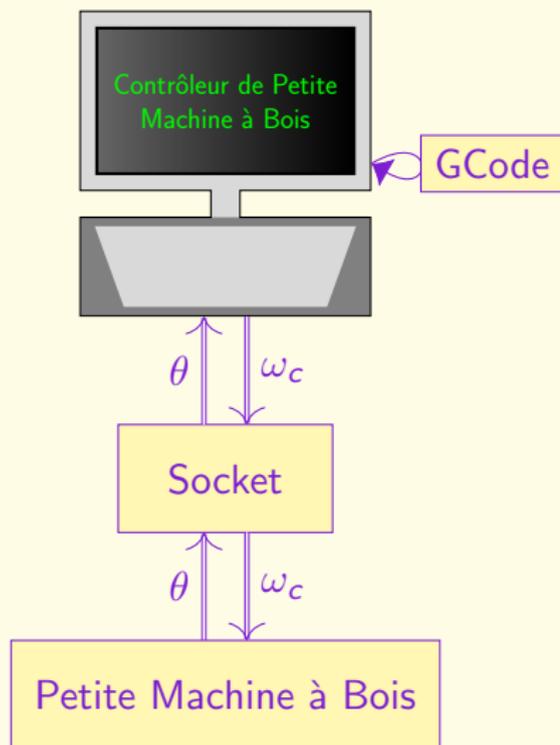
## Format d'Entrée

On convertit un fichier `.svg` en fichier `.gc` qui est une abstraction des commandes à exécuter.



Le format GCode est standardisé, et indépendant des capacités de la machine.

# Communication



**Contrôleur** : traduit le GCode et envoie les vitesses de commande  $\omega_c$  pour les moteurs.

**Socket** : sert d'intermédiaire pour la connexion (CNC).

**Petite Machine à Bois** : envoie les positions  $\theta$  de ses moteurs pas-à-pas.

# Plan

Positionnement du Problème

Implémentation

Réalisme

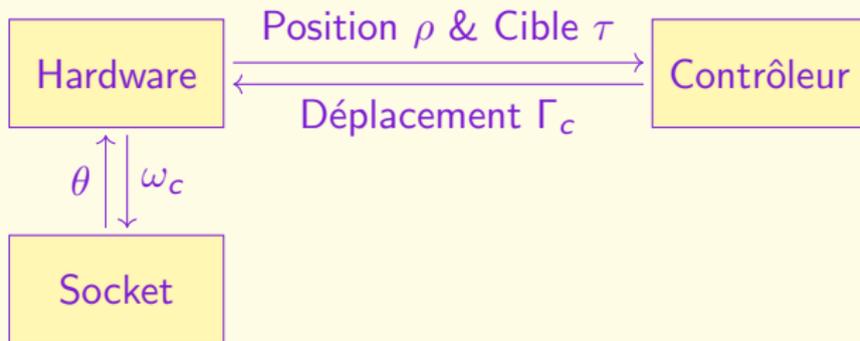
# Traducteur

On agit simplement ligne par ligne, voici quelques commandes :

- |   |  |
|---|--|
| <b>G0</b> : Aller au point donné ;                                      | d'en suivre plusieurs  |
| <b>G1</b> : Aller au point donné, en coupant au passage ;               | d'affilée de manière dérivable ;                                   |
| <b>G2,G3</b> : Faire un arc de cercle, en sens trigonométrique ou non ; | <b>G20</b> : WTF IS A KILOMETER ;                                  |
| <b>G4</b> : Dormir ;  | <b>G21</b> : Changer d'unité en mm ;                               |
| <b>G5</b> : Suivre une courbe de Bézier cubique. Permet                 | <b>G28</b> : Retourner à la maison ;                               |
|   | <b>G30</b> : Retourner à la maison, en passant par un point donné. |

## Module de Contrôle Hardware

Le module Hardware du contrôleur maintient une représentation mémoire de la position *théorique* de la tête, transmise au contrôleur. Il demande à la socket (et donc au *vrai* hardware) les positions mesurées des moteurs pas-à-pas.



Il envoie ensuite à la socket pour envoyer les vitesses de commandes traduites dans la spécification des moteurs.

## Contrôleur en Vitesse

Dans une première approximation, on poursuit le point de contrôle à vitesse constante. On calcule  $\delta = \rho - \tau$ . Selon le signe de  $\delta_x$  (resp.  $y, z$ ) on définit  $\omega_c$  la vitesse de commande de sorte que :

$$\nu\omega_c = 1 \text{ et } (\omega_c)_x \leq \text{FAST\_XY\_FEEDRATE (resp. } y, z)$$

Un contrôleur plus poussé est obtenu en ajoutant une proportionnalité par rapport à l'écart *théorie - mesure*.

## Simulateur

Pour modéliser la Petite Machine à Bois, on construit un simulateur physique simple. On discrétise le temps par :

$$\Delta t = t(\text{dernière requête reçue}) - t(\text{dernière requête traitée})$$

On a ensuite par le PFD :

$$\dot{j}_\omega = \underbrace{-f\omega}_{\text{Inertie}} + \underbrace{(\omega_c - \omega_t)}_{\text{Couple Commande}}, \text{ d'où } \omega_{t+1} = \omega_t + \Delta t(-f\omega + (\omega_c - \omega_t))$$

On introduit par ailleurs un terme d'erreur lié aux frottements, et on peut encore raffiner le modèle.

# Plan

Positionnement du Problème

Implémentation

Réalisme

## Connexion et Précision

L'utilisation d'une socket et d'un modèle théorique restreint comme intermédiaire permet de prendre en compte, durant les phases de test, les différents facteurs d'erreurs pouvant intervenir durant l'exécution :

- ▶ Une connexion trop volatile ;
- ▶ Des erreurs liées au glissement des moteurs pas-à-pas ;
- ▶ Des erreurs liées à la dureté du matériau à couper, et à son anisotropie.