

Concepts importants pour l'utilisateur

Table des matières

- [1. La configuration machine](#)
- [2. Contrôle de trajectoire](#)
 - [2.1. La planification de trajectoire](#)
 - [2.2. Le suivi du parcours](#)
 - [2.3. La programmation du planificateur](#)
 - [2.4. Planification des mouvements](#)
- [3. G-code](#)
 - [3.1. Par défaut](#)
- [4. Vitesse d'avance](#)
- [5. Compensation de rayon d'outil](#)
- [6. Prise d'origine machine](#)
- [7. Changement d'outil](#)
- [8. Systèmes de coordonnées](#)
 - [8.1. G53 Coordonnées machine](#)
 - [8.2. G54 à 59.3 Coordonnées utilisateur](#)
 - [8.3. Quand vous êtes perdu](#)

1. La configuration machine

Le dessin suivant montre les directions de déplacement de l'outil et la position des fins de course de limite sur une fraiseuse classique. Noter le diagramme cartésien représentant les directions de déplacement de l'outil (Tool Direction). La direction de déplacement de la table et en opposition du système de coordonnées cartésiennes. Le système de coordonnées cartésiennes représente le sens de déplacement de l'outil. C'est toujours les déplacements de l'outil qui doivent être programmés pour que l'outil se déplace dans les directions correctes par rapport au matériel.

Noter également la position des fins de course et le sens d'activation de leurs cames. Plusieurs combinaisons sont possibles, par exemple il est possible, à l'inverse du dessin, de placer un seul fin de course fixe au milieu de la table et deux cames mobiles pour l'actionner. Dans ce cas les limites seront inversées, +X sera à droite de la table et -X à gauche. Cette inversion ne change rien du point de vue du sens de déplacement de l'outil.

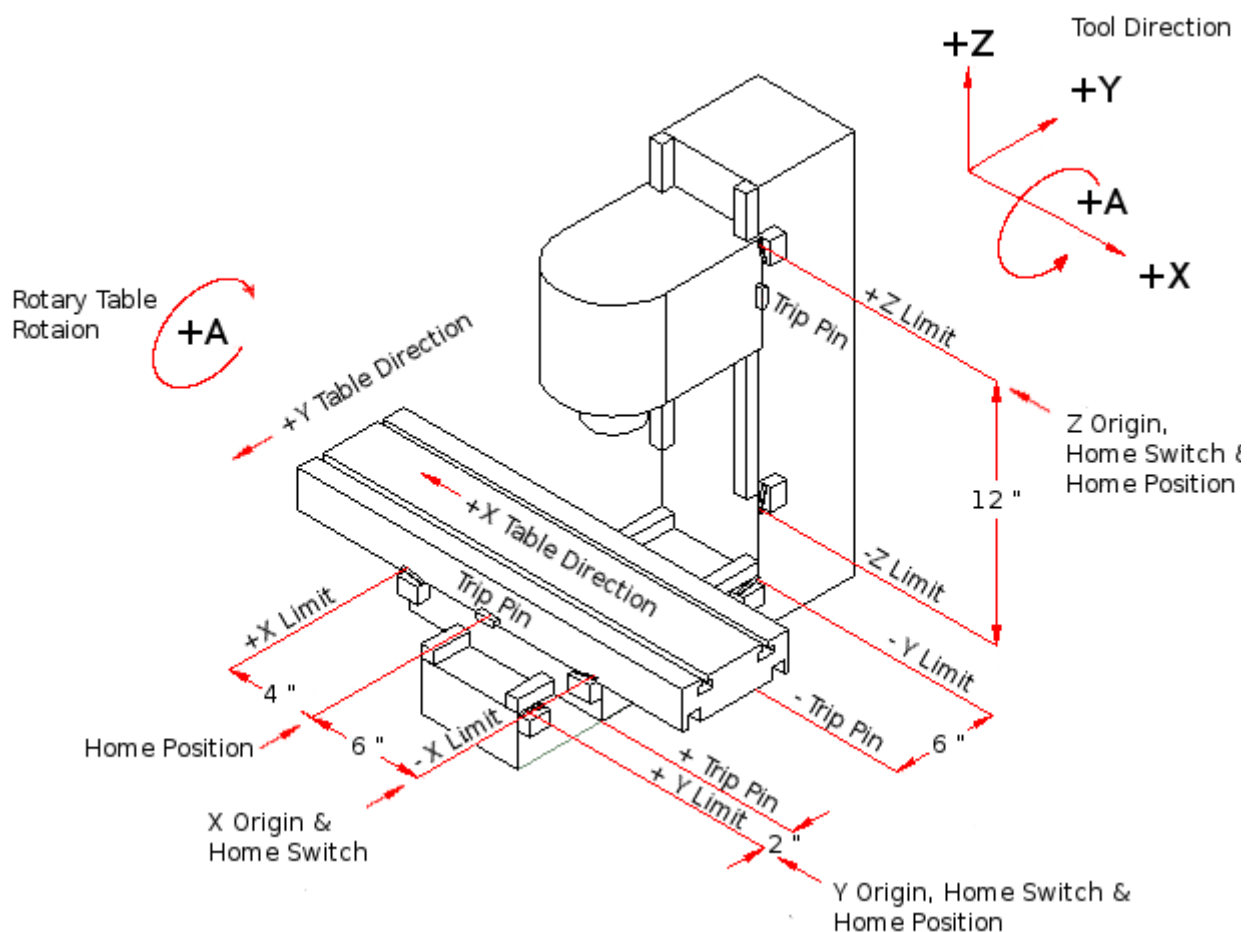


Figure 1. Configuration typique d'une fraiseuse

Le dessin suivant montre les directions de déplacement de l'outil et la position des fins de course de limite sur un tour classique.

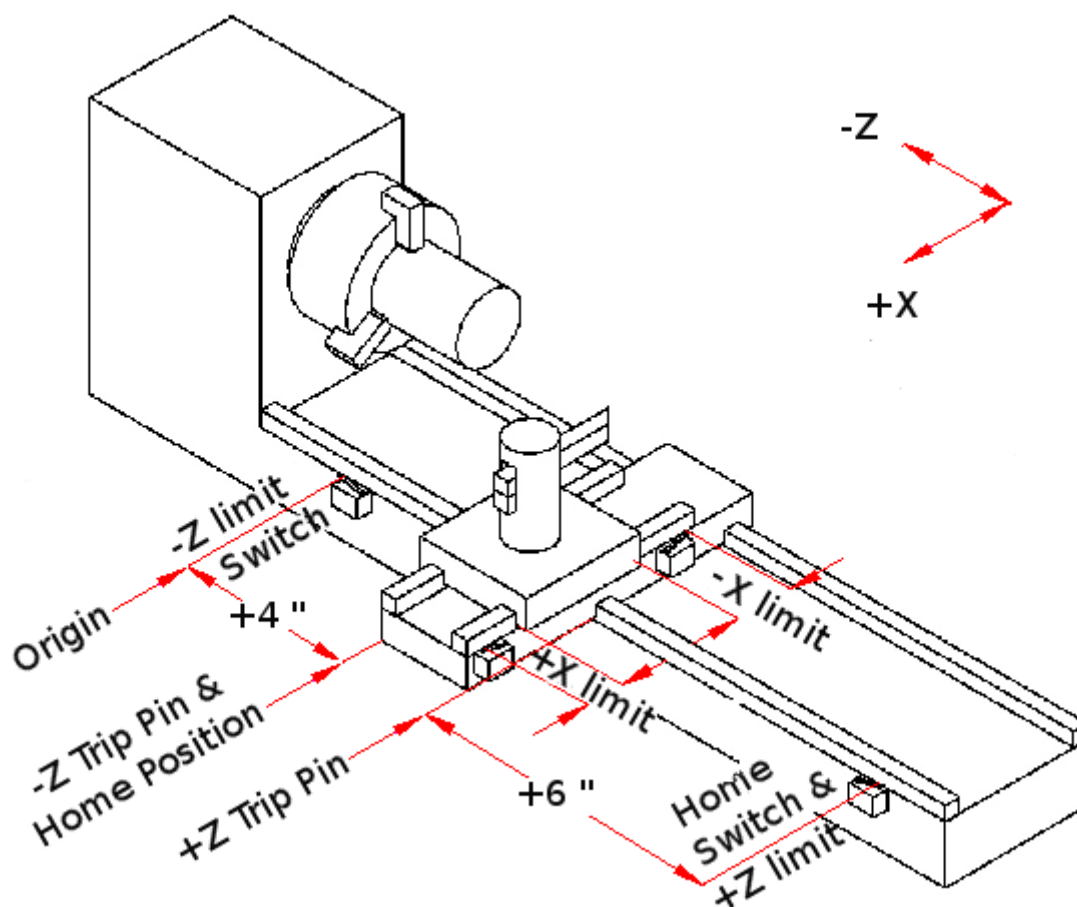


Figure 2. Configuration typique d'un tour

2. Contrôle de trajectoire

2.1. La planification de trajectoire

La planification de trajectoire est en général, le moyen qui permet à LinuxCNC de suivre le chemin spécifié par le programme G-code, tout en restant dans les limites permises par la machine.

Un programme en G-code ne peut jamais être exactement suivi. Par exemple imaginez que vous spécifiez dans une ligne du programme les mouvements suivants:

`G1 X10 F100` (G1 un mouvement linéaire, X10 la destination, F100 la vitesse)

En réalité, la totalité du mouvement ne peut pas être effectuée à F100, puisque la machine commence le mouvement à une vitesse nulle, elle doit accélérer pour se déplacer vers X=10, puis décélérer pour revenir à une vitesse nulle en fin de mouvement. Parfois une portion du mouvement se fera bien à F100, mais pour beaucoup de mouvements, spécialement les petits mouvements, la vitesse spécifiée

ne sera jamais atteinte.

Les accélérations et décélérations de base décrite ici ne sont pas complexes et ne nécessite pas de compromis. Les contraintes des axes de la machine sont placés dans le fichier INI, comme la vitesse maximum de l'axe et l'accélération ne devant pas être dépassées par le planificateur de trajectoire.

2.2. Le suivi du parcours

Un problème plus compliqué est posé par le suivi du parcours. Quand vous programmez un angle droit en G-code, le planificateur de trajectoire peut suivre différents parcours, tous sont bons dans certains cas; il peut décélérer et s'arrêter exactement sur les coordonnées du sommet de l'angle, puis accélérer dans la direction perpendiculaire. Il peut également faire ce qui est appelé le mode *trajectoire continue*, qui consiste à maintenir la vitesse d'avance en passant vers le sommet de l'angle, ce qui nécessite d'arrondir l'angle de façon à respecter les contraintes machine. Vous pouvez remarquer qu'il y a dans ce cas un compromis: vous pouvez ralentir pour avoir un meilleur suivi du parcours, ou conserver une vitesse d'avance élevée au détriment de la finesse des angles, du fait d'un moins bon suivi du parcours. Selon les particularités de l'usinage, du matériau, de l'outillage, etc., le programmeur devra décider du bon compromis.

2.3. La programmation du planificateur

Les commandes de contrôle de trajectoire sont les suivantes:

G61

(mode trajectoire exacte) G61 indique au planificateur de suivre exactement la trajectoire prévue.

G61.1

(mode Arrêt exact) G61.1 demande au planificateur de s'arrêter exactement à la fin de chaque segment. Le parcours sera suivi avec exactitude mais les arrêts complets de l'avance peuvent se révéler destructeurs pour la pièce ou l'outillage, selon les particularités de l'usinage.

G64

(mode trajectoire continue sans tolérance) Le mode G64 est le mode par défaut au démarrage de LinuxCNC. G64 est juste une trajectoire continue, le *Détecteur naive CAM* n'est pas activé. G64 et G64 P0 indiquent au planificateur de sacrifier la précision de suivi du parcours pour conserver une vitesse d'avance élevée. Ce mode est nécessaire pour certains types de matériaux ou d'outillages pour lesquels l'arrêt exact est dangereux. Il peut très bien fonctionner tant que le programmeur garde à l'esprit que le parcours d'outil pourra être plus arrondi que celui indiqué par le programme. Dans le cas d'un mouvement en G0 (rapide) avec G64, faire preuve de prudence sur les mouvements de dégagement et prévoir suffisamment de distance pour éviter les obstacles selon les capacités d'accélérations de la machine.

G64 Px.xxx

(mode trajectoire continue avec tolérance) Ce mode active le *Détecteur naive*

CAM" et active le mode trajectoire continue avec tolérance. Si vous utilisez le millimètre comme unité et programmez G64 P1.27, vous dites au planificateur que vous souhaitez une vitesse d'avance continue, mais qu'aux coins programmés vous voulez un ralentissement suffisant pour que le parcours de l'outil puisse rester à moins de 1.27mm du parcours programmé. L'amplitude exacte du ralentissement dépend de la géométrie de l'angle programmé et des contraintes machine, mais la seule chose dont le programmeur ait à se soucier est la tolérance, ce qui lui donne le contrôle complet des compromis du suivi de parcours. La tolérance de ce mode peut être modifiée tout au long du programme si nécessaire. Attention: spécifier un G64 PO aura le même effet qu'un G64 seul (voir ci-dessus), c'est rendu nécessaire pour conserver la compatibilité ascendante avec les anciens programmes G-code. Voir le chapitre sur le G-code pour plus d'information sur G64 P- Q-.

Trajectoire continue sans tolérance

Le point contrôlé touchera chaque mouvement spécifié à au moins un point. La machine ne pourra jamais se déplacer à une vitesse d'avance telle qu'elle ne puisse pas s'arrêter avec précision à la fin du mouvement en cours (ou du prochain mouvement, si vous mettez en pause lorsque la trajectoire est déjà commencée). La distance avec le point final du mouvement est aussi grande que nécessaire pour maintenir la meilleure vitesse d'avance possible pendant le parcours.

Détecteur Naive Cam

Les mouvements successifs en G1, concernant uniquement les axes XYZ, dont la déviation par rapport à une ligne droite est inférieur à P, sont fusionnés en une seule ligne droite. Ce mouvement fusionné remplace les mouvements individuels en G1 pour obtenir une nouvelle trajectoire avec tolérance. Entre les mouvements successifs, le point contrôlé ne passera jamais à plus de P- du point final du mouvement en cours. Le point contrôlé touchera au moins un point de chacun des mouvements. La machine ne pourra jamais se déplacer à une vitesse ne lui permettant pas de venir s'arrêter exactement à la fin du mouvement actuel (ou du prochain mouvement, si vous mettez en pause lorsque la trajectoire est déjà commencée). En mouvement G2/3 dans le plan G17 (XY) quand la déviation maximale entre un arc et une ligne droite est plus petite que la tolérance G64 Q- l'arc est brisé en deux lignes (du début de l'arc à son milieu et du milieu à la fin de l'arc). Ces deux tronçons sont ensuite soumis à l'algorithme Naïve cam des lignes. Ainsi, les cas ligne-arc, arc-arc et arc-ligne, comme les cas ligne-ligne bénéficient du traitement *Détecteur naive CAM*. Les performances de contournage sont accrues grâce à la simplification de la trajectoire.

Dans la figure suivante la ligne bleue représente la vitesse machine actuelle. La ligne rouge représente la capacité d'accélération de la machine. La ligne horizontale sous chaque tracé est le mouvement planifié. Le tracé supérieur montre comment le planificateur de trajectoire ralentit la machine quand des petits mouvements sont rencontrés. Ceci pour rester dans les limites fixées par les paramètres d'accélération de la machine et être capable de s'arrêter exactement à la fin du prochain mouvement. Le tracé du bas montre l'effet du détecteur Naive Cam pour combiner les mouvements et fournir une amélioration conséquente

dans le suivi de la vitesse programmée.

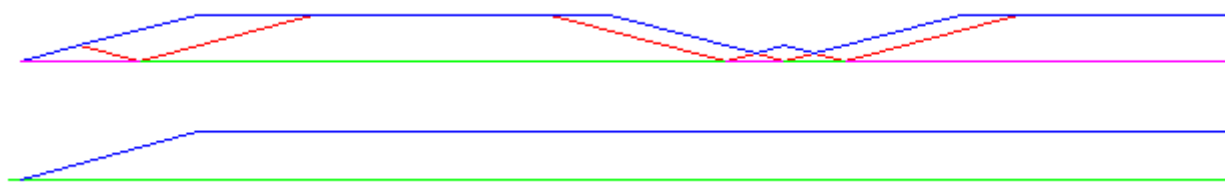


Figure 3. Détecteur Naive Cam

2.4. Planification des mouvements

Assurez-vous que les mouvements soient *assez longs* pour convenir à votre machine/matériel. Principalement en raison de la règle selon laquelle "la machine ne pourra jamais se déplacer à une vitesse ne lui permettant pas de venir s'arrêter complètement à la fin du mouvement actuel", il y a une longueur minimale de déplacement permettant à la machine d'atteindre la vitesse demandée avec un réglage d'accélération donné.

Les phases d'accélération et de décélération utilisent chacune la moitié de la variable `MAX_ACCELERATION` du fichier `.ini`. Avec une trajectoire continue c'est exactement inversé, ce qui fait que l'accélération totale de l'axe est égal à la variable `MAX_ACCELERATION`. Dans d'autres cas, l'accélération actuelle de la machine est un peu inférieure à celle du fichier `ini`.

Pour maintenir la vitesse d'avance, le mouvement doit être plus long que la distance qui lui est nécessaire pour accélérer de zéro à la vitesse souhaitée, puis de décélérer pour s'arrêter. En utilisant A comme étant $1/2$ de la variable `MAX_ACCELERATION` du fichier `ini` et F comme étant la vitesse d'avance *en unités par seconde*, le temps d'accélération sera $t_a = F/A$ et la distance d'accélération sera $d_a = F \cdot t_a / 2$. Les temps et distance de décélération sont les mêmes, ce qui fait que la distance critique $d = d_a + d_d = 2 \cdot d_a = F^2/A$.

Par exemple, pour une vitesse d'avance de *25mm par seconde* et une accélération de *250 mm/sec²*, la distance critique sera de $10^2/100 = 100/100 = 1mm$. Pour une vitesse d'avance de *5mm par seconde*, la distance critique ne serait que de $5^2/100 = 25/100 = 0.25mm$.

3. G-code

3.1. Par défaut

Quand LinuxCNC démarre pour la première fois beaucoup de G et M codes sont chargés par défaut. Les codes actifs courants sont visibles dans l'interface Axis, dans l'onglet *Données manuelles* dans le champ *G-codes actifs*. Ces codes G et M définissent le comportement de LinuxCNC et il est important de bien comprendre

la signification de chacun avant de démarrer LinuxCNC. Ces codes par défaut peuvent être modifiés lors du lancement d'un fichier de G-codes puis laissés dans différents états qui seront identiques lors d'une nouvelle session de LinuxCNC. La bonne pratique consiste à mettre dans le préambule de chaque fichier de G-codes les codes nécessaires pour le travail demandé et ne pas supposer que ceux par défaut conviendront. Imprimer la page des références rapides du G-code peut aider à se rappeler la signification de chacun d'eux.

4. Vitesse d'avance

Si vous avez un tour ou un axe rotatif, pour savoir comment la vitesse d'avance s'applique selon que l'axe est linéaire ou rotatif, lire et comprendre la section [vitesse d'avance](#) du manuel de l'utilisateur.

5. Compensation de rayon d'outil

La compensation de rayon d'outil (G41/G42) nécessite que l'outil puisse usiner tout au long de la trajectoire programmée sans interférer avec les mouvements d'entrée ou de sortie. Si c'est impossible avec le diamètre de l'outil courant, une erreur est signalée. Un diamètre d'outil inférieur est peut être utilisable sans erreur pour le même parcours. Ce qui signifie que quand ce type de problème se présente, il est possible de programmer un outil plus petit pour usiner le même parcours sans erreur. Voir la section compensation de rayon d'outil pour plus d'informations.

6. Prise d'origine machine

Après le démarrage de LinuxCNC chaque axe doit être référencé sur son point d'origine machine avant tout mouvement ou commande MDI.

Pour déroger à ce comportement par défaut, ou pour utiliser l'interface Mini, il est possible d'ajuster l'option `NO_FORCE_HOMING = 1` dans la section [TRAJ] du fichier ini.

7. Changement d'outil

Il existe plusieurs options pour effectuer un changement d'outil. Voir la section [EMCIO] dans le manuel de l'intégrateur pour les informations sur la configuration de ces options. Voir également les sections G28 et G30 du manuel de l'utilisateur.

8. Systèmes de coordonnées

Les systèmes de coordonnées peuvent être déroutant au premier abord. Avant de démarrer une machine CNC, il est important de bien comprendre les bases des systèmes utilisés par LinuxCNC. Pour explorer plus en profondeur les systèmes de coordonnées utilisés par LinuxCNC, voir la section xxxxx de ce manuel.

8.1. G53 Coordonnées machine

Quand vous réalisez une prise d'origine de plusieurs axes de LinuxCNC, vous passez G53, les coordonnées système, à 0 pour chacun des axes concernés.

- La prises d'origine ne modifient en rien les autres systèmes de coordonnées, ni les compensations d'outil.

La seule façon de se déplacer en mode G53, en coordonnées machine, c'est de programmer un G53 sur la même ligne que celle d'un mouvement. En fonctionnement normal, vous êtes dans le système de coordonnées G54.

8.2. G54 à 59.3 Coordonnées utilisateur

Normalement vous utilisez le système de coordonnées G54. Quand un décalage est appliqué au système de coordonnées utilisateur courant, dans Axis, une petite sphère bleue avec des rayons est affichée à l'emplacement de l'origine machine quand la visu affiche *Position: Relative Actuelle*. Si votre décalage utilise temporairement les coordonnées machine, depuis le menu Machine ou en programmant *G10 L2 P1 Xo Yo Zo* à la fin du programme G-Code. Modifiez la valeur du mot *P* en fonction du système de coordonnées dont vous voulez effacer le décalage.

- Les décalages stockés dans un système de coordonnées utilisateur sont conservés à l'arrêt de LinuxCNC.
- Dans Axis, utiliser le bouton *Toucher* décalera le système de coordonnées utilisateur choisi.

8.3. Quand vous êtes perdu

Si vous avez des difficultés pour obtenir 0,0,0 sur la visu alors que vous pensez que vous devriez l'avoir, c'est peut être provoqué par plusieurs décalages programmés et qu'il conviendrait de supprimer. Pour cela:

- Placez vous sur l'origine machine avec *G53 Go Xo Yo Zo*
- Supprimez tous les décalages *G92* avec *G92.1*
- Utilisez les coordonnées utilisateur avec *G54*
- Rendez les coordonnées utilisateur *G54*, identiques aux coordonnées machine avec *G10 L2 P1 Xo Yo Zo Ro*
- Annulez les offsets d'outil avec *G49*
- Activez l'affichage des coordonnées relatives depuis le menu.

Maintenant vous devriez être, à l'origine machine *Xo Yo Zo* et le système de

coordonnées relatives devrait être le même que le système de coordonnées machine.

Dernière mise à jour 2018-02-01 12:09:49 MST