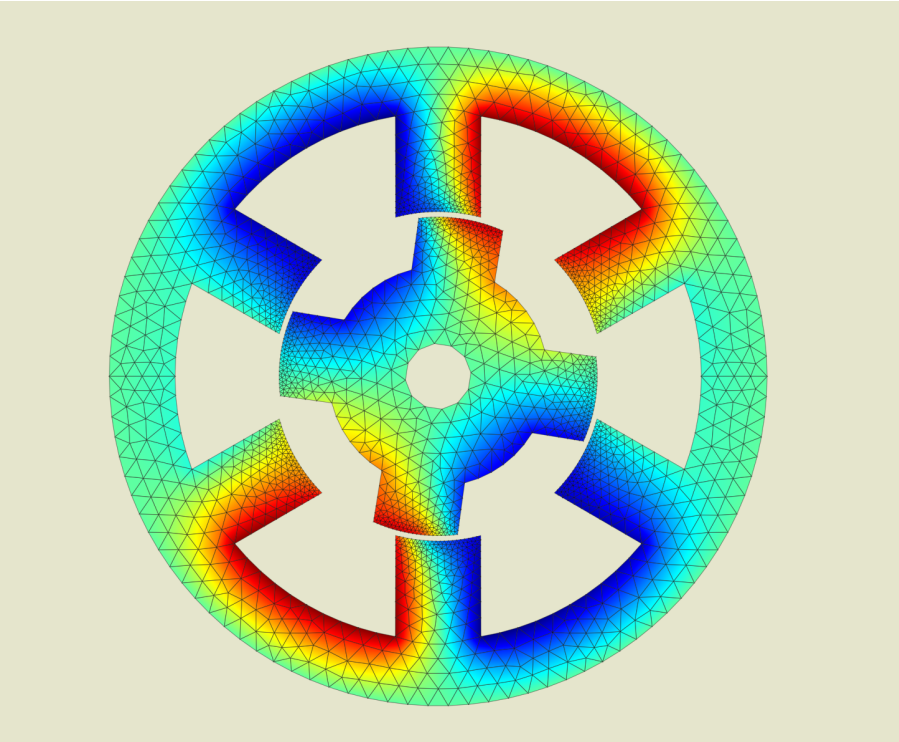
ÉCOLE POLYTECHNIQUE DE LOUVAIN

UCLouvain

LEPL1110 - Éléments finis - 2021

Simulation d'un moteur à reluctance variable par la méthode des éléments finis



Merlin Camberlin 0944-17-00

5 mai 2021

Professeurs :
V Legat

J-F Remacle

Introduction

Le moteur à réluctance variable est un type de moteur électrique relativement récent. Son apparition a été permise par l'apparition de dispositifs électroniques permettant la commutation des courants dans les bobines statoriques.

Le moteur à réluctance variable est composé de 2 parties :

- d'un stator entourés d'encoches autour desquelles des bobines statoriques sont parfois parcourues par un courant continus positivement ou négativement.
- d'un rotor entourés d'encoches d'un nombre inférieur à celles dans le stator et autour desquelles sont enroulées des bobines.

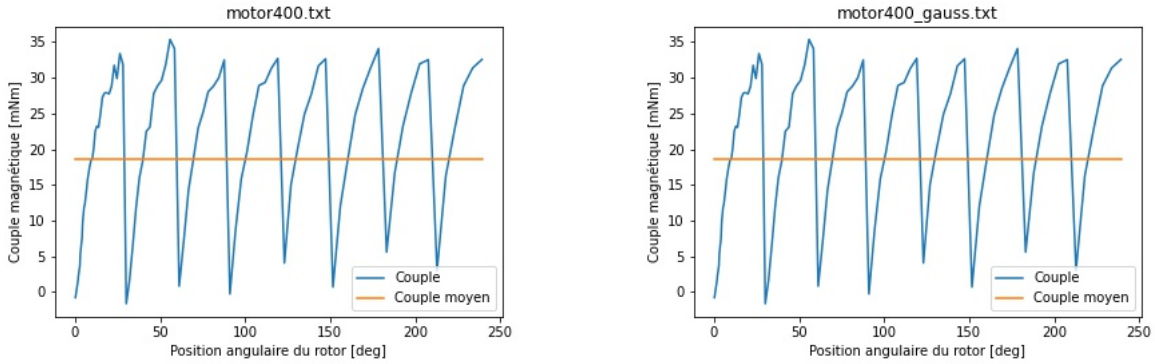
Dans le modèle étudié, le nombre d'encoches au stator est de 6 autour desquelles 3 paires de bobines sont enroulées tandis que le nombre d'encoches au rotor est de 4 autour desquelles 2 paires d'encoches sont enroulées.

Ce rapport contient les détails de la solution implémentée pour simuler le comportement de ce moteur par la méthode des éléments finis.

1 Validation du code

Pour valider la simulation numérique plusieurs techniques ont été réalisées.

Tout d'abord, l'évolution des différents paramètres dynamiques de la simulation ont été comparés entre celle obtenue par une élimination gaussienne et celle obtenue par un solveur bande. Comme l'illustre la FIGURE 3B, les deux graphes sont en tout point identique.



(a) Solveur bande

(b) Elimination gaussienne

FIGURE 1 – Évolutions du couple en fonction de la position angulaire du rotor pour le maillage `motor400.txt` avec une élimination gaussienne et un solveur bande

En outre, les évolutions du couple en fonction de la position angulaire ressemblent sensiblement à celle attendue et illustrée sur la FIGURE 2. De plus, les valeurs de couple obtenue sont du même ordre de grandeur.

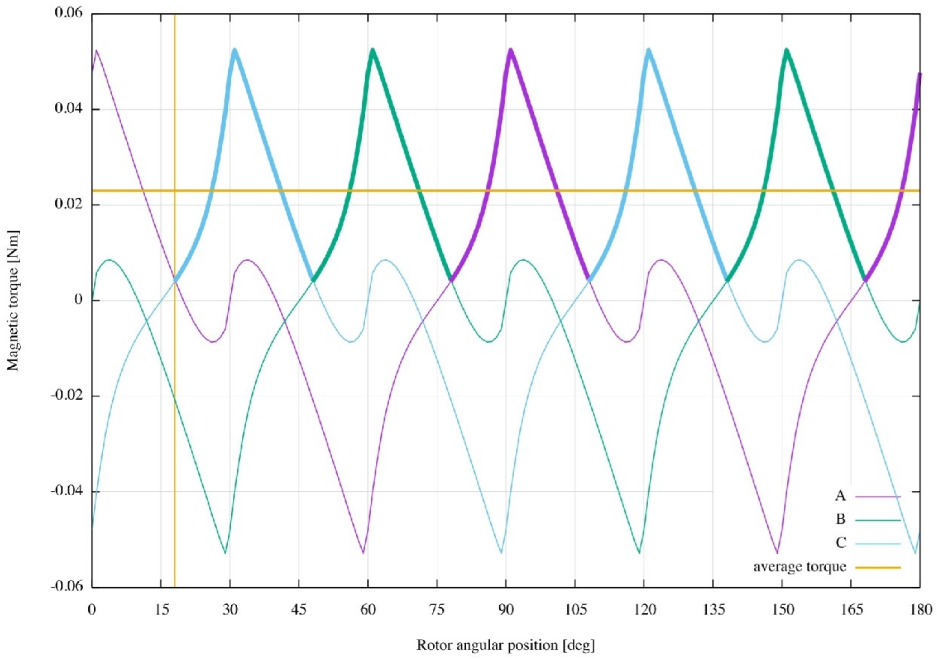
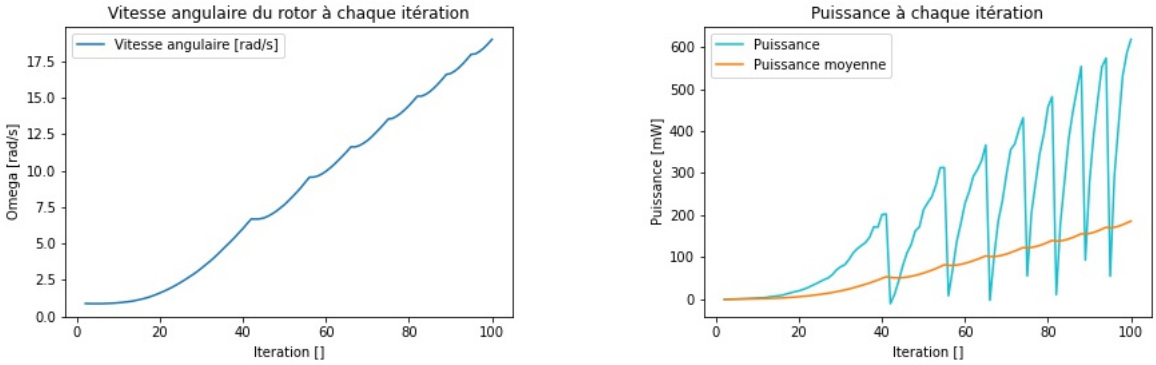


FIGURE 2 – Évolution du couple magnétique en fonction de la position angulaire du rotor attendue pour une machine à réluctance variable (tirée du cours d'introduction aux machines électriques)

Enfin, d'un point de vue physique, la simulation obtenue illustre le phénomène attendu. Sans l'abscence de couple résistant et de couple de frottement, le moteur électrique simulé ne devrait cesser d'accélérer. Comme son couple moyen est presque constant (comme l'illustrait la FIGURE 3B), la puissance moyenne du moteur devrait également continûment croître. C'est exactement le phénomène décrit par la simulation et illustré sur les figures suivantes.



(a) Évolution de la vitesse angulaire oméga au fur et à mesure des itérations de la simulation

(b) Évolution de la puissance au fur et à mesure des itérations de la simulation

FIGURE 3 – Comportement physique décrit par la simulation

2 Optimisations réalisées

2.1 Point de vue implémentation

Pour pouvoir percevoir les optimisations logicielles de la simulation, il faut d'abord déterminer quelles sont les parties du code les plus consommatrices de CPU. A l'aide de la commande `perf` disponible sur *Linux*, il apparaît clairement que la fonction la plus consommatrice de CPU est la fonction effectuant une élimination gaussienne pour déterminer le potentiel magnétique en tout point :

```
double* femFullSystemEliminate(femFullSystem *mySystem);
```

Comme l'illustre la FIGURE 4, à elle seule, elle représente plus de 90% de l'utilisation du processeur. Pour que des optimisations soient perceptibles, c'est donc cette fonction qu'il faut améliorer. Sur le maillage `motor400.txt`, environ 69,18270 secondes sont prises par la simulation pour que le moteur atteigne la vitesse de 150 [rad/s] (avec un pas de temps de 0,005).

```
Samples: 237K of event 'cycles', Event count (approx.): 176573916321
Overhead  Command       Shared Object           Symbol
  92,23%  myFem         myFem                   [.] femFullSystemEliminate
   0,60%  myFem         myFem                   [.] femFullSystemInit
   0,59%  myFem         [kernel.kallsyms]       [k] 0xffffffff8bb6ec2b
   0,42%  myFem         myFem                   [.] femEdgesCompare
   0,41%  myFem         myFem                   [.] motorComputeMagneticPotentialFullSolver
   0,24%  myFem         myFem                   [.] femFullSystemConstrain
```

FIGURE 4 – Performances de la simulation avec une élimination gaussienne sur le maillage `motor400.txt`

Pour améliorer les performances, un solveur bande a été implémenté. Toute chose étant égale par ailleurs (même maillage et même pas de temps), la consommation du CPU chute à environ 58 % (voir FIGURE 5) et le temps d'exécution pris sur la même machine à environ 12.17303 secondes.

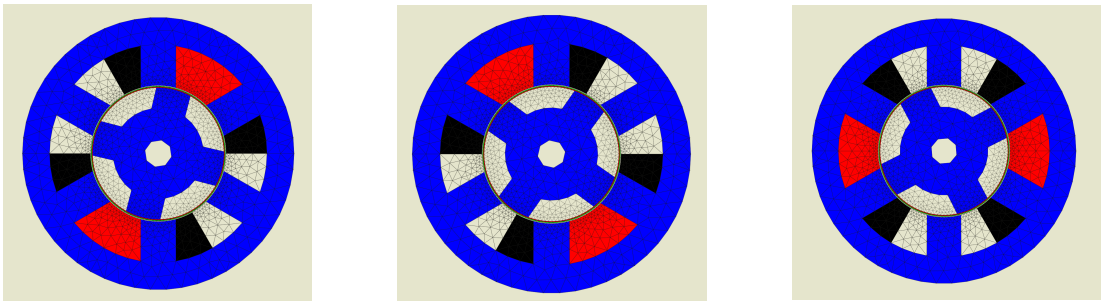
```
Samples: 42K of event 'cycles', Event count (approx.): 29779074959
Overhead  Command       Shared Object           Symbol
  58,45%  myFem         myFem                   [.] femBandSystemEliminate
   3,54%  myFem         [kernel.kallsyms]       [k] 0xffffffff8bb6ec2b
   2,65%  myFem         myFem                   [.] femEdgesCompare
   2,51%  myFem         myFem                   [.] motorComputeMagneticPotential
   1,20%  myFem         libc-2.31.so            [.] msort_with_tmp.part.0
   1,18%  myFem         [kernel.kallsyms]       [k] 0xffffffff8bc0002d
   0,95%  myFem         myFem                   [.] femBandSystemInit
```

FIGURE 5 – Performances de la simulation avec un solveur bande sur le maillage `motor400.txt`

2.2 Point de vue modélisation

Concernant la modélisation, pour obtenir un couple optimal, 2 bobines sont simultanément allumées, de sorte que l'une attire une encoche du rotor pour l'aligner et l'autre la repousse. Sur les 3 illustrations de la simulation sur la FIGURE 6, les bobines alimentées par :

- une densité de courant js sont de couleur rouge
- une densité de courant $js = 0$ ne sont pas colorées
- une densité de courant $-js$ sont de couleur noire.



(a) (b) (c)

FIGURE 6 – Commutation des courants dans les bobines

Ensuite, les valeurs de courant imposées dans les bobines sont constantes ($-js$, 0, $+js$). Bien qu' idéalement, les valeurs de courant devraient dépendre de l'angle de rotation du rotor, des valeurs échelons ont été utilisées pour modéliser au mieux un moteur réel. Ce choix se veut car dans la pratique, c'est ce qui est qui effectué car imposer une valeur du courant variant sinusoïdalement avec l'angle de rotation du rotor nécessite une électronique complexe et avancée.

Conclusion

Pour terminer ce rapport, voici quelques points qui résument le projet. Tout d'abord, l'objectif était de concevoir un programme C permettant de simuler un moteur à reluctance variable. Pour ce faire, la méthode des éléments

finis appliquée à des triangles continus linéaires a été utilisée et pour optimiser ses performances, c'est un solveur bande qui a été utilisé.

Ensuite, pour valider le comportement numérique décrit, une seconde résolution numérique (certes moins performante) a également été implémentée. Les résultats fournies par ces 2 méthodes étant identiques, l'implémentation réalisée et d'autant plus crédible. De plus les valeurs et évolutions des grandeurs décrites suivent celles attendues.