

# Optimisation paramétrique d'un modèle de moteur diesel à injections multiples <sup>★</sup>

Quang N. Tran <sup>\*,\*\*</sup> Emmanuel Witrant <sup>\*\*</sup> Nicolas Bordet <sup>\*\*\*</sup>

*\* Department of Electrical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands  
(e-mail : n.q.tran@tue.nl)*

*\*\* UJF-Grenoble 1/CNRS, Grenoble Image Parole Signal Automatique (GIPSA-lab), UMR 5216, B.P. 46, F-38402 St Martin d'Hères, France  
(e-mail : emmanuel.witrant@ujf-grenoble.fr)*

*\*\*\* Université d'Orléans, Château de La Source, Avenue du Parc Floral, BP 6749, 45067 Orléans cedex 2, France (e-mail : nicolas.bordet@etu.univ-orleans.fr)*

---

Résumé : La simulation numérique joue un rôle crucial pour modéliser les phénomènes physiques. Pour le moteur diesel à injections multiples, un modèle 0D permet de simuler les phases de combustion et d'améliorer la prédiction de la pression dans le cylindre. Chaque sous-modèle, lié à un phénomène physique spécifique, nécessite un certain nombre de paramètres. La plupart des sous-modèles sont calées grâce à des paramètres fournis par la littérature mais il reste néanmoins un jeu de six paramètres (pour chaque injection) difficilement modélisables, liés principalement au modèle de combustion. Ces paramètres sont à identifier pour l'ensemble des points de fonctionnement du moteur. Le but de cet article est d'appliquer des algorithmes d'optimisation au modèle de combustion et de comparer leur performance afin de trouver les paramètres permettant d'avoir une sortie identique à celle mesurée. Il s'agit d'une identification de type "boîte grise", fournissant le modèle nécessaire au réglage de lois de commande pour le moteur.

Mots clés : optimisation non-linéaire, estimation paramétrique, moteur diesel, modèle boîte grise.

---

## 1. INTRODUCTION

La modélisation du moteur diesel à injections multiples est un problème complexe et d'actualité, du fait des objectifs de réduction des émissions d'hydrocarbures et d'oxydes d'azote imposés par la norme Euro 6. Nous considérons dans cet article le modèle 0D des phases de combustion du moteur diesel proposé par Bordet et al. (2009, 2010). Le modèle global est non-linéaire et divisé en deux sous-modèles : un modèle thermodynamique et un modèle de combustion. Le modèle thermodynamique prend en compte les phénomènes physiques impliquant la température et la pression dans le cylindre. Le modèle de combustion décrit l'échange d'énergie et de masse du carburant. Les sous-modèles sont en général modélisés par des équations différentielles et des cartographies.

L'estimation paramétrique de ce modèle implique une stratégie d'identification non-linéaire de type boîte-grise avec contraintes. L'identification paramétrique du modèle de combustion est aussi traitée dans Molinar-Monterrubio et al. (2009) où l'auteur a utilisé la dynamique du système pour établir un modèle discret. Ce modèle a été établi en vue de l'identification paramétrique pour la détection des

fautes du moteur, réalisée grâce à une stratégie d'estimation récursive (moindres carrés). Dans Houpt et al. (1981), la méthode de maximum de vraisemblance est utilisée pour trouver les paramètres d'un modèle de combustion pour la pression cylindre. En pratique, l'auteur a annulé les dérivées de la fonction de vraisemblance par rapport aux paramètres afin d'obtenir une simple minimisation scalaire, qui était traitée numériquement. Par ailleurs, Gambino et al. (1994) ont résolu un problème de minimisation non-linéaire avec contraintes en utilisant l'approche lagrangienne augmentée avec la technique Quasi-Newton. Dans ces travaux, les auteurs ont identifié des paramètres non mesurable de la dynamique de l'écoulement du carburant.

Dans cet article, notre objectif est d'optimiser les paramètres du modèle de moteur diesel à injections multiples à partir d'un algorithme de simulation incluant les dynamiques non-linéaires, cartographies et contraintes. Différentes méthodes d'optimisation (région de confiance, Levenberg-Marquardt, point intérieur et ensembles actifs) classiquement disponibles dans des solveurs numériques (i.e. *Optimization Toolbox* de Matlab<sup>®</sup>) sont évaluées sur la base de données expérimentales. Une comparaison entre différentes normes dans la fonction objectif est aussi traitée, ainsi que l'influence du réglage de la précision.

---

★. Cette recherche a bénéficié de l'aide financière du Septième programme-cadre (7e PCRD) de l'Union Européenne, sous la convention de subvention numéro 257059, le projet 'Autoprofit' (www.fp7-autoprofit.eu).

## 2. MODÈLE DE RÉFÉRENCE

Les détails du modèle de référence et des équations différentielles associées se trouvent dans Bordet et al. (2009, 2010). Nous synthétisons les variables d'état et les paramètres à identifier comme suit. Cinq variables d'état constituent le modèle thermodynamique : la masse du gaz brûlé, celle du gaz non-brûlé, la température du gaz brûlé, celle du gaz non-brûlé et la pression dans le cylindre. Par ailleurs, huit variables d'état sont associées au modèle de combustion : la masse d'air entraînée dans le jet de carburant, l'énergie cinétique associée au mouvement du jet, l'énergie cinétique des mouvements moyen et turbulent dans le jet, la masse de carburant vaporisé, la variance de la fraction du mélange, la masse de carburant consommée par la combustion prémélange et la masse de carburant consommée par la combustion de diffusion. Les entrées du système dépendent du régime du moteur : la vitesse du moteur, l'angle de l'atomiseur, le volume du cylindre, etc. La sortie considérée est la pression dans le cylindre  $P$ .

Nous avons six paramètres à identifier : la température des parois  $T_p$ , la constante de temps du carburant consommé durant la phase de flamme froide  $\tau_C$ , la constante dans l'équation de combustion de prémélange  $C_{phy}$ , les constantes de vitesse et de turbulence  $C_{vmp}$  et  $C_{kd}$  dans l'équation de combustion de diffusion, et un critère spécial  $\kappa$  qui décide de la distribution du carburant vaporisé entre les phases de combustion de prémélange et de diffusion. Les paramètres  $T_p$ ,  $C_{vmp}$  et  $C_{kd}$  sont modélisés par des équations non-linéaires dynamiques dont la classe générale est décrite par :

$$\dot{x} = f(x, u, \theta) \quad (1)$$

où  $f$  est une fonction non-linéaire en l'entrée  $u$ , l'état  $x$  et les paramètres à identifier  $\theta$ . De plus,  $\tau_C$  et  $C_{phy}$  sont fournis par les cartographies du modèle chimique  $M(x, u, \theta)$  qui implique d'autres variables d'état et leur inclusion implique de considérer le modèle :

$$\dot{x} = f(x, u, \theta) + M(x, u, \theta) \quad (2)$$

La dynamique du système a finalement un comportement hybride :

- si  $f(x) \geq \kappa$ ,  $\dot{x} = f_1(x)$ ;
- si  $f(x) < \kappa$ ,  $\dot{x} = f_2(x)$ ;

où  $\kappa$  est le critère de commutation. La détermination des propriétés de commandabilité et d'observabilité de ce système est rendue impossible de par sa complexité. Nous devons donc nous concentrer d'une analyse numérique des méthodes d'identification et de leur capacité à générer une sortie du modèle (i.e. la pression dans le cylindre) identique à la pression mesurée.

## 3. FORMULATION DU PROBLÈME D'OPTIMISATION

Le système de référence est un modèle Simulink® de la combustion (basé sur Bordet et al. (2010) et Bordet et al. (2009)), dont il s'agit d'estimer les paramètres. Ce modèle

est considéré comme une fonction dont les entrées sont les paramètres à identifier et la sortie est la pression dans le cylindre. Il s'agit d'une fonction non-linéaire multi-entrées et simple-sortie. Les algorithmes d'optimisation visent à diminuer l'erreur entre la sortie de cette fonction et celle mesurée. Cette erreur peut être calculée grâce à différentes normes mathématiques.

L'identification des paramètres du système est de type "boîte-grise", du fait des lois physiques combinées à des cartographies expérimentales composant le système de référence. Il s'agit d'un problème de minimisation de la fonction objectif, soumis à des contraintes sur les variables d'optimisation :

$$\theta^* = \arg \min_{\theta \in \mathcal{D}} \{f = \|P_{modèle}(\theta) - P_{mesurée}\|\} \quad (3)$$

où  $f$  est la fonction objectif à minimiser (i.e. l'erreur entre la pression obtenue du modèle et celle de la mesure),  $\theta$  est l'ensemble des six paramètres inconnus,  $\mathcal{D} \subset \mathbb{R}$  est l'ensemble qui décrit les limites physiques des paramètres,  $P_{modèle}(\theta)$  est le vecteur de la pression en sortie du modèle et dépend donc des paramètres (les éléments du vecteur sont associés à des angles de vilebrequin),  $P_{mesurée}$  est le vecteur de pression mesurée. La fonction objectif est calculée en utilisant les normes 1, 2 (moindres carrés) ou infini.

Quatre algorithmes d'optimisation sont considérés : *région de confiance*, *Levenberg-Marquardt*, *point intérieur* et *ensembles actifs*. La performance d'un algorithme peut être déterminée par le nombre d'itérations (i.e. la vitesse de convergence) et par la fonction objectif relative, définie par :

$$f_{relative} = \frac{f_{finale}}{f_{initiale}} \quad (4)$$

où  $f_{initiale}$  est évaluée pour une valeur de référence de  $\theta$  et  $f_{finale}$  correspond à la solution de (3) pour une méthode d'optimisation donnée. Cette fonction représente la diminution de la différence entre la sortie mesurée et la sortie simulée et est exprimée en %.

Des valeurs initiales (réglage manuel) ont été proposées dans Bordet et al. (2009, 2010). La sortie du modèle avec ces valeurs est proche de celle mesurée et sera par la suite mentionnée comme *Pression avec les paramètres initiaux* ou *Pression avec les paramètres du modèle Simulink*.

Lorsque ces valeurs sont utilisées comme celles initiales pour l'identification, le but est de raffiner la sortie du modèle pour que les deux pressions coïncident plus précisément. Il s'agit du premier scénario de simulation. Ces valeurs initiales sont :

$$[T_p \ \tau_C \ C_\phi \ C_{vmp} \ C_{kd} \ \kappa] = [400 \ 10^{-3} \ 0,3 \ 0,5 \ 0,5 \ 10^{-3}]$$

La second scénario a pour but d'évaluer la performance des algorithmes lorsque nous initions les itérations à partir d'un point loin de l'optimal (proche des contraintes physiques). Le but de ce scénario est d'examiner la sensibilité des méthodes d'optimisation aux minimas locaux. Les valeurs fictives sont :

$$[T_p \ \tau_C \ C_\phi \ C_{vmp} \ C_{kd} \ \kappa] = [700 \ 0,01 \ 1 \ 0,005 \ 5 \ 10^{-3}]$$

La sortie du modèle avec ces valeurs sera par la suite mentionnée comme *Pression avec les paramètres fictifs*.

Les contraintes physiques sur les paramètres sont :

$$400 \leq T_p \leq 700, \quad 10^{-5} \leq \tau_C \leq 10^{-2}, \quad 0,05 \leq C_\phi \leq 1, \\ 10^{-3} \leq C_{vmp} \leq 5, \quad 10^{-3} \leq C_{kd} \leq 5, \quad 10^{-9} \leq \kappa \leq 1.$$

#### 4. MÉTHODES D'OPTIMISATION

Dans cette section, les valeurs de la fonction objectif absolue (3) ainsi que celle relative (4) sont fournies afin d'évaluer la performance des algorithmes. De plus, les critères d'arrêt des itérations (e.g. la tolérance et le nombre d'itérations) sont pris en compte lorsqu'ils sont remarquables en vue de la comparaison.

##### 4.1 Région de confiance et Levenberg-Marquardt

L'algorithme *région de confiance* (Yuan, 2000) considère le problème d'optimisation dans un sous-ensemble des paramètres. Si cet ensemble est fiable, il est élargi. Sinon, il est contracté afin d'obtenir la région fiable. L'algorithme *Levenberg-Marquardt* peut être considéré comme étant celui de Gauss-Newton combiné avec la méthode de *région de confiance*. Ces deux algorithmes sont utilisés avec le solveur *lsqnonlin* de Matlab<sup>®</sup> (résolution du problème non-linéaire des moindres carrés). Les méthodes de moindres carrés (e.g. Gauss-Newton et Levenberg-Marquardt) sont présentées et synthétisées dans Nocedal et al. (1999); Björck (1996) et Golub et al. (1996). Madsen et al. (2004) produit une vue d'ensemble des méthodes des moindres carrés pour les systèmes non-linéaires. La fonction objectif dans (3) est la somme des carrés des erreurs :

$$f(\theta) = \sum_{k=1}^N |P_{\text{modèle}}(\theta, \varphi_k) - P_{\text{mesurée}}(\varphi_k)|^2 \quad (5)$$

où  $\varphi_k$  est l'angle de vilebrequin de l'essai moteur et  $N$  le nombre de mesures.

*Scénario 1 :* Les résultats issus du premier scénario sont présentés Figure 1a. La fonction objectif est de  $4,93 \times 10^{13}$  avec les paramètres initiaux et de  $3,01 \times 10^{13}$  après optimisation. La fonction objectif relative (4) est de 61%. Nous avons donc amélioré la ressemblance entre les sorties. Les valeurs des paramètres obtenus sont :

$$[T_p \ \tau_C \ C_\phi \ C_{vmp} \ C_{kd} \ \kappa] = [440 \ 0,0055 \ 0,37 \ 0,0021 \ 0,91 \\ 10^{-3}]$$

Le solveur a effectué dix-neuf itérations pour obtenir la convergence. Nous avons ici fixé une tolérance de  $10^{-4}$  pour les paramètres et de  $10^{-4}$  pour la sortie. Le critère d'arrêt associé à la tolérance sur les paramètres est défini par :

$$|\theta_l - \theta_{l+1}| < \epsilon_\theta * (1 + |\theta_l|) \quad (6)$$

où  $\theta$  est le vecteur des paramètres,  $l$  est l'indice d'itération et  $\epsilon_\theta$  est la tolérance des paramètres. Le critère d'arrêt associé à la tolérance sur la sortie est défini par :

$$|f(\theta)_l - f(\theta)_{l+1}| < \epsilon_f * (1 + |f(\theta)_l|) \quad (7)$$

où  $f(\theta)$  est la fonction objectif absolue,  $l$  est l'indice et  $\epsilon_f$  est la tolérance pour la fonction. Les tolérances utilisées ci-dessus sont les plus petites admissibles pour une amélioration significative du résultat. Par ailleurs, les paramètres trouvés nous montrent que les contraintes de borne sont respectées.

*Scénario 2 :* Le second scénario (valeurs initiales fictives) est présenté Figure 1b. La fonction objectif avec ces valeurs est de  $1,76 \times 10^{15}$ . Après optimisation, la fonction objectif est de  $2,94 \times 10^{13}$  après quinze itérations (soit 1,67% de celle initiale). Ces résultats sont proches du scénario 1 et valident la robustesse vis-à-vis de la condition initiale. La vitesse de convergence n'augmente pas lorsque nous commençons par un point proche des bornes. Les valeurs trouvées sont :

$$[T_p \ \tau_C \ C_\phi \ C_{vmp} \ C_{kd} \ \kappa] = [531 \ 0,0053 \ 0,75 \ 10^{-3} \ 0,50 \\ 10^{-3}]$$

Ces valeurs sont relativement distinctes des valeurs trouvées dans le cas précédent malgré la similitude des coûts, ce qui illustre la présence de minima locaux associés aux non-linéarités du système.

Nous venons de vérifier la performance de l'algorithme *région de confiance* pour le solveur *lsqnonlin*. En ce qui concerne l'algorithme *Levenberg-Marquardt*, nous constatons qu'il existe très peu de différence entre ces deux algorithmes. Néanmoins, l'algorithme *Levenberg-Marquardt* exige beaucoup plus d'itérations pour trouver l'optimum. Les résultats numériques sont présentés Table 1.

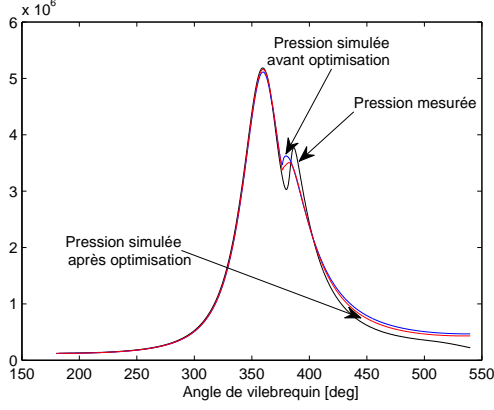
##### 4.2 Ensembles actifs et point intérieur

L'algorithme *ensembles actifs* est basé sur la solution de l'équation de Karush-Kuhn-Tucker (KKT). Si le problème est convexe, cette équation est la condition nécessaire et suffisante pour trouver l'optimum global (cf. Nocedal et al. (1999), ch. 16). La méthode du *point intérieur* approxime le problème d'optimisation par une séquence de problèmes avec des variables de ressort (slack variables) ajoutées. Une description détaillée de la méthode se trouve dans Byrd et al. (1999); Forsgren et al. (2002) et le chapitre dix-neuf de Nocedal et al. (1999). Ces deux algorithmes sont utilisés avec le solveur Matlab<sup>®</sup> *fmincon*. Ce solveur nous permet de minimiser la norme de l'erreur. Nous testons ici les algorithmes ainsi que différentes normes (1, 2 et  $\infty$ ).

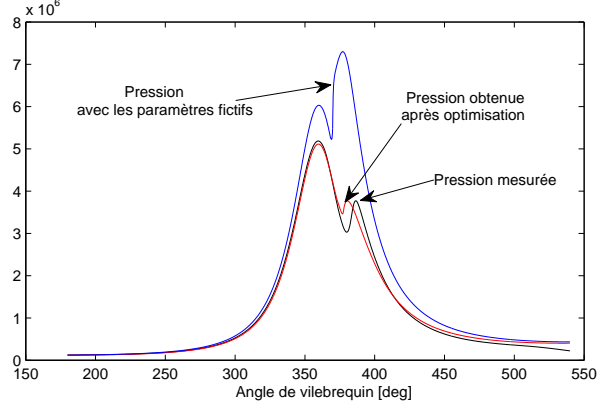
##### Norme 2

La norme 2 (fonction objectif (5)) est tout d'abord optimisée avec l'algorithme *point intérieur* et le Scénario 1. Nous obtenons la convergence après quinze itérations, et une fonction objectif finale de  $2,65 \times 10^{13}$  (54% de celle initiale). Cet algorithme converge plus vite que l'algorithme *région de confiance* précédent. En revanche, cette méthode est moins robuste : nous ne trouvons plus le point optimal dans le cadre du Scénario 2 (cf. Figure 2a).

En utilisant toujours la norme 2, nous appliquons l'algorithme *ensembles actifs*. Les valeurs initiales des paramètres sont celles du second scénario. Nous trouvons un bon résultat au niveau de la ressemblance entre la pression mesurée et celle simulée. La convergence est cependant obtenue très lentement : après 65 itérations, nous n'obtenons pas le point optimal, mais l'algorithme s'arrête à cause du grand nombre de calculs de fonction objectif (605 fois). Le gain en performance à partir d'une initialisation fictive peut être associé au fait que cet algorithme essaye de créer et calculer des matrices pleines (algorithme à échelle moyenne), tandis que l'algorithme de point intérieur traite seulement les matrices éparées. Ceci implique par contre la



(a) Scénario 1



(b) Scénario 2

FIGURE 1. Effet des conditions initiales sur l'optimisation par région de confiance (solveur *lsqnonlin*, pression en [Pa]).

nécessité de plus d'itérations. Le problème de convergence est résolu en imposant les tolérances des conditions d'arrêt (6) et (7). Nous utilisons la tolérance  $10^{-3}$  pour les paramètres et  $5 \times 10^{-3}$  pour la fonction objectif (valeurs par défaut de  $10^{-6}$ ). Avec ces tolérances, l'algorithme s'arrête après 33 itérations et la fonction objectif obtenue est acceptable :  $2,04 \times 10^{13}$  (1,16% de celle initiale). La courbe de la pression simulée est très proche de celle mesurée (cf. Figure 2b), avec les valeurs :

$$[T_p \tau_C C_\phi C_{vmp} C_{kd} \kappa] = [674 \ 6,85 \times 10^{-4} \ 0,60 \ 10^{-3} \ 0,091 \ 10^{-3}]$$

### Norme 1

La norme 1 calcule la somme des modules des éléments de l'écart entre la pression mesurée et celle simulée. Le problème d'optimisation (5) est donc formulé avec la fonction de coût :

$$f(\theta) = \sum_{k=1}^N |P_{\text{modèle}}(\theta, \varphi_k) - P_{\text{mesurée}}(\varphi_k)| \quad (8)$$

L'algorithme *point-intérieur* s'avère efficace pour le premier scénario (cf. Figure 2c). La fonction objectif obtenue est de  $1,37 \times 10^8$  par rapport à  $2,49 \times 10^8$  initiale (fonction objectif relative 55%). La vitesse de convergence est acceptable (15 itérations).

Comme dans le cas d'utilisation de norme 2, l'algorithme *ensembles actifs* converge plus lentement mais permet une meilleure précision (cf. Figure 2d), avec une fonction objectif de  $1,27 \times 10^8$  après 31 itérations (51% de celle initiale). Si nous diminuons la tolérance, le nombre d'itérations augmente et la fonction objectif baisse légèrement (i.e. le point obtenu n'est pas encore le point initial du scénario 1, mais en est très proche).

### Norme $\infty$

La norme  $\infty$  calcule le maximum des éléments du vecteur d'erreur. La fonction de coût s'écrit dans ce cas :

$$f(\theta) = \max_k |P_{\text{modèle}}(\theta, \varphi_k) - P_{\text{mesurée}}(\varphi_k)| \quad (9)$$

Le fait que seule l'erreur maximale soit considérée permet à la norme infini de corriger les gros écarts entre les pressions mesurée et simulée, avec les différents algorithmes. Sur la Figure 2e, nous constatons un meilleur résultat que précédemment, au niveau de la ressemblance entre les sorties, avec une meilleure représentation du second pic.

En revanche, lorsque la pression simulée est loin de celle mesurée (Scénario 2), l'utilisation de la norme infini cherchera à minimiser les écarts les plus grands et donc négligera les écarts petits. Ceci conduit à des résultats moins bons qu'avec les méthodes précédentes (cf. Figure 2f) et la norme  $\infty$  devra donc être utilisée pour affiner les résultats proches d'un point optimal.

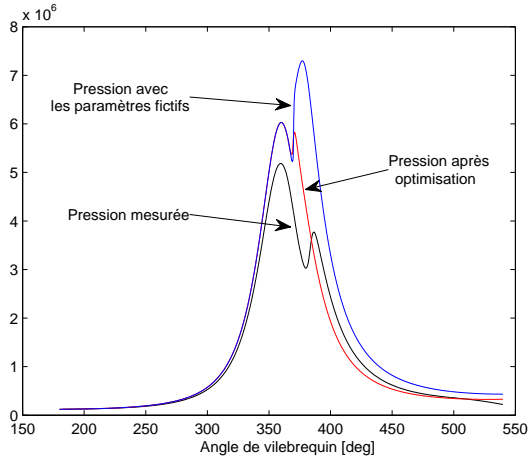
### 4.3 Synthèse des performances et choix de l'algorithme

Trois tables de performance sont établies, pour les trois normes considérées (Tables 1, 2 et 3). Les valeurs dans les tables sont les fonctions objectif absolues et relatives après optimisation. Nous constatons que pour le premier scénario, l'algorithme *point-intérieur* converge plus vite, quelle que soit la norme utilisée. Cependant, il est moins efficace que l'algorithme *ensembles actifs* pour le second scénario. Cela peut être expliqué par le fait que l'algorithme *point-intérieur* est à grande échelle (i.e. il calcule des matrices éparées) alors que l'algorithme *ensembles actifs* est à échelle moyenne (i.e. il calcule des matrices pleines). L'utilisation de la norme  $\infty$ , permettant d'affiner la courbe d'un point proche de l'optimum, est donc appropriée au Scénario 1. Les autres normes sont plus appropriées pour le Scénario 2, en raison de leur robustesse. A partir des résultats ci-dessus, nous pouvons faire un choix sur l'algorithme approprié comme suit.

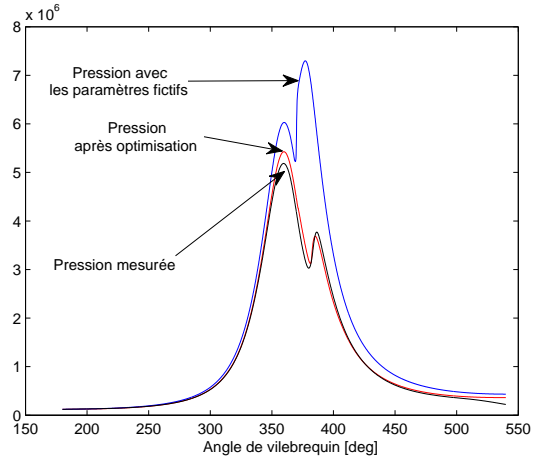
Pour le Scénario 1, l'algorithme fonctionnant le mieux pour notre cas est le *point-intérieur* en utilisant la norme infini (cf. Figure 2e). Les valeurs proposées des paramètres sont :

$$[T_p \tau_C C_\phi C_{kd} C_{vmp} \kappa] = [551 \ 0,0089 \ 1 \ 0,0012 \ 0,70 \ 0,0046]$$

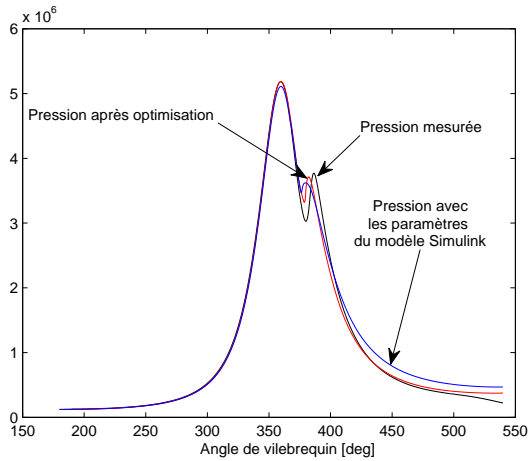
Cet algorithme est efficace lorsque le point initial n'est pas trop loin du point optimal. Cependant, dans le cas où nous connaissons très peu les paramètres (i.e. le point initial est loin de l'optimum), l'algorithme *ensembles actifs* avec la



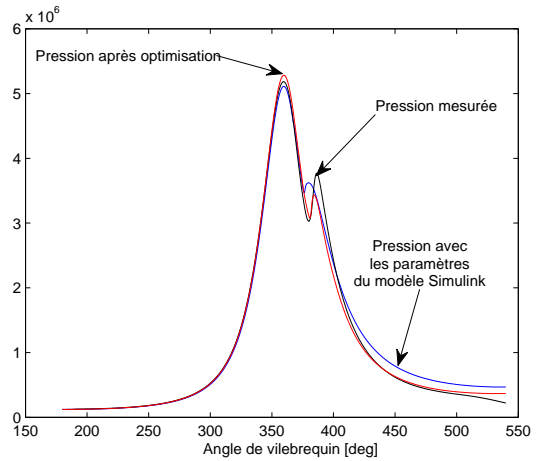
(a) Méthode du point intérieur, norme 2 pour le scénario 2



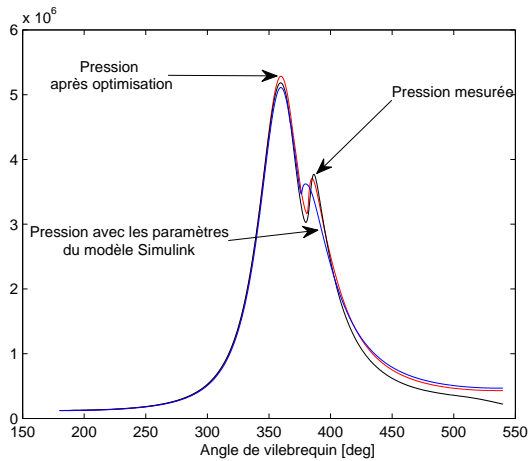
(b) Méthode "ensembles actifs", norme 2 pour le scénario 2



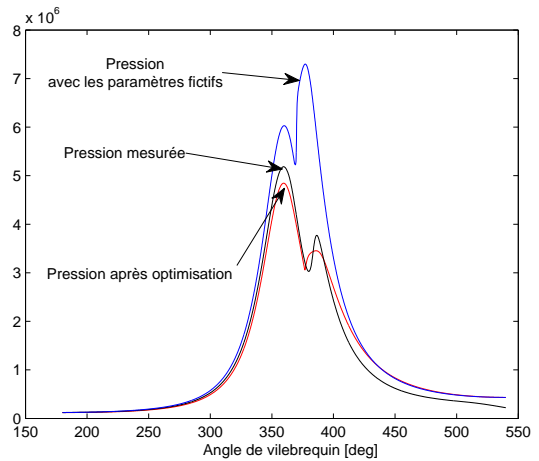
(c) Méthode du point intérieur, norme 1 pour le scénario 1



(d) Méthode "ensembles actifs", norme 1 pour le scénario 1



(e) Méthode du point intérieur, norme  $\infty$  pour le scénario 1



(f) Méthode "ensembles actifs", norme  $\infty$  pour le scénario 2

FIGURE 2. Optimisation des normes 1, 2 et  $\infty$  avec les méthodes du point intérieur et "ensembles actifs" (solveur *fmincon*, pression en [Pa]).

norme 2 (ou la norme 1) est préférable, du fait de ses caractéristiques de robustesse (cf. Figure 2b).

## 5. CONCLUSIONS

Une application de différentes méthodes d'optimisation et la comparaison de leur performance ont été effectuées

TABLE 1. Performance des algorithmes en utilisant la norme 2

Solveur-algorithme	A partir des valeurs Simulink	nombre d'itérations	A partir des valeurs fictives	nombre d'itérations
lsqnonlin-région de confiance	$3,0137 \times 10^{13}$ 61,15%	19	$2,9448 \times 10^{13}$ 1,67%	15
lsqnonlin-Levenberg-Marquardt	$3,0137 \times 10^{13}$ 61,15%	85	$2,8904 \times 10^{13}$ 1,64%	85
fmincon-point intérieur	$2,6533 \times 10^{13}$ 53,84%	15	$3,9800 \times 10^{14}$ 2,26%	5
fmincon-ensembles actifs	$3,4115 \times 10^{13}$ 69,23%	39	$2,0433 \times 10^{13}$ 1,16%	33

TABLE 2. Performance des algorithmes en utilisant la norme 1

Solveur-algorithme	A partir des valeurs Simulink	nombre d'itérations	A partir des valeurs fictives	nombre d'itérations
fmincon-point intérieur	$1,3728 \times 10^8$ 55,24%	15	$2,7911 \times 10^8$ 29,32%	28
fmincon-ensembles actifs	$1,2743 \times 10^8$ 51,40%	31	$1,3753 \times 10^8$ 14,45%	41

TABLE 3. Performances des algorithmes en utilisant la norme infini

Solveur-algorithme	A partir des valeurs Simulink	nombre d'itérations	A partir des valeurs fictives	nombre d'itérations
fmincon-point intérieur	$2,4357 \times 10^5$ 41,05%	15	$9,6390 \times 10^5$ 23,08%	17
fmincon-ensembles actifs	$3,6500 \times 10^5$ 61,51%	30	$3,4365 \times 10^5$ 8,23%	33

sur le modèle diesel à injections multiples. Six paramètres du modèle ont été trouvés en vue d'obtenir une pression dans le cylindre modélisée identique de celle mesurée, en utilisant une modélisation de type boîte-grise. Notre analyse a permis de construire un protocole d'identification pouvant être appliqué à la plupart des simulateurs de moteur à injections multiples. La capacité des algorithmes à traiter les non-linéarités et les couplages entre variables nous autorise à travailler directement avec un modèle de connaissance, fortement non-linéaire et complexe.

Les performances des méthodes d'optimisation ont été évaluées sur un essai moteur à 2400 tr/min et quatre bars. D'autres essais moteur sont envisagés dans le futur pour calibrer le modèle sur d'autres points d'opération. Concernant la comparaison des algorithmes, une étude plus approfondie sur les tolérances serait nécessaire, car l'utilisation efficace de celles-ci diminuerait le temps d'exécution ainsi que le nombre d'itérations, notamment dans le cas où la fonction objectif montre une convergence lente (e.g. les algorithmes *lsqnonlin - région de confiance* et *fmincon - ensembles actifs* décrits ci-dessus).

## RÉFÉRENCES

- Bordet, N., Caillol, C., Higelin P. et Talon V. A physical 0D combustion model using tabulated chemistry with presumed probability density function approach for multi-injection diesel engines. *International Powertrains, Fuels and Lubricants Meeting*, Rio De Janeiro, Brazil, 2010.
- Bordet, N., Caillol, C., Higelin P. et Talon V. Zero-combustional model of premixed combustion in diesel engines. *6th Mediterranean Combustion Symposium*, Porticcio - Ajaccio, Corsica - France, 2009.
- Nocedal, J. et Wright, S. *Numerical optimization*. Springer, New York, 1999.
- Björck, A. *Numerical methods for least squares problems*. SIAM, Philadelphia, 1996.
- Golub, G.H. and Van Loan, C.F. *Matrix computations*. The Johns Hopkins University Press, Baltimore, 1996.
- Madsen, K., Nielsen, H.B. et Tingleff, O. *Methods for Non-Linear Least Squares Problems*. Lecture Notes, Technical University of Denmark, 2004.
- Forsgren, A., Gill, P.E. et Wright, M.H. Interior Methods for Nonlinear Optimization. *SIAM Review*, vol. 44, n° 4, pp. 525-597, 2002.
- Molinar-Monterrubio, J.M. et Castro-Linares, R. Internal combustion engine parametric identification scheme for misfire fault detection : Experimental results. *IEEE ICIT*, Gippsland, Australia, 2009.
- Haupt, P.K. et Tan, H.N. Identification of a thermodynamic model for spark ignited internal combustion engines. *Decision and Control including the Symposium on Adaptive Processes, 20th IEEE Conference*, San Diego, CA, USA, 1981.
- Gambino, M., Pianese, C. et Rizzo, G. Experimental and computational analysis of a dynamic model for control strategy optimization in a spark ignition engine. *American Control Conference*, Baltimore, MD, USA, 1994.
- Yuan Y. A review of trust region algorithms for optimization. *Proceedings of the Fourth International Congress on Industrial & Applied Mathematics*, Edinburgh, 2000 Oxford University Press, USA.
- Byrd, R.H., Hribar, M.E. et Nocedal, J. An Interior Point Algorithm for Large-Scale Nonlinear Programming. *SIAM Journal on Optimization*, vol. 9, n° 4, pp. 877-900, 1999.