
Détection de micro-défauts process, situés à l'intérieur des limites de surveillance d'une carte de contrôle

Application à la fabrication de wafer en microélectronique

Ikram Doumani¹, Patrick Pujo², Nacer M'Sirdi³

1. IC STAR, Aix-Marseille Université

Technopole de Château Gombert, 60 rue Joliot Curie, F-13013 Marseille
ikram.doumani@univ-amu.fr

2. CRET-LOG, Aix-Marseille Université

Av. Esc. Normandie Niémen, 13397 Marseille cedex 20
patrick.pujo@univ-amu.fr

3. LSIS, Aix-Marseille Université

Av. Esc. Normandie Niémen, 13397 Marseille cedex 20
nacer.msirdi@univ-amu.fr

RÉSUMÉ. Après avoir décrit la complexité des processus de production dans l'industrie du semi-conducteur et les difficultés qui en résultent en termes de contrôle qualité, nous présentons une méthode de détection de variations et changements brusques dans l'évolution d'une grandeur présente dans un processus. Cette méthode est utilisée pour observer des évolutions de grandeurs intrinsèques à un produit en cours de production ou à un processus, évolutions qui ne pourraient être observées classiquement par des cartes de contrôle. Cette méthode automatique, inspirée du traitement de signal, permet de modéliser, d'estimer et de prédire l'état du processus ainsi qu'élaborer un modèle de détection de rupture basé sur des tests d'hypothèse. Ces derniers utilisent l'état prédit et l'état actuel du processus afin de conclure sur la présence d'une variation correspondant à la détection d'un évènement, susceptible de motiver une prise de décision ou d'empêcher un dysfonctionnement dans le processus.

ABSTRACT. A detection method of abrupt variations and changes in the evolution of a greatness in a process is presented. For that, at first the complexity of production processes in the semiconductor industry and the resulting difficulties in terms of quality control are described. The proposed detection method is used to observe changes in intrinsic variables of a product being produced or of a process. These changes could not be observed by classical control charts. This automatic method, inspired from signal processing, allows to model, estimate and predict the state of the process. It also allows to develop a break detection model

based on hypothesis tests. These latter use the predicted state and the current state of the process to conclude on the presence of a variation corresponding to the detection of an event. This detection may motivate a decision making or prevent a dysfunction in the process.

MOTS-CLÉS : détection, rupture, prédiction, ARMA, tests d'hypothèse.

KEYWORDS: detection, abrupt change, prediction, ARMA, hypothesis tests.

DOI:10.3166/JESA.49.161-180 © Lavoisier 2016

1. Introduction

En génie industriel, il est courant d'observer l'évolution de grandeurs en fonction du temps, et de relever dans ces observations des changements de comportements significatifs, représentant des événements susceptibles d'engendrer des prises de décision. Dans le domaine plus particulier de la maîtrise statistique des procédés, depuis la création des cartes de contrôle (Shewhart, 1931), de nombreuses théories ont été établies (Taylor, 1967 ; Aroian et Levene, 1950 ; Barnard, 1959) pour associer observation stochastique et validation qualitative de la production. Le principe général consiste à mesurer sur un échantillon représentatif d'une population une grandeur soumise à aléas, puis à calculer la position de cette grandeur, sous la forme d'une médiane ou d'une moyenne, et sa dispersion, sous la forme d'une étendue ou d'un écart type (Vachette, 1990 ; Pillet, 2000). Différents types de cartes de contrôle existent (Gibra, 1975 ; Vance, 1983 ; Chakraborti *et al.*, 2008), et de nouvelles approches voient le jour (Box et Narasimhan, 2010), mais le plus souvent, il s'agit d'observer dans le temps, *via* des échantillons successivement mesurés, l'évolution du couple (position, dispersion) représentatif de la grandeur considérée, relativement à des limites de contrôle ou de surveillance. Lorsque ces limites sont franchies, des alertes sont déclenchées et des actions de pilotage du processus sont mises en œuvre : ce type de techniques constitue toujours un des outils fondamentaux pour le pilotage par la qualité des processus (Pujo et Pillet, 2002 ; Trip et Does, 2010).

Toutefois, ces techniques ne sont pas adaptées à certains processus de production complexes, où il existe une problématique de détection de phénomènes furtifs se produisant à l'intérieur des limites de surveillance. Nous verrons que c'est le cas en production de semi-conducteurs : des produits peuvent ne pas passer le contrôle final, alors que tous les contrôles intermédiaires se sont avérés valides, qu'aucun dysfonctionnement n'a été observé et que toutes les cartes de contrôle sont correctement paramétrées. La détection de tels phénomènes au moment où ils se produisent pourraient facilement permettre de corriger le défaut, mais la réduction de l'intervalle entre les limites de surveillance n'est pas une bonne solution : cela engendrerait essentiellement un nombre considérable de fausses alarmes, fortement pénalisantes à traiter.

Après une présentation plus complète de l'ensemble de cette problématique, nous proposons une présentation des différents modèles de prédiction, en vue de leur application au pilotage d'atelier, puis nous expliquons le modèle choisi pour notre étude. Ensuite, nous présentons les méthodes statistiques utilisées dans la détection

de rupture, avec seuillage automatique. Enfin, et avant de conclure, nous présentons quelques applications et la mise en œuvre de ce travail.

2. Contexte et problématique

L'unité de fabrication "Front End" de ST Microelectronics à Rousset supporte la fabrication d'une quinzaine de technologies différentes, correspondant à environ 350 produits génériques actifs (c'est-à-dire pouvant être différenciés par un paramétrage de différentes données). En moyenne, 250 équipements de production sont visités par les produits, certains plusieurs fois pour des dépôts-gravures successifs, afin de réaliser 360 opérations sur un temps de cycle moyen d'environ 60 jours. Les gammes de fabrication des circuits, appelées *Route*, décrivent l'itinéraire que suit le produit de la première opération à la dernière, nécessaire à l'obtention d'un produit particulier. Les routes diffèrent selon leur nature, le nombre et la durée des opérations. Une *Operation* est la brique élémentaire d'une route reliée à un travail (i.e. diffusion, photolithographie, gravure, implantation, etc.). Chaque opération est composée par une séquence ordonnée d'*Events*, nommée *Script*. Un script donne la définition claire et détaillée des tâches : c'est l'équivalent du contrat de phase. Les Events (événements) appelés *Steps* sont toutes les actions réalisées, décrites *via* le CAM system (*Computer Aided Manufacturing*). Ce dernier gère l'état des lots en cours de fabrication et envoie à chaque équipement les *recipes* (recette qui correspond à une étape de traitement, dans un équipement donné : il s'agit des paramètres nécessaires à la réalisation de l'opération). Tous les éléments principaux de processus de fabrication sont pilotés, coordonnés et maîtrisés par *WorkStream*, un outil de pilotage de type *Manufacturing Execution System* (MES). Nous pouvons rajouter à cela que la production s'effectue par lot contenant plusieurs wafers, mais que sur un wafer peuvent être gravés plusieurs types de circuits. Par ailleurs, la localisation d'un circuit sur son wafer peut influencer la qualité du résultat final, certains procédés de diffusion pouvant générer des défauts physiques d'homogénéité dans la fabrication.

Conjointement aux activités de production, une large partie des activités correspondent à des activités de contrôle en vue de la surveillance du processus et du maintien de la qualité des produits. 20 000 contrôles sur équipement et 60 000 contrôles sur produit sont ainsi réalisés hebdomadairement ! Cela concerne des grandeurs telles que la résistivité d'un liquide, la pression d'une chambre, sa température... Le système de pilotage doit donc intégrer cette dimension : analyser tous ces résultats, détecter ceux qui sont représentatifs ou potentiellement porteurs d'un futur dysfonctionnement, émettre un diagnostic et concevoir une stratégie corrective, puis enfin déployer les actions précédemment décidées. Un grand nombre d'ingénieurs sont inclus dans la prise de décision de ce système de pilotage : au-delà de l'observation de caractéristiques événementielles de production par la détection de variations d'état ou de franchissement de seuils ou de limites de surveillance, le produit ou le lot concerné est bloqué en attendant la décision d'invalider l'incident observé ou de déclencher une procédure de retouche. Ceci nécessite différentes compétences scientifiques et techniques qui peuvent

difficilement être automatisées, car relevant d'un haut niveau d'expertise. Il faut impérativement, lors d'un éventuel incident, être capable de parcourir à nouveau l'intégralité de la route suivie par le produit, et d'en analyser tous les paramètres. Pour répondre à ce défi, relevant de la complexité quantitative, les unités de fabrication sont dotées de systèmes d'information performants, capables d'emmagasiner toute la dynamique du système, avec des moyens de sauvegarde considérables, de l'ordre de plusieurs centaines de téraoctets. Ainsi, toutes les évolutions sont sauvegardées, mais également tous les contrôles effectués, au moins le temps de fabrication du lot en question (60 derniers jours en moyenne). Pour accélérer et fiabiliser ce processus de prise de décision par expertise humaine, appelé ECN pour *Engineering Change Notification* et auparavant traité *via* un dossier papier transmis tour à tour à chaque expert, la mise en place d'une procédure utilisant des workflows (Zacharewicz *et al.*, 2009) a permis une amélioration substantielle de la réaction de pilotage et la réduction des temps de blocage.

Dans ce contexte déjà compliqué, il apparaît impossible de réduire l'intervalle entre les limites de surveillance, car cela augmenterait considérablement le nombre d'incidents à traiter *via* les ECN, sans pour autant une meilleure efficacité. Il faut savoir que les processus de fabrication en microélectronique sont souvent évolutifs, c'est-à-dire que leurs caractéristiques évoluent au fur et à mesure de leur utilisation. Par exemple, dans une enceinte de dépôt, à chaque utilisation, le dépôt se fait sur le wafer, mais aussi sur toute la surface interne de l'enceinte. Petit à petit, l'enceinte change de propriétés, son utilisation doit être ajustée en fonction, et à terme, une calibration doit être effectuée. Une carte de contrôle sur un tel équipement sert avant tout à maîtriser cette singularité. Réduire l'intervalle entre les limites de surveillance pour observer un phénomène rare, de faibles valeurs et sans trop d'impact sur le fonctionnement de l'enceinte, mais pouvant éventuellement engendrer des conséquences significatives sur le produit serait contreproductif, du fait d'un trop grand nombre de faux incidents déclenchant des ECN.

Aussi, nous proposons une technique visant à détecter automatiquement les changements de comportement dans la suite des valeurs chronologiques prises par une grandeur mesurée. Nous sommes dans un cas de figure où il est difficile d'obtenir des modèles particuliers pouvant être mis sous forme d'équations. Nous utiliserons comme modélisation une technique courante consistant à prendre en compte le comportement antérieur pour élaborer ensuite, par un algorithme *ad hoc*, une estimation du comportement futur, dite prédiction (Landau *et al.*, 1986). Cette prédiction va renseigner sur l'état du processus à n'importe quel instant ultérieur, et donc renforcer le contrôle du comportement prévisionnel du système. Notre objectif est d'utiliser cette prédiction pour détecter automatiquement les variations de la dynamique représentant un événement vécu par le système, par exemple susceptible d'engendrer un dysfonctionnement dans le processus. Il s'agit de multiplier les informations de haut niveau afin de prendre des décisions multicritères *ad hoc*, comme générer des alertes et lancer des actions correctives. Une telle approche de détection de ruptures comportementales observées a déjà été appliquée au domaine de l'automobile (M'Sirdi *et al.*, 1991). La détection d'une rupture comportementale s'effectue en comparant l'évolution estimée/prédite avec l'état actuel de la grandeur.

Cela permet de réaliser une étude statistique en utilisant des tests d'hypothèse qui pourraient être basés sur le rapport de vraisemblance et/ou un intervalle de confiance, tout en appliquant des méthodes statistiques, telles que la loi χ^2 ou la loi Fischer. Ceci présente l'avantage d'éviter le choix des seuils de détection. À partir des résultats de l'étude statistique, nous déduisons s'il existe une rupture entre la prédiction et l'état actuel, et donc le fait qu'à ce moment-là, il y a bien une apparition d'un événement correspondant à cette variation détectée. Ainsi nous réalisons un signal d'information basé sur les résidus (de comparaison état-prédiction). Ce signal, ou résidu, doit rester confiné dans un intervalle (de confiance) tant qu'il n'y a pas de rupture.

3. Méthodes de détection

3.1. Modèle de prédiction

La modélisation d'un processus consiste à trouver un modèle paramétré dont le comportement dynamique approche celui du processus. Ce modèle sera utilisé pour effectuer des prédictions de la sortie du processus, ou encore pour simuler le processus au sein d'un système de commande. La première étape de la modélisation consiste à observer le processus afin de recueillir toutes les informations descriptives de son comportement dynamique. Ces informations permettent de formuler plusieurs hypothèses de structures de modèles capables d'expliquer le comportement. Ces structures ou architectures de modèles diffèrent selon le nombre et la nature des variables d'entrée, d'état et de sortie ainsi que par les relations entre ces variables (figure 1). L'étape suivante appelée « Identification » repose sur l'estimation des paramètres du modèle choisi en mettant en place un prédicteur de la sortie du processus relatif au modèle choisi. À l'issue de l'identification, l'estimation des différents paramètres du modèle est calculée à partir de l'écart entre les valeurs des variables de sortie mesurées et les valeurs des variables de sortie prédites. Le niveau de précision de cette estimation dépend du choix de la structure du modèle, du critère d'estimation et des données utilisées.

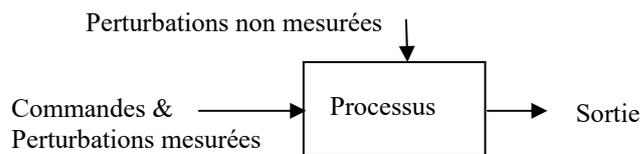


Figure 1. Modélisation d'un processus avec perturbations non mesurées (bruit)

En automatique, il existe une structure générale de modèle d'identification (figure 2). Ainsi, suivant le choix des polynômes A,B,C,D et F et/ou le choix de la modélisation du bruit présent dans le processus ' $e(t)$ ', nous obtenons plusieurs structures de modèle (Landau *et al.*, 1986).

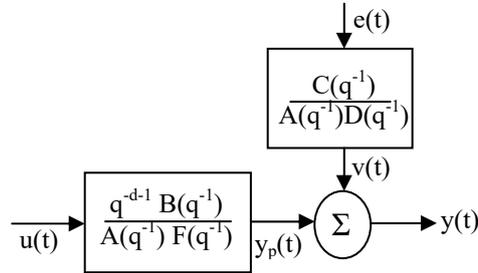


Figure 2. Structure générale d'un modèle d'identification et de prédiction

avec les polynômes :

$$\left. \begin{aligned}
 A(q^{-1}) &= 1 + a_1 q^{-1} + a_2 q^{-2} + \dots + a_{na} q^{-na} \\
 B(q^{-1}) &= b_0 + b_1 q^{-1} + b_2 q^{-2} + \dots + b_{nb} q^{-nb} \\
 C(q^{-1}) &= 1 + c_1 q^{-1} + c_2 q^{-2} + \dots + c_{nc} q^{-nc} \\
 D(q^{-1}) &= 1 + d_1 q^{-1} + d_2 q^{-2} + \dots + d_{nd} q^{-nd} \\
 F(q^{-1}) &= 1 + f_1 q^{-1} + f_2 q^{-2} + \dots + f_{nf} q^{-nf}
 \end{aligned} \right\} \quad (1)$$

où na , nb , nc , nd , et nf représentent les ordres des polynômes A , B , C , D et F . Le signal étudié est un signal à temps discret, noté x_k ou $x(k)$ qui est égal à la valeur du signal analogique $x(t)$ au temps $t = kT_e$ avec T_e la période d'échantillonnage et $-\infty < k < \infty$.

Les structures les plus utilisées (Basseville, 1988) sont :

– le modèle de la réponse impulsionnelle finie (RIF) : il fait appel au polynôme B et ne nécessite aucun modèle de bruit :

$$y_k = q^{-d} B(q^{-1}) u_k + e_k \quad (2)$$

avec : q^{-1} l'opérateur retard ($q^{-1}.x_k = x_{k-1}$), ' u_k ' l'entrée auxiliaire ou la perturbation mesurable, ' y_k ' la sortie du processus à l'instant k , d le retard du processus à la réponse ' u_k ' que nous considérons égale à zéro par la suite, et ' e_k ' le bruit blanc générateur non mesurable (Landau *et al.*, 1986).

– le modèle autorégressif (AR) : ce modèle représente le signal étudié uniquement en fonction de ses états précédents, excité par un bruit blanc :

$$A(q^{-1}) y_k = e_k \quad (3)$$

– le modèle autorégressif avec une entrée auxiliaire (ARX) : il s'agit de la méthode des moindres carrés ordinaires en utilisant les polynômes A et B , le bruit est supposé blanc centré :

$$A(q^{-1}) y_k = q^{-d} B(q^{-1}) u_k + e_k \quad (4)$$

– le modèle ARARX : il utilise la méthode des moindres carrés généralisés avec les polynômes A, B et D, le modèle du bruit étant $1/(A.D)$:

$$A(q^{-1}) y_k = q^{-d} B(q^{-1}) u_k + e_k / D(q^{-1}) \quad (5)$$

le modèle autorégressif à moyenne ajustée avec une entrée auxiliaire (ARMAX) : ce modèle utilise la méthode des moindres carrés étendus pour les polynômes A, B, C avec un modèle de bruit en C/A :

$$A(q^{-1}) y_k = q^{-d} B(q^{-1}) u_k + C(q^{-1}) e_k \quad (6)$$

le modèle autorégressif à moyenne ajustée (ARMA) : il s'agit d'un modèle pôle-zéro avec seule entrée le bruit e_k :

$$A(q^{-1}) y_k = C(q^{-1}) e_k \quad (7)$$

Nous avons choisi de travailler avec le modèle ARMA, qui est le plus adapté à notre problème. En réécrivant l'équation précédente sous forme de fraction rationnelle de polynômes, nous obtenons un rapport qui représente le bruit affectant le processus.

$$y_k = \frac{C(q^{-1})}{A(q^{-1})} e_k \quad (8)$$

En général, les paramètres a_i et c_i ne sont pas connus, ils seront donc estimés, d'où l'utilisation des prédicteurs adaptatifs.

Après la définition du modèle d'identification et de prédiction, il faut définir la prédiction du signal de sortie à un instant futur. Ceci revient à construire un prédicteur optimal (Landau *et al.*, 1986) donnant ' \hat{y}_k ' (le signal de sortie estimé) à partir de la connaissance de ' y_{k-i} ' avec ($i=1 \dots n$). La structure d'un prédicteur optimal associé à un modèle ARMA s'écrit sous la forme :

$$\hat{y}_k = -\sum_{i=1}^{na} a_i y_{k-i} + \sum_{i=1}^{nc} c_i e_{k-i} \quad (9)$$

Afin de calculer ' \hat{y}_k ', nous reprenons l'équation du signal du modèle ARMA (6) en utilisant la forme explicite des polynômes A et C, nous obtenons :

$$y_k = -\sum_{i=1}^{na} a_i y_{k-i} + \sum_{i=1}^{nc} c_i e_{k-i} + e_k \quad (10)$$

La différence ' $y_k - \hat{y}_k$ ' nous donne le résidu ou l'erreur de prédiction ' ϵ_k '. Le bruit ' e_k ' est inaccessible à la mesure. Nous savons qu'à l'instant k, toutes les données à k-1, k-2, ... sont connues, et en tenant compte du critère (M'Sirdi *et al.*, 1991), l'espérance mathématique ' $E[(y_k - \hat{y}_k)^2]$ ' est minimum, voire nulle dans l'idéal (sans bruit), nous obtenons à la fin, asymptotiquement :

$$E\{\epsilon_k^2\} = E\{e_k^2\} \quad (11)$$

Cela signifie que l'erreur de prédiction tend asymptotiquement et en moyenne quadratique vers le terme du bruit. Si à $k-1$, nous estimons par ' ϵ_{k-1} ' les valeurs antérieures de ' ϵ_{k-1} ', nous pouvons réaliser un prédicteur à l'instant k et donc avoir une valeur du bruit au même instant :

$$\epsilon_k = y_k - \hat{y}_{k/(k-1)} = e_k \quad (12)$$

avec ' $\hat{y}_{k/(k-1)}$ ', la prédiction du signal ' y_k ' à l'instant k en connaissant les valeurs de l'équation de ' y_k ' à l'instant $k-1$. Avec cette méthode, nous pouvons accéder alors, à chaque instant, à l'estimation ' ϵ_k ' de ' e_k ' de façon récursive. En écrivant l'équation du signal sous la forme matricielle :

$$y_k = \varphi_{k-1}^T \cdot \theta + e_k \quad (13)$$

avec ' φ_k^T ' le vecteur des observations :

$$\varphi_k^T = [-y_{k-1} -y_{k-2} \dots e_{k-1} e_{k-2} \dots] \quad (14)$$

et ' θ ' le vecteur des paramètres :

$$\theta = [a_1 a_2 \dots c_1 c_2 \dots]^T \quad (15)$$

D'après ce que nous avons expliqué précédemment, à l'instant k , nous déterminons à la fois une estimation ' $\hat{\theta}_k$ ' d'un jeu de paramètres ' θ ' et nous en déduisons une estimation ϵ_k de la valeur du bruit ' e_k '. Le vecteur des observations s'écrira donc :

$$\hat{y}_k = \varphi_k^T \cdot \hat{\theta}_{k-1} \quad (16)$$

avec ' φ_k^T ' le vecteur des observations :

$$\varphi_k^T = [-y_{k-1} -y_{k-2} \dots e_{k-1} e_{k-2} \dots] \quad (17)$$

avec ' $\hat{\theta}_{k-1}$ ' est le vecteur des paramètres estimés à l'instant $k-1$:

$$\hat{\theta}_{k-1} = [\hat{a}_1 \hat{a}_2 \dots \hat{c}_1 \hat{c}_2 \dots]^T \quad (18)$$

et

$$\epsilon_k = y_k - \hat{y}_{k/(k-1)} \quad (19)$$

En termes de validation, pour savoir pratiquement si l'algorithme a correctement fonctionné et si les résidus ' ϵ_k ' convergent vers ' e_k ', qui par hypothèse est un bruit blanc, nous pouvons effectuer *a posteriori* un test de blancheur sur le signal ' ϵ_k '.

3.2. Modèle de détection de rupture

La détection des ruptures de modèle a fait l'objet d'études dans divers domaines (Basseville et Nikiforov, 1993). La détection de changements brusques ou de ruptures de modèle nécessite de :

1. Choisir l'information à évaluer et définir le signal de test ou d'information de rupture. Cette information peut être la sortie du système, la variance de la sortie, le vecteur paramètres ou l'erreur des estimations de paramètres. En ce qui concerne notre étude, nous choisissons de travailler sur l'erreur de prédiction et sa variance.

2. Choisir le test de décision (un test d'hypothèse).

3. Observer une rupture.

4. Dater et estimer la rupture

Détecter une rupture de modèle fait appel à un test d'hypothèse qui peut consister à :

1. Établir l'hypothèse nulle $H_0 =$ il n'y a pas de rupture, et considérer l'hypothèse alternative $H_1 =$ il y a rupture.

2. Choisir le test statistique approprié pour tester H_0 .

3. Spécifier un niveau de signification et la taille de l'échantillon.

4. Définir la région de rejet.

5. Calculer la valeur du test statistique.

6. Conclure sur le résultat du test.

3.2.1. Test d'hypothèse et intervalle de confiance

Dans la partie précédente, nous avons estimé le signal de sortie ' y_k ' par une valeur ' \hat{y}_k '. Chaque réalisation de ' \hat{y}_k ' doit être donc proche de la vraie valeur de ' y_k '. De ce fait, nous devons vérifier la vraisemblance de ces deux variables. Pour cela, nous définissons un intervalle de confiance ayant une probabilité $((1-\alpha) = 95\%$ par exemple) de contenir la vraie valeur de ' y_k '. $\alpha \in [0,1]$ représente la densité de probabilité de ' \hat{y}_k '. La probabilité que ' \hat{y}_k ' soit en dehors de l'intervalle de confiance avec $\alpha = 0.05$, est :

$$\int_{-\infty}^{\alpha/2} p(x)dx + \int_{1-\alpha/2}^{+\infty} p(x)dx = \alpha \quad (20)$$

Il s'agit d'une probabilité d'erreur qui doit être assez petite. Les valeurs usuelles de α sont 10%, 5% et 1%. En effet, un intervalle de confiance défini pour un estimateur ' \hat{y} ' (voir figure 3) se ramène à la définition d'une région d'acceptation :

$$\text{Prob} [y_{1-\alpha/2} < \hat{y} < y_{\alpha/2}] = 1-\alpha \quad (21)$$

et d'une région de rejet de l'hypothèse :

$$\text{Prob} [\hat{y} > y_{\alpha/2}] = \alpha/2 \ \& \ \text{Prob} [\hat{y} \leq y_{1-\alpha/2}] = \alpha/2 \quad (22)$$

La loi de χ^2 (Basseville et Nikiforov, 1993) se base sur le calcul de la différence entre les valeurs théoriques du signal et ses valeurs observées, les résidus ' ϵ_k ', ainsi que sur l'estimation de l'importance de l'écart résultant selon le seuil α et le degré de liberté des variables en se ramenant sur une table du χ^2 .

$$\chi^2 = \frac{(y_{t-3} - \hat{y}_{t-3})^2}{\hat{y}_{t-3}} + \frac{(y_{t-2} - \hat{y}_{t-2})^2}{\hat{y}_{t-2}} + \frac{(y_{t-1} - \hat{y}_{t-1})^2}{\hat{y}_{t-1}} + \frac{(y_t - \hat{y}_t)^2}{\hat{y}_t} \quad (23)$$

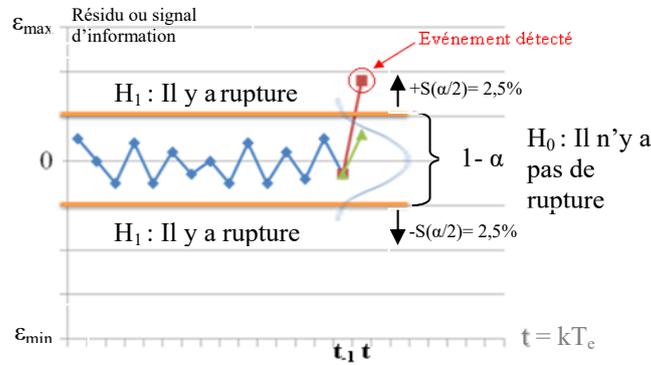


Figure 3. Région d'acceptation et région de rejet

Si la valeur de χ^2 est plus supérieure ou très petite par rapport à la valeur indiquée sur le tableau, l'hypothèse H_0 sera rejetée, ce qui signifie qu'il y a une rupture d'hypothèse, et donc une variation dans le signal. Selon (M'Sirdi *et al.*, 1991), 's²', l'estimation de la variance 'σ²', suit une loi du χ^2 à n-1 degrés de liberté avec :

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - m)^2 \tag{24}$$

$$(n-1) \frac{s^2_{n-1}}{\sigma^2} \sim \chi^2_{n-1} \tag{25}$$

ce qui nous donne l'intervalle de confiance pour la variance :

$$\frac{ns^2}{\chi^2_{\alpha/2}(n-1)} \leq \sigma^2 < \frac{ns^2}{\chi^2_{1-\alpha/2}(n-1)} \tag{26}$$

La loi de Fisher est le rapport entre deux variables indépendantes qui suivent la loi du χ^2 , divisé par le rapport des degrés de liberté. Pour l'estimation du rapport de variance, on considère que le rapport des variances suit la loi du χ^2 et la distribution de Fisher.

$$F_{n_1 n_2} = \frac{\frac{n_1}{n_1-1} \frac{s^2_{n_1}}{\sigma^2_{n_1}}}{\frac{n_2}{n_2-1} \frac{s^2_{n_2}}{\sigma^2_{n_2}}} = \frac{\chi^2_{n_1} / n_1 - 1}{\chi^2_{n_2} / n_2 - 1} \tag{27}$$

On rejettera H_0 si la valeur observée pour $F_{n_1 n_2}$ est trop grande ou trop petite par rapport à la valeur indiquée dans la table de Fisher, suivant le seuil et le degré de liberté choisis.

3.2.2. Rapport de vraisemblance

La vraisemblance a été utilisée dans le but de quantifier l'adéquation entre une distribution de probabilité et un échantillon de données : plus grande est la vraisemblance de l'échantillon, meilleure est l'adéquation. De ce fait, le test permet de discriminer entre deux groupes de distributions celle qui contient la probabilité

qu'il y ait rupture et la distribution qui contient la probabilité qu'il n'y ait pas de rupture, chacune affirmant contenir la distribution qui a engendré l'événement.

Nous considérons la famille des distributions normales $N(\mu, \sigma^2)$ (Ycart, 2002), μ étant la moyenne et σ^2 la variance des variables. Les deux groupes de distributions sont définis par l'hypothèse nulle $H_0 : \mu = \mu_0$ contre l'hypothèse alternative $H_1 : \mu \neq \mu_0$. De quelle distribution est-il le plus vraisemblable que l'échantillon soit issu ?

Si H_0 est vraie, la distribution ayant engendré l'échantillon appartient à H_0 et la vraisemblance de l'échantillon pour cette distribution serait élevée. Dans ce cas, aucune des distributions de H_1 n'a engendré l'échantillon. Finalement, la plus grande vraisemblance rencontrée dans H_0 ne serait pas beaucoup plus petite que la plus grande vraisemblance dans l'ensemble complet des distributions $H_0 \cup H_1$ (Ycart, 2002), nous déduisons alors :

$$\lambda = \max_{H_0} L(\mathbf{x}; \theta) / \max_{H_0 \cup H_1} L(\mathbf{x}; \theta) \tag{28}$$

avec $\max_{H_0} L(\mathbf{x}; \theta)$ qui représente la plus grande des vraisemblances rencontrées dans H_0 et $\max_{H_0 \cup H_1} L(\mathbf{x}; \theta)$ qui représente la plus grande des vraisemblances rencontrées dans $H_0 \cup H_1$.

De ce fait, λ est certainement plus petit que 1, et probablement proche de 1. Le rapport de vraisemblance permet de tester H_0 contre H_1 , des petites valeurs de λ , proches de 0, signifient probablement que H_1 est vraie, alors que de grandes valeurs de λ signifient probablement que H_0 est vraie.

4. Modélisation du mécanisme de détection

L'approche réalisée et proposée dans cet article est la combinaison des trois méthodes présentées (modèle de prédiction, tests statistiques et carte de contrôle) afin d'aboutir à un seul modèle capable de détecter tout changement d'état. Le synoptique général de notre approche est présenté figure 4.

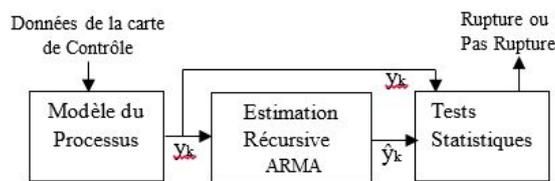


Figure 4. Schéma synoptique de la méthode étudiée

4.1. Modèle de processus

Le modèle de processus est en fait dans notre cas un modèle de type 'boite noire' dont nous connaissons seulement la suite de données correspondant à la grandeur à observer. Une carte de contrôle aux mesures fournit donc l'ensemble des données

nécessaires pour l'observation de la caractéristique correspondante. Le bloc 'Modèle du Processus' représente donc une carte de contrôle aux mesures qui a comme entrée les données du système.

4.2. Modèle de prédiction

Le bloc « estimation récursive ARMA » contient le modèle d'identification et de prédiction que nous avons élaboré afin d'avoir une estimation de l'état du processus à un instant futur. Le bloc ARMA peut être représenté suivant plusieurs méthodes sur le logiciel MATLAB. La méthode la plus basique consiste à utiliser les blocs *Discrete Filter* de la Librairie de Simulink, qui représentent les polynômes de l'équation (7), reliés à l'entrée ' u_k ' et ' e_k ' (figure 5). Ceci dit, ce schéma reste partiel, puisqu'il nous manque les valeurs des paramètres des polynômes A et C. Dans ce cas, le logiciel MATLAB propose un exemple d'algorithme afin d'estimer ces paramètres. Pour représenter la perturbation ' e_k ' nous utilisons le bloc *Band-limited White Noise* qui génère un bruit blanc. Le choix du bloc de l'entrée dépend en fait du type de signal que nous disposons.

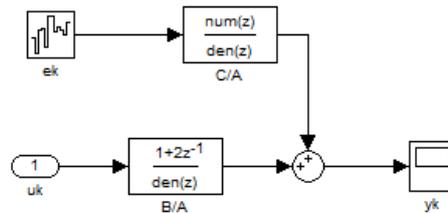


Figure 5. Schéma Simulink – Modèle basique

La seconde possibilité est offerte par la librairie *System Identification Toolbox* du logiciel MATLAB, qui nous permet d'utiliser directement le bloc *General Model Estimator Using Predictive Error Method*. Ce bloc général permet de faire appeler n'importe quelle structure de modèle de prédiction. En effet, ce type de bloc estime immédiatement les paramètres du modèle. L'avantage de l'utilisation de ce bloc est le fait qu'il génère à chaque estimation une figure montrant l'état mesuré actuel et l'état prédit ainsi que l'erreur de prédiction entre les deux états. Sur Simulink, nous pouvons lui associer le bloc *Iddata Source* comme entrée pour relié 'u' et 'y' (figure 6), dans notre cas, 'u' représente ' y_k ' et 'y' représente la sortie estimée ' \hat{y}_k '. Ce dernier bloc permet d'importer des données que nous appelons *IDDATA* et que nous pouvons créer sur l'espace de travail de MATLAB. L'utilisation de ce bloc ne nécessite que l'indication de l'ordre des polynômes, le nombre d'échantillons de données d'entrée qui vont spécifier l'intervalle pour estimer un nouveau modèle, le temps d'échantillonnage pour le modèle, et le nombre d'échantillons de données antérieures pour l'estimation de chaque modèle.

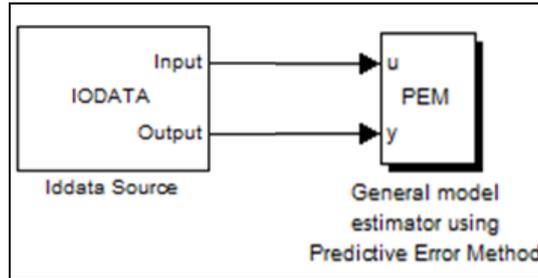


Figure 6. Schéma Simulink – Bloc ARMA

La dernière solution de réalisation du modèle ARMA est celle que nous avons adoptée pour cette étude, afin d'obtenir une prédiction de la sortie ' y_k '. Elle consiste à exécuter dans l'espace de travail de MATLAB un algorithme mettant en œuvre les différentes équations mathématiques traitées. En développant les équations mathématiques précédentes, et en tenant compte de nouvelles hypothèses pour l'élaboration du modèle d'estimation paramétrique de notre système, nous avons introduit de nouveaux termes dans le raisonnement mathématique, ce qui nous permet d'avoir un programme performant dans la prédiction.

La méthode de l'erreur de prédiction, utilisant la méthode des moindres carrés, sur laquelle se base notre étude, prenant en compte la connaissance des mesures antérieures du système permet de minimiser le critère quadratique qui se porte sur l'erreur de prédiction, qui à son tour donne lieu à une meilleure estimation récursive des paramètres du modèle à réaliser. Nous considérons que notre système est dynamique, linéaire, stochastique, mono-variable et à paramètres inconnus constants. Considérant (17), l'erreur de prédiction ε_k peut s'écrire aussi sous la forme :

$$\varepsilon_k = y_k - \varphi_k^T \cdot \hat{\theta}_{k-1} \quad (29)$$

$\hat{\theta}_k$ représente le vecteur de paramètres estimés à l'instant k .

À partir des équations (13) et (16), nous pouvons encore réécrire l'expression de ε_k sous la forme :

$$\varepsilon_k = \varphi_k^T [\theta - \hat{\theta}_{k-1}] + e_k \quad (30)$$

De cette écriture, nous pouvons conclure que si $\hat{\theta}$ représente une meilleure estimation du vecteur des paramètres θ , l'erreur de prédiction sera la meilleure estimation du bruit. Aussi, les valeurs de ces paramètres à chaque nouvelle prédiction seront calculées à partir des précédentes valeurs estimées auxquelles nous ajoutons une correction C_r :

$$\hat{\theta}_k = \hat{\theta}_{k-1} + C_r \quad (31)$$

$$P^{-1}_{k+1} = \sum_{i=N+1}^k \varphi_i \varphi_i^T + \varphi_{k+1} \varphi_{k+1}^T$$

De ce fait, nous nous intéresserons à la réalisation d'un prédicteur optimal $\hat{\theta}$ du vecteur des paramètres θ , en minimisant le critère quadratique (Ycart, 2002) à partir d'un instant $N+1$ jusqu'à un instant k , qui est défini par :

$$J_k = \sum_{i=N+1}^k [y_i - \varphi_i^T \hat{\theta}_k]^2 \quad (32)$$

Le minimum de ce critère est donné par la résolution de :

$$\frac{\partial J_k}{\partial \hat{\theta}_k} = -2 \sum_{i=N+1}^k [y_i - \hat{\theta}_k^T \varphi_i] \varphi_i = 0 \quad (33)$$

Ceci donne, avec l'hypothèse de l'existence de la somme des matrices $\varphi_i \varphi_i^T$:

$$\hat{\theta}_k \left[\sum_{i=N+1}^k \varphi_i \varphi_i^T \right] = \sum_{i=N+1}^k y_i \varphi_i \quad (34)$$

et donc :

$$\hat{\theta}_k = \left[\sum_{i=N+1}^k \varphi_i \varphi_i^T \right]^{-1} \sum_{i=N+1}^k y_i \varphi_i \quad (35)$$

De cette équation, nous pouvons écrire :

$$\hat{\theta}_{k+1} = P_{k+1} \sum_{i=N+1}^k y_i \varphi_i \quad (36)$$

$P_{k+1} = \left[\sum_{i=N+1}^k \varphi_i \varphi_i^T \right]^{-1}$ est la matrice de gain d'adaptation, symétrique définie positive.

$$\text{et :} \quad \sum_{i=N+1}^k y_i \varphi_i = \sum_{i=N+1}^k y_i \varphi_i + y_{k+1} \varphi_{k+1} \quad (37)$$

ce qui résulte, en tenant compte l'équation (33) :

$$\sum_{i=N+1}^k y_i \varphi_i = \left[\sum_{i=N+1}^k \varphi_i \varphi_i^T \right] \hat{\theta}_k + y_{k+1} \varphi_{k+1} \quad (38)$$

A partir de l'équation (29) et celle de la matrice de gain d'adaptation :

$$\sum_{i=N+1}^k y_i \varphi_i = P^{-1}_{k+1} \hat{\theta}_k + \varphi_{k+1} \varepsilon_{k+1} \quad (39)$$

Le prédicteur optimal $\hat{\theta}_{k+1}$ que nous obtenons en fin de compte, d'après (36) et (39), s'écrit alors :

$$\hat{\theta}_{k+1} = \hat{\theta}_k + P_{k+1} \varphi_{k+1} \varepsilon_{k+1} \quad (40)$$

Cependant, il nous reste à calculer la matrice de gain d'adaptation d'une manière récursive pour avoir une meilleure estimation :

$$P^{-1}_{k+1} = \sum_{i=N+1}^k \varphi_i \varphi_i^T + \varphi_{k+1} \varphi_{k+1}^T \quad (41)$$

ou

$$P^{-1}_{k+1} = [P^{-1}_k + \varphi_{k+1} \varphi_{k+1}^T] \quad (42)$$

Nous appliquons la propriété d'inversion matricielle sur cette dernière équation, ce qui permet d'écrire la matrice de gain d'adaptation sous une forme récursive :

$$P_k = P_{k-1} - \frac{P_{k-1} \varphi_k \varphi_k^T P_{k-1}}{1 + \varphi_k^T P_{k-1} \varphi_k} \quad (43)$$

Nous obtenons ainsi l'algorithme de l'estimation récursive de notre étude :

$$\left. \begin{aligned} \hat{\theta}_k &= \hat{\theta}_{k-1} + P_k \varphi_k \varepsilon_k \\ P_k &= P_{k-1} - \frac{P_{k-1} \varphi_k \varphi_k^T P_{k-1}}{1 + \varphi_k^T P_{k-1} \varphi_k} \\ \varepsilon_k &= y_k - \varphi_k^T \cdot \hat{\theta}_k \\ \hat{y}_k &= -\sum_{i=1}^n \hat{a}_i y_{k-i} + \sum_{i=1}^n \hat{c}_i \varepsilon_{k-i} \end{aligned} \right\} \quad (44)$$

Le terme de correction des valeurs du vecteur de paramètres s'écrit :

$$Cr_k = P_k \varphi_k \varepsilon_k \quad (45)$$

Dans cette étude, nous avons effectué le calcul de la sortie estimée a priori \hat{y}_k en tenant compte des paramètres prédits à l'instant k-1. Nous pouvons aussi recalculer la sortie estimée *a posteriori* \hat{y}_{Ok} en tenant compte cette fois-ci des valeurs des paramètres prédits à l'instant k.

4.3. Tests statistiques : modèle de rupture

Le bloc final « Tests statistiques » comporte le ou les test(s) d'hypothèse que nous avons présentés. Il détecte les variations entre l'état mesuré et l'état prédit, et donc l'importance de l'erreur de prédiction. En utilisant la Librairie *Statistics Toolbox* sur le logiciel MATLAB, nous obtenons tous les algorithmes de tests d'hypothèse qui nous intéressent. Ainsi, nous pouvons réaliser des *MATLAB Fonction*, des fonctions à base de ces algorithmes éditées sur l'éditeur de MATLAB, afin de les importer et de les intégrer dans le schéma Simulink. Sinon, nous pouvons récupérer les résultats de la prédiction à la sortie du modèle (figure 7) ou en exploitant les figures du modèle (figure 8), afin de les intégrer directement dans les algorithmes des tests d'hypothèse du logiciel MATLAB.

5. Simulations et résultats

Nous avons présenté et expliqué une méthode qui consiste à récupérer un signal, le signal de sortie du système de production, issu d'une carte de contrôle aux mesures, afin de l'introduire dans un algorithme qui calcule la prédiction du signal à un instant futur. Après cette étape, nous soumettons le signal prédit à des tests d'hypothèse afin de vérifier sa vraisemblance avec le signal de sortie mesuré et ainsi de le maintenir dans une zone de valeurs acceptables et proches du signal initial. Ceci nous amène à détecter une rupture lorsque celui-là se retrouve dans la région de rejet.

Cependant, nous proposons de réaliser le modèle présenté auparavant, en ajoutant une nouvelle phase avant la prédiction. Cette étape consiste à traiter l'information que représente le signal de sortie fourni par la carte de contrôle. En effet, nous calculons la moyenne estimée du signal et nous la retranchons au signal initial pour récupérer le signal y_k . Ensuite, nous créons un nouveau signal nommé z_k qui est en fonction du signal y_k et de son écart-type. Par la suite, nous réalisons deux simulations : une avec le signal d'entrée y_k et l'autre avec le signal d'entrée z_k , afin de comparer les résultats des deux modèles.

$$z_k = y_k / \sqrt{\sigma^2} \quad (46)$$

Pour la validation de notre approche, notre partenaire industriel nous a fournis différents jeux de données à expérimenter, issus de cartes de contrôle réelles relatives à des cas où il eut été judicieux de détecter un micro-défaut.

5.1. Premier jeu de données

Nous allons détailler ici le traitement du premier jeu de données et l'application de la méthode proposée. Le signal issu de la carte de contrôle est enregistré dans fichier Excel et lu dans le programme de MATLAB comme un vecteur colonne de 1000 valeurs (figure 7).

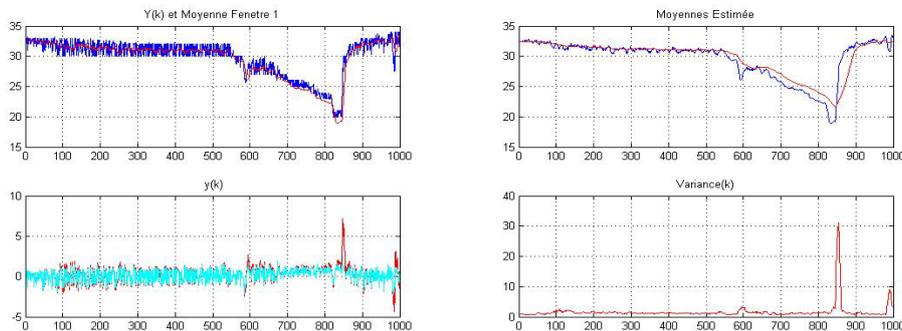


Figure 7. Simulation du signal Y_k , sa moyenne estimée, y_k et sa variance

Tout d'abord, nous calculons la moyenne estimée du signal d'entrée, afin de créer le signal y_k , ensuite et une fois la variance calculée (figure 7), nous obtenons z_k en fonction du signal y_k et de l'écart type.

La simulation du programme de prédiction est effectuée, d'abord avec le signal y_k et ensuite avec le signal z_k . Nous obtenons les résultats (figures 8 et 9).

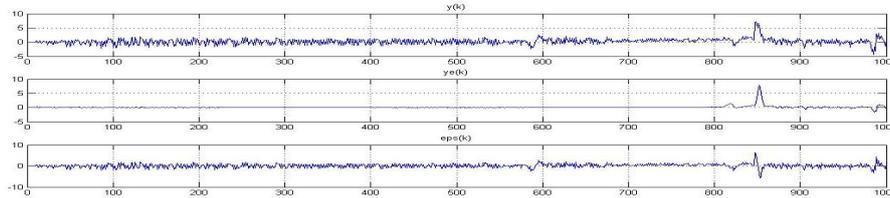


Figure 8. Simulation du signal y_k , sa prédiction $y_{e,k}$, et l'erreur de prédiction

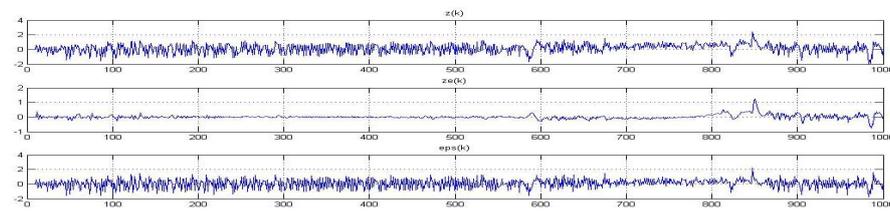


Figure 9. Simulation du signal z_k , sa prédiction $z_{e,k}$, et l'erreur de prédiction

Nous remarquons qu'une simulation avec le signal z_k est nettement meilleure que le résultat avec le signal y_k , puisque l'erreur de prédiction de ce dernier présente une variation peu importante par rapport à celle de z_k . Cette faible variance risque fort de ne pas être détecté lors des tests d'hypothèse. Par contre les fortes variations de l'erreur de prédiction du signal z_k nous permet de mieux traiter ainsi que d'appliquer les tests statistiques pour vérifier la détection de présence de ces variations.

La partie de la détection de rupture du modèle ou encore les variations présentes sur le signal de l'erreur de prédiction du premier jeu de données est réalisée à partir de la moyenne et la moyenne estimée (figures 10 et 11).

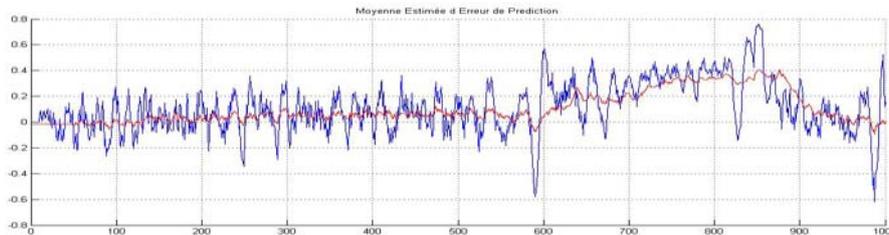


Figure 10. Simulation de la moyenne estimée de l'erreur de prédiction

À ce stade du modèle, nous calculons l'erreur standard, afin de définir l'intervalle de confiance de la moyenne de l'erreur de prédiction. Nous effectuons à la fin les tests statistiques. Les résultats des tests d'hypothèse sont directement affichés dans l'espace de travail du logiciel MATLAB (figure 11).

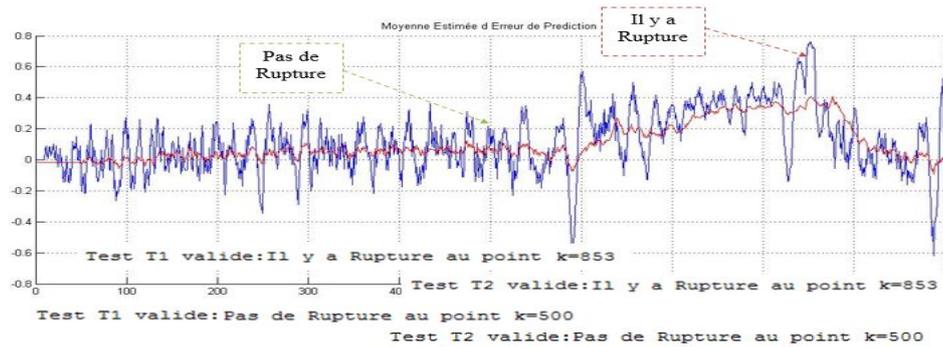


Figure 11. Simulation de la détection du changement d'état du signal

5.2. Autres jeux de données

Parmi les autres jeux de données, nous présenterons ici les résultats obtenus avec un phénomène de micro-défaut extrêmement fugitif (figure 12) et ceux relatifs à un phénomène extrêmement bruité noyant le micro-défaut (figure 13).

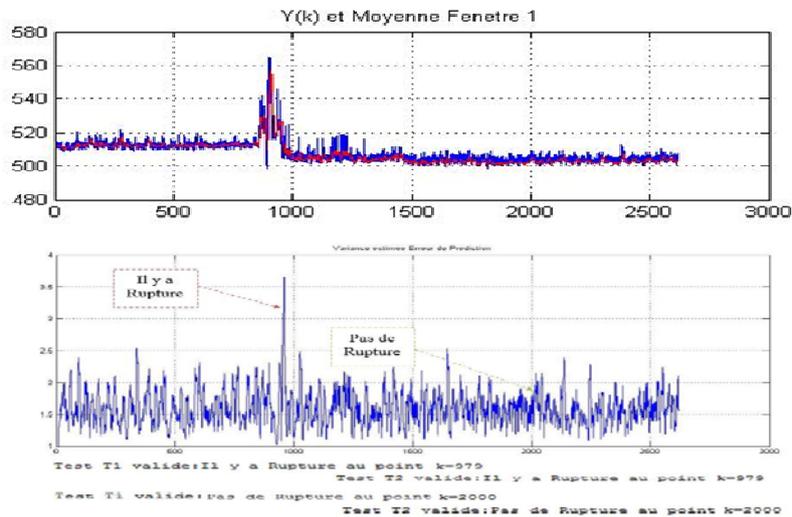


Figure 12. Résultats avec un micro-défaut fugitif

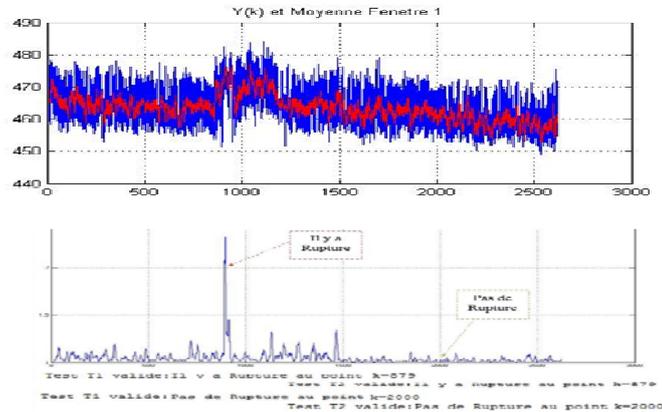


Figure 13. Résultats avec un micro-défaut bruité

Nous constatons que la méthode est tout autant efficace avec un phénomène fugitif qu'avec un phénomène bruité, ce qui lui confère une grande robustesse.

6. Conclusion

La méthode présentée ici associe un modèle de prédiction (ARMA) à des tests d'hypothèses statistiques pour automatiser la surveillance de cartes de contrôle aux mesures. Cette approche, aboutissant à un seul modèle capable de détecter tout changement d'état dans une série de données, permet de prévenir lors de l'apparition de dysfonctionnements, même lorsque ceux-ci n'entraînent pas le dépassement des limites de surveillance : ceci permet la détection de micro-défauts qui pourraient engendrer la perte de lots de produits. Un tel processus de surveillance automatique de chaque carte de contrôle peut alors déclencher, lors de la détection d'un micro-défaut, le lancement d'un workflow type ECN qui alertera immédiatement l'ingénieur-expert en charge du pilotage de la caractéristique considérée. Ainsi, il devient possible d'intervenir si nécessaire par des actions correctives avant que le lot ne soit perdu. Cette méthode a été implémentée sous MATLAB et testée de manière académique *via* différents jeux de données réelles. Les résultats sont encourageants, dans la mesure où la détection des micro-défauts s'effectue correctement dans différentes conditions (micro-défaut fugitif, micro-défaut bruité...), ce qui dénote d'une bonne robustesse. Il reste donc à définir quelles sont les cartes de contrôle qui doivent être ainsi traitées.

Remerciements

Les auteurs remercient la société ST Microelectronics, qui a fourni les données nécessaires à ces travaux.

Bibliographie

- Aroian L.A., Levene H., (1950). The effectiveness of quality control procedures. *Jal American Statistical Association*, vol. 45, p. 520-529.
- Barnard G.A. (1959). Control charts and stochastic processes. *Jal Royal Statistical Society*, vol. B.21, p. 239-271.
- Basseville M., (1988). Detecting Changes in Signals and Systems-A Survey. *Automatica*, vol. 24, n° 3, p. 309-326.
- Basseville M., Nikiforov I.V. (1993). *Detection of Abrupt Changes: Theory and Application*. Prentice Hall
- Box G., Narasimhan S., (2010). Rethinking Statistics for Quality Control, *Quality Engineering*, vol. 22, n° 2, p. 60-72.
- Chakraborti S., Human S. W., Graham M. A. (2008). Phase I Statistical Process Control Charts: An Overview and Some Results, *Quality Engineering*, vol. 21, n° 1, p. 52-62.
- Gibra I.N. (1975). Recent developments in control charts techniques. *Jal Quality Technology*, vol. 7, p. 183-192.
- Landau I.D., M'Sirdi N.K., M'Saad M. (1986). Techniques de modélisation récursives pour l'analyse spectrale paramétrique adaptative. *Traitement du Signal*, vol. 3, n° 4-5.
- M'Sirdi N.K., Ramdani R., Delanne Y. (1991). Détection de ruptures et application à la caractérisation de signaux de chaussée. *13^e Colloque GRETSI*, Juan-Les-Pins, 16-20 Septembre.
- Pillet M. (2000). *Appliquer la Maîtrise Statistique des Procédés*. Les éditions d'organisation.
- Pujo P., Pillet M. (2002). Control by quality: proposition of a typology. *International Journal of Quality Assurance: Good Practice, Regulation, and Law*, vol. 9, n° 2, p. 99-126.
- Shewhart W.A. (1931) *Economic Control of Quality of Manufactured Products*. New York: Van Nostrand
- Taylor H.M. (1967). Statistical control of a gaussian process. *Technometrics*, vol. 9, n° 1, p. 29-41.
- Trip A., Does R.J.M.M. (2010). Quality Quandaries: Interpretation of Signals from Runs Rules in Shewhart Control Charts, *Quality Engineering*, vol. 22, n° 4, p. 351-357.
- Vachette J.L. (1990). *Amélioration continue de la qualité SPC*. Les éditions d'organisations.
- Vance L.C. (1983). A bibliography of statistical quality control charts techniques, 1970-1980. *Jal Quality Technology*, vol. 15, p. 59-62.
- Ycart B. (2002). Tests statistiques, *Cahier de Mathématiques Appliquées*, n° 6, p. 51-107.
- Zacharewicz G., Pujo P., Frydman C., Giambiasi N. (2009). Environnement G-DEVS/HLA pour la simulation distribuée de systèmes de production multiprocessus. *Journal of Decision Systems*, vol. 18, n° 3, p. 375-402.

Article soumis le 13/05/2015

Accepté le 14/10/2015