 FR8100277

2. Colloque international sur la fiabilité et la maintenabilité.
Perros-Guirec, France, 8 - 12 Septembre 1980.
CEA - CONF 5556

- OPTIMISATION DE LA POLITIQUE DE TEST DES SYSTEMES EN ATTENTE DE FONCTIONNEMENT

- OPTIMISATION OF THE TEST POLICY OF STAND-BY SYSTEMS

par J.P. SIGNORET C.E.A.
O. MURON
G. COHEN

Il existe au sein des installations nucléaires de nombreux systèmes en attente de fonctionnement comme, par exemple, les systèmes de sauvegarde. Normalement à l'arrêt, ces systèmes doivent être prêts à démarrer lorsqu'une situation nécessitant leur intervention se produit. Pour connaître l'état dans lequel il se trouve et réparer ses défaillances éventuelles, un système en attente est généralement soumis à une certaine politique de test.

Les tests ont habituellement lieu à intervalle régulier selon une "grille" prévue à l'avance et nous avons montré au cours du IIIème Congrès National de fiabilité qui s'est tenu à Perros-Guirec en 1976 comment il était possible de trouver l'intervalle entre test optimum permettant d'obtenir la meilleure disponibilité moyenne. On peut en fait faire mieux si on considère qu'un intervalle entre test fixé a priori ne permet pas de tenir compte des informations que l'on a à chaque instant sur le système. C'est ce que nous nous proposons de montrer au cours de cette conférence.

Pour un système simple l'information connue à chaque instant, lorsqu'il est en attente, est la suivante :

- durée z qui s'est écoulée depuis la dernière fois où on a constaté qu'il était en bon état de marche (dernier test ou fin de la dernière réparation).

- durée restant avant l'arrêt programmé (et où le problème devient caduque).

Il est possible, en fonction de cette information, de prendre à chaque instant la meilleure décision : tester ou ne pas tester. Ce problème entre dans le cadre de la détermination d'une stratégie optimale et fait appel, pour être résolu, aux techniques de programmation dynamique.

Après avoir exposé la méthode dans le cas d'un système simple, nous l'appliquerons à un cas particulier afin de voir quel gain peut être escompté par rapport à la grille fixe traditionnelle.

Nous aborderons aussi succinctement le cas d'un système formé de 2 sous-systèmes.

In a nuclear power plant there are several systems which are on stand-by position as, for instance, safety systems. These stand-by systems have to be ready to start as soon as any situation for which they have be designed occurs. A test policy is then necessary in order to know the state of such a system and repair the possible failures.

The tests are normally performed at regular intervals of time according to a scheduled sequence. We have already shown during the "IIIème Congrès National de Fiabilité" held in Perros-Guirec in 1976 how it is possible in this case to find the optimum test interval leading to the best mean availability of the system. It is, in fact, possible to improve that by considering that if the test interval is fixed "a priori" this does not allow to take into account information we have on the system state every time.

For a simple system, being on stand-by condition this information is the next one :

- the duration z elapsed from the last time the system has been known in good state (last test or end of last repair).

- the duration left until the scheduled shut-down will be achieved (after which the problem disappears).

It is possible, by using this information, to take the best decision -performing a test or not- every time. This problem is a problem of optimal control and needs to be solved dynamic programmation technics.

The method is first described in the case of a simple system. It is then applied to a particular case in order to bring to light which gain can be obtained in comparaison with a traditionnal test policy.

After that the case of system composed of 2 sub-systems is briefly studied.

INTRODUCTION

Il existe dans l'industrie et en particulier dans l'industrie nucléaire des systèmes à l'arrêt en temps normal mais qui doivent être prêts à démarrer lorsque certaines situations se produisent au niveau de l'installation à laquelle ils appartiennent. Ces systèmes sont généralement des systèmes de sauvegarde et de nombreux travaux [1, 2, 3, 4, 5] leur ont été consacrés en vue d'optimiser la politique de test à laquelle ils doivent être soumis. Cependant il a toujours été considéré que les instants de tests étaient prévus à l'avance et qu'ils ne pouvaient en aucune manière être modifiés au cours du temps. Il est en fait possible d'envisager un autre type de politique de test où, à l'inverse, aucun instant de test n'est fixé à l'avance mais où l'opérateur peut, à chaque instant, décider de tester ou de ne pas tester et ceci en fonction des informations concernant le système qu'il a à sa disposition. Les 2 points de vue sont totalement différents et les modèles à mettre en oeuvre le sont aussi bien entendu. Les spécialistes disent dans le cas où les tests sont fixés à l'avance que c'est un problème en "boucle ouverte" et dans le cas où ils peuvent être décidés à chaque instant que c'est un problème en "boucle fermée". Ceci traduit bien le fait que dans le premier cas l'information qui s'acquiert au cours du temps n'est pas utilisée alors que dans le second cas elle est utilisée constamment pour pouvoir choisir la meilleure décision à prendre (tester ou ne pas tester).

Nous avons déjà exposé la méthode de traitement en boucle ouverte au cours du IIIème Congrès National de Fiabilité qui s'est tenu à Perros-Guirec en septembre 1976. Notre but est d'exposer ici la méthode de traitement du problème en boucle fermée. Ceci est un problème de stratégie optimale qui fait appel, pour être résolu, aux techniques de programmation dynamique. Nous étudierons tout d'abord le cas d'un système formé d'un seul module puis ensuite, beaucoup plus succinctement celui d'un système formé de 2 modules identiques.

I - SYSTEME COMPOSE D'UN SEUL MODULE

I.1 - Paramètres probabilistes

Les paramètres retenus sont les paramètres classiques :

- λ : taux de défaillance en attente
- γ : probabilité de défaillance à la demande
- μ : taux de réparation.

Nous remarquerons ici la nécessité d'avoir à la fois un paramètre λ qui modélise des défaillances dépendant du temps passé en attente et un paramètre γ qui modélise des défaillances causées par le test lui-même. En effet, si λ était nul il n'y aurait aucun intérêt à faire des tests et si γ était nul on aurait intérêt à tester tout le temps ; dans ces 2 cas il n'y aurait pas de problème d'optimisation de la politique de test.

D'autre part, le modèle développé ici permet de prendre en compte les lois de réparation quelconques G(.) donc, le paramètre μ peut dépendre du temps.

I.2 - Modélisation du système

Au lieu de considérer un écoulement continu du temps, nous considérerons ici que celui-ci est découpé en une succession d'intervalles égaux de longueur h. Ce pas de calcul doit, bien entendu, être choisi judicieusement pour être suffisamment petit par rapport à la durée des phénomènes mis en jeu.

A l'instant n ($t = n.h$), le système peut être dans 3 états distincts :

- en attente et en bon état de marche (M)
- en attente et en défaillance cachée (P)
- en réparation (R)

Ceci se représente facilement à l'aide d'un processus stokastique X_n prenant les valeurs M, P ou R à l'instant n selon l'état dans lequel se trouve le système. On a :

$$P\ (X_{n+1} = P / X_n = M) = 1 - e^{-\lambda h} = p$$

$$P\ (X_{n+1} = M / X_n = M) = e^{-\lambda h} = 1 - p$$

Si on introduit la variable aléatoire T_R représentant la durée d'une réparation, on peut poser :

$$P\ (T_R = i.h) = q_i \qquad q_i \geq 0 \quad ; \quad \sum_i q_i = 1$$

Ce qui permet de définir le taux de défaillance μ_i de la manière suivante :

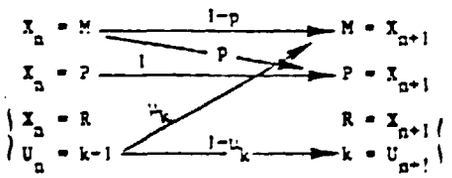
$$\mu_i = P\ (T_R = i.h / T_R > (i-1)h) = \frac{q_i}{1 - \sum_{j=0}^{i-1} q_j}$$

On constate que μ_i dépend de i c'est-à-dire du temps et ceci correspond au cas d'une loi de réparation G(.) quelconque.

Si à l'instant n le système est en réparation, il l'est depuis un certain temps U_n. On peut alors écrire :

$$P\ (X_{n+1} = M / X_n = R,\ U_n = k - 1) = \mu_k$$

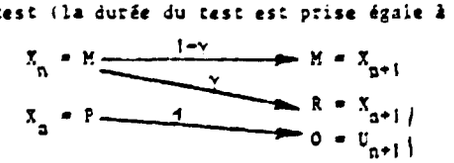
Finalement les différentes transitions entre états du système entre les instants n et n+1 (lorsqu'il n'y a pas de tests) se représentent de la manière suivante :



Plaçons nous maintenant à l'instant où un test est effectué à l'instant n :

- si $X_n = R$ Le test ne peut avoir lieu, la réparation se continue
- si $X_n = P$ L'effet du test est de mettre cette défaillance en évidence et d'initier une réparation : $X_{n+1} = R$, $U_{n+1} = 0$
- si $X_n = M$ 2 cas se présentent :
  - Le test met le système en panne et on est ramené au cas ci-dessus : $X_{n+1} = R$, $U_{n+1} = 0$
  - Le test montre que le système est en bon état, il en résulte $X_{n+1} = M$

D'où le diagramme suivant des transitions au cours d'un test (la durée du test est prise égale à h) :



Quand le système est en attente de fonctionnement, à l'instant n, on ne sait pas dans quel état X_n il se trouve. Par contre on connaît ce qui s'est passé avant :

- date des tests précédents (s'il y en a eu),
- résultats de ces tests.

Si le système est en réparation on le sait forcément et on connaît donc de ce fait les instants où les réparations se terminent.

On connaît aussi la durée restant à courir avant d'arriver à l'instant T qui caractérise la fin du problème (arrêt complet de l'installation par exemple).

Si on note par Θ_n l'ensemble de cette information on peut ainsi définir la probabilité de bon fonctionnement du système à l'instant n conditionnellement à Θ_n :

$$Y_n = P\,(X_n = M/\Theta_n)$$

Cette probabilité peut s'évaluer facilement :

1er cas : aucun test n'a eu lieu depuis l'instant 0 ; le système est alors dans l'état M s'il n'y a eu aucune défaillance depuis l'instant initial

$$Y_n = (1-p)^n$$

2ème cas : le dernier test a eu lieu à $n = k$ et le système a été trouvé en parfait état de marche. Pour être encore en bon état de marche à l'instant n le système a dû y rester pendant $k - 1$:

$$Y_n = (1-p)^{k-1}$$

3ème cas : le système est revenu de réparation à n - k et aucun test n'a eu lieu depuis :

$$Y_n = (1-p)^k$$

Si on définit par Z_n la durée qui s'est écoulée depuis la dernière fois où on a su que le système était en bon état de marche ceci se résume par :

$$Y_n = (1-p)^{Z_n}$$

Il existe un 4ème cas : c'est celui où le système est en réparation à l'instant n. On a, dans ce cas $Y_n = 0$ et ceci nous conduit à poser $Z_n = +\infty$ pour symboliser ce cas.

Si à l'instant n le vrai état X_n du système n'est pas connu avec certitude, la valeur de Y_n est par contre très bien définie. Y_n est en quelque sorte un résumé de toute la connaissance concernant le système ; c'est donc sur Y_n que l'on se basera pour décider de tester ou de ne pas tester le système et ceci à chaque instant.

I.3 - Optimisation de la disponibilité

Pour que le système soit "disponible" à l'instant n il faut qu'il soit dans l'état M c'est-à-dire que X_n soit égal à M. Donc la disponibilité du système sur l'intervalle n° n et de longueur h, au sens probabiliste s'écrit simplement : $P\,(X_n = M)$. Si on considère que la durée T se compose de N+1 intervalles de ce type on obtient facilement la disponibilité moyenne $\bar{A}$ du système sous la forme suivante :

$$\bar{A} = \frac{\sum_{n=0}^{N} P(X_n = M)}{N+1}$$

Il nous faut maintenant relier cette expression aux processus Y_n et Z_n qui sont, seuls, connus à l'instant n. Pour ce faire un théorème sur les espérances conditionnelles nous sera fort utile :

$$E\,[P(X_n = M/\Theta_n)] = P\,(X_n = M)$$

soit $E\,[Y_n] = P\,(X_n = M)$

soit $E\,[(1-p)^{Z_n}] = P\,(X_n = M)$

d'où : $$\bar{A} = \frac{E\{\sum_{n=0}^{N} (1-p)^{Z_n}\}}{N+1}$$

A l'instant n 2 décisions sont possibles :

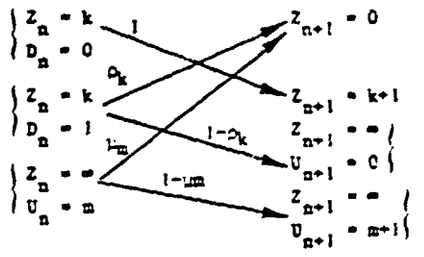
- $D_n = 0$ l'opérateur ne teste pas le système
- $D_n = 1$ l'opérateur teste le système

Si $D_n = 0$ la valeur de Z_n s'incrémente simplement d'une unité pour passer à Z_{n+1}, par contre, si $D_n = 1$ deux cas peuvent se présenter :

- Le système est trouvé en bon état de marche et Z_{n+1} retombe à zéro,
- Le système est trouvé en panne et le système est mis en réparation (symbolisé par $Z_{n+1} = +\infty$).

Bien entendu, si à l'instant n, le système est déjà en réparation, il n'y a pas de décision à prendre. La probabilité de transition de Z_n à Z_{n+1} ne dépend alors que de U_n.

On peut donc écrire toutes les probabilités de transition de Z_n à Z_{n+1} :



On note : $(1-\gamma)(1-p)^k = \rho_k$

Considérons maintenant le terme $E\sum_{n=0}^{n} (1-p)^{Z_n}$, il est égal à $\bar{A}$ à un facteur de proportionnalité près et il dépend, bien entendu, de la procédure de test $D = D_1, D_2, \ldots$ appliquée du système. Nous indiquerons cela en notant E_D.

Plaçons-nous à l'instant n ; il reste alors une durée N-n à courir avant d'arriver à la fin de la période T. Vue de cet instant, la disponibilité moyenne dépend à la fois de cette durée restante, de la procédure D et de l'état dans lequel se trouve le système à l'instant n (Z_n). On est ainsi amené à introduire les 2 fonctions suivantes :

$$V_n^1(D, k) = E_D\left\{\sum_{i=n}^{N} (1-p)^{Z_i}/Z_n = k\right\}$$

$$V_n^2(D, m) = E_D\left\{\sum_{i=n}^{N} (1-p)^{Z_i}/Z_n = \infty,\ U_n = m\right\}$$

Comme il existe une multitude de procédures D possibles posons :

$$V_n^{1*}(k) = \max_D V_n^1(D, k)$$

$$V_n^{2*}(m) = \max_D V_n^2(D, m)$$

On constate que la disponibilité maximum possible obtenue en employant la meilleure politique de test se déduit directement des formules ci-dessus :

$$\bar{A}_{max} = \frac{V_0^{1*}(0)}{N+1}$$

Il reste maintenant à raisonner par récurrence pour déduire les V_n^* des V_{n+1}^*. Ce raisonnement est typique de la méthode de programmation dynamique.

Imaginons qu'à l'instant n le système est dans l'état $Z_n = k$ et que l'opérateur décide de ne pas tester ($D_n = 0$) on aura :

$$V_n^1(D, k) = E_D\left\{\sum_{i=n}^{N} (1-p)^{Z_i}/Z_n = k\right\}$$

$$= (1-p)^k + E_D\left\{\sum_{i=n+1}^{N} (1-p)^{Z_i}/Z_n = k+1\right\}$$

Mais on considère qu'à partir du rang n+1 on connaît la procédure optimale ; il en résulte :

$$V_n^1(D, k) = (1-p)^k + V_{n+1}^{1*}(k+1)$$

On peut imaginer aussi que l'opérateur fasse le test et ceci va nous conduire à :

$$V_n^1(D, k) = (1-p)^k + \rho_k V_{n+1}^{1*}(0) + (1-\rho_k) V_{n+1}^{2*}(0)$$

La comparaison des 2 résultats conduit à choisir la meilleure décision D_n à l'instant n. Il faut tester uniquement si :

$$\rho_k V_{n+1}^{1*}(0) + (1-\rho_k) V_{n+1}^{2*}(0) > V_{n+1}^{1*}(k+1)$$

On trouve ainsi :

$$V_n^{1*}(k) = (1-p)^k + \text{Max}\left\{V_{n+1}^{1*}(k+1),\ \rho_k V_{n+1}^{1*}(0) + (1-\rho_k) V_{n+1}^{2*}(0)\right\}$$

De manière analogue on trouve :

$$V_n^{2*}(m) = \mu_m V_{n+1}^{1*}(0) + (1-\mu_m) V_{n+1}^{2*}(m+1)$$

On a donc trouvé 2 équations de récurrence ; il ne manque plus que les conditions initiales qui sont très simples :

$$\begin{cases} V_{N+1}^{1*}(k) = 0 & \forall k \\ V_{N+1}^{2*}(m) = 0 & \forall m \end{cases}$$

Il est possible d'exprimer les $V_n^{2*}(0)$ en fonction des V_{n+i+1}^{1*} et ceci permet de simplifier le système à résoudre :

$$\begin{cases} V_n^{1*}(k) = (1-p)^k + \text{Max}\left[V_{n+1}^{1*}(k+1),\ \rho_k V_{n+1}^{1*}(0) + (1-\rho_k)\sum_{i=0}^{N-n} q_i V_{n+i+1}^{1*}(0)\right] \\ V_{N+1}^{1*}(k) = 0 \qquad \forall k \end{cases}$$

Ce système se résoud par itérations successives en partant de n = N pour aller jusqu'à n = 0.

Plaçons-nous à l'instant n ; k varie de zéro jusqu'à n, il est évident que si k est petit (on a constaté le bon état de marche du système il y a peu de temps) il n'y a pas de raison de tester, par contre si k devient grand il faudra le faire. Il existe donc une valeur k_n telle que pour $k < k_n$ il ne faille pas tester à l'instant n et $k \geq k_n$ il faille effectuer un test. Lorsque n varie de zéro à N+1, k_n décrit une courbe frontière séparant l'espace en 2 régions : une où il est mieux de ne pas tester, l'autre où il est mieux de tester. L'allure typique de cette frontière est représentée sur la figure n° 1.

I.4 - Application et conclusion

Comme cela est montré sur la figure n° 1, la frontière k_n permet de déterminer les instants de test optimaux. Loin de l'arrêt programmé T on constate que s'il n'y a pas de réparation on obtient une succession régulière d'intervalles entre tests. Lorsqu'une réparation survient les tests se décalent d'une durée égale à celle de la réparation. Lorsque on se trouve à proximité de l'arrêt programmé on peut faire l'économie du (ou des) dernier(s) test(s). Ceci correspond bien à l'idée intuitive que si on fait un test trop près de l'arrêt programmé celui-ci est inutile car on risque de créer une défaillance qui de toutes façons n'aura pas le temps d'être réparée.

Nous avons comparé les résultats obtenus par cette méthode à ceux obtenus dans le cas où les instants de test sont fixés a priori. On constate numériquement que le pseudo-intervalle entre tests déterminé ici correspond pratiquement à l'intervalle entre tests optimum déterminé pour l'autre cas (cf. /3_/). Il en résulte que la disponibilité moyenne maximum obtenue est pratiquement la même dans les 2 cas. Il en résulte aussi que la méthode consistant à optimiser une grille de test fixée a priori est très bonne car elle conduit à des résultats proches de l'optimum optimorum tout en étant beaucoup plus facile à mettre en oeuvre que celle exposée ici.

II - SYSTEME COMPOSE DE 2 MODULES IDENTIQUES

Nous ne développerons pas ici la théorie permettant de traiter le cas d'un système formé de 2 modules en série ou en parallèle. Elle est analogue à ce qui a été expliqué ci-dessus.

Dans le cas de 2 modules on peut donc aussi chercher des courbes frontières séparant l'espace en plusieurs zones :

- ne pas tester
- tester le module n° 1
- tester le module n° 2

Mais on obtient alors un équivalent à 3 dimensions de la figure n° 1 qui est à 2 dimensions. Pour pouvoir représenter cela nous avons effectué une coupe à n = constante dans la région où l'équilibre est atteint (assez loin de l'arrêt programmé et assez loin de l'origine des temps). Les figures ainsi obtenues sont différentes selon que l'on étudie le système série (figure n° 2) ou le système parallèle (figure n° 3). Sur ces figures les axes représentent les ages respectifs (c'est-à-dire Z_n^1 et Z_n^2) de chacun des modules. On a d'autre part, reporté parallèlement à ces axes d'autres axes qui représentent les cas où un module ou l'autre est en réparation.

Considérons par exemple la figure n° 3. A l'instant t = 0 on se trouve à l'origine (0) des axes, c'est-à-dire dans la zone où il ne faut pas tester. Le temps s'écoule de la même façon pour Z_n^1 et Z_n^2 et il en résulte que l'on se déplace sur la diagonale de la figure ; on se déplace ainsi jusqu'à ce que l'on touche le point faisant la jonction entre les trois zones (1). A ce moment là il faut tester chacun des modules. Si aucune défaillance n'est décelée Z_n^1 et Z_n^2 prennent ensemble la valeur zéro, on est ramené à l'origine (0) et on recommence. Ainsi, tant qu'aucune défaillance n'est mise en évidence, les 2 modules doivent être testés en même temps et à intervalle régulier. Imaginons maintenant que le module n° 1 soit trouvé en panne, pour savoir ce qu'il faut faire sur le second, il faut aller voir sur l'axe correspondant au cas où le module n° 1 est en réparation. On constate que l'on tombe dans la zone où il ne faut pas tester le second module. Puis, l'age de 2 augmentant on arrive sur le point (2) où il faut le tester. On voit que tant que le premier module est en réparation les tests du second doivent s'effectuer à intervalles plus grands (intervalles 3 - 2) que lorsque les 2 modules sont en attente normale. Lorsque la réparation se termine Z_n^2 a une certaine valeur, par contre Z_n^1 se trouve remis à zéro. On repart donc, sur le diagramme central en un point (4) qui n'est pas

diagramme on atteint la zone (5) où il faut tester le module n° 2 ; si ce test ne détecte pas de défaillance Z_B^2 est remis à zéro ; on repart donc d'un point (6) situé sur l'axe Z_A^1. Au bout d'un certain temps on rencontre la zone (7) où il faut tester le module n° 1. Et ainsi de suite. Dès la première réparation les tests se désynchronisent et il n'y a aucune raison qu'ils se resynchronisent par la suite.

L'utilisation de la figure n° 2 est parfaitement analogue. La grosse différence provient du fait que lorsqu'un module est trouvé en panne il est nécessaire de tester l'autre plus fréquemment. Ceci paraît logique, en effet, dans le cas d'un système série dès qu'un module est défaillant, le système global est défaillant donc si on doit faire aussi une réparation aussi sur le second module autant la faire le plus vite possible. Le chevauchement des réparations permet ainsi de gagner en disponibilité. Dans le cas d'un système parallèle c'est l'inverse qui se produit. Ceci est logique aussi car le test du second module présente un risque de mettre le système global en défaillance lorsque le premier module est en réparation. Donc pour diminuer ce risque il faut diminuer la fréquence des tests.

Ces résultats sont très intéressants et mettent bien en évidence que l'optimisation de chacun des modules pris séparément ne conduit pas à l'optimisation du système global. Un calcul que nous avons fait montre un écart de 16 % ce qui n'est pas négligeable du tout. On peut penser que sur des systèmes plus compliqués cet écart ne pourrait que se creuser.

CONCLUSION

La méthode exposée ici permet de traiter le problème des systèmes en attente de fonctionnement d'une manière complètement différente de ce qui est fait habituellement. Elle conduit à l'optimisation de la disponibilité moyenne mais les politiques de test qui en découlent peuvent être difficiles à mettre en oeuvre. Le grand intérêt de ces calculs est de montrer comment se situent les politiques de test plus simples (intervalle entre tests fixé a priori, par exemple) par rapport à la politique optimum. Ainsi on a pu constater que dans le cas d'un module simple le gain est insignifiant (si ce n'est que l'on peut faire l'économie du ou ces dernier(s) test(s). Par contre dans le cas d'un système formé de 2 modules l'écart est plus grand et il est probable que cet écart devienne de plus en plus grand au fur et à mesure que la complexité du système augmente.

Cette technique est donc très intéressante car elle permet de déterminer la stratégie de test optimale c'est-à-dire d'évaluer la limite de la disponibilité moyenne au-dessus de laquelle il est impossible d'aller, en agissant simplement sur la politique de test.

REFERENCES

1. Jacobs, I.M. Marriott, P.W., Avril 1969, "Guideline for determining Safe tests intervals and repair times for engineered safeguards" APED 5736.

2. Vaurio, J.K., 1979, "Unavailability of components with inspection and repair", Nuclear Engineering and Design 54 (1979) 309-324.

3. Signoret, J.P., 1976 "Disponibilité d'un système en attente périodiquement testé", C.E.A., rapport DSN n° 113.

4. Signoret, J.P., 1976, "Disponibilité d'un système en attente périodiquement testé - calculs approchés Optimisation" C.E.A., rapport DSN n° 129.

5. Signoret, J.P., 1978, "Disponibilité d'un système en attente périodiquement testé - durée du test non négligeable - test non efficace à 100 % - système redondant d'ordre 2", C.E.A., rapport DSN n° 206.

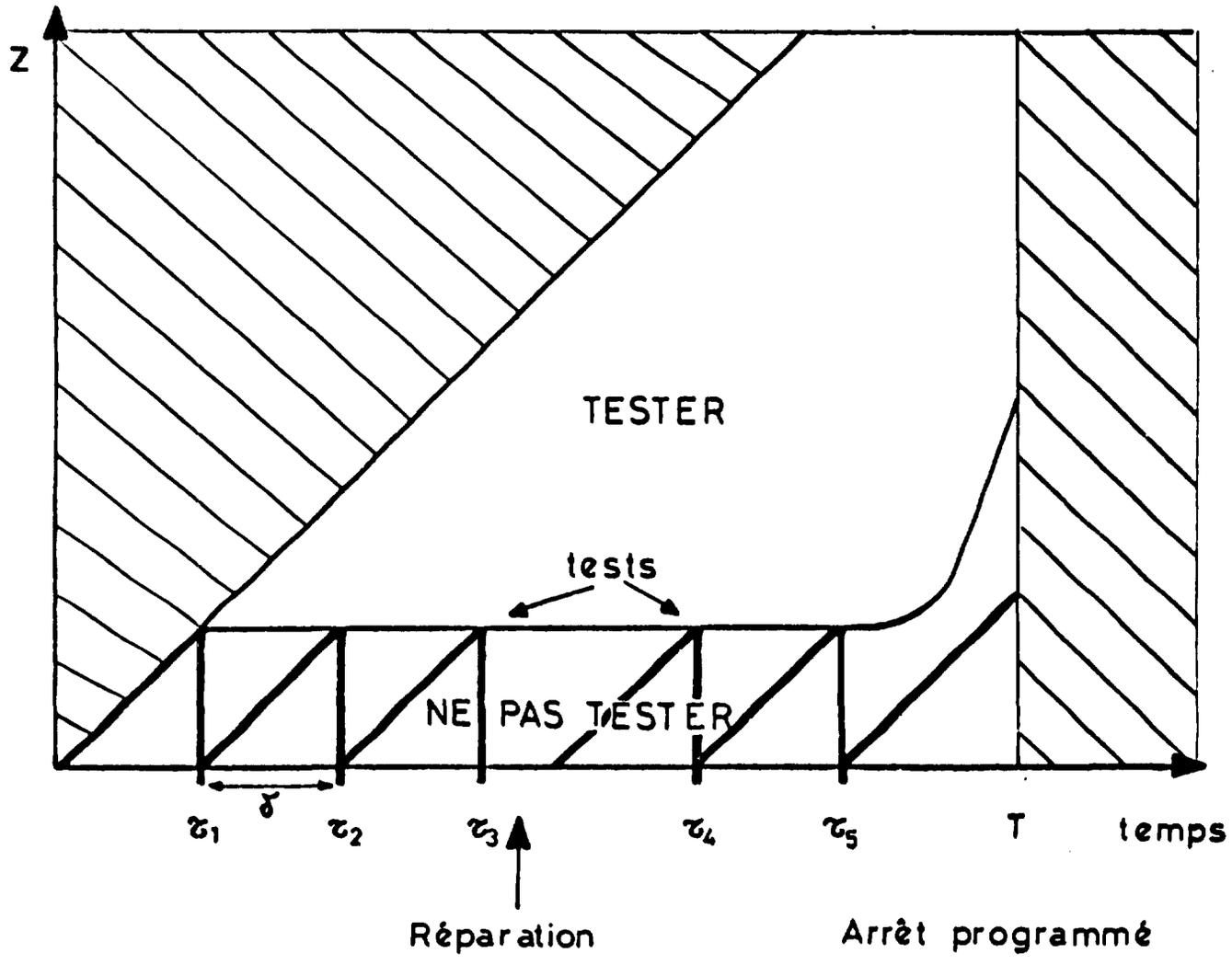

Z
TESTER
tests
NE PAS TESTER
γ
τ_1
τ_2
τ_3
τ_4
τ_5
T
temps
Réparation
Arrêt programmé

Figure n° 1

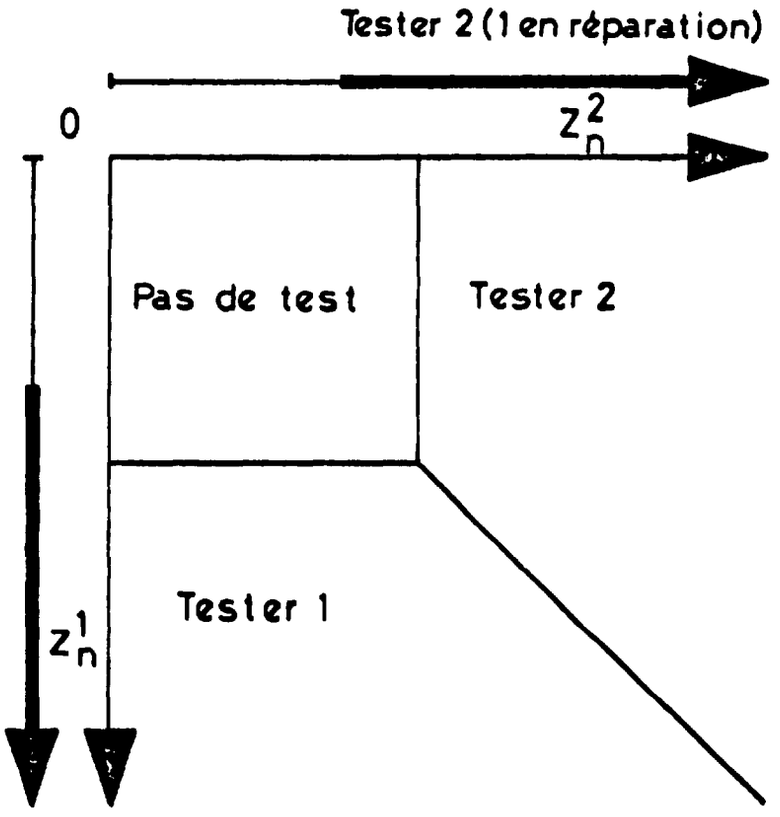


Figure n°2 - Système série

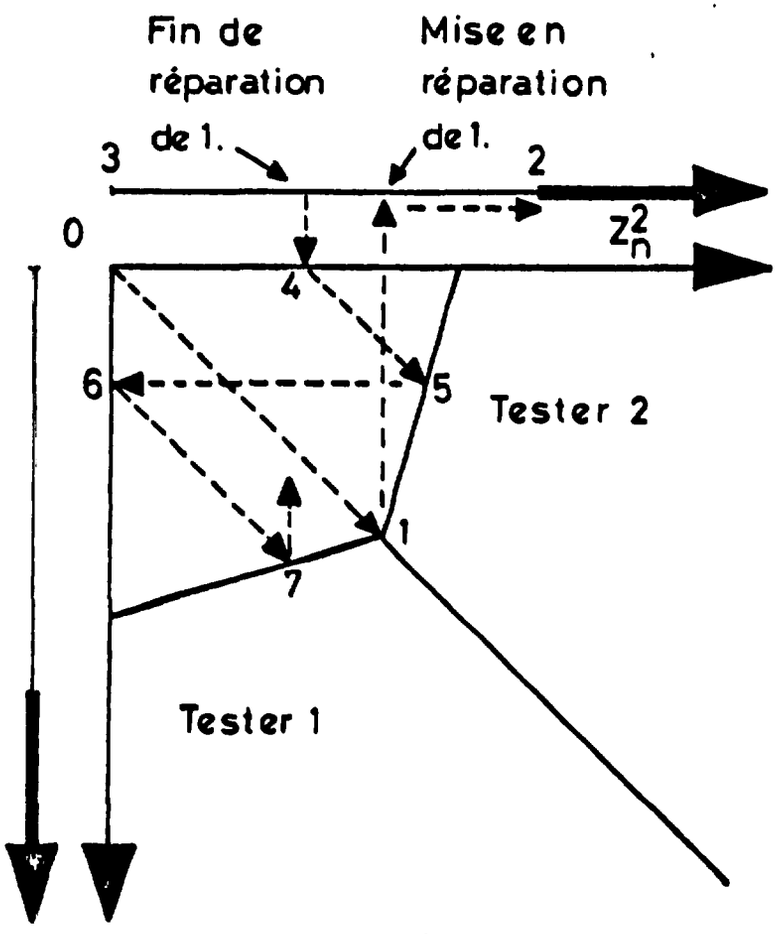


Figure n°3 - Système parallèle