

**IRSN**

INSTITUT  
DE RADIOPROTECTION  
ET DE SÛRETÉ NUCLÉAIRE

*Faire avancer la sûreté nucléaire*

# Principes relatifs à la démarche de conception du contrôle-commande numérique

2017

---

## RÉSUMÉ

*La conception du contrôle-commande des installations nucléaires utilise des systèmes numériques offrant des capacités croissantes de calcul et d'interconnexion : elles permettent de réaliser des fonctions avancées comme le calcul du rapport de flux thermique critique, de détecter en temps réel les défaillances du matériel ou encore d'offrir aux opérateurs des interfaces riches et souples. Toutefois, ces fonctions évoluées peuvent être entachées de défauts rendant leur logique systématiquement inadéquate dans certains cas, ce qui introduit des sources de défaillances autres que les pannes aléatoires du matériel et suscite des interrogations liées à la notion informelle de « complexité » croissante du contrôle-commande. Des principes de conception adaptés doivent donc être appliqués pour que cette logique soit autant que possible exempte de défaut et évaluable par une entité indépendante telle que l'IRSN.*

*Ce document présente les principaux problèmes associés à la conception du contrôle-commande numérique d'une installation complexe, ainsi que les principes généraux à respecter pour démontrer l'atteinte d'un niveau de sûreté satisfaisant. Les éléments de doctrine présentés dans ce document sont issus de l'expérience acquise lors des évaluations menées pour le parc électronucléaire français, nourries des échanges avec les experts du secteur nucléaire, et reflètent la pratique française ; ils s'appliquent à d'autres secteurs dans lesquels un haut niveau de confiance doit pouvoir être accordé au contrôle-commande.*

*Les textes normatifs cités dans ce document fournissent des exigences détaillées nécessitant une large part d'interprétation, car la nature du problème posé ne permet pas la définition de critères mesurables pertinents dans tous les cas. Ce document vise à expliciter les principes qui sous-tendent ces exigences détaillées et à donner ainsi les moyens de les interpréter dans chaque situation.*

## TABLE DES MATIERES

REFERENCES.....	4
DEFINITIONS.....	5
ABREVIATIONS.....	6
1 CONTEXTE.....	7
2 MISSIONS ET ORGANISATION DU CONTROLE-COMMANDE.....	7
3 TEXTES REGLEMENTAIRES ET NORMATIFS - CONSENSUS INTERNATIONAL.....	8
4 APPROCHE GENERALE DE SURETE.....	9
5 PARTICULARITES DU CONTROLE-COMMANDE.....	9
5.1 INTERET DES TECHNIQUES NUMERIQUES.....	9
5.2 PARTICULARITES DES DEFAILLANCES.....	10
5.2.1 Defaillances matérielles.....	10
5.2.2 Defaillances résultant d'une logique inadéquate.....	11
5.2.3 Nécessité d'une démarche spécifique pour la logique.....	12
5.3 PARTICULARITES EN MATIERE D'INDEPENDANCE.....	13
<b>6 DEMARCHE ET PRINCIPES DE CONCEPTION.....</b>	<b>13</b>
6.1 DEMARCHE GENERALE DE CONCEPTION DU CONTROLE-COMMANDE.....	13
6.2 SPECIFICATION DU CONTROLE-COMMANDE.....	15
6.3 CONCEPTION DE L'ARCHITECTURE.....	15
6.4 CONCEPTION D'UN SYSTEME.....	17
6.4.1 Evitement des défauts.....	17
6.4.2 Suppression des défauts.....	21
6.4.3 Tolérance d'un système aux défauts résiduels.....	23
6.5 PRISE EN COMPTE DANS L'ARCHITECTURE DES DEFAILLANCES POSTULEES - DIVERSIFICATION.....	23
6.5.1 Principes généraux.....	23
6.5.2 Impact de la diversification sur la complexité de l'architecture.....	24
6.5.3 Application au cas du réacteur EPR Flamanville 3.....	24
6.5.4 Inefficacité de la diversification d'un logiciel.....	25
6.6 PRISE EN COMPTE DE LA MALVEILLANCE.....	26
<b>7 ACTIONS INTEMPESTIVES OU INAPPROPRIEES DU CONTROLE-COMMANDE.....</b>	<b>27</b>
<b>8 CONCLUSION.....</b>	<b>29</b>
<b>ANNEXE 1 - EXEMPLES D'EXIGENCES DE CONCEPTION DETAILLEES.....</b>	<b>31</b>
<b>ANNEXE 2 - RETOUR D'EXPERIENCE DES SYSTEMES NUMERIQUES.....</b>	<b>33</b>
<b>ANNEXE 3 - EXPERIENCE DE KNIGHT ET LEVESON.....</b>	<b>34</b>
<b>ANNEXE 4 - APPROCHES PROBABILISTES.....</b>	<b>35</b>
<b>ANNEXE 5 - TECHNIQUES DE REALISATION PARTICULIERES (INTERRUPTIONS).....</b>	<b>37</b>

## REFERENCES

- [1] Règle Fondamentale de Sûreté II.4.1.a, Logiciel des systèmes électriques classés de sûreté
- [2] Règle Fondamentale de Sûreté I.3.a, Utilisation du critère de défaillance unique dans les analyses de sûreté
- [3] Règle Fondamentale de Sûreté IV.2.b, Exigences à prendre en compte dans la conception, la qualification, la mise en œuvre et l'exploitation des matériels électriques appartenant aux systèmes électriques classés de sûreté
- [4] Directives techniques pour la conception et la construction de la nouvelle génération de réacteurs nucléaires à eau sous pression adoptées pendant les réunions plénières du GPR et d'experts allemands les 19 et 26 octobre 2000
- [5] Guide de l'ASN n° 22 réalisé conjointement avec l'Institut de Radioprotection et de Sûreté Nucléaire - Conception des réacteurs à eau sous pression
- [6] RCC-E, Recueil des règles de Conception et de Construction des matériels Electriques des îlots nucléaires (2012)
- [7] IAEA Safety Standards, SSG-39 Design of Instrumentation and Control Systems for Nuclear Power Plants
- [8] IEC 61226, ed3.0, Nuclear power plants - Instrumentation and control important to safety - Classification of instrumentation and control functions
- [9] IEC 61513, ed2.0, Nuclear power plants - Instrumentation and control important to safety - General requirements for systems
- [10] IEC 60880, ed2.0, Nuclear power plants - Instrumentation and control systems important to safety - Software aspects for computer-based systems performing category A functions
- [11] IEC 62138, Nuclear power plants - Instrumentation and control important for safety - Software aspects for computer-based systems performing category B or C functions
- [12] IEC 62566, Nuclear power plants - Instrumentation and control important to safety - Development of HDL-programmed integrated circuits for systems performing category A functions
- [13] IEC 62340, Nuclear power plants - Instrumentation and control systems important to safety - Requirements for coping with common cause failure (CCF)
- [14] NUREG/IA-0254, Suitability of Fault Modes and Effects Analysis for Regulatory Assurance of Complex Logic in Digital Instrumentation and Control Systems, NRC et IRSN, avril 2011
- [15] Assessment of the overall Instrumentation and Control architecture of the EPR FA3 project, J. Gassino et P. Régnier, IRSN, EUROSAFE 2010
- [16] An experimental evaluation of the assumption of independence in multi-version programming, IEEE Transactions on Software Engineering, January 1986, John C. Knight, Nancy G. Leveson
- [17] J. Bickel, Risk Implications of Digital RPS Operating Experience, 2007, AIEA
- [18] EPRI, Operating Experience Insights on Common-Cause Failures in Digital Instrumentation and Control Systems, 2008
- [19] « Multi-domain comparison of safety standards », ERTS 2010, P. Baufreton, JP. Blanquart, JL. Boulanger, H. Delseny, JC. Derrien, J. Gassino, G. Ladier, E. Ledinet, M. Leeman, P. Quéré, B. Ricque

## DEFINITIONS

**Architecture de contrôle-commande** : « structure organisant les systèmes de contrôle-commande de la centrale importants pour la sûreté » [extrait de la référence [9]].

**Classe d'un système de contrôle-commande** : « l'une des () affectations possibles () des systèmes de contrôle-commande importants pour la sûreté, résultant de la nécessité pour ces systèmes d'exécuter des fonctions d'importances pour la sûreté différentes. Une affectation « Non Classé » est délivrée si le système n'exécute pas de fonction importante pour la sûreté » [extrait de la référence [9], modifiée pour supprimer le nombre et les noms des classes spécifiques à l'IEC (voir parenthèses vides)].

**Composant** : « l'une des pièces constituant un système. Un composant peut être matériel ou logiciel et peut être subdivisé en plusieurs autres composants » [extrait de la référence [9]].

**Critère de défaillance unique** : « critère (ou contrainte) appliqué à un système, en vertu duquel ce dernier doit être capable de remplir sa (ses) fonctions en cas de défaillance unique » [extrait de la référence [9]].

**Défaillance** : « perte de la capacité d'une structure, d'un système ou d'un composant de fonctionner conformément aux critères d'acceptation » [extrait de la référence [9]].

**Défaillance unique** : « perte de la capacité d'un composant à remplir sa (ses) fonction(s) de sûreté prévue(s) et toute autre défaillance qui peut en résulter » [extrait de la référence [9]].

**Défaillance de cause commune** : « défaillance de plusieurs structures, systèmes ou composants due à un événement ou à une cause unique » [extrait de la référence [9]].

**Défaut** : « imperfection dans un composant matériel, logiciel ou système »  
« NOTE Les défauts peuvent provenir de défauts aléatoires, par exemple suite au vieillissement du matériel, et peuvent être systématiques, par exemple des défauts logiciels, suite à des erreurs de conception » [extrait de la référence [9]].

**Défaut systématique** : « défaut relié de façon déterministe à une certaine cause, ne pouvant être éliminé que par une modification de la conception, du processus de fabrication, des procédures d'exploitation, de la documentation ou d'autres facteurs appropriés » [extrait de la référence [9]].

**Fonction de contrôle-commande** : « fonction permettant de commander, exploiter et/ou surveiller une partie définie du procédé » [extrait de la référence [9]].

**Système** : « ensemble de composants qui interagissent conformément à une conception donnée, un élément d'un système pouvant être un autre système, appelé sous-système » [extrait de la référence [9]].

**Système de contrôle-commande** : « système exécutant des fonctions de contrôle-commande ainsi que des fonctions de service et d'affichage liées au fonctionnement du système lui-même. Sa technologie est électrique et/ou électronique et/ou électronique programmable » [extrait de la référence [9]].

## **ABREVIATIONS**

AAR	Arrêt automatique du réacteur
AIEA	Agence Internationale de l'Énergie Atomique
CEI	Commission Electrotechnique Internationale (IEC en anglais)
DCC	Défaillance de cause commune
IEC	International Electrotechnical Commission (CEI en français)
MCP	Moyen de Conduite Principal
MCS	Moyen de Conduite de Secours
PAS	Système d'automatisme de tranche
PS	Système de Protection (Protection System)
RFTC	Rapport de flux thermique critique
SAS	Système d'automatisme de sûreté

# **1 CONTEXTE**

La conception du contrôle-commande est un domaine important dans l'évaluation de la sûreté nucléaire, comme le montre l'actualité internationale au Royaume-Uni et en Finlande, où elle a suscité une forte activité dans le cadre de l'évaluation de l'EPR. Cette importance se traduit aussi par une activité soutenue des groupes de travail internationaux et des organismes de normalisation dans ce domaine.

Le développement des technologies numériques permet de mettre en œuvre des contrôles-commandes performants mais pose des difficultés spécifiques en termes de démonstration de sûreté, ce qui a conduit les parties intéressées en France (AREVA, EDF, IRSN) à développer progressivement une approche particulière.

Ces difficultés sont apparues soudainement dans les pays dont le programme nucléaire avait été peu actif depuis les années 1970. La situation en France est différente dans la mesure où ces techniques ont été introduites progressivement dans les paliers P4, P'4 et N4, ce qui a permis aux industriels et à l'IRSN d'acquérir une expérience conséquente, créant une situation de fait plus favorable ; l'évaluation de sûreté du contrôle-commande numérique du réacteur EPR Flamanville 3 a toutefois soulevé des problèmes difficiles dont la résolution a nécessité une forte implication de l'exploitant, des constructeurs, de l'Autorité de sûreté nucléaire et de l'IRSN.

Le présent document explicite l'approche développée en France en présentant la démarche et les principes que l'IRSN estime devoir être mis en œuvre lors de la conception de l'architecture et des systèmes de contrôle-commande numérique (en dehors des interfaces avec le procédé et des interfaces homme-machine, pour lesquelles le document ne détaille pas les principes spécifiques). Il s'applique aux réacteurs électronucléaires. Ses principes peuvent être pertinents pour d'autres installations nucléaires mais leur application détaillée dépendra de la complexité de l'installation et des risques qu'elle présente, sur la base d'une approche proportionnée à l'importance de ces risques. Il précise les fondements de l'approche précitée et la situe par rapport au consensus international.

Il expose la nécessité de principes de conception spécifiques pour démontrer l'atteinte d'un niveau de sûreté satisfaisant. Il ne reproduit pas les exigences détaillées qui traduisent en pratique la mise en œuvre de ces principes car cela conduirait à un document trop volumineux et redondant avec les textes normatifs cités au chapitre 3. Cependant, il en fait apparaître les motivations et permet ainsi de les interpréter et de les appliquer de façon adéquate.

Il relève ainsi du référentiel d'expertise de sûreté de l'IRSN et ne constitue pas un guide d'expertise.

## **2 MISSIONS ET ORGANISATION DU CONTROLE-COMMANDE**

Le contrôle-commande de sûreté participe à des fonctions de surveillance, de régulation et de protection de l'installation. Il comporte pour ce faire :

- des interfaces avec le procédé : capteurs et actionneurs, soit « tout ou rien », soit « continu » ;
- des automates chargés d'acquérir les mesures et les commandes des opérateurs, de les traiter, de commander les actionneurs et d'élaborer les informations nécessaires à l'exploitation ;
- des interfaces avec les opérateurs (moyens de conduite) et avec les équipes de maintenance.

Le contrôle-commande est organisé en systèmes réalisant des fonctions homogènes, parmi lesquels on peut distinguer, pour une centrale nucléaire :

- un système de protection, qui réalise entre autres les fonctions automatiques d'arrêt du réacteur (AAR) et de mise en service des systèmes de sauvegarde (pour EPR : Système de Protection, PS) ;
- un système participant aux fonctions nécessaires à l'atteinte de l'état sûr en cas de situation accidentelle (pour EPR : Système d'automatisme de sûreté, SAS) ;
- un système participant aux fonctions automatiques et manuelles utilisées en situation normale, ainsi qu'à certaines fonctions de limitation (pour EPR : Système d'automatisme de tranche, PAS) ;
- un moyen de conduite principal, informatisé, utilisable (dans toutes les situations) tant qu'il est disponible (pour EPR : Moyen de Conduite Principal, MCP) ;
- un moyen de conduite de secours, non informatisé, permettant de gérer les situations incidentelles et accidentelles lorsque le moyen de conduite principal est indisponible (pour EPR : Moyen de Conduite de Secours, MCS).

Les systèmes de contrôle-commande sont organisés selon une architecture visant à satisfaire des exigences fonctionnelles (par exemple certains systèmes comme le MCP doivent communiquer avec d'autres) et des exigences de sûreté (par exemple l'indépendance).

### **3 TEXTES REGLEMENTAIRES ET NORMATIFS - CONSENSUS INTERNATIONAL**

En France, pour les réacteurs à eau sous pression, la RFS II.4.1.a [1] traite des exigences applicables aux logiciels, en particulier celles de déterminisme et de prédictibilité qui jouent un rôle essentiel dans la conception des logiciels de sûreté (voir le chapitre 6). La RFS I.3.a [2] traite de l'utilisation du critère de défaillance unique et la RFS IV.2.b [3] des exigences applicables aux matériels électriques classés de sûreté. Les directives techniques en référence [4] et le guide en référence [5] incluent également des exigences pour le contrôle-commande.

Le document RCC-E [6] (règles de conception et de construction des matériels électriques) élaboré par l'AFCEM (Association française pour les règles de conception et de construction des matériels des chaudières électronucléaires) décrit les pratiques adoptées par les constructeurs pour développer le contrôle-commande de sûreté en respectant les objectifs fixés par les RFS et l'Agence internationale de l'énergie atomique (AIEA), ainsi que les exigences des normes CEI.

L'AIEA recommande des objectifs concernant le contrôle-commande, les travaux les plus récents sont précisés dans le guide SSG-39 « *Design of Instrumentation and Control Systems for Nuclear Power Plants* » [7]. Des exigences permettant d'atteindre ces objectifs sont détaillées dans les normes du comité 45A de la Commission électrotechnique internationale (CEI, IEC en anglais), en vertu d'un accord entre ces deux organismes.

Les principaux textes de la CEI concernant le contrôle-commande de sûreté sont : l'IEC 61226 [8] (classement des fonctions de contrôle-commande), l'IEC 61513 [9] (systèmes), l'IEC 60880 [10] et l'IEC 62138 [11] (logiciels), l'IEC 62566 [12] (électronique programmable) et l'IEC 62340 [13] (défaillance de cause commune).

**Ces textes normatifs essentiels dans le domaine du contrôle-commande nucléaire sont récents et sont totalement cohérents avec la démarche et les principes décrits dans ce document, qui font donc l'objet d'un large consensus international.**

La doctrine présentée dans le présent document explicite et justifie les fondements de ces textes.



## 4 APPROCHE GENERALE DE SURETE

Le principe de défense en profondeur, tel que mentionné dans l'arrêté du 7 février 2012 fixant les règles générales relatives aux installations nucléaires de base, est le principe fondamental de sûreté nucléaire. Pour la démonstration de sûreté nucléaire, différents niveaux de défense en profondeur sont définis. Des moyens doivent être mis en œuvre en vue de garantir le bon « fonctionnement » de chaque niveau, dont on suppose néanmoins ensuite la défaillance ; d'autres moyens, suffisamment indépendants des précédents, sont définis pour limiter les conséquences de cette défaillance. Du principe de défense en profondeur découlent notamment des exigences relatives à :

- la prévention des défaillances des systèmes, y compris des défaillances de cause commune (DCC) ;
- l'identification des défaillances plausibles des systèmes ;
- la définition de dispositions visant à limiter les conséquences de ces défaillances ;
- l'indépendance des systèmes participant à différents niveaux de la défense en profondeur.

Les particularités du contrôle-commande en termes d'évolutions technologiques, de modes de défaillance et d'indépendance, rappelées au chapitre 5, ont motivé l'élaboration de la démarche et des principes présentés aux chapitres 6 et 7.

## 5 PARTICULARITES DU CONTROLE-COMMANDE

### 5.1 INTERET DES TECHNIQUES NUMERIQUES

Dans la technologie dite « câblée », utilisée par exemple pour les réacteurs de 900 MWe, la fonctionnalité voulue (telle que la validation d'un signal de déclenchement par un permissif) est réalisée de façon concrète par le câblage physique de composants élémentaires comme des relais. La taille de ces composants élémentaires, de l'ordre du centimètre, limite leur nombre dans un système et donc la complexité des fonctions réalisables.

La technologie numérique utilise des circuits intégrés contenant typiquement des millions ou des milliards de structures élémentaires<sup>1</sup> équivalentes à des relais, ce qui lui donne une capacité fonctionnelle bien illustrée par la comparaison des objets de la vie courante (tels que les téléphones) avec leurs homologues des années 1960-70. Dans le domaine nucléaire, les évolutions, bien que plus modestes pour conserver une maîtrise appropriée des systèmes, sont tout de même importantes. Elles portent en particulier sur :

- **les fonctions automatiques** : la technologie numérique autorise des fonctions plus nombreuses et plus avancées que la logique câblée, par exemple le calcul du rapport de flux thermique critique (RFTC) en temps réel à partir de centaines de capteurs et de paramètres de conception et de calibration ; elle permet donc de mieux connaître l'état de l'installation ;

---

<sup>1</sup> Les concepteurs ne définissent pas directement le câblage de ces structures, mais développent des programmes qui décrivent la logique à réaliser de façon abstraite, au moyen de « boucles », « variables », etc. Des outils traduisent ces programmes en interactions des structures élémentaires contenues dans des circuits intégrés de type « microprocesseur » ou « électronique programmable » tels que les « Field Programmable Gate Array » (FPGA).

- **les interfaces de conduite** : les opérateurs ont accès à un grand nombre d'informations et de moyens d'action, dont la présentation peut être adaptée à l'état de l'installation et à leur demande en prenant en compte les exigences liées à l'ergonomie ;
- **la tolérance aux aléas d'exploitation** : les architectures numériques permettent d'effectuer davantage d'opérations de maintenance et de test périodique en fonctionnement, en détectant et en traitant des situations anormales telles que la simultanéité d'actions de maintenance sur plusieurs voies redondantes ;
- **la disponibilité des systèmes** : les systèmes numériques de sûreté surveillent en permanence le fonctionnement de leurs composants matériels, en testant continuellement les cases mémoires et les ressources de calcul, en émettant des « signes de vie » surveillés par d'autres systèmes ou sous-systèmes, etc. ; ils détectent ainsi la plupart de leurs défaillances, les signalent et mettent, dans ce cas, leurs sorties dans des positions prédéfinies ;
- **le diagnostic et la maintenance** : grâce aux réseaux de communication qui permettent de transmettre des milliers d'informations élémentaires sur un même support physique, les unités de diagnostic peuvent surveiller en temps réel les comptes rendus d'autosurveillance des systèmes et comparer de nombreux résultats globaux et intermédiaires d'entités redondantes pour détecter les discordances.

## 5.2 PARTICULARITES DES DEFAILLANCES

En raison de la richesse fonctionnelle mentionnée ci-avant et de la complexité technologique qui la rend possible, des erreurs de spécification ou de réalisation peuvent être commises, ce qui induit des défauts permanents dans la logique du système concerné : dans certains cas, sa réponse diffère de la réponse spécifiée (cas d'une erreur de réalisation) ou de la réponse souhaitable (cas d'une erreur de spécification).

Un système de contrôle-commande peut donc être défaillant du fait de défaillances de ses composants matériels (exemple : un relais ou un circuit intégré tombe en panne) ou du fait de défauts de conception rendant sa logique inadéquate dans certains cas. Les paragraphes ci-après montrent que l'utilisation des techniques numériques tend à réduire les premières causes (ou leurs conséquences) et à favoriser les secondes.

### 5.2.1 DEFAILLANCES MATERIELLES

La fiabilité matérielle des systèmes de contrôle-commande doit être conforme aux besoins de disponibilité des fonctions qu'ils exécutent. Elle doit être estimée au moment de la conception, puis faire l'objet d'un suivi en exploitation pour détecter toute situation nécessitant des dispositions correctives : en effet, une fiabilité moins bonne que prévu ou en baisse peut dénoter une erreur de conception du matériel électronique conduisant un composant à fonctionner en dehors de ses spécifications (par exemple application d'une tension excessive lors des commutations, dissipation thermique insuffisante), un problème d'approvisionnement de composants, un défaut de fabrication ou encore une exploitation incorrecte.

La qualité de la conception, de la fabrication et de l'exploitation doit garantir que chaque composant matériel fonctionne dans la plage physique spécifiée par son fabricant (tension, température, humidité, vibrations, etc.). Dans ces conditions, chaque composant a une probabilité de défaillance spontanée en principe connue et documentée dans des tables de fiabilité, que le concepteur utilise pour calculer la probabilité d'apparition d'une défaillance rendant le système concerné indisponible, totalement ou partiellement (dans le cas de systèmes possédant des redondances).

Cependant, les tables de fiabilité réellement disponibles, constituées à partir du retour d'expérience, ne comprennent pas les données relatives aux circuits récents : certaines tables peuvent encore être largement utilisées bien que non actualisées depuis 1991. De plus, ces tables ne définissent pas les modes de défaillance à prendre en compte pour les circuits intégrés offrant un grand nombre de fonctionnalités. Les concepteurs sont donc amenés à raisonner par analogie et par extrapolation, ce qui affaiblit la crédibilité des valeurs obtenues. Cette faiblesse théorique peut être compensée, pour les composants anciens, par un retour d'expérience favorable montrant le conservatisme de ces estimations.

Comme indiqué au sous-chapitre 5.1, les capacités d'autosurveillance des systèmes numériques permettent de détecter et de signaler la plupart des défaillances matérielles du contrôle-commande, de telle sorte que seule une faible fraction d'entre elles peut occasionner une indisponibilité cachée d'un système numérique de sûreté.

Par ailleurs, certains systèmes importants pour la sûreté doivent remplir leurs missions malgré une défaillance unique affectant un de leurs composants, tout en limitant les risques d'actions intempestives, ce qui nécessite des architectures redondantes. Les systèmes numériques autorisent des architectures complexes qui permettent d'atteindre ces objectifs même en présence d'indisponibilités supplémentaires, dues par exemple à l'accomplissement de la maintenance préventive ou des tests périodiques en fonctionnement.

Dans une architecture à trois voies redondantes combinées dans un vote en 2 sur 3 (utilisée pour le système de protection « câblé » des réacteurs de 900 MWe), aucune défaillance unique en amont du voteur ne peut empêcher l'action de sûreté ni la déclencher de façon intempestive. La défaillance du voteur lui-même peut être considérée comme faisant partie de celle de l'actionneur concerné.

Une architecture à quatre redondances combinées dans un vote en 2 sur 4 (utilisée pour les systèmes de protection numériques des réacteurs de 1300 MWe, de 1400 MWe et EPR) permet de tolérer une indisponibilité supplémentaire (pour maintenance, test périodique, etc.) sans perte de mission ni action intempestive. Ce vote plus complexe (qu'il faut reproduire pour chacune des centaines de sorties du système) est souvent réalisé en technologie numérique. Celle-ci permet également de détecter des situations anormales comme l'exécution d'opérations de maintenance sur plus d'une redondance à la fois, et de traiter ces situations de façon conforme aux exigences de sûreté.

Ainsi, les technologies numériques améliorent la tolérance des systèmes de contrôle-commande aux défaillances matérielles, au prix d'une complexification des architectures et des traitements.

### **5.2.2 DEFAILLANCES RESULTANT D'UNE LOGIQUE INADEQUATE**

Comme indiqué au sous-chapitre 5.2, des défaillances peuvent survenir du fait d'un défaut affectant la logique du contrôle-commande, induisant un comportement erroné dans certains cas parfaitement déterminés par les caractéristiques du défaut mais inconnus tant que ces cas ne sont pas exécutés (sinon le défaut serait corrigé).

La technologie numérique est plus propice que la technologie câblée à des défauts affectant la logique, parce qu'elle permet de réaliser des fonctions plus nombreuses et plus complexes (voir le sous-chapitre 5.1). Ceci constitue un enjeu de sûreté important pour le secteur nucléaire dans la mesure où la technologie numérique est aujourd'hui largement mise en œuvre dans les systèmes de sûreté d'installations nouvelles ou lors de rénovations.

### 5.2.3 NECESSITE D'UNE DEMARCHE SPECIFIQUE POUR LA LOGIQUE

Les défauts de logique des systèmes numériques ne sont pas de même nature que les défaillances matérielles et ne peuvent pas être analysés, prévenus ou tolérés avec les moyens adaptés au cas des matériels tels que :

- les analyses de type AMDE (Analyse des Modes de Défaillance et de leurs Effets) : elles sont utilisées avec succès pour évaluer la fiabilité d'un matériel à partir des défaillances de ses composants (relais collé ouvert, relais collé fermé, etc.) et de leurs chemins de propagation vers les sorties, qui suivent le câblage physique des composants, lui-même image de leurs relations fonctionnelles ; mais le cas d'une logique est différent de celui d'un matériel : ses défauts ne sont pas connus (sinon ils seraient corrigés) et le nombre de défauts envisageables est trop grand pour pouvoir les analyser de façon enveloppe. De plus, les chemins de propagation de chaque défaut envisageable ne sont pas connus a priori ; en particulier ils ne sont pas limités aux relations fonctionnelles car un défaut dans un traitement peut indûment affecter une mémoire utilisée par un autre, fonctionnellement indépendant du premier. Les analyses de type AMDE ne sont donc pas applicables aux défauts de logique [14] ;
- la vérification par test en fin de conception : les tests permettent de vérifier selon une démarche simple une fonction réalisée en logique câblée, qui possède typiquement peu d'entrées et de mémoires internes. Le nombre de cas possibles augmentant exponentiellement avec le nombre d'entrées et le nombre de mémoires, il est bien plus élevé dans le cas des systèmes numériques et la vérification de ces derniers ne peut pas reposer simplement sur des tests en fin de conception.

15 entrées analogiques numérisées chacune sur 10 bits (soit  $2^{10}=1024$  valeurs possibles) et 15 mémoires internes de la même taille, soit  $30 \times 10$  bits d'entrée en tout, peuvent définir jusqu'à  $2^{300}$ , soit environ  $10^{90}$  cas différents ; pour fixer les idées, l'univers comporte environ  $10^{80}$  atomes. Le test exhaustif est donc irréalisable, quels que soient les moyens alloués.

Ainsi, les tests ne permettent pas de vérifier un système numérique de conception quelconque ou inconnue ; par contre, ils constituent un moyen de vérification pertinent dans le cadre d'une conception prévue à cet effet et permettant en particulier de regrouper les cas possibles en familles homogènes (voir les principes du sous-chapitre 6.4) ;

- la diversification : les défauts de logique ne peuvent pas être éliminés aisément, comme le confirment les nombreuses défaillances de produits industriels courants. De plus, aucune approche simple ne permet à ce jour de contourner leur présence et le recours à la diversification du logiciel (voir le paragraphe 6.5.4 et l'annexe 3) s'avère inadéquat. L'adjonction de systèmes de secours diversifiés ne peut pas être généralisée sans complexifier excessivement le contrôle-commande au risque d'introduire de nouveaux défauts et d'augmenter les risques d'actions intempestives (voir le sous-chapitre 6.5 et le chapitre 7) ; la diversification ne peut donc pas se substituer à la justification du bon fonctionnement des systèmes de contrôle-commande ;
- la prise en compte d'un taux de défaillance, qui s'avère inadéquate pour contourner la présence de défauts (voir l'annexe 4).

Une démarche de sûreté spécifique est donc nécessaire pour construire et démontrer le caractère correct de la logique d'un système de contrôle-commande.

## 5.3 PARTICULARITES EN MATIERE D'INDEPENDANCE

Le contrôle-commande fait l'objet d'exigences fonctionnelles qui peuvent a priori venir en contradiction avec la recherche d'indépendance entre classes de sûreté et entre niveaux de défense. Par exemple :

- la limitation des déclenchements intempestifs d'actions de sauvegarde nécessite un vote à partir d'informations élaborées dans des voies redondantes ; des communications entre voies « indépendantes » sont donc nécessaires ;
- pour simplifier la conception des systèmes fluides, certaines fonctions de contrôle-commande de classes de sûreté différentes ou appartenant à des niveaux de défense différents doivent piloter un même actionneur ; cette dépendance entre classes différentes ou entre niveaux de défense différents se retrouve alors inévitablement dans le contrôle-commande : par exemple, sur le parc en exploitation, l'alimentation de secours des générateurs de vapeur est utilisée par les opérateurs en fonctionnement normal, dans l'état d'arrêt à chaud, mais aussi par le système de protection en cas d'incident ;
- la conduite par le même moyen informatisé (MCP) dans toutes les situations est à privilégier pour des raisons liées à l'ergonomie. Ce moyen de conduite doit donc communiquer avec les systèmes de contrôle-commande utilisés dans différentes situations, ce qui induit des liens entre systèmes de classes de sûreté différentes ou participant à des niveaux de défense différents.

L'exigence d'indépendance entre classes de sûreté et entre niveaux de défense doit donc être précisée au cas par cas par le type d'indépendance requis : fonctionnelle (par exemple, absence d'échange de données ou de données provenant d'une source commune), matérielle (absence de composant matériel commun, tel qu'un processeur), physique (par exemple, absence de liaison électrique) ou logique (par exemple, absence de connexion à un même réseau, même si les calculateurs concernés n'échangent pas d'information à travers lui).

## 6 DEMARCHE ET PRINCIPES DE CONCEPTION

### 6.1 DEMARCHE GENERALE DE CONCEPTION DU CONTROLE-COMMANDE

Les particularités exposées ci-avant nécessitent la mise en œuvre d'une démarche spécifique afin notamment de prévenir les défaillances à toutes les étapes du processus de développement du contrôle-commande et de prendre en compte les défaillances résiduelles, en particulier de cause commune, de façon systématique afin d'en limiter les conséquences. Cette démarche repose sur la définition et le respect :

- de principes de spécification du contrôle-commande (voir le sous-chapitre 6.2) ;
- de principes de conception de l'architecture du contrôle-commande (voir le sous-chapitre 6.3) ;
- de principes de conception des systèmes de contrôle-commande (voir le sous-chapitre 6.4) :
  - la mise en œuvre d'un processus d'ingénierie strict ainsi que des principes techniques doivent permettre **d'éviter l'introduction de défauts** durant tout le cycle de vie de chaque système de contrôle-commande (voir le paragraphe 6.4.1),
  - cette démarche d'évitement doit être complétée par une approche méthodique de **suppression des défauts**, incluant des procédures non formelles (comme des inspections, des relectures, des

audits, des revues), des procédures formelles (raisonnements mathématiques visant à prouver le respect de propriétés attendues, l'absence de défauts prédéfinis, l'équivalence avec un modèle réputé correct, etc.) et des essais de validation dont la couverture doit être justifiée (voir le paragraphe 6.4.2),

- de plus, le système doit **tolérer des défauts résiduels** qui subsisteraient malgré ce qui précède (voir le paragraphe 6.4.3).

La démarche de prise en compte des défauts résiduels au niveau d'un système doit être complétée par des dispositions d'architecture du contrôle-commande incluant la diversification (voir le sous-chapitre 6.5), sachant que ces défauts peuvent également se manifester par des actions intempestives, qui doivent être analysées (voir le chapitre 7).

Enfin, toutes les phases du développement du contrôle-commande doivent également prendre en compte la protection contre les actes de malveillance (voir le sous-chapitre 6.6).

Dans le cas du réacteur EPR Flamanville 3, des choix fondamentaux ont été effectués dès les premières étapes de la conception du contrôle-commande afin de tenir compte des principes issus des enseignements apportés par les précédentes conceptions et leurs évaluations par l'IRSN :

- la suppression de certaines fonctions des outils de diagnostic, dont l'interaction avec le SAS (Système d'automatisme de sûreté) aurait pu empêcher de démontrer le respect des exigences de temps de réponse de ce système ;
- l'introduction d'un réseau interne au SAS pour pouvoir démontrer que les communications entre ses unités s'effectuent toujours dans un délai connu, condition indispensable à la démonstration du respect des exigences de temps de réponse global du SAS ;
- une refonte de l'architecture des outils d'ingénierie, pour pouvoir démontrer l'absence d'influence excessive de leur part sur le fonctionnement du SAS et du Système d'automatisme de tranche (PAS) et ainsi éliminer d'éventuelles DCC dues à ce facteur commun ;
- la limitation des débits de certains calculateurs sur les réseaux, de façon à éliminer des DCC dues au « bavardage » d'un calculateur sur un réseau, qui empêcherait d'autres calculateurs d'y accéder ;
- l'introduction d'un bouton matériel de validation des commandes de permissifs émises par le MCP (Moyen de Conduite Principal, informatisé) vers le PS, pour empêcher la propagation de défaillances du MCP vers le PS ;
- la création d'un moyen de surveillance permettant de détecter les défaillances du MCP et d'alerter les opérateurs de la nécessité de passer au MCS (Moyen de Conduite de Secours, non informatisé), avec une confiance correspondant au classement de ce dernier.

## 6.2 SPECIFICATION DU CONTROLE-COMMANDE

Le contrôle-commande doit être développé à partir d'exigences écrites, complètes (chaque aspect important doit être traité), claires (compréhensibles par une personne possédant la connaissance générale du domaine mais pas nécessairement celle du projet particulier) et précises (sans ambiguïté). Elles doivent en particulier couvrir :

- la description détaillée de chaque fonction à réaliser, y compris la précision, le temps de réponse, les modes dégradés (par exemple le calcul du RFTC avec un ou plusieurs capteurs détectés défaillants), les positions de repli en cas de détection de défaillance, etc. ;
- les interfaces entre fonctions ;
- les principes de priorité entre fonctions agissant sur un même actionneur ou sur des actionneurs fonctionnellement liés ;
- les classes de sûreté des fonctions, qui induisent des exigences de conception, de fabrication et d'exploitation des systèmes qui les réalisent ;
- l'indépendance des fonctions, par exemple lorsqu'elles participent à des niveaux de défense différents ;
- les contraintes imposées par la conception de l'installation, par exemple les interfaces avec les équipements électromécaniques et les systèmes supports, les plages d'évolution des signaux, la localisation des équipements et le cheminement des câbles ;
- les contraintes d'exploitation et de maintenance ;
- les conditions d'ambiance et les agressions à considérer, ainsi que les exigences de qualification correspondantes.

Ces exigences doivent être cohérentes avec celles issues des études de sûreté nucléaire.

## 6.3 CONCEPTION DE L'ARCHITECTURE

Le contrôle-commande doit être organisé en systèmes, en veillant en particulier à :

- définir des systèmes fonctionnellement cohérents, de façon à limiter les communications entre systèmes ;
- vérifier que le nombre de systèmes ou de sous-systèmes redondants permet de satisfaire le critère de défaillance unique lorsqu'il s'applique ;
- respecter les exigences induites par la classe de sûreté de chaque fonction, ainsi que les exigences de séparation et d'indépendance des fonctions relevant de niveaux de défense différents ;
- réaliser la transmission de signaux entre systèmes ainsi que les votes et priorités entre leurs sorties conformément aux exigences de séparation et d'indépendance ;
- organiser les interactions entre les hommes et les machines (moyens de conduite, élaboration des alarmes, outils de diagnostic et de maintenance, etc.) ;
- exploiter la signalisation des défaillances pour informer les opérateurs des indisponibilités de systèmes et donner les moyens de basculer, automatiquement ou manuellement, sur des systèmes de secours ;
- spécifier les exigences des différents systèmes, leurs interfaces et les communications entre eux, de sorte que leur fonctionnement combiné respecte les exigences de traitements à réaliser, de précision, de temps de réponse, etc. (voir le sous-chapitre 6.2).

L'architecture doit être conçue selon une démarche d'ingénierie planifiée, accordant une attention particulière aux activités de vérification. Les exigences imposées à chaque système doivent comporter l'utilisation d'un

processus d'ingénierie adéquat et, en fonction de sa classe de sûreté, l'utilisation de techniques favorisant leur fonctionnement correct et leur vérification (voir le sous-chapitre 6.4).

Une architecture idéale éviterait toute communication entre niveaux de défense différents et entre systèmes de classes de sûreté différentes, et utiliserait des solutions diversifiées pour les différents niveaux de défense. Toutefois, cet idéal n'est pas atteignable à cause des contraintes fonctionnelles imposées au contrôle-commande (par exemple celles présentées au sous-chapitre 5.3) et des possibilités limitées de diversification technologique : il n'existe que très peu de solutions industrielles pouvant réellement être classées de sûreté ; de plus, multiplier les technologies (donc les outillages et procédures de conception et d'exploitation, ainsi que les interfaces avec les utilisateurs) complexifie la conception et l'exploitation, ce qui peut affecter la sûreté (voir le paragraphe 6.5.2 et l'annexe 2).

Le concepteur doit donc faire des choix et les justifier (voir [15]), en particulier par rapport à l'objectif de prévention des DCC entre systèmes ou sous-systèmes redondants ou accueillant des fonctions indépendantes.

Les DCC dues à la propagation d'une défaillance ou à une interaction inadéquate entre systèmes doivent être prévenues, du point de vue matériel, par la séparation physique et électrique des entités concernées et de leurs systèmes supports (notamment leurs alimentations électriques). Cela implique de découpler électriquement les communications, par exemple avec des transmissions par fibre optique. Du point de vue de la logique, chaque interaction doit être analysée sous ses aspects :

- fonctionnel, par l'analyse du rôle des données échangées, qui concerne également les domaines en amont du contrôle-commande ;

L'analyse du rôle fonctionnel de chaque donnée reçue par un système permet de déterminer les conséquences possibles d'une valeur erronée de cette donnée, due à la défaillance de l'entité émettrice ou de la transmission ; si ces conséquences ne sont pas acceptables, le concepteur doit typiquement avoir recours à des redondances et à des votes pour garantir un résultat correct.

- technologique, par l'analyse des protocoles de communication et de leur influence sur le fonctionnement des calculateurs ; la nécessité de maîtriser cette influence contraint la conception des systèmes (voir le sous-chapitre 6.4 et l'annexe 1) et peut empêcher d'utiliser des automates industriels courants ; en effet, la conception de ces automates tend à généraliser les interconnexions pour augmenter la flexibilité, ce qui complique l'analyse et peut même la rendre irréalisable.



Dans les systèmes industriels usuels, lorsque plusieurs unités sont reliées par un réseau, la défaillance d'une unité peut perturber les autres même si elles n'échangent pas d'informations ; par exemple, si une unité défaillante « bavarde » en permanence sur le réseau, les autres ne peuvent plus ni émettre ni recevoir des informations par ce réseau. Dans les systèmes de sûreté, le concepteur doit donc utiliser un plus grand nombre de réseaux, séparés les uns des autres, pour limiter les conséquences d'une défaillance.

Entre deux unités qui dialoguent, les protocoles de communication courants peuvent conduire au blocage d'une unité saine si elle attend indéfiniment une information qui n'arrivera jamais lorsque l'unité émettrice est défaillante. Typiquement, dans un système de protection en redondance d'ordre quatre, quatre unités calculent une condition d'arrêt automatique et un voteur reçoit les informations des quatre unités via quatre réseaux séparés. Cette architecture est fiable, mais il faut éviter que le voteur n'attende en permanence les informations des quatre unités et ne se bloque si l'une d'entre elles n'arrive pas. Les systèmes de sûreté nécessitent donc des protocoles de communication particuliers, garantissant la non-propagation des défaillances.

Les DCC dues à la coïncidence de défaillances doivent être prévenues, en ce qui concerne les matériels, par leur fiabilité, leurs essais périodiques de fréquences cohérentes avec leurs probabilités de défaillance et leur aptitude au fonctionnement dans les conditions d'environnement spécifiées. En ce qui concerne la logique, aucun moyen reconnu ne permet d'évaluer ni même de définir convenablement la probabilité de la présence de défauts (voir l'annexe 4) : cet aspect ne peut donc être traité que de façon qualitative. La conception des systèmes doit donc assurer qu'un défaut résiduel potentiellement commun à plusieurs entités ne peut pas y être activé simultanément (voir le sous-chapitre 6.4) ; concernant l'architecture, le concepteur doit identifier les causes résiduelles plausibles de DCC, postuler leur existence et limiter leurs conséquences au moyen de fonctions ou de systèmes diversifiés (voir le sous-chapitre 6.5). L'identification des causes résiduelles de DCC de plusieurs systèmes doit tenir compte de leurs principes de fonctionnement et de communication (par exemple cyclique ou à la demande), des langages et outils informatiques utilisés et de leurs principaux composants matériels et logiciels (processeurs, interfaces avec les réseaux, bibliothèques, etc.).

Un plan de mise en service doit permettre de s'assurer de la bonne intégration sur site des systèmes de contrôle-commande validés séparément en usine. Ce plan doit justifier la capacité des essais prévus à solliciter suffisamment les systèmes pour confirmer qu'ils fonctionnent correctement sur site et interagissent, entre eux et avec l'installation, conformément aux exigences spécifiées.

## 6.4 CONCEPTION D'UN SYSTEME

### 6.4.1 EVITEMENT DES DEFAUTS

La conception doit éviter d'introduire des défauts, en respectant des principes (détaillés dans les documents cités au chapitre 3) :

- d'organisation du développement du système en phases bien identifiées, ayant des documents d'entrée et de sortie définis au préalable, faisant l'objet de revues et de vérifications systématiques ; les phases

doivent être en nombre suffisant pour limiter l'étendue des activités effectuées dans chacune, afin de réduire les risques d'erreur et de faciliter la vérification (voir la figure 1 ci-après) ;

- d'identification systématique des exigences (fonctionnelles et non fonctionnelles) ainsi que de vérification de leur cohérence avec celles de l'installation ;
- de conception pour structurer la logique, exclure les constructions propices aux défauts et faciliter la vérification par des analyses manuelles ou automatiques ;
- de documentation et de justification systématiques des choix de conception ;
- de gestion des configurations des produits techniques et de la documentation des activités, pour assurer leur disponibilité et le contrôle de leurs modifications.

La figure 1 illustre l'organisation typique des phases, dite « cycle en V » ; la phase de conception est en fait divisée en autant de sous-phases que nécessaire pour limiter l'étendue des activités effectuées dans chacune. Ces activités visent à diviser le problème initial en sous-problèmes suffisamment simples, affectés à des composants matériels ou logiciels.

Ces composants sont ensuite réalisés (en bas du cycle en V) : il s'agit par exemple de la programmation des composants logiciels. Dans la branche remontante du cycle en V, les composants sont progressivement assemblés (« intégration ») en s'assurant qu'ils interagissent comme prévu lors de la conception. L'ordre dans lequel les composants matériels et logiciels sont intégrés dépend en général de leurs interactions et donc de la conception.

Toutes les phases qui précèdent sont vérifiées, comme l'indique la figure 1, et le système entièrement intégré fait l'objet d'une validation (voir le paragraphe 6.4.2).

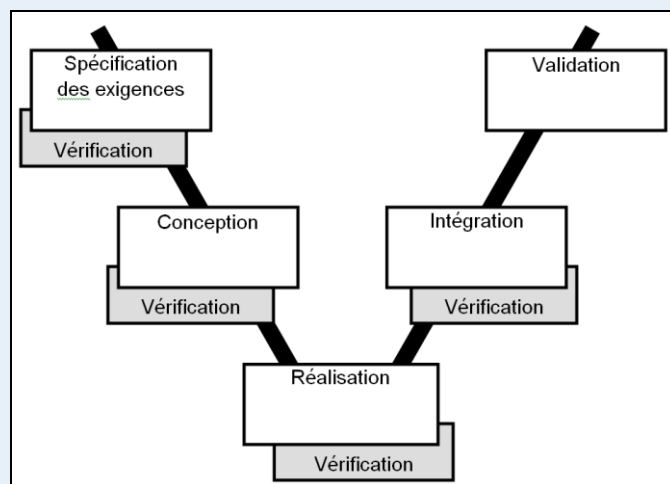


Figure 1 : principe d'un cycle de développement « en V », d'après la CEI

Des modifications peuvent être effectuées, par exemple en cas d'évolution des exigences fonctionnelles ou de découverte d'une anomalie. Toute modification doit être faite en respectant une procédure comportant la formalisation de la demande, l'évaluation de sa pertinence et l'analyse de son impact sur les documents, sur le logiciel, sur le reste du système et sur ses performances. Si la demande est acceptée, le processus de développement doit être repris à partir de la première phase concernée par la modification, en respectant les mêmes principes et exigences détaillées de conception, de vérification, de documentation et de gestion des

configurations que lors du développement initial. Les documents modifiés doivent faire référence à la demande de modification et un document doit synthétiser les actions effectuées.

Les principes concernant le processus sont complétés par d'autres portant sur le produit lui-même. Pour les systèmes classés au plus haut niveau de sûreté (1E dans le cas du palier N4, F1A dans le cas du réacteur EPR), le concepteur doit en particulier démontrer que, conformément à la RFS II.4.1.a [1], chaque calculateur fonctionne de façon déterministe (le cycle de calcul et l'utilisation des ressources telles que la mémoire, les réseaux, les entrées et les sorties, doivent être déterminés lors de la conception) et ses résultats ne dépendent d'aucun facteur autre que les entrées spécifiées. En pratique, ce principe contraint fortement la conception et rend la vérification plus efficace.

Dans le domaine de la bureautique, le comportement d'un logiciel donné dépend souvent d'une base de données interne à l'ordinateur, qui est modifiée en permanence par d'autres logiciels et par les actions de l'utilisateur, ce qui rend chaque ordinateur unique. Ainsi, le comportement d'un tel logiciel dépend d'un environnement instable, ce qui diminue drastiquement la représentativité des tests (puisque'une même action peut avoir des conséquences différentes, en fonction d'un contexte non maîtrisable) et provoque finalement la plupart des dysfonctionnements constatés dans ce domaine.

A contrario, le déterminisme implique d'effectuer toujours les mêmes calculs, à partir des mêmes entrées clairement spécifiées et en utilisant exclusivement les ressources prévues et allouées à la conception ; cette propriété permet d'analyser les logiciels indépendamment de leur contexte et d'élaborer des tests représentatifs.

De plus, les principes suivants concernent plus particulièrement l'évitement des DCC (ils sont présentés plus précisément dans l'annexe 1 et détaillés dans les normes [10] et [13]) :

- la défaillance d'un calculateur (due à un défaut résiduel ou à une panne matérielle) ne doit pas pouvoir se propager à des calculateurs redondants ou indépendants, même s'il communique avec eux ; pour cela, ils doivent être mutuellement asynchrones, sans jamais attendre d'action d'un autre pendant un temps supérieur à une limite prise en compte dans la démonstration de déterminisme ;

Sous cette condition, les données reçues par un calculateur peuvent être erronées ou absentes en cas de défaillance de l'émetteur, mais son cycle de calcul n'est pas affecté ; il peut alors traiter ce cas de façon fonctionnellement correcte, par exemple au moyen d'entrées redondantes et de votes.

- aucune condition de fonctionnement ne doit pouvoir simultanément activer un éventuel défaut dans plusieurs calculateurs redondants ou indépendants ; en particulier, l'enchaînement des opérations et les besoins en ressources ne doivent dépendre ni de l'état ou des transitoires du procédé, ni d'informations explicites ou implicites pouvant être communes à plusieurs calculateurs, comme la date calendaire (« bug de l'an 2000 ») ou la durée écoulée depuis leur démarrage.

Par exemple, si la durée écoulée depuis le démarrage intervenait dans un traitement non fonctionnel (relatif à l'autosurveillance, à la maintenance, etc.) un défaut (tel que le « débordement » non prévu d'un compteur de temps après une certaine durée) pourrait provoquer la DCC de calculateurs indépendants exécutant des fonctions de contrôle-commande différentes.

De même, si la charge des réseaux ou les besoins en ressources de calcul variaient en fonction des transitoires du procédé, une DCC pourrait survenir lors de transitoires non prévus.

*La RFS II.4.1.a [1] demande que les systèmes classés au deuxième niveau de sûreté (2E dans le cas du palier N4, F1B dans le cas du réacteur EPR) soient « prédictibles » ; ce principe laisse davantage de liberté au concepteur que celui de déterminisme présenté ci-avant, en permettant des variations du cycle de calcul ou de l'utilisation des ressources (telles que la mémoire, les réseaux, les entrées et les sorties), à condition que les résultats soient conformes aux exigences fonctionnelles dans toute la plage de ces variations ; en pratique, cela nécessite que la conception soit suffisamment maîtrisée pour permettre de déterminer cette plage.*

Tous les facteurs (entrées fonctionnelles et ressources) dont dépend un calcul peuvent être identifiés : le respect du principe de déterminisme est donc vérifiable. Cependant, le déterminisme (et dans une moindre mesure la « prédictibilité ») contraignent fortement la conception, conduisent à sous-utiliser les ressources et nécessitent un important travail de justification, de sorte qu'ils sont exigés uniquement dans les secteurs soumis à une réglementation technique très contraignante, comme le nucléaire et l'avionique.

Du point de vue de l'évaluateur, cela implique que les produits acceptés dans d'autres industries ne sont pas acceptables sans démonstration complémentaire pour les systèmes classés de sûreté : ils sont, jusqu'à preuve du contraire (et de fait souvent), fondés sur une conception ni déterministe ni prédictible, et par conséquent non vérifiable. En effet, le très grand nombre de cas possibles pour une logique même simple (voir le paragraphe 5.2.3) affaiblit les justifications empiriques fondées sur le retour d'expérience : même en observant des millions d'exemplaires d'un système pendant des millions d'années, la proportion de cas possibles couverte resterait très faible et sa représentativité pour une utilisation différente serait incertaine<sup>2</sup>, faute de maîtrise suffisante de la conception. Cela confirme la nécessité d'une démarche spécifique pour concevoir les systèmes de contrôle-commande de sûreté nucléaire.

Les principes de conception qui précèdent sont déclinés en exigences plus détaillées dans les textes normatifs tels que [7], [10], [11], [12] et [13] (voir l'annexe 1). Ces textes donnent également des informations au niveau encore plus fin des techniques de réalisation, qui peuvent être adéquates dans certains cas mais pas dans d'autres en fonction de détails dont la discussion sort du cadre du présent document (voir, pour illustration, le cas des « logiciels à interruptions » dans l'annexe 5).

---

<sup>2</sup> L'échec du tir de la fusée Ariane 501 est typique : sa centrale inertielle, développée pour Ariane 4 selon de bonnes pratiques industrielles, fonctionnait parfaitement dans ce lanceur malgré la présence d'un défaut qui ne se manifestait pas dans ce contexte ; le profil de vitesse différent d'Ariane 5 a conduit de façon indirecte mais déterministe à l'activation du défaut et à l'échec du tir.

## 6.4.2 SUPPRESSION DES DEFAUTS

Les défauts qui auraient malgré tout été introduits dans les logiciels doivent être recherchés et éliminés en respectant les principes suivants :

- pour les systèmes classés au plus haut niveau de sûreté (1E dans le cas du palier N4, F1A dans le cas du réacteur EPR), les produits de chaque phase de développement d'un logiciel doivent être vérifiés par une équipe indépendante de celle chargée de la conception ; pour les autres classes de sûreté, les principaux produits doivent être vérifiés par des personnes n'ayant pas participé à leur élaboration (des vérifications croisées au sein d'une équipe unique sont donc admises dans ce cas) ;
- les activités, moyens et objectifs de la vérification doivent être déterminés et justifiés à l'avance ; la vérification doit être fondée sur un ensemble cohérent d'analyses de documents, de tests et d'analyses du logiciel ; des revues techniques doivent être menées pour faire le bilan des activités de conception et de vérification accomplies et conclure formellement quant à leur acceptation ;
- le produit final doit être validé par des tests montrant qu'il fonctionne conformément à ses exigences ; pour la plus haute classe de sûreté, ces tests doivent être élaborés par l'équipe de vérification indépendante et leur capacité à solliciter pleinement le produit final doit être démontrée.

### Stratégie de test

La justification de l'adéquation des tests est un sujet difficile. En effet, les logiciels de sûreté sont affectés de peu de défauts, contrairement aux logiciels courants, pour lesquels des tests définis sans méthode rigoureuse permettent d'en détecter ; l'adéquation des tests des logiciels de sûreté ne peut donc pas être justifiée par le nombre de défauts détectés, mais uniquement par une analyse de leur capacité à détecter d'éventuels défauts.

Cette justification nécessite que les exigences du logiciel définissent un nombre limité de familles de cas d'exécution (par exemple, l'exigence « si la pression est supérieure à 150 bars, déclencher l'AAR » définit les deux familles « pression supérieure à 150 bars » et « pression non supérieure à 150 bars ») et que la conception garantisse l'unicité du traitement pour tous les cas d'une famille donnée ; sinon, les cas ne peuvent pas être regroupés en familles homogènes et la vérification est impossible.

Des cas de test en nombre limité, par exemple un cas par famille et des cas aux frontières entre familles, peuvent alors être suffisamment représentatifs : pour l'exemple précédent, des cas avec les valeurs de pression 75, 149, 150, 151 et 160 bars pourraient ainsi être choisis. Les tests doivent être élaborés par l'équipe de vérification et les résultats attendus doivent être déterminés avant l'exécution des tests.

Dans ces conditions, l'exécution des tests avec des résultats conformes à l'attendu permet de confirmer que les traitements des familles de cas spécifiées par les exigences sont correctement réalisés. Si les résultats ne sont pas conformes à l'attendu, une analyse doit être effectuée pour rechercher toutes les causes du défaut, y compris dans le processus de développement, afin de les éliminer.

Cependant, la réalisation pourrait comporter des traitements non spécifiés par les exigences, pouvant produire des résultats incorrects. Par exemple, la réalisation pourrait utiliser dans les calculs la valeur absolue d'une entrée au lieu de l'entrée elle-même ; le résultat serait alors erroné lorsque cette entrée est négative, et les tests pourraient ne comporter aucun cas de cette famille puisque qu'elle n'est pas spécifiée par les exigences. Ces tests ne permettraient donc pas nécessairement de détecter le défaut.

Pour faire face à cette éventualité, l'équipe de vérification doit analyser la « couverture structurelle » des tests, c'est-à-dire leur capacité à solliciter les éléments de la structure interne du logiciel concerné tels que les traitements ou les variables. Dans l'exemple mentionné ci-avant, cette analyse détecterait la présence d'un traitement de valeur absolue, tel que « changer le signe de l'entrée si elle est négative », non sollicité par les tests puisque ceux-ci ne comportent pas de cas avec l'entrée négative. La suite de l'analyse montrerait que la présence de ce traitement constitue un défaut. Dans d'autres situations, l'analyse pourrait montrer que le traitement non sollicité est spécifié par les exigences mais que les tests sont incomplets, ce qui conduirait à réexaminer la méthode d'élaboration des tests.

Ainsi, l'analyse de la capacité des tests à solliciter la structure du logiciel concerné permet, s'ils ont été élaborés sans connaître cette structure, de confirmer l'absence de traitement non spécifié par les exigences<sup>3</sup>.

La double confirmation d'une réalisation correcte des traitements spécifiés et de l'absence de traitement non spécifié constitue un résultat essentiel pour la vérification, à condition que les tests soient élaborés à partir des exigences, avant l'analyse de leur couverture structurelle. Les tests ne doivent donc pas être élaborés en connaissant la structure du logiciel concerné et en visant par exemple à solliciter les traitements effectivement présents : cette démarche (dite « test structurel ») est souvent adoptée dans l'industrie car elle nécessite peu d'effort, mais elle n'est pas acceptable pour des systèmes classés de sûreté car elle revient à contrôler un produit par rapport à lui-même et non par rapport à ses exigences.

L'analyse de la couverture structurelle nécessite de décider quels types d'éléments de la structure doivent être sollicités par les tests élaborés au préalable ; en effet, les types (instructions, traitements conditionnels, traitements entre l'écriture et l'utilisation de chaque donnée, etc.) pertinents pour cette analyse dépendent des choix de conception et de programmation effectués pour le logiciel concerné et ne peuvent donc pas être identifiés une fois pour toutes. Les types retenus pour le logiciel concerné doivent donc être documentés et justifiés.

Le respect des conditions mentionnées ci-avant nécessite une grande rigueur ; mais, sous cette condition, les tests constituent un moyen de vérification pertinent et incontournable. La norme [10] détaille les principes décrits dans ce paragraphe et contient une annexe illustrant les domaines d'application de différentes techniques de test.

### **Vérification formelle**

Une autre approche de vérification, en cours de développement, repose sur le constat que la logique d'un système numérique est un système d'équations entre les entrées et les sorties, donc un objet mathématique ; comme pour un théorème, démontrer ses propriétés se fait essentiellement en raisonnant sur des ensembles de cas et non en tentant de les examiner individuellement. Ce constat constitue un des fondements de la vérification formelle, qui consiste à analyser un programme, aussi automatiquement que possible, pour démontrer qu'il possède certaines

---

<sup>3</sup> La présence de traitements non spécifiés par les exigences peut toutefois être justifiée, par exemple dans le cas de l'utilisation de bibliothèques préexistantes. Ainsi, un logiciel nécessitant la fonction « sinus » peut utiliser une bibliothèque de fonctions trigonométriques incluant d'autres fonctions telles que « tangente » : le logiciel complet inclut alors des fonctions non spécifiées par les exigences, ce qui peut être acceptable si la bibliothèque incluse avait été convenablement développée et vérifiée.

propriétés, par exemple qu'une certaine relation entre entrées et sorties est toujours vraie ou qu'une variable constituant le dénominateur d'une division n'est jamais nulle, quelles que soient les entrées du programme. Malgré certaines limitations théoriques<sup>4</sup> et pratiques<sup>5</sup>, cette approche a déjà été utilisée avec succès par EDF pour les systèmes de protection du réacteur EPR et des réacteurs de 1300 MWe à l'occasion de la rénovation associée à leur troisième visite décennale. Sa mise en œuvre doit être clairement spécifiée pour que sa portée et sa complémentarité avec les autres moyens de vérification puisse être évaluée.

### **6.4.3 TOLERANCE D'UN SYSTEME AUX DEFAUTS RESIDUELS**

Malgré toutes les précautions prises pour éviter les défauts de logique et pour fiabiliser les matériels, des défaillances peuvent se produire. Elles doivent autant que possible être prises en compte dans le système concerné par :

- des redondances au regard des défaillances du matériel ;
- des mécanismes d'autosurveillance détectant des comportements anormaux (voir le sous-chapitre 5.1) sans complexifier excessivement la conception, ce qui implique de ne pas chercher à identifier la cause de l'anomalie dans le système de sûreté lui-même ; cette recherche peut être effectuée par un système de diagnostic externe, recevant des informations du système de sûreté sans lui en transmettre ;
- la définition préalable, fonctionnellement justifiée, des valeurs que les sorties du système doivent prendre lorsqu'un comportement anormal est détecté par les mécanismes d'autosurveillance.

## **6.5 PRISE EN COMPTE DANS L'ARCHITECTURE DES DEFAILLANCES POSTULEES - DIVERSIFICATION**

### **6.5.1 PRINCIPES GENERAUX**

Malgré le respect des principes décrits au sous-chapitre 6.4 pour éviter les défaillances des systèmes, de telles défaillances doivent être postulées et des dispositions d'architecture du contrôle-commande doivent les prendre en compte, en se limitant aux défaillances plausibles pour éviter une complexité excessive : par exemple, la DCC de tous les systèmes numériques de sûreté n'est pas plausible au seul titre de leur caractère numérique s'ils n'échangent pas d'information et utilisent des calculateurs de types différents, conformes aux principes de conception permettant de montrer que le comportement de chacun est correct et indépendant des autres et de l'environnement. Dès lors, le concepteur doit postuler :

- l'existence d'un défaut dans la logique réalisant une fonction de contrôle-commande, à cause d'une exigence fonctionnelle erronée ou d'une erreur de conception ; ce défaut pourrait rendre cette fonction simultanément défaillante dans les voies redondantes ;

---

<sup>4</sup> Par exemple l'indécidabilité de certaines propriétés, au sens logique (la théorie mathématique utilisée ne permet de démontrer ni qu'elles sont justes, ni qu'elles sont fausses) ou au sens algorithmique (il n'existe pas de programme capable de faire la démonstration en un temps fini).

<sup>5</sup> Par exemple la grande taille des équations qui représentent un logiciel donné. Les logiciels de sûreté conçus selon les principes du présent document facilitent l'utilisation des méthodes formelles, car la relative simplicité de leurs exigences fonctionnelles et la régularité de leur conception limitent la taille de ces équations.

- l'existence d'un défaut dans la logique réalisant les fonctions de base d'un type de calculateurs (par exemple les calculateurs du PS), conduisant à une DCC de plusieurs calculateurs de ce type (bien que les mécanismes pouvant déclencher simultanément ou propager des défaillances aient été éliminés par l'application des principes de conception du paragraphe 6.4.1).

La tolérance à ces défauts postulés repose sur l'ajout de systèmes de secours adéquatement diversifiés.

### **6.5.2 IMPACT DE LA DIVERSIFICATION SUR LA COMPLEXITE DE L'ARCHITECTURE**

Les défauts potentiels de la logique d'un système de contrôle-commande provenant principalement d'un excès de complexité, la démarche de diversification doit éviter l'excès de systèmes de secours, dont la présence pourrait rassurer dans une représentation simplifiée du contrôle-commande, mais qui le complexifieraient en réalité, au risque d'introduire des défauts dans une architecture ou dans des systèmes jusqu'alors corrects. Ainsi :

- un nombre accru de matériels et de types de matériels augmente les difficultés de maintenance ; or, les erreurs de maintenance constituent en pratique la principale source de DCC des systèmes de sûreté numériques, loin devant les défauts de logique (voir l'annexe 2) ;
- l'ajout d'interfaces différentes avec les opérateurs, découlant de l'adjonction de systèmes différents, peut augmenter le risque d'erreur en exploitation, particulièrement dans les situations difficiles ;
- un actionneur piloté par plusieurs systèmes (le système normal et le système de secours) peut recevoir des commandes contradictoires en cas de défaillance ; ce conflit doit être résolu par la logique de pilotage de l'actionneur, qui constitue un point unique de défaillance de l'architecture ; ceci augmente la complexité de cette logique, au risque d'introduire des défauts.

Toutes les conséquences de l'ajout de systèmes doivent donc être examinées, afin de déterminer une solution satisfaisante pour la sûreté.

### **6.5.3 APPLICATION AU CAS DU REACTEUR EPR FLAMANVILLE 3**

Dans ce cadre, le concepteur du réacteur EPR Flamanville 3 a notamment mis en place plusieurs diversifications (voir la figure 2 ci-après).

Une diversification fonctionnelle au sein du PS permet de pallier un défaut qui affecterait une fonction de protection au moyen d'autres fonctions, en permettant d'atteindre les mêmes objectifs au moyen de signaux physiques et de traitements différents. Par exemple, la fonction « d'AAR sur bas niveau dans un générateur de vapeur » est diversifiée par la fonction « d'AAR sur bas RFTC ». Cette diversification est effective car ces fonctions sont indépendantes malgré leur implantation dans un même système. Cette indépendance repose sur le déterminisme du fonctionnement (voir le paragraphe 6.4.1) : les résultats de chaque fonction ne dépendent que des ressources qui lui sont attribuées et de ses entrées spécifiées, et les fonctions diversifiées n'ont pas de ressource ou d'entrée spécifiée en commun ; elles sont donc indépendantes. La démonstration de séparation des ressources attribuées aux fonctions diversifiées a été facilitée en les implantant dans des calculateurs séparés.

Une diversification technologique permet de pallier une défaillance de cause commune des calculateurs du PS (réalisé en technologie Teleperm XS) dans les situations dont l'importance a notamment été mise en évidence par les études probabilistes de sûreté ; il s'agit de fonctions de protection en cas d' « ATWS » (Anticipated Transient



Without Scram) implantées dans le SAS, réalisé dans une technologie (SPPA-T2000) différente de celle du PS pour ce qui concerne les mécanismes de fonctionnement, les composants matériels et logiciels, les langages, les protocoles de réseau, etc.

Une autre diversification technologique permet de pallier une défaillance de cause commune des calculateurs du SAS : de façon symétrique à la précédente, les fonctions du CCND (Contrôle-commande noyau dur, réalisé en technologie Teleperm XS) diversifient certaines fonctions du SAS réalisé en technologie SPPA-T2000.

Une troisième diversification technologique permet de pallier une défaillance de cause commune des calculateurs du MCP (informatisé) par le MCS (non informatisé).

Des liaisons directes entre la salle de commande et certains capteurs et actionneurs complètent ces diversifications.

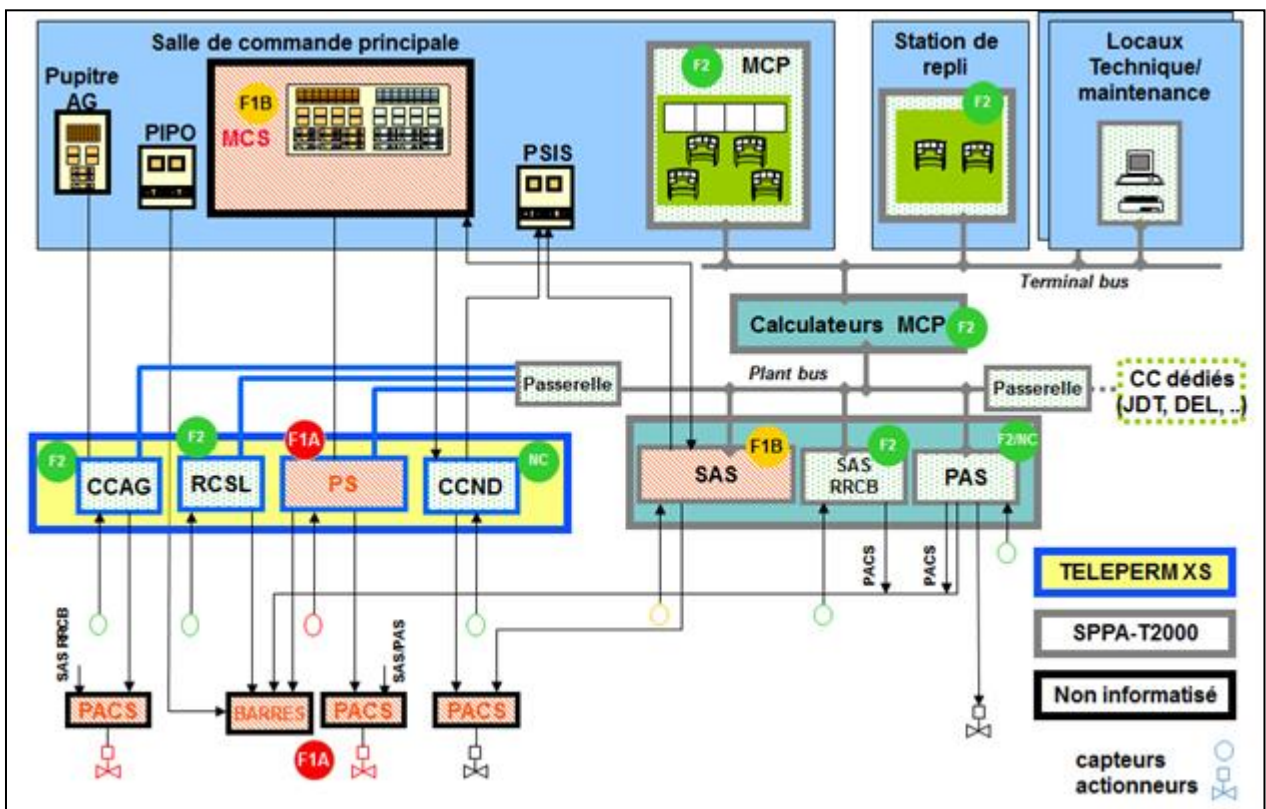


Figure 2 : schéma d'architecture du contrôle-commande du réacteur EPR Flamanville 3

#### 6.5.4 INEFFICACITE DE LA DIVERSIFICATION D'UN LOGICIEL

La diversification d'un logiciel a été proposée comme moyen de pallier les conséquences de défauts logiques. Cette approche consiste à exécuter en parallèle plusieurs programmes (« versions ») réalisant la même fonction mais développés indépendamment, en faisant l'hypothèse que leurs défauts éventuels ne concernent pas les mêmes cas et peuvent donc être masqués par un vote approprié.

Knigh et Leveson [16] ont expérimenté cette approche avec 27 versions d'une même fonction, dans des conditions favorables à la diversification (voir l'annexe 3). Pourtant, le nombre de cas où plusieurs versions ont simultanément fourni des résultats erronés s'est avéré incompatible avec l'hypothèse d'indépendance statistique entre défauts qui motivait l'approche, conduisant avec 99% de confiance au rejet de cette hypothèse.

**La diversification d'un logiciel ne peut donc pas, sans autre justification, être considérée comme une parade à la présence d'hypothétiques défauts.**

Ainsi, une exigence de diversification généralisée des logiciels pourrait dégrader la qualité de chaque version de ces logiciels par la dilution des moyens, nécessiterait une logique de décision entre versions elle-même susceptible d'être affectée de défauts et, plus généralement, complexifierait le contrôle-commande dans l'espoir d'un gain non confirmé par l'expérience.

## 6.6 PRISE EN COMPTE DE LA MALVEILLANCE

Certains principes de conception des systèmes de sûreté présentés dans ce document favorisent la protection contre la malveillance, par exemple :

- les principes de conception simple et justifiée, de vérification indépendante et de gestion rigoureuse des configurations compliquent l'introduction et la dissimulation de logiciels malveillants ;
- le fait de chercher à éviter les défauts dans les logiciels mis en œuvre empêche leur exploitation pour réaliser des attaques visant à provoquer des fonctionnements anormaux (exécution de programmes non prévus, accès à des ressources normalement inaccessibles, etc.).

Par exemple, lorsque l'utilisateur est invité à saisir une information (nom, adresse, etc.), celle-ci est écrite dans une zone mémoire, souvent de taille fixée (par exemple 500 caractères) ; si l'information saisie excède cette taille et si le programme ne se protège pas contre ce cas (ce qui constitue un défaut), l'écriture peut déborder sur les zones mémoires voisines ; selon le rôle de ces zones (par exemple désigner le traitement suivant à effectuer), cela peut conduire à exécuter des fonctions non prévues par le concepteur. Ce type d'attaque ne nécessite pas de compétence dans des domaines difficiles comme la cryptographie et, bien qu'ancien, reste efficace dans la bureautique et les secteurs industriels courants parce que leurs logiciels sont affectés de nombreux défauts.

Cependant, la prise en compte des actes de malveillance peut influencer le choix des organisations à mettre en place tout au long des différentes phases de développement et d'exploitation, ainsi que des architectures, des composants, des logiciels et de leur localisation, afin d'augmenter la robustesse intrinsèque du contrôle-commande. Aussi, dans toutes ses phases, la conception doit prendre en compte la protection contre les actes de malveillance. Au-delà des choix précités, des dispositions spécifiques doivent être prévues pour détecter, empêcher ou pour le moins retarder et minimiser les conséquences d'une action de malveillance et ainsi préserver la disponibilité et l'intégrité des fonctions du contrôle-commande.

Par exemple, la conception doit empêcher l'accès aux systèmes depuis l'extérieur de l'installation, interdire techniquement l'accès physique aux équipements sensibles aux personnes qui n'y sont pas autorisées et interdire autant que possible les modifications inappropriées de programmes ou de données.

Ces moyens spécifiques nécessaires à la protection contre les actes de malveillance peuvent augmenter la complexité du contrôle-commande et introduire des risques nouveaux de défauts préjudiciables à la fonctionnalité des systèmes. Il convient donc de rechercher un équilibre entre l'apport de tels moyens pour la protection contre la malveillance et l'augmentation de la complexité du système. La prise en compte des actes de malveillance dès le début de la conception favorise l'atteinte d'un équilibre satisfaisant.

## **7 ACTIONS INTEMPESTIVES OU INAPPROPRIÉES DU CONTRÔLE-COMMANDE**

Les conséquences de chaque action intempestive unique doivent être étudiées : en pratique, une telle action est supposée conduire à un fonctionnement intempestif de l'actionneur piloté, ce qui constitue un cas particulier de défaillance active au sens de la RFS I.3.a [2] ; elle doit être prise en compte dans la conception des systèmes et dans les études de sûreté.

Pour le réacteur EPR Flamanville 3, une action intempestive du contrôle-commande ne doit provoquer ni un incident de référence (PCC3) ni un accident de référence (PCC4) ; en revanche elle peut conduire à un transitoire de référence (PCC2). Pour atteindre cet objectif, l'actionnement d'un composant mécanique dont le fonctionnement intempestif provoquerait une condition de fonctionnement plus sévère qu'un transitoire (PCC2) nécessite une combinaison de demandes de plusieurs divisions du contrôle-commande.

Ainsi, l'ouverture d'une vanne d'isolement de la décharge à l'atmosphère (« Vanne » dans la figure 3) nécessite l'ouverture de deux électrovannes de commande : EV1 et EV2, ouvertes respectivement par les divisions de contrôle-commande 1 et 2, ou EV3 et EV4, ouvertes respectivement par les divisions de contrôle-commande 3 et 4. Ainsi, une action intempestive ou plus généralement une défaillance unique du contrôle-commande ne peut ni déclencher l'ouverture intempestive de la vanne ni empêcher son ouverture en cas de besoin.

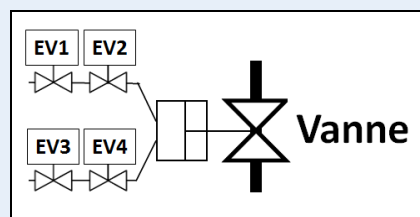


Figure 3 : principe de la commande d'une vanne d'isolement de la décharge à l'atmosphère

Avec l'utilisation croissante des techniques numériques, les architectures de contrôle-commande tendent à regrouper des commandes d'actionneurs différents dans un même calculateur ou dans des calculateurs communiquant entre eux, ce qui peut introduire la possibilité d'actions intempestives multiples de cause commune. Il est important de s'assurer que cela ne peut pas conduire à une situation non prévue par les études des conditions de fonctionnement.

Le respect des principes présentés dans ce document permet de montrer que l'indépendance requise entre fonctions et entre systèmes est effective et de considérer comme non plausible une DCC entraînant des actions

intempestives ou inappropriées de multiples actionneurs correspondant à des fonctions indépendantes. Les actions intempestives multiples restant à considérer sont :

- celles dues à une éventuelle défaillance matérielle unique (par exemple la défaillance d'une carte électronique supportant plusieurs sorties vers des actionneurs), dans le cas où cela est plausible ;
- la mise en service intempestive d'une fonction dans une division de contrôle-commande (qui ne conduit pas nécessairement au fonctionnement intempestif d'un système fluide, comme le montre l'exemple de la figure 3).

Les conséquences fonctionnelles de ces cas doivent être analysées ; en particulier, la mise en service intempestive d'une fonction de sûreté peut bloquer une autre fonction de sûreté et donc ne pas aller dans le sens de la sûreté.

Par exemple, la mise en service et l'isolement de l'alimentation de secours d'un générateur de vapeur donné sont deux fonctions antagonistes classées au plus haut niveau de sûreté ; ceci a nécessité de choisir à l'avance l'action effectuée en cas de demandes simultanées : en l'occurrence l'isolement est prioritaire sur la mise en service, de façon à garantir l'isolement d'un générateur de vapeur lors de la rupture d'un de ses tubes.

Le déclenchement intempestif de la fonction d'isolement pourrait donc bloquer la fonction de mise en service dans une situation nécessitant cette dernière pour évacuer la puissance résiduelle. Cette situation a été étudiée du point de vue fonctionnel, au-delà du contrôle-commande : elle est acceptable parce que la puissance résiduelle peut être évacuée par les trois autres générateurs de vapeur.

Le groupe de travail MDEP (Multilateral Design Evaluation Programme) de l'Agence pour l'énergie nucléaire a interrogé les USA, le Royaume-Uni, la Chine, la France, l'Inde et la Finlande sur leurs positions en matière de prise en compte d'actions intempestives. Leurs positions sont similaires à celle exprimée ci-avant, quoique parfois moins détaillées : la démonstration doit étudier le cas d'une action intempestive plausible, et les justifications relatives à la conception (qualité résultant du classement et indépendance) permettent de limiter l'étendue des actions intempestives multiples à considérer.

Un système de protection du réacteur (pour les paliers N4, P4/P'4 rénovés lors de leur troisième visite décennale et Flamanville 3) utilise plusieurs calculateurs, d'un type donné pour un palier donné ; les calculateurs d'un même type comportent le même logiciel de base. L'exploitant a démontré que ces différents logiciels de base respectent les exigences strictes de la norme [10] et fonctionnent de façon indépendante de leur environnement (procédé, dates calendaires, durée écoulée depuis la mise sous tension, etc.). Chaque exemplaire d'un logiciel de base fonctionne ainsi à son rythme propre, sans couplage avec un environnement susceptible d'activer une hypothétique erreur de façon simultanée dans les différents exemplaires.

Cela permet de ne pas considérer le cas où des fonctions différentes, exécutées par différents calculateurs d'un système de protection, seraient mises en service de façon intempestive et simultanée à cause d'une DCC activée par le couplage du logiciel de base commun à ces calculateurs avec l'environnement.

## **8 CONCLUSION**

Les techniques numériques permettent de réaliser des fonctions avancées comme le calcul du RFTC, de détecter en temps réel les défaillances des matériels ou encore d'offrir aux opérateurs des interfaces riches et souples. Ces fonctions peuvent toutefois être entachées de défauts rendant leur logique inadéquate dans certains cas, ce qui introduit des sources de défaillance des systèmes autres que les pannes aléatoires des matériels et suscite des interrogations relatives à leurs conséquences.

Les pannes matérielles affectant les systèmes de sûreté sont prises en compte par des architectures redondantes et par la réalisation d'essais selon des périodicités cohérentes avec les fréquences estimées des défaillances des matériels. Mais les défauts affectant la logique ne sont pas de même nature que les pannes matérielles, car ils affectent les sorties de façon systématique et non aléatoire, et ne peuvent pas être prévenus ou étudiés avec les mêmes moyens.

L'approche classique de développement des logiques, utilisée par exemple en informatique bureautique, ne maîtrise pas suffisamment la conception et conduit à des produits non vérifiables et affectés de nombreux défauts. De plus, les tentatives faites pour maîtriser la fiabilité d'un système sans viser en priorité à éliminer les défauts dans sa logique se sont avérées inadéquates : par exemple, la diversification d'un logiciel par la mise en parallèle de plusieurs versions, dans l'espoir de masquer les défauts de chacune par un vote majoritaire, est peu praticable dans les faits et son efficacité a été mise en échec expérimentalement (voir l'annexe 3) ; les analyses probabilistes visant à estimer un taux de défaillance ne sont pas applicables à l'évaluation du caractère correct de la logique (voir l'annexe 4) et les analyses de propagation de défaillances utilisées avec succès pour les matériels ne sont pas non plus applicables à la logique (voir la référence [14]).

C'est pourquoi les constructeurs, les exploitants et les organismes techniques comme l'IRSN ont reconnu le besoin de définir une démarche spécifique pour la conception de systèmes de contrôle-commande numériques de réacteurs nucléaires permettant de démontrer le caractère correct de leur logique. La vérification et la validation indépendantes, menées selon une stratégie techniquement justifiée, constituent une précaution supplémentaire.

Cette démarche est complétée par une diversification fonctionnelle qui permet de pallier un hypothétique défaut de la spécification ou de la réalisation de certaines fonctions au moyen d'autres fonctions atteignant les mêmes objectifs en utilisant des signaux physiques et des traitements différents. De plus, une hypothétique défaillance technologique d'une famille de calculateurs est palliée par un moyen fondé sur des mécanismes et des composants logiciels et matériels différents.

Cette démarche a évolué au cours du temps pour prendre en compte les évolutions technologiques telles que les communications par réseaux, ainsi que les progrès scientifiques et techniques comme les méthodes de vérification « formelles », fondées sur une approche mathématique. Elle est entièrement conforme au consensus international exprimé dans les textes de l'AIEA (guide de sûreté SSG-39 relatif au contrôle-commande) et de la CEI (normes concernant le contrôle-commande des centrales nucléaires), et similaire à celles d'autres secteurs industriels où le contrôle-commande exécute des fonctions importantes pour la sûreté, comme l'avionique, le spatial ou le ferroviaire.

La pertinence de cette démarche est confirmée par le retour d'expérience des systèmes numériques de sûreté développés depuis plus de 30 ans (voir l'annexe 2).

Cependant, il ne serait pas souhaitable d'augmenter la complexité et l'intégration du contrôle-commande au-delà de ce qui a été réalisé pour le réacteur EPR Flamanville 3, car la démonstration de l'accomplissement des actions requises et de l'évitement d'actions intempestives généralisées pourrait alors devenir trop difficile pour les méthodes actuellement connues.

Les systèmes numériques sont organisés en une partie générique préexistante (matériel et logiciel de base) et une partie spécifique à l'application visée. Cela permet d'anticiper l'instruction technique de la partie générique, souvent plus complexe que la partie spécifique du point de vue du contrôle-commande.

## ANNEXE 1 - EXEMPLES D'EXIGENCES DE CONCEPTION DETAILLEES

Les deux exemples qui suivent illustrent la manière dont les principes de conception de haut niveau comme ceux relatifs à l'évitement des DCC (voir le paragraphe 6.4.1) se déclinent en exigences détaillées.

### Prévention des DCC par propagation ou interaction

Des communications entre calculateurs sont indispensables (par exemple, quatre calculateurs redondants doivent transmettre leurs résultats à un cinquième chargé du vote), mais la défaillance d'un ordinateur ne doit pas pouvoir perturber le fonctionnement d'un autre, même s'ils communiquent par un réseau de transmission.

Des moyens matériels tels que des transmissions par fibre optique permettent de prévenir les DCC pouvant résulter de la propagation de perturbations physiques ; de plus la conception doit prévenir les DCC pouvant résulter d'un couplage logique : un ordinateur ne doit pas être perturbé si son correspondant ne communique plus ou communique de façon incorrecte. Ainsi, pour les systèmes classés 1E (ou F1A) :

- le fonctionnement des réseaux de transmission doit être déterministe : les ressources nécessaires telles que l'accès au support physique de transmission et la mémoire utilisée dans les équipements d'émission et de réception, ainsi que les calculs effectués dans ces équipements, doivent être prédéfinis et ne doivent en particulier dépendre ni des valeurs transmises ni des variations du procédé ;
- l'émission sur un réseau doit s'effectuer de façon cyclique, sans tenir compte de l'état du (ou des) récepteur(s) et sans même connaître cet état ;
- la lecture sur un réseau doit s'effectuer à l'initiative du ordinateur récepteur, sans connaître l'état de l'émetteur et sans attente due à l'état du réseau ;
- cette lecture doit être possible à tout instant et doit fournir des valeurs correctes, y compris lorsqu'une émission sur le même réseau survient simultanément.

Par exemple, un ordinateur A peut acquérir et émettre cycliquement sur un réseau les valeurs de deux capteurs X et Y. Ces valeurs sont reçues par un ordinateur B, qui les utilise dans un calcul dont l'exactitude nécessite que les valeurs de X et de Y utilisées aient été acquises à des instants très proches.

B doit pouvoir lire à tout instant les valeurs transmises par le réseau, sans attendre, afin de préserver le caractère déterministe de son propre fonctionnement ; si cette lecture s'effectue en même temps que l'émission par A, juste après la transmission de X et juste avant celle de Y, les valeurs les plus récentes disponibles pour B sont temporellement incohérentes car celle de X est très récente et celle de Y date du cycle d'émission précédent : pour garantir la cohérence des calculs, l'interface avec le réseau doit fournir à B l'ensemble cohérent le plus récent (dans ce cas les valeurs de X et de Y du cycle d'émission précédent) et non les valeurs individuelles les plus récentes.

Si ces conditions sont respectées, la défaillance d'un ordinateur émetteur peut avoir pour conséquence l'émission de données incorrectes sur le réseau, l'absence d'émission ou même la monopolisation du réseau (« bavardage »), mais le fonctionnement des ordinateurs récepteurs n'en est pas affecté : ils lisent les données reçues, qui

peuvent être erronées ou absentes, et poursuivent normalement leur cycle de calcul. Ceci permet de tolérer l'absence ou le caractère erroné des données au moyen de données redondantes provenant de réseaux différents et d'une sélection appropriée, telle que le choix de la valeur médiane pour des données numériques ou le vote pour des données tout ou rien. Les calculs peuvent ensuite se poursuivre sans perturbation en utilisant cette donnée « consolidée ».

Ainsi, la défaillance d'un calculateur élaborant la demande partielle d'action de sûreté dans une voie ne peut pas se propager au calculateur effectuant le vote entre les demandes partielles des voies redondantes, pourvu que ces demandes soient transmises par des réseaux séparés respectant les conditions mentionnées ci-avant. De même, la défaillance du voteur ne peut pas se propager aux calculateurs élaborant les demandes partielles d'action de sûreté, ce qui permet à ces calculateurs de transmettre également leurs demandes, sans perturbation, à un voteur redondant.

#### Prévention des DCC par coïncidence

Les calculateurs redondants pourraient être simultanément défectueux si leur fonctionnement dépendait de phénomènes extérieurs tels que les transitoires du procédé et si l'un de ces phénomènes activait un défaut de conception. Par exemple, les automates industriels standards sont souvent configurés pour transmettre les valeurs de capteurs uniquement lorsqu'elles évoluent, afin de minimiser la charge des réseaux : les transitoires créent alors des pics d'émissions qui pourraient saturer les capacités de transmission et de calcul d'unités de traitement redondantes.

La conception doit prévenir ce type de DCC en garantissant un fonctionnement des calculateurs de sûreté indépendant de leur environnement. Ainsi :

- un calculateur doit lire les valeurs données par les capteurs (ou reçues d'autres calculateurs), effectuer ses calculs et émettre les ordres vers les actionneurs (ainsi que les données destinées à d'autres calculateurs) cycliquement, à son propre rythme, de façon invariante et indépendante de son environnement, dans toutes les situations où les fonctions qu'il exécute sont requises ;
- le nombre de données traitées à chaque cycle (capteurs, actionneurs, communications) et la succession des calculs doivent être invariants et ne doivent pas dépendre des valeurs numériques des données ;
- le fonctionnement d'un calculateur ne doit dépendre d'aucune condition extérieure autre que celles spécifiées (par exemple l'inhibition d'un calculateur avec une clé physique, dans le cadre d'une procédure adéquate) ;
- le fonctionnement d'un calculateur ne doit dépendre d'aucun événement ou information non spécifié, comme des dates particulières ou le nombre de cycles de calcul effectués depuis le démarrage du calculateur ;
- le fonctionnement de l'infrastructure du calculateur (lecture des entrées et écriture des sorties, communications, gestion du cycle, autosurveillance, etc.) ne doit pas pouvoir être influencé par les fonctions de contrôle-commande qu'il exécute ou leurs données.

Dans ces conditions, la succession des opérations du calculateur est invariante et indépendante de son environnement, ce qui empêche ce dernier de pouvoir entraîner la DCC de plusieurs calculateurs.



## ANNEXE 2 - RETOUR D'EXPERIENCE DES SYSTEMES NUMERIQUES

Dès que l'utilisation de techniques numériques a été envisagée pour des systèmes de sûreté, les exploitants, les constructeurs et les organismes techniques comme l'IRSN ont reconnu le besoin d'une démarche différente des pratiques industrielles courantes, visant l'absence de défaut. Le retour d'expérience des systèmes numériques développés selon cette démarche apparaît très bon : par exemple, les systèmes de protection numériques des centrales françaises n'ont jamais connu de défaillance non sûre à cause d'un défaut de logiciel.

L'analyse du retour d'expérience présenté dans la référence [17] porte sur une dizaine de millions d'heures de fonctionnement de calculateurs de systèmes de protection. Il montre que la principale contribution aux DCC latentes (c'est-à-dire qui auraient causé des DCC si certains événements initiateurs étaient apparus) est la maintenance et non la conception : sur les 26 DCC latentes identifiées, la plupart concernent des calibrations, des réglages erronés ou, plus généralement, des erreurs de maintenance. Les erreurs de maintenance ne sont d'ailleurs pas spécifiques aux systèmes numériques, qui aident au contraire à les prévenir. Une seule DCC latente a résulté d'un défaut de logiciel, introduit lors d'une modification consistant à ajouter des capteurs diversifiés aux capteurs déjà présents dans les voies redondantes d'un système de protection. La logique ajoutée pour gérer ces capteurs était erronée pour un cas de défaillances particulières des deux capteurs diversifiés d'une même voie : elle fournissait alors une mesure incorrectement considérée comme valide, malgré la détection des défaillances affectant les deux capteurs. Ceci illustre le risque de dégrader le contrôle-commande en croyant l'améliorer, par suite d'un accroissement de complexité.

De son côté, l'Electric Power Research Institute (EPRI) a étudié le retour d'expérience de systèmes de sûreté exploités aux USA depuis plus de 20 ans et en a tiré une conclusion similaire [18] : parmi les 49 incidents ayant affecté ces systèmes, 6 auraient pu conduire à une DCC et un seul est dû à un défaut de logiciel. Deux leçons importantes peuvent être tirées de cet incident :

- le défaut est dû à une complexité excessive et inutile du logiciel, qui gérait des séquences d'autotests dépendant de la position d'un commutateur ;
- la conception du logiciel n'était pas conforme aux principes décrits dans le présent document concernant l'exécution cyclique et le fonctionnement indépendant de l'environnement, ce qui confirme le bien-fondé de ces principes.

Les deux cas précités de DCC latentes dues au logiciel, qui sont marginaux par rapport aux autres cas de défaillances, ont été favorisés par des exigences excessives, qui semblaient aller dans le sens de l'amélioration de la sûreté, mais qui complexifiaient le système et produisaient finalement l'effet inverse de celui attendu.

L'intérêt de systèmes de secours doit ainsi être apprécié en tenant systématiquement compte de la complexification du contrôle-commande, en particulier de la maintenance qui, à ce jour, apparaît en pratique comme la contribution principale aux DCC du contrôle-commande de sûreté.

### **ANNEXE 3 - EXPERIENCE DE KNIGHT ET LEVESON**

La diversification d'un logiciel consiste à exécuter en parallèle plusieurs programmes réalisant la même fonction mais développés indépendamment, en faisant l'hypothèse que leurs défauts éventuels ne concernent pas les mêmes cas et peuvent donc être masqués par un vote approprié.

Knight et Leveson ont mené l'expérience décrite dans la référence [16] pour évaluer la pertinence de cette hypothèse d'indépendance des défauts : 27 versions d'un logiciel répondant à des exigences fonctionnelles précises ont été développées par 27 personnes différentes.

Chaque version a été testée par comparaison avec une version de référence sur un million de cas. Comme les auteurs l'ont indiqué, cette méthode a pu masquer d'éventuels défauts communs à toutes les versions et à la référence, et a donc pu influencer les résultats en faveur de l'approche diversifiée.

D'autres conditions de l'expérience étaient favorables à cette approche. Par exemple la nature mathématique des exigences (favorisant l'absence d'ambiguïté) et l'absence de contraintes de réalisation (concernant la mémoire, le temps d'exécution, le format des sorties, etc.) ont éliminé des sources potentielles de défauts communs à plusieurs versions.

Les 27 versions réalisées étaient globalement bonnes : les tests ont révélé 1,6 défaillances par version en moyenne, quatre et sept défaillances dans les deux plus mauvaises, et aucune dans les six meilleures.

En dépit des conditions favorables, l'expérience a montré un nombre de défaillances multiples (affectant plusieurs versions pour une même entrée) incompatible avec l'hypothèse d'indépendance statistique qui motivait l'approche. Cette hypothèse a donc dû être rejetée, à 99% de confiance.

Aucune corrélation susceptible de biaiser l'expérience n'a été trouvée entre les défaillances des versions et les profils de leurs auteurs (formation, expérience, localisation, etc.).

La diversification d'un logiciel ne peut donc pas, sans autre justification, être considérée comme une parade à la présence d'hypothétiques défauts.

## **ANNEXE 4 - APPROCHES PROBABILISTES**

La capacité d'une logique à accomplir ou non sa fonction résulte de sa conception et ne change pas pendant sa durée de service : une logique ne peut donc pas défaillir, à proprement parler, puisque la défaillance est définie comme la perte de cette capacité (voir le paragraphe « Définitions »). L'expression probabilité de défaillance n'est donc pas adaptée au cas des logiques.

Cependant, certains organismes définissent des exigences spécifiques concernant la quantification de la probabilité de défaillance due au logiciel de systèmes de contrôle-commande. Ces exigences spécifiques n'ont été retenues ni dans le guide AIEA [7] consacré au contrôle-commande ni dans les normes du comité 45A de la CEI ([8] à [13]). Le test statistique étant souvent associé à cette quantification, il est succinctement présenté ci-après.

### **Tests statistiques et profil opérationnel**

Les tests statistiques d'une logique visent à estimer, non la probabilité mal définie de « défaillance de la logique », mais la probabilité que, dans un environnement donné, les entrées appliquées à la logique activent un défaut : la logique est testée par des séquences d'entrées tirées « aléatoirement » dans le « profil opérationnel », défini comme l'ensemble des séquences indépendantes d'entrées, pondérées par leurs fréquences d'apparition dans l'environnement considéré.

Un point essentiel est que cette probabilité est égale au poids total des séquences pour lesquelles la logique est défectueuse. Supposons qu'une logique donnée soit fautive pour une séquence d'entrée particulière ; si cette séquence apparaît une fois sur deux dans un certain environnement, la probabilité de défaillance vaut un demi ; si elle apparaît une fois sur un milliard, la probabilité de défaillance vaut un milliardième, pour la même logique et le même défaut.

La confiance dans l'estimation de cette probabilité dépend donc directement de la confiance dans l'identification et la pondération des séquences indépendantes d'entrées produites par l'environnement considéré.

Comme indiqué dans ce document, les séquences d'entrées possibles d'un système de contrôle-commande sont beaucoup plus nombreuses que les atomes dans l'univers. Il est extrêmement difficile de savoir lesquelles sont effectivement possibles dans un environnement donné et a fortiori de les pondérer selon leur fréquence d'apparition, avec la précision nécessaire pour obtenir une estimation crédible de la probabilité cherchée.

### **Impossibilité pratique de connaître le profil opérationnel**

Il a été proposé d'identifier le profil opérationnel en observant les entrées d'un système identique déjà en fonctionnement, ce qui est inapplicable à un nouveau réacteur. De plus, les sorties dépendent en général des valeurs courantes des entrées mais aussi de mémoires qui ont pu être positionnées à différentes valeurs dans le passé, à des instants inconnus ; par exemple, un même dépassement de seuil peut produire des valeurs différentes d'une sortie donnée selon qu'un certain événement a été détecté ou non auparavant, même si l'occurrence de cet événement n'a pas modifié cette sortie entre-temps et que, par conséquent, rien n'indique cette dépendance à un observateur. L'historique continu des entrées ne peut donc pas être découpé a priori en séquences indépendantes pour constituer un profil opérationnel correct.

Enfin, le nombre de séquences possibles est tel que très peu d'entre elles seront observées en pratique : ainsi, un profil opérationnel fondé sur l'observation de tous les réacteurs du monde depuis leur démarrage ne contiendrait

même pas un cas pour chaque type d'accident, ce qui suscite des doutes lorsque l'objet à tester est un système de protection dont la principale mission est la détection des accidents.

Il a aussi été proposé d'identifier le profil opérationnel à partir des grandeurs physiques utilisées dans les études d'accidents. Mais une grandeur physique comme le RFTC étant calculée dans le système de protection à partir de centaines d'entrées et de paramètres, un nombre gigantesque de combinaisons des entrées et des paramètres conduisent à une même valeur du RFTC. La valeur du RFTC à un instant donné d'une étude d'accident ne permet donc pas de déterminer toutes les combinaisons d'entrées physiquement possibles correspondant à cette valeur, ni a fortiori de les pondérer par leurs fréquences d'apparition. Or, le calcul pouvant être erroné pour une combinaison et pas pour les autres, elles ne peuvent pas être considérées comme équivalentes et elles devraient donc être pondérées correctement pour donner un caractère objectif à la probabilité estimée ; comme on l'a vu plus haut, si la combinaison « fautive » apparaît une fois sur deux dans un certain environnement, la probabilité de défaillance vaut un demi ; si elle apparaît une fois sur un milliard, la probabilité de défaillance vaut un milliardième. Il reste donc indispensable de les pondérer correctement.

### Conclusion sur les tests statistiques

Aucun moyen n'a été proposé à ce jour pour identifier le profil opérationnel d'un environnement réel (complétude des cas et exactitude de la pondération) et donc pour réaliser un test statistique pertinent. La « *probabilité pour que, dans un environnement donné, les entrées appliquées à un système activent un défaut de sa logique* » a donc un sens théorique, mais aucun des moyens proposés ne permet d'en fournir une estimation crédible.

La norme de référence des logiciels de sûreté [10] synthétise ce constat : « *The validity of the calculated (probability) depends upon the similarity of the profile of the test inputs to the profile of the actual inputs experienced by the system in operation. If (...) used on an unrealistic operational profile (...) a (probability) will be estimated that may be very different to the actual system availability (...). This is a fundamental weakness of the statistical testing approach as it is generally very difficult to accurately determine the operational profile that a system will experience in use, and this is particularly true for systems with large numbers of inputs* ».

### Etudes probabilistes de sûreté

Pour les études probabilistes de sûreté (EPS), des valeurs forfaitaires consensuelles de probabilité de défaillance, justifiées qualitativement, peuvent être utilisées pour les systèmes de contrôle-commande. Par exemple, les experts du contrôle-commande nucléaire du comité 45A de la CEI estiment [8] que « *pour un système individuel, (...) une probabilité de l'ordre de  $10^{-4}$  échec par demande peut être revendiquée comme une limite de fiabilité globale légitime, considérant que toutes les sources potentielles de défaillances dues aux spécifications, à la conception, à la fabrication, à l'installation, à l'environnement d'exploitation et aux pratiques de maintenance sont prises en compte* ». La combinaison de systèmes indépendants permet d'améliorer cette fiabilité (par exemple, le système de protection de l'EPR comporte en fait deux sous-systèmes indépendants, réalisant des fonctions d'arrêt automatique diversifiées, pour atteindre une probabilité de l'ordre de  $10^{-5}$  échec par demande). La même approche est utilisée dans d'autres secteurs industriels comme l'avionique, le spatial, le ferroviaire, l'automatisation, l'automobile, etc. [19]

## **ANNEXE 5 - TECHNIQUES DE REALISATION PARTICULIERES (INTERRUPTIONS)**

Les normes telles que [7], [10], [11] et [12] décrivent des techniques de réalisation spécifiques, qui peuvent être adéquates dans certains cas, mais pas dans d'autres, en fonction de détails concernant les exigences fonctionnelles, l'architecture du système et le fonctionnement du matériel. Ce document n'a pas vocation à analyser ces détails et par conséquent il ne prend pas position sur les techniques de réalisation correspondantes.

Cette annexe illustre ce sujet sur l'exemple important des « interruptions ». Une interruption est un signal matériel transmis à un processeur, qui le force à suspendre le traitement en cours (dit « principal » dans la suite de cette annexe) pour en exécuter un autre avant de reprendre le cours du traitement principal.

Des interruptions peuvent être transmises au processeur par exemple lorsqu'une information arrive d'un réseau, pour que le processeur la mémorise et la traite. Des interruptions peuvent aussi être transmises au processeur à intervalles de temps prédéfinis, par exemple pour que celui-ci exécute des autotests pendant une partie de son temps.

En général, le traitement principal dépend des données manipulées pendant l'interruption : c'est le cas par exemple lorsque ce traitement consiste à calculer une fonction à partir de mesures provenant de capteurs et de réseaux, et que les variables contenant ces mesures sont mises à jour au moyen d'interruptions. Les variables utilisées par le traitement principal peuvent alors être modifiées à tout instant par les interruptions.

Des résultats faux peuvent en découler si des précautions ne sont pas prises. Considérons par exemple une variable du programme principal codée sur deux caractères, valant « 09 » à un instant donné. Les deux caractères « 1 » et « 0 » formant la nouvelle valeur « 10 » sont alors transmis successivement au processeur et génèrent chacun une interruption. Si le processeur utilise cette variable avant l'arrivée des interruptions, il effectue ses calculs avec la valeur « 09 » ; s'il l'utilise après, il calcule avec la valeur « 10 », qui est également correcte et correspond à un instant d'échantillonnage du procédé un peu ultérieur. Mais s'il l'utilise entre les deux interruptions, il calcule avec la valeur « 19 », qui ne correspond à aucun état réel du procédé et conduit à un résultat faux.

Cet exemple simplifié mais représentatif illustre la nécessité de prendre des précautions lorsque deux processus différents utilisent une même donnée ou plus généralement une même ressource. Dans une conception non maîtrisée, il est très difficile d'éviter de telles erreurs de synchronisation ; ceci a entraîné la mauvaise réputation des interruptions et certains textes proposent d'ailleurs de bannir leur utilisation.

Cependant, dans un système de sûreté, plusieurs calculateurs doivent généralement échanger des informations, et ils fonctionnent le plus souvent de façon asynchrone pour éviter la propagation de défaillances (chacun émet vers l'autre sans connaître son état). Pour un calculateur donné, la réception des informations et l'exécution des calculs qui utilisent ces informations sont donc asynchrones, ce qui requiert l'existence de deux traitements asynchrones pour prendre en charge ces deux aspects.

Comme indiqué plus haut, l'utilisation d'une interruption remplit ce besoin en permettant au même processeur d'exécuter le calcul principal et de mémoriser les données reçues lorsqu'elles arrivent, de façon à les utiliser au prochain cycle de calcul. Cependant, si ces traitements sont mal synchronisés, des incohérences peuvent survenir, comme dans l'exemple présenté ci-avant.

Un autre moyen pour effectuer les deux traitements asynchrones consiste à utiliser, en plus du processeur chargé du traitement principal, un processeur auxiliaire pour recevoir les informations et les ranger dans une mémoire dite « à double accès » ; le processeur principal accède de son côté à cette même mémoire, par exemple au début de chaque cycle de calcul, pour lire les informations rangées et les utiliser dans son calcul. Cette technique « à mémoire partagée » permet d'éviter les « logiciels à interruptions », puisque chaque processeur effectue un seul traitement ; elle est de ce fait parfois recommandée dans des documents prenant le parti d'imposer des détails de réalisation. Cependant, utilisée sans précaution, la technique « à mémoire partagée » souffre exactement du même défaut que celle fondée sur les interruptions : si le processeur principal lit les deux caractères de l'exemple présenté ci-avant pendant que le processeur auxiliaire les modifie, une incohérence peut survenir.

Il ne s'agit donc pas ici de préconiser ou d'interdire l'une ou l'autre de ces techniques, mais plutôt de montrer que le problème de fond lié à l'exécution parallèle de deux processus interdépendants, doit être résolu rigoureusement. Le principe d'une telle résolution, connu depuis les travaux d'Edsger Dijkstra (« Solution of a Problem in Concurrent Programming Control », Communications of the ACM, 1965), repose sur l'utilisation de protocoles rigoureux pour attribuer à un seul processus à la fois le droit d'utiliser une ressource commune. En tant que principe mathématique, il s'applique aussi bien à une technique qu'à l'autre : si un concepteur est capable de résoudre correctement le problème correspondant à l'exemple cité plus haut avec une mémoire partagée, il est tout aussi capable de le résoudre avec une interruption. L'une ou l'autre des solutions peut alors conduire à une conception plus simple et mieux vérifiable, en fonction de détails comme la quantité de données à transférer, leur fréquence d'arrivée par rapport à la fréquence du traitement principal, la charge des processeurs, etc.

**Siège social**

31, avenue de la Division Leclerc  
92260 Fontenay-aux-Roses  
RCS Nanterre B 440 546 018

**Téléphone**


+33 (0)1 58 35 88 88

**Courrier**

B.P. 17 - 92262 Fontenay-aux-Rose Cedex

**Site Internet**

[www.irsn.fr](http://www.irsn.fr)

 [@IRSNFrance](https://twitter.com/IRSNFrance), [@suretenucleaire](https://twitter.com/suretenucleaire)