

Master 1, Conceptions Formelles
Projet du module ALTARICA
Synthèse (assistée) d'un contrôleur du niveau d'une cuve

CATALDI

ERRARD

Chapitre 1

Le sujet

1.1 Cahier des charges

Le système que l'on souhaite concevoir est composé :

- d'un réservoir contenant **toujours** suffisamment d'eau pour alimenter l'exploitation,
- d'une cuve,
- de deux canalisations parfaites amont reliant le réservoir à la cuve, et permettant d'amener l'eau à la cuve,
- d'une canalisation parfaite aval permettant de vider l'eau de la cuve,
- chaque canalisation est équipée d'une vanne commandable, afin de réguler l'alimentation et la vidange de la cuve,
- d'un contrôleur.

1.1.1 Détails techniques

La vanne

Les vannes sont toutes de même type, elles possèdent trois niveaux de débits correspondant à trois diamètres d'ouverture : 0 correspond à la vanne fermée, 1 au diamètre intermédiaire et 2 à la vanne complètement ouverte. Les vannes sont commandables par les deux instructions **inc** et **dec** qui respectivement augmente et diminue l'ouverture. Malheureusement, la vanne est sujet à défaillance sur sollicitation, auquel cas le système de commande devient inopérant, la vanne est désormais pour toujours avec la même ouverture.

La Cuve

Elle est munie de $nbSensors$ capteurs (au moins quatre) situés à $nbSensors$ hauteurs qui permettent de délimiter $nbSensors + 1$ zones. La zone 0 est comprise entre le niveau 0 et le niveau du capteur le plus bas ; la zone 1 est comprise entre ce premier capteur et le second, et ainsi de suite.

Elle possède en amont un orifice pour la remplir limité à un débit de 4, et en aval un orifice pour la vider limité à un débit de 2.

Le contrôleur

Il commande les vannes avec les objectifs suivants ordonnés par importance :

1. Le système ne doit pas se bloquer, et le niveau de la cuve ne doit jamais atteindre les zones 0 ou $nbSensors$.
2. Le débit de la vanne aval doit être le plus important possible.

On fera également l'hypothèse que les commandes ne prennent pas de temps, et qu'entre deux pannes et/ou cycle *temporel*, le contrôleur à toujours le temps de donner au moins un ordre. Réciproquement, on fera l'hypothèse que le système à toujours le temps de réagir entre deux commandes.

Les débits

Les règles suivantes résument l'évolution du niveau de l'eau dans la cuve :

- Si ($amont > aval$) alors au temps suivant, le niveau aura augmenté d'une unité.
- Si ($amont < aval$) alors au temps suivant, le niveau aura baissé d'une unité.
- Si ($amont = aval = 0$) alors au temps suivant, le niveau n'aura pas changé.
- Si ($amont = aval > 0$) alors au temps suivant, le niveau pourra :
 - avoir augmenté d'une unité,
 - avoir baissé d'une unité,
 - être resté le même.

1.2 L'étude

1.2.1 Rappel méthodologique

Comme indiqué en cours, le calcul par point fixe du contrôleur est exact, mais l'opération de projection effectuée ensuite peut perdre de l'information et générer un contrôleur qui n'est pas satisfaisant. Plus précisément, le contrôleur ALTARICA généré :

- ne garanti pas la non accessibilité des *Situations Redoutées*.
- ne garanti pas l'absence de *nouvelles situations de blocages*.

Dans le cas où il existe toujours *des situations de blocages ou redoutées*, vous pouvez au choix :

1. Corriger manuellement le contrôleur calculé (sans doute très difficile).
2. Itérer le processus du calcul du contrôleur jusqu'à stabilisation du résultat obtenu.
 - Si le contrôleur obtenu est sans blocage et sans situation redoutée, il est alors correct.
 - Si le contrôleur obtenu contient toujours des blocages ou des situations redoutées, c'est que le contrôleur initial n'est pas assez performant, mais rien ne garanti que l'on soit capable de fournir ce premier contrôleur suffisamment performant.

Remarque : Pour vos calculs, vous pouvez utiliser au choix les commandes :

- `altarica-studio xxx.alt xxx.spe`
- `arc -b xxx.alt xxx.spe`
- `make` pour utiliser le fichier GNUmakefile fourni.

1.2.2 Le travail à réaliser

Avant de calculer les contrôleurs, vous devez répondre aux questions suivantes.

1. Expliquez le rôle de la constante `nbFailures` et de la contrainte, présente dans le composant `System`, $nbFailures \geq (V[0].fail + V[1].fail + V[2].fail)$.
2. Expliquez le rôle du composant `ValveVirtual` et de son utilisation dans le composant `CtrlVV`, afin de remplacer le composant `Ctrl` utilisé initialement.

L'étude consiste à étudier le système suivant deux paramètres :

1. `nbFailures` : une constante qui est une borne pour le nombre de vannes pouvant tomber en panne.
2. Le contrôleur initial qui peut être soit `Ctrl`, soit `CtrlVV`.

Pour chacun des huit systèmes étudiés, vous devez décrire votre méthodologie pour calculer les différents contrôleurs et répondre aux questions suivantes :

1. Est-il possible de contrôler en évitant les blocages et les situations critiques ?
2. Si oui, donnez quelques caractéristiques de ce contrôleur, si non, expliquez pourquoi.
3. Est-il possible de contrôler en optimisant le débit aval et en évitant les blocages et les situations critiques ?
4. Si oui, donnez quelques caractéristiques de ce contrôleur, si non, expliquez pourquoi.

Chapitre 2

Le rapport

Le rapport est sur 20 points.

2.1 Rôle du fichier GNUmakefile (2 points)

Les fichiers GNUmakefile permettent de générer différents contrôleurs (classique et optimisés, avec et sans vannes virtuelles). Les différents contrôleurs sont donc générés en fonction d'un nombre maximum d'erreurs possible (allant de 0 à 3). On optimise ensuite ses contrôleurs en itérant 5 fois sur le contrôleur précédent généré. Il retourne également les résultats des spécifications pour chacun des contrôleurs générés.

2.2 Rôle de la constante nbFailures et de l'assertion associée (1 point)

La constante `nbFailures` permet de représenter le nombre maximum d'erreur du système possible. L'assertion associée permet donc de définir que le nombre d'erreurs du système ne peut pas dépasser `nbFailures`.

2.3 Résultats avec le contrôleur initial Ctrl

2.3.1 Calcul d'un contrôleur

Avec 0 défaillance (0.5 point)

```
/*
 * Properties for node : System0FCtrl
 * # state properties : 7
 *
 * any_s = 247
 * deadlock = 0
 * NC = 86
 * SR = 86
 * out0 = 80
 * out1 = 83
 * out2 = 84
 *
 * # trans properties : 4
 *
 * any_t = 3472
 * dec21 = 9750
```

```

* dec10 = 9500
* CCoupGagnant = 1134
*/

/*
* Properties for node : System0FCtrl0F1I
* # state properties : 7
*
* any_s = 94
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 26
* out1 = 34
* out2 = 34
*
* # trans properties : 4
*
* any_t = 858
* dec21 = 2870
* dec10 = 1910
* CCoupGagnant = 712
*/

/*
* Properties for node : System0FCtrl0F2I
* # state properties : 7
*
* any_s = 94
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 26
* out1 = 34
* out2 = 34
*
* # trans properties : 4
*
* any_t = 858
* dec21 = 2870
* dec10 = 1910
* CCoupGagnant = 712
*/

```

Interprétation des résultats

On constate que pour un système avec 0 erreur, on constate qu'à partir de la 1^{ère} itération on diminue de 153 le nombre d'état total et de 3225 le nombre de transition. On obtient également plus aucun état dans un niveau critique ou une situation redouté. De plus dans le contrôleur de base, environ 33 % des transitions sont des coups gagnants. A partir de la première itération 83% des transition sont gagnant. On remarque également qu'il n'y a aucune différences entre le résultats à la 1^{ère} et à la deuxième itération sont identique. On peut donc déterminé qu'on a générer le contrôleur optimal au bout de la première itération.

Avec 1 défaillance (0.5 point)

```

/*
 * Properties for node : System1FCtrl
 * # state properties : 7
 *
 * any_s = 958
 * deadlock = 0
 * NC = 329
 * SR = 329
 * out0 = 300
 * out1 = 326
 * out2 = 332
 *
 * # trans properties : 4
 *
 * any_t = 19540
 * dec21 = 26025
 * dec10 = 25225
 * CCoupGagnant = 4950
 */

/*
 * Properties for node : System1FCtrlF1I
 * # state properties : 7
 *
 * any_s = 508
 * deadlock = 93
 * NC = 69
 * SR = 93
 * out0 = 120
 * out1 = 188
 * out2 = 200
 *
 * # trans properties : 4
 *
 * any_t = 5230
 * dec21 = 8205
 * dec10 = 5285
 * CCoupGagnant = 2941
 */

/*
 * Properties for node : System1FCtrlF2I
 * # state properties : 7
 *
 * any_s = 508
 * deadlock = 96
 * NC = 69
 * SR = 96
 * out0 = 120
 * out1 = 188
 * out2 = 200
 *
 * # trans properties : 4
 *
 * any_t = 5161
 * dec21 = 8205

```

```

* dec10 = 5245
* CCoupGagnant = 2909
*/

/*
* Properties for node : System1FCtrl1F3I
* # state properties : 7
*
* any_s = 508
* deadlock = 96
* NC = 69
* SR = 96
* out0 = 120
* out1 = 188
* out2 = 200
*
* # trans properties : 4
*
* any_t = 5161
* dec21 = 8205
* dec10 = 5245
* CCoupGagnant = 2909
*/

```

Interprétation des résultats

On constate que pour un système avec 1 erreur, on constate que entre le contrôleur de base et la dernière itération on diminue de 450 le nombre d'état total et de 14 379 le nombre de transition. On obtient également 260 état de moins dans un niveau critique et 233 état de moins dans une situation redouté. Mais le nombre de d'état puis est passé de 0 à 96. De plus dans le contrôleur de base, environ 25 % des transitions sont des coups gagnants. A la dernière itération 56% des transition sont gagnant. On remarque qu'on obtient le nombre optimal d'état dès la première itération. Et qu'au fil des itération on augmente le pourcentage de coup gagnant mais on augmente aussi le nombre de situation redouté.

Avec 2 défaillances (0.5 point)

```

/*
* Properties for node : System2FCtrl
* # state properties : 7
*
* any_s = 1627
* deadlock = 0
* NC = 551
* SR = 551
* out0 = 506
* out1 = 553
* out2 = 568
*
* # trans properties : 4
*
* any_t = 44608
* dec21 = 34725
* dec10 = 33475
* CCoupGagnant = 7533
*/

```



```

/*
 * Properties for node : System2FCtrl2F1I
 * # state properties : 7
 *
 * any_s = 790
 * deadlock = 239
 * NC = 107
 * SR = 239
 * out0 = 200
 * out1 = 306
 * out2 = 284
 *
 * # trans properties : 4
 *
 * any_t = 7168
 * dec21 = 4520
 * dec10 = 5335
 * CCoupGagnant = 3029
 */

```

```

/*
 * Properties for node : System2FCtrl2F2I
 * # state properties : 7
 *
 * any_s = 774
 * deadlock = 271
 * NC = 107
 * SR = 271
 * out0 = 190
 * out1 = 302
 * out2 = 282
 *
 * # trans properties : 4
 *
 * any_t = 6547
 * dec21 = 3675
 * dec10 = 5280
 * CCoupGagnant = 2826
 */

```

```

/*
 * Properties for node : System2FCtrl2F3I
 * # state properties : 7
 *
 * any_s = 772
 * deadlock = 270
 * NC = 107
 * SR = 270
 * out0 = 190
 * out1 = 302
 * out2 = 280
 *
 * # trans properties : 4
 *
 * any_t = 6534
 * dec21 = 3675

```

```

* dec10 = 5280
* CCoupGagnant = 2826
*/

/*
* Properties for node : System2FCtrl2F4I
* # state properties : 7
*
* any_s = 772
* deadlock = 270
* NC = 107
* SR = 270
* out0 = 190
* out1 = 302
* out2 = 280
*
* # trans properties : 4
*
* any_t = 6534
* dec21 = 3675
* dec10 = 5280
* CCoupGagnant = 2826
*/

```

Interprétation des résultats

On constate que pour un système avec 2 erreurs, on constate que entre le contrôleur de base et la dernière itération on diminue de 855 le nombre d'état total et de 38 074 le nombre de transition. On obtient également 444 état de moins dans un niveau critique et 281 état de moins dans une situation redouté. Mais le nombre de d'état puis est passé de 0 à 270. De plus dans le contrôleur de base, environ 16% des transitions sont des coups gagnants. A la dernière itération 43% des transition sont gagnant. On constate qu'on obtient le contrôleur le plus optimal à partir de la 3ème itération car aucune des valeur ne change.

Avec 3 défaillances (0.5 point)

```

/*
* Properties for node : System3FCtrl
* # state properties : 7
*
* any_s = 1832
* deadlock = 0
* NC = 617
* SR = 617
* out0 = 570
* out1 = 622
* out2 = 640
*
* # trans properties : 4
*
* any_t = 57696
* dec21 = 36210
* dec10 = 34870
* CCoupGagnant = 7908
*/

```

```

/*
 * Properties for node : System3FCtrl3F1I
 * # state properties : 7
 *
 * any_s = 240
 * deadlock = 112
 * NC = 0
 * SR = 112
 * out0 = 48
 * out1 = 120
 * out2 = 72
 *
 * # trans properties : 4
 *
 * any_t = 1568
 * dec21 = 420
 * dec10 = 480
 * CCoupGagnant = 343
 */

```

```

/*
 * Properties for node : System3FCtrl3F2I
 * # state properties : 7
 *
 * any_s = 62
 * deadlock = 27
 * NC = 0
 * SR = 27
 * out0 = 36
 * out1 = 26
 * out2 = 0
 *
 * # trans properties : 4
 *
 * any_t = 609
 * dec21 = 0
 * dec10 = 0
 * CCoupGagnant = 343
 */

```

```

/*
 * Properties for node : System3FCtrl3F3I
 * # state properties : 7
 *
 * any_s = 62
 * deadlock = 27
 * NC = 0
 * SR = 27
 * out0 = 36
 * out1 = 26
 * out2 = 0
 *
 * # trans properties : 4
 *
 * any_t = 609
 * dec21 = 0

```

```

* dec10 = 0
* CCoupGagnant = 343
*/

```

Interprétation des résultats

On constate que pour un système avec 3 erreurs, on constate que entre le contrôleur de base et la dernière itération on diminue de 855 le nombre d'état total et de 38 074 le nombre de transition. On obtient également 1770 état de moins dans un niveau critique et 281 état de moins dans une situation redouté. Mais le nombre de d'état puis est passé de 0 à 270. De plus dans le contrôleur de base, environ 16% des transitions sont des coups gagnants. A la dernière itération 43% des transition sont gagnant. On constate qu'on obtient le contrôleur le plus optimal à partir de la 3ème itération car aucune des valeur ne change.

2.3.2 Calcul des contrôleurs optimisés (2 points)

```

event
/*
* les priorites dependent des actions sur la vanne aval
* inc > nop > dec
*/
{ddi, dii, dni, idi, iii, ini, ndi, nii, nni} >
{ddn, din, dnn, idn, iin, inn, ndn, nin, nnn};
{ddn, din, dnn, idn, iin, inn, ndn, nin, nnn} >
{ddd, did, dnd, idd, iid, ind, ndd, nid, nnd};
edon

/*
* Properties for node : System0FCtrl0F2I_Opt
* # state properties : 7
*
* any_s = 49
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 1
* out1 = 14
* out2 = 34
*
* # trans properties : 4
*
* any_t = 220
* dec21 = 40
* dec10 = 0
* CCoupGagnant = 142
*/

```

2.4 Construction d'un contrôleur initial plus performant

2.4.1 Rôle du composant ValveVirtual(2 points)

Le composant ValveVirtual est une alternative au composant Valve présent dans Ctrl. La ValveVirtual possède non seulement un rate mais aussi un rateReal, le premier étant modifier par la valve et le second par le contrôleur. Contrairement à la Valve classique qui utilise un booléen pour savoir si elle est coincé, celle-ci compare sa valeur rate et rateReal, la ValveVirtual utilise ces deux valeurs comme garde de ses transitions, si les deux valeurs sont les mêmes alors la ValveVirtual n'est pas coincé et peut donc effectuer des transitions, si les deux valeurs sont différentes alors la ValveVirtual ne fait rien.

2.4.2 Rôle du composant CtrlVV (5 points)

Le composant CtrlVV est une alternative au Ctrl qui au lieu d'utiliser des Valve, utilise des ValveVirtual. Ce composant vérifie par le biais d'assertion que les valeurs de ses rate et les rateReal de ses ValveVirtual soient les mêmes. Ces assertions, un fois combiner avec les gardes des ValveVirtual garantisse que si tout va bien alors les valeurs des rate des ValveVirtual et celle du CtrlVV soient les mêmes. En plus, à la différences du Ctrl, le CtrlVV fait aussi la synchronisation entre les différentes ValveVirtual lors des actions du contrôleur.

2.5 Résultats avec le contrôleur CtrlVV

2.5.1 Calcul d'un contrôleur

Avec 0 défaillance (0.5 point)

```
/*
 * Properties for node : System0FCtrlVV
 * # state properties : 7
 *
 * any_s = 247
 * deadlock = 0
 * NC = 86
 * SR = 86
 * out0 = 80
 * out1 = 83
 * out2 = 84
 *
 * # trans properties : 4
 *
 * any_t = 1863
 * dec21 = 5270
 * dec10 = 5140
 * CCoupGagnant = 548
 */

/*
 * Properties for node : System0FCtrlVV0F1I
 * # state properties : 7
 *
 * any_s = 94
```

```

* deadlock = 0
* NC = 0
* SR = 0
* out0 = 26
* out1 = 34
* out2 = 34
*
* # trans properties : 4
*
* any_t = 508
* dec21 = 1670
* dec10 = 1110
* CCoupGagnant = 362
*/

/*
* Properties for node : System0FCtrlVV0F2I
* # state properties : 7
*
* any_s = 94
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 26
* out1 = 34
* out2 = 34
*
* # trans properties : 4
*
* any_t = 508
* dec21 = 1670
* dec10 = 1110
* CCoupGagnant = 362
*/

```

Interprétation des résultats

On constate que pour un système utilisant le contrôleur CtrlVV avec 0 erreur, on constate que entre le contrôleur de base et la dernière itération on diminue de 153 le nombre d'état total et de 1355 le nombre de transition. De plus, on supprime entièrement les états puits et les états de niveaux critiques ce qui fait que nous n'avons plus de situation redouté. De plus dans le contrôleur de base, environ 29 % des transitions sont des coups gagnants. A la dernière itération 71 % des transitions sont gagnantes.

Avec 1 défaillance (0.5 point)

```

/*
* Properties for node : System1FCtrlVV
* # state properties : 7
*
* any_s = 1201
* deadlock = 0
* NC = 413
* SR = 413
* out0 = 350
* out1 = 463
* out2 = 388
*

```

```

* # trans properties : 4
*
* any_t = 8370
* dec21 = 9050
* dec10 = 8820
* CCoupGagnant = 1866
*/

/*
* Properties for node : System1FCtrlVV1F1I
* # state properties : 7
*
* any_s = 316
* deadlock = 16
* NC = 0
* SR = 16
* out0 = 68
* out1 = 138
* out2 = 110
*
* # trans properties : 4
*
* any_t = 1076
* dec21 = 1130
* dec10 = 740
* CCoupGagnant = 546
*/

/*
* Properties for node : System1FCtrlVV1F2I
* # state properties : 7
*
* any_s = 232
* deadlock = 3
* NC = 0
* SR = 3
* out0 = 46
* out1 = 104
* out2 = 82
*
* # trans properties : 4
*
* any_t = 787
* dec21 = 670
* dec10 = 500
* CCoupGagnant = 413
*/

/*
* Properties for node : System1FCtrlVV1F3I
* # state properties : 7
*
* any_s = 224
* deadlock = 0
* NC = 0
* SR = 0

```

```

* out0 = 46
* out1 = 104
* out2 = 74
*
* # trans properties : 4
*
* any_t = 745
* dec21 = 620
* dec10 = 500
* CCoupGagnant = 392
*/

/*
* Properties for node : System1FCtrlVV1F4I
* # state properties : 7
*
* any_s = 224
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 46
* out1 = 104
* out2 = 74
*
* # trans properties : 4
*
* any_t = 745
* dec21 = 620
* dec10 = 500
* CCoupGagnant = 392
*/

```

Interprétation des résultats

On constate que pour un système utilisant le contrôleur CtrlVV avec 1 erreur, on constate que entre le contrôleur de base et la dernière itération on diminue de 977 le nombre d'état total et de 7625 le nombre de transition. De plus, on supprime entièrement les états puits et les états de niveaux critiques ce qui fait que nous n'avons plus de situation redouté. De plus dans le contrôleur de base, environ 22 % des transitions sont des coups gagnants. A la dernière itération 53 % des transitions sont gagnantes.

Avec 2 défaillances (0.5 point)

```

/*
* Properties for node : System2FCtrlVV
* # state properties : 7
*
* any_s = 2398
* deadlock = 0
* NC = 812
* SR = 812
* out0 = 651
* out1 = 1005
* out2 = 742
*
* # trans properties : 4
*
* any_t = 15894

```



```

* dec21 = 9690
* dec10 = 9450
* CCoupGagnant = 2360
*/

/*
* Properties for node : System2FCtrlVV2F1I
* # state properties : 7
*
* any_s = 274
* deadlock = 70
* NC = 0
* SR = 70
* out0 = 52
* out1 = 130
* out2 = 92
*
* # trans properties : 4
*
* any_t = 725
* dec21 = 170
* dec10 = 170
* CCoupGagnant = 155
*/

/*
* Properties for node : System2FCtrlVV2F2I
* # state properties : 7
*
* any_s = 2
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 2
* out1 = 0
* out2 = 0
*
* # trans properties : 4
*
* any_t = 4
* dec21 = 0
* dec10 = 0
* CCoupGagnant = 1
*/

/*
* Properties for node : System2FCtrlVV2F3I
* # state properties : 7
*
* any_s = 2
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 2
* out1 = 0
* out2 = 0

```

```

*
* # trans properties : 4
*
* any_t = 4
* dec21 = 0
* dec10 = 0
* CCoupGagnant = 1
*/

```

Interprétation des résultats

On constate que pour un système utilisant le contrôleur CtrlVV avec 2 erreur, on constate que entre le contrôleur de base et la dernière itération on diminue de 2396 le nombre d'état total et de 15890 le nombre de transition. De plus, on supprime entièrement les états puits et les états de niveaux critiques ce qui fait que nous n'avons plus de situation redouté. De plus dans le contrôleur de base, environ 14 % des transitions sont des coups gagnants. A la dernière itération 25 % des transitions sont gagnantes.

Avec 3 défaillances (0.5 point)

```

/*
* Properties for node : System3FCtrlVV
* # state properties : 7
*
* any_s = 2889
* deadlock = 0
* NC = 970
* SR = 970
* out0 = 764
* out1 = 1253
* out2 = 872
*
* # trans properties : 4
*
* any_t = 18776
* dec21 = 9690
* dec10 = 9450
* CCoupGagnant = 2384
*/

/*
* Properties for node : System3FCtrlVV3F1I
* # state properties : 7
*
* any_s = 210
* deadlock = 97
* NC = 0
* SR = 97
* out0 = 36
* out1 = 114
* out2 = 60
*
* # trans properties : 4
*
* any_t = 565
* dec21 = 130
* dec10 = 80
* CCoupGagnant = 27

```

```

*/

/*
* Properties for node : System3FCtrlVV3F2I
* # state properties : 7
*
* any_s = 2
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 2
* out1 = 0
* out2 = 0
*
* # trans properties : 4
*
* any_t = 4
* dec21 = 0
* dec10 = 0
* CCoupGagnant = 1
*/

/*
* Properties for node : System3FCtrlVV3F3I
* # state properties : 7
*
* any_s = 2
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 2
* out1 = 0
* out2 = 0
*
* # trans properties : 4
*
* any_t = 4
* dec21 = 0
* dec10 = 0
* CCoupGagnant = 1
*/

```

Interprétation des résultats

On constate que pour un système utilisant le contrôleur CtrlVV avec 3 erreur, on constate que entre le contrôleur de base et la dernière itération on diminue de 2887 le nombre d'état total et de 18772 le nombre de transition. De plus, on supprime entièrement les états puits et les états de niveaux critiques ce qui fait que nous n'avons plus de situation redouté. De plus dans le contrôleur de base, environ 13 % des transitions sont des coups gagnants. A la dernière itération 25 % des transitions sont gagnantes.

2.5.2 Calcul des contrôleurs optimisés (2 points)

Avec 0 défaillance

```

/*
* Properties for node : System0FCtrlVV0F2I_Opt
* # state properties : 7

```

```

*
* any_s = 49
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 1
* out1 = 14
* out2 = 34
*
* # trans properties : 4
*
* any_t = 174
* dec21 = 40
* dec10 = 0
* CCoupGagnant = 96
*/

```

Avec 1 défaillance

```

/*
* Properties for node : System1FCtrlVV1F4I_Opt
* # state properties : 7
*
* any_s = 191
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 21
* out1 = 96
* out2 = 74
*
* # trans properties : 4
*
* any_t = 580
* dec21 = 180
* dec10 = 140
* CCoupGagnant = 277
*/

```

Avec 2 défaillances

```

/*
* Properties for node : System2FCtrlVV2F3I_Opt
* # state properties : 7
*
* any_s = 2
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 2
* out1 = 0
* out2 = 0
*
* # trans properties : 4

```

```

*
* any_t = 4
* dec21 = 0
* dec10 = 0
* CCoupGagnant = 1
*/

```

Avec 3 défaillances

```

/*
* Properties for node : System3FCtrlVV3F3I_Opt
* # state properties : 7
*
* any_s = 2
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 2
* out1 = 0
* out2 = 0
*
* # trans properties : 4
*
* any_t = 4
* dec21 = 0
* dec10 = 0
* CCoupGagnant = 1
*/

```

2.6 Conclusion (2 points)