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

Garion GOUBARD Kristo DHIMA

# 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 précisemment, 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 ou 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 suffisemment 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 a réaliser

L'étude consiste à étudier le système suivant trois 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`.
3. Une éventuelle optimisation pour améliorer le débit aval.

Les questions auxquelles vous devez répondre sont dans le fichier `fichier rapport.tex`. Elles correspondent aux interrogations 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.

Vous écrirez vos réponses dans ce même fichier.

# Chapitre 2

# Le rapport

Le rapport est sur 20 points.

## 2.1 Processus

### 2.1.1 Rôle du fichier `GNUmakefile` (1.5 points)

Le fichier `GNUmakefile` sert à fabriquer le modèle `tank`. Il fabrique tous les différents controleurs, par exemple avec une défaillance, avec deux défaillances ou avec une vanne virtuelle par exemple.

### 2.1.2 Rôle de la constante `nbFailures` et de l'assertion associée (0.5 point)

La constante nbFailures est utilisé dans le fichier `GNUmakefile` pour définir le nombre de contrôleurs à fabriquer, tandis que l'assertion associée sert a borner le nombre de vannes pouvant tomber en panne simultanement.

## 2.2 Résultats avec le contrôleur initial `Ctrl`

### 2.2.1 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 = 351
 * dec10 = 342
 * 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 = 102
 * dec10 = 68
 * 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 = 102
 * dec10 = 68
 * CCoupGagnant = 712
 */
```

**Interprétation des résultats** En ce qui concerne le système System0FCtrl, nous pouvons constater que le système est stable et efficace. Il n'y a pas de défaillance, et il n'y a pas de blocages. Meme si dans l'etat initial on a 86 sitations critiques, le controleur est capable de les regler sans probleme. Ce controleur n'a donc pas besoin d'optimisation, car il propose le debit ideal en evitant les blocages et situations critiques.

### 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 = 2205
 * dec10 = 2139
 * CCoupGagnant = 4950
 */

/*
 * Properties for node : System1FCtrl1F1I
 * # 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 = 695
 * dec10 = 443
 * CCoupGagnant = 2941
 */

/*
 * Properties for node : System1FCtrl1F2I
 * # 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 = 695
 * dec10 = 439
 * 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 = 695
 * dec10 = 439
 * CCoupGagnant = 2909
 */
```

**Interprétation des résultats** Le système System1FCtrl commence par ne pas avoir de blocages, mais des situations critiques. Apres une iteration, il est capable de diminuer le nombre de situations critiques, mais en contrepartie il en cree des blocages. Apres deux et trois iterations, le resultat devient un peu mauvais. 3 situations de blocage sont apparues. Il est important aussi de noter que tous les cas redoutés sont des blocages, dont environ 2/3 sont des situations critiques. Des optimisations sont nécessaires afin d'éviter ce grand nombre de blocages et de situations critiques.

### 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 = 5418
 * dec10 = 5228
 * 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 = 725
 * dec10 = 834
 * 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 = 578
 * dec10 = 824
 * 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 = 578
 * dec10 = 824
 * 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 = 578
 * dec10 = 824
 * CCoupGagnant = 2826
 */
```

**Interprétation des résultats** Pareillement au système System1FCtrl, le système System2FCtrl commence par ne pas avoir de blocages, mais des situations critiques. Après une iteration, il est capable de diminuer le nombre de situations critiques, mais en contrepartie il créé des blocages. Apres plusieurs iterations, le résultat devient un peu mauvais. Environ 40 situations de blocage sont apparues. Il est aussi important de noter que tous les cas redoutés sont des blocages, dont environ 1/3 sont des situations critiques, comparé a 2/3 du système System1FCtrl.Des optimisations sont nécessaires afin d'éviter ce grand nombre de blocages et de situations critiques.

**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 = 7242
 * dec10 = 6974
 * 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 = 84
 * dec10 = 96
 * 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
 */
```

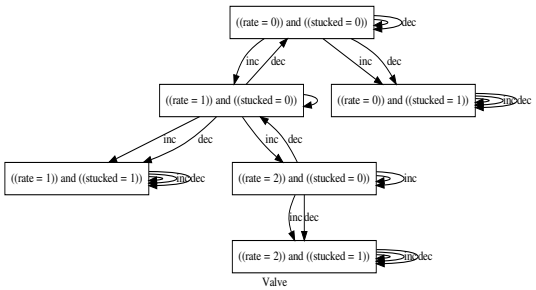
**Interprétation des résultats** Le système System3FCtrl commence par avoir beaucoup de sitations critiques, 617 pour être exact. Après une iteration, il est capable de neutraliser les situations critiques, mais en contrepartie il créé un petit nombre de blocages, comparé avec les autres systèmes. Après plusieurs iterations, le résultat devient plutôt bon. Le nombre de blocages diminue de 112 a 27. Des petites optimiations sont nécessaires afin d'éviter ce petit nombre de blocages, mais le système est encore assez efficace.

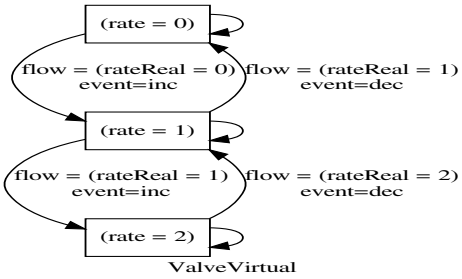
### 2.2.2 Bilan avec le contrôleur initial (1 point)

Le controleur initial permet de regler une bonne partie de situations critiques, mais il n'est pas capable de les regler toutes. En plus, il cree en contrepartie un bon nombre de blocages, suivant le nombre de situations critiques.

## 2.3 Construction d'un contrôleur initial plus performant

### 2.3.1 Rôle du composant `ValveVirtual`(2 points)




La semantique du composant `ValveVirtual` est le fait que nous pouvons pas utiliser la valve si elle est bloquée. Elle simule donc une valve reele parfaite. Le role de ce composant est donc d'eviter les utilisations inutiles de la valve tant qu'on ne peut pas changer le rate avec et de simuler les defaillances.

### 2.3.2 Rôle du composant `CtrlVV` (4 points)

Le rôle du composant `CtrlVV` est de simuler le contrôleur initial `Ctrl` avec une valve virtuelle. L'utilité de ce composant est de permettre de tester le contrôleur initial `Ctrl` avec une valve virtuelle, en evitant donc de faire des coups de changement qui menent a une erreur, par exemple un blogage ou une situation critique.

## 2.4 Résultats avec le contrôleur `CtrlVV`

### 2.4.1 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 = 213
 * dec10 = 208
 * 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 = 65
 * dec10 = 44
 * 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 = 65
 * dec10 = 44
 * CCoupGagnant = 362
 */
```

**Interprétation des résultats** Le système System0FCtrlVV est stable et efficace. Il commence par avoir 86 situations critiques. Au bout de la première itération, il est capable de régler toutes ces situations critiques sans générer de blocage ni de défaillance. Ce système est tres efficace, donc il n'y a pas besoin de l'optimiser.

### 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 = 910
 * dec10 = 888
 * 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 = 92
 * dec10 = 64
 * 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 = 52
 * dec10 = 36
 * 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 = 44
 * dec10 = 36
 * 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 = 44
 * dec10 = 36
 * CCoupGagnant = 392
 */
```

**Interprétation des résultats** Le système System1FCtrlVV commence par avoir beaucoup de situations critiques, 413 pour être exact. Au bout de la première itération, il est capable de régler presque toutes ces situations critiques en generant qu'un petit nombre de blocages. Au bout de la 3ème itération, il est capable de neutraliser le nombre de blocages. Ce système est assez efficace, mais il peut avoir une défaillance, donc il y a quand meme un besoin d'optimiser, afin de le rendre un peu plus stable et rapide.

### 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 = 1666
 * dec10 = 1625
 * 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 = 30
 * dec10 = 27
 * 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** Le système System2FCtrlVV commence par avoir vraiment beaucoup de situations critiques, 812 pour être exact. Au bout de la première itération, il est capable de régler presque toutes ces situations critiques en generant quand meme un bon nombre de blocages, 70 pour etre exact. Au bout de la 2ème itération, il est capable de neutraliser le nombre de blocages. Ce système est assez efficace, plus efficace que le systeme System1FCtrlVV mais il presente un risque plus haut de defaillance. Le besoin d'optimiser est donc plus important.

### 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 = 1938
 * dec10 = 1890
 * 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 = 26
```

```
 * dec10 = 16
 * 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** Le système System3FCtrlVV commence par avoir beaucoup de situations critiques, 970 pour être exact. Au bout de la première itération, il est capable de régler presque toutes ces situations critiques en génerant quand même un bon nombre de blocages, 97 blocages. Au bout de la 2ème itération, il est capable de neutraliser le nombre de blocages. Ce système est assez efficace, presque aussi efficace que le systeme System2FCtrlVV, mais il presente un risque plus haut de defaillance. Le besoin d'optimiser est donc encore plus important.

### 2.4.2 Bilan avec le contrôleur CtrlVV (1 point)

Le contrôleur CtrlVV permet d'avoir de meilleurs résultats que le contrôleur initial grâce a la Valve Virtuelle. Il est beaucoup plus stable et donne des résultats beaucoup plus cohérents. Il permet de totalement neutraliser les situations critiques et les blocages. Cela est dû au fait qu'il est capable de prevoir une situation de blocage, grace a la valve virtuelle.

## 2.5 Une première optimisation des contrôleurs pour améliorer le débit aval

### 2.5.1 Une optimisation basée sur les priorités (1 point)

```
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
```

### 2.5.2 Calcul des contrôleurs optimisés avec `CtrlVV`

**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 = 1
 * 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 = 16
 * dec10 = 14
 * 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
 */
```

### 2.5.3 Bilan avec la première optimisation du contrôleur CtrlVV (1 point)

## 2.6 Une deuxième optimisation (3 points)

Il est possible d'obtenir de meilleurs résultats que les précédents par au moins deux façons.

1. En utilisant un meilleur ordre pour les priorités entre événements.
2. En introduisant cet objectif dans le système de calcul de point fixe des actions du contrôleur.

Vous devez proposer une des deux optimisations conduisant à une solution dans laquelle le débit de la vanne aval est le moins souvent possible décrémenter, voire jamais. Pour cela, vous pouvez :

— Meilleur ordre sur les événements.

1. Modifier le fichier `ControleursOpt/Optimisation.alt`.
2. Faites `make`.
3. Quand vos résultats sont satisfaisants, notez les, puis copiez votre fichier dans `ControleursOpt/Optimisation-2.alt`.
4. Remettez le fichier `ControleursOpt/Optimisation.alt` d'origine.
5. Faites `make`.

— Meilleur système d'équations au point fixe.

1. Modifier le fichier `Spec/System.spe`.
2. Faites `make`.
3. Quand vos résultats sont satisfaisants, notez les, puis copiez votre fichier dans `Spec/System-2.spe`.
4. Remettez le fichier `Spec/System.spe` d'origine.
5. Faites `make`.

### Le nouvel ordre

```
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
```

### Le nouveau système d'équations

```
with SystemNbPannesFNomDuControleur do
  deadlock := any_s - src(any_t - self_epsilon);
  notResettable := any_s -coreach(initial,any_t);
  /* output */
  out0 := any_s & [V[2].rate=0];
  out1 := any_s & [V[2].rate=1];
  out2 := any_s & [V[2].rate=2];
  dec21 := any_t & label V[2].dec & rsrc(out2);
  dec10 := any_t & label V[2].dec & rsrc(out1);
  /* Niveaux critiques et Situations redoutees */
  NC := any_s & [T.level=0 | T.level=nbSensors];
  SR := deadlock | NC;
  EPerdu := any_s - SR;
  CGagne := any_s - SR;
  /* systeme d'equation au point fixe
  * CGagnant = CGagne & src(CCoupGagnant)
  * CCoupGagnant = CCoup & rtgt(EPerdant)
  * EPerdant = EPerdu & (src(ECoupPerdant) - src(ECoupNonPerdant))
  * ECoupPerdant = ECoup & rtgt(CGgagnant)
  * ECoupNonPerdant = ECoup & rtgt(any_s - CGagnant)
  */
  CCoup := any_t & label cmd;
  ECoup := any_t & label env;
  CCoupGagnant -= CCoup & rtgt(EPerdu & (src(ECoup & rtgt(CGagne & src(CCoupGagnant))) - src(EC
  /* Controller projection */
  /* Syntax and semantic of the "project" command
  * param 1 : les configurations projetees
  * param 2 : les transitions projetees
  *  - arg1 : projection existentielle
  *  - arg2 : projection universelle
  * param 3 : nom du noeud AltaRica genere
  * param 4 : simplification ou non des expressions booleennes
  * param 5 (optionel) : noeud de la hierarchie servant a la projection
  */
  /* Universal projection is use */
  project(any_s, (empty_t,CCoupGagnant), 'CtrlInitialNbPannesFNumIter', true,C)
```

```
        > 'Controleurs/CtrlInitialNbPannesFNumIter.alt';
  /* record results */
  show(any_s, any_t, deadlock, NC, SR , out0, out1, out2, dec21, dec10, CCoupGagnant) > 'Res/$N
done
```

## 2.7 Conclusion sur la synthèse de contrôleur (1 point)

Le contrôleur initial n'est pas tres efficace, mais il donne quand même des résultats convenables. Il peut être amélioré grâce a la valve virtuelle que le CtrlVV implémente. On peut voir que ce deuxième contrôleur est bien plus efficace et plus stable. Il permet de neutraliser les situations critiques et de régler les blocages. Il y a quand même plusieurs optimisations à faire afin d'éviter des défaillances, ainsi qu'améliorer le nombre d'itérations nécessaire pour régler les situations critiques et les blocages.