

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

HOCINI Mohamed Fouad

MAURICE Bastien



# 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 de 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 en travaux dirigés.

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

### 2.1 Rôle de la constante nbFailures (2 points)

La constante nbFailures permet de définir le nombre maximal de pannes pouvant survenir. Elle permet ainsi lors des tests, de limiter le nombre de configuration possibles. La condition  $\text{nbFailures} \geq (\text{V}[0].\text{stucked} + \text{V}[1].\text{stucked} + \text{V}[2].\text{stucked})$  ; permet d'assurer que le nombre de valves en panne ne peut pas dépasser nbFailures. Elle nous permettra de générer les contrôleurs avec une possibilité de défaillance contrôlé.

### 2.2 Résultats avec le contrôleur initial Ctrl

#### 2.2.1 Calcul d'un contrôleur

Avec 0 défaillance (1 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
 * CtrlCanControl = 27
 * CCoupGagnant = 1134
 * CCoupGagnantUtile = 712
 */

/*
 * 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
* CtrlCanControl = 27
* CCoupGagnant = 712
* CCoupGagnantUtile = 712
*/

```

**Interprétation des résultats** On observe que le contrôleur de base ne comporte pas de situation de blocage (DL) mais admet 86 état de niveau critique (NC), cependant dès la premières itérations sur le contrôleur celle ci sont supprimer et le contrôleur est correct; les itérations suivantes ne sont donc pas nécessaire.

#### Avec 1 défaillance (1 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
* CtrlCanControl = 81
* CCoupGagnant = 4950
* CCoupGagnantUtile = 3043
*/

/*
* Properties for node : System1FCtrlF1I
* # state properties : 7
*
* any_s = 244
* deadlock = 16
* NC = 0
* SR = 16
* out0 = 60
* out1 = 94
* out2 = 90
*
* # trans properties : 4
*
* any_t = 2022
* CtrlCanControl = 71
* CCoupGagnant = 1589
*/

```

```
* CCoupGagnantUtile = 1589
*/
```

**Interprétation des résultats** Dès l'apparition de défaillance sur les valves on constate que les NC augmente significativement (+243), et les itérations effectuées sur le contrôleur dans le but de réduire les NC provoquent l'apparition de DL, inévitable dû au fait que par exemple si la valve sortante se bloque avec un débit nul le système se bloque.

#### Avec 2 défaillances (1 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
* CtrlCanControl = 117
* CCoupGagnant = 7533
* CCoupGagnantUtile = 4654
*/

/*
* Properties for node : System2FCtrl2F1I
* # state properties : 7
*
* any_s = 196
* deadlock = 55
* NC = 0
* SR = 55
* out0 = 48
* out1 = 84
* out2 = 64
*
* # trans properties : 4
*
* any_t = 1410
* CtrlCanControl = 72
* CCoupGagnant = 801
* CCoupGagnantUtile = 618
*/
```

**Interprétation des résultats** Comme dans le cas précédent où le système a une défaillance, on observe aussi l'apparition de blocage et de situation critique, donc les remarques précédentes s'appliquent aussi dans ce cas.

### Avec 3 défaillances (1 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
 * CtrlCanControl = 125
 * CCoupGagnant = 7908
 * CCoupGagnantUtile = 5029
 */

/*
 * Properties for node : System3FCtrl3F1I
 * # state properties : 7
 *
 * any_s = 156
 * deadlock = 70
 * NC = 0
 * SR = 70
 * out0 = 36
 * out1 = 76
 * out2 = 44
 *
 * # trans properties : 4
 *
 * any_t = 1124
 * CtrlCanControl = 64
 * CCoupGagnant = 343
 * CCoupGagnantUtile = 343
 */
```

**Interprétation des résultats** On remarque que les itérations amène le contrôleur à complètement éviter les cas où la vanne de sortie est ouverte à 2, car les deux vannes d'entrée pourraient tomber en panne. On note que les deadlocks doivent correspondre aux cas où toutes les vannes sont fermées et bloquées.

### 2.2.2 Calcul des contrôleurs optimisés (2 points)

Il a fallu intégrer la gestion des pannes en observant les pannes survenant lors du fonctionnement du système. Une simple assert dans le composant Système et notre contrôleur est capable de détecter les pannes.

## 2.3 Rôle des composants ValveVirtual et CtrlVV (4 points)

Le composant ValveVirtual comme son nom l'indique sert à simuler le comportement d'une vanne en parfait état et stopper son utilisation lorsque cette dernière est défaillante. L'utilisation des ValveVirtual



au sein du CtrlVV permet de gérer les pannes pouvant intervenir lors de l'utilisation d'une vanne afin d'éviter des situations de blocages et critiques (atteinte de la zone 0 ou de la zone nbSensors). Il est préférable de l'utiliser par rapport au contrôleur initial car le comportement du contrôleur est aléatoire et ne gère pas le cas des pannes.

Il existe moins de transitions dans un système avec valve virtuel car le contrôleur est capable de détecter les pannes et donc ne cherche pas à activer une valve bloquée ; on peut le vérifier en observant par exemple que pour le System2FCtrlVV any\_t = 15894 alors que pour le System2FCtrl any\_t = 44608.

## 2.4 Résultats avec le contrôleur initial CtrlVV

### 2.4.1 Calcul d'un contrôleur

Avec 0 défaillance (1 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
 * CtrlCanControl = 8
 * CCoupGagnant = 548
 * CCoupGagnantUtile = 362
 */

/*
 * 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
 * CtrlCanControl = 8
 * CCoupGagnant = 362
 * CCoupGagnantUtile = 362
 */
```

**Interprétation des résultats** On peut observer que l'utilisation de valve virtuelle n'impacte pas les débits ou le nombre de situations redoutées mais seulement le nombre de transitions possible sur le cas de 0 défaillance.

Dans le cas où les vannes ne sont pas défaillantes, on constate qu'il est possible d'éviter que le système se bloque, ainsi que les situations critiques.

On peut donc déduire qu'une des caractéristique de ce contrôleur est qu'il est capable de gérer les pannes.

#### Avec 1 défaillance (1 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
 * CtrlCanControl = 20
 * CCoupGagnant = 1866
 * CCoupGagnantUtile = 1186
 */

/*
 * 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
 * CtrlCanControl = 17
 * CCoupGagnant = 546
 * CCoupGagnantUtile = 546
 */
```

**Interprétation des résultats** D'après les résultats, il est possible de détecter que le système soit bloquer et les situations critiques pouvant survenir au cours du fonctionnement du système.

Cette situation est dû au faite que le contrôleur à vannes simulées gère le cas où le système est défaillant en bloquant la vanne en cas de défaillance.

#### Avec 2 défaillances (1 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
* CtrlCanControl = 26
* CCoupGagnant = 2360
* CCoupGagnantUtile = 1529
*/

/*
* 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
* CtrlCanControl = 12
* CCoupGagnant = 155
* CCoupGagnantUtile = 99
*/

```

**Interprétation des résultats** Le contrôleur n'a plus que 2 états, et dans les deux, la valve 2 est toujours sur 0. C'est donc que le tank ne reçoit jamais d'eau. Il ne trouve qu'un seul coup gagnant qui est de ne rien faire.

**Avec 3 défaillances (1 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
* CtrlCanControl = 27
* CCoupGagnant = 2384
* CCoupGagnantUtile = 1553
*/

/*
* 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
* CtrlCanControl = 8
* CCoupGagnant = 27
* CCoupGagnantUtile = 27
*/

```

**Interprétation des résultats** Comme dans les deux cas précédents où le système a une ou deux défaillance(s), il est possible de détecter l'apparition de blocages et de situations critiques, donc les remarques précédentes s'appliquent aussi dans ce cas.

### 2.4.2 Calcul des contrôleurs optimisés (2 points)

Pour optimiser ce contrôleur nous avons du intégrer une vanne qui ne tombe jamais en panne pour rendre le système plus robuste au lieu de bloquer la vanne comme le fait le contrôleur précédent.

#### Avec 0 défaillance

```

/*
* Properties for node : System0FCtrlVR
* # state properties : 7
*
* any_s = 5481
* deadlock = 0
* NC = 1988
* SR = 1988
* out0 = 1578
* out1 = 1875
* out2 = 2028
*
* # trans properties : 4
*
* any_t = 75924
* CtrlCanControl = 27
* CCoupGagnant = 20356
* CCoupGagnantUtile = 13255
*/

```

```

/*
 * Properties for node : System0FCtrlVR0F1I
 * # state properties : 7
 *
 * any_s = 1710
 * deadlock = 0
 * NC = 0
 * SR = 0
 * out0 = 366
 * out1 = 630
 * out2 = 714
 *
 * # trans properties : 4
 *
 * any_t = 15955
 * CtrlCanControl = 27
 * CCoupGagnant = 13255
 * CCoupGagnantUtile = 13255
 */

```

```

/*
 * Properties for node : System0FCtrlVR0F2I
 * # state properties : 7
 *
 * any_s = 1710
 * deadlock = 0
 * NC = 0
 * SR = 0
 * out0 = 366
 * out1 = 630
 * out2 = 714
 *
 * # trans properties : 4
 *
 * any_t = 15955
 * CtrlCanControl = 27
 * CCoupGagnant = 13255
 * CCoupGagnantUtile = 13255
 */

```

```

/*
 * Properties for node : System0FCtrlVR0F3I
 * # state properties : 7
 *
 * any_s = 1710
 * deadlock = 0
 * NC = 0
 * SR = 0
 * out0 = 366
 * out1 = 630
 * out2 = 714
 *
 * # trans properties : 4
 *
 * any_t = 15955
 * CtrlCanControl = 27

```

```

* CCoupGagnant = 13255
* CCoupGagnantUtile = 13255
*/

/*
* Properties for node : System0FCtrlVR0F4I
* # state properties : 7
*
* any_s = 1710
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 366
* out1 = 630
* out2 = 714
*
* # trans properties : 4
*
* any_t = 15955
* CtrlCanControl = 27
* CCoupGagnant = 13255
* CCoupGagnantUtile = 13255
*/

```

#### **Avec 1 défaillance**

```

/*
* Properties for node : System1FCtrlVR
* # state properties : 7
*
* any_s = 10986
* deadlock = 0
* NC = 3911
* SR = 3911
* out0 = 3182
* out1 = 3772
* out2 = 4032
*
* # trans properties : 4
*
* any_t = 284886
* CtrlCanControl = 81
* CCoupGagnant = 64784
* CCoupGagnantUtile = 42279
*/

/*
* Properties for node : System1FCtrlVR1F1I
* # state properties : 7
*
* any_s = 1838
* deadlock = 115
* NC = 0
* SR = 115
* out0 = 406
* out1 = 746
* out2 = 686

```

```

*
* # trans properties : 4
*
* any_t = 17741
* CtrlCanControl = 71
* CCoupGagnant = 13911
* CCoupGagnantUtile = 13911
*/

/*
* Properties for node : System1FCtrlVR1F2I
* # state properties : 7
*
* any_s = 1588
* deadlock = 47
* NC = 0
* SR = 47
* out0 = 310
* out1 = 656
* out2 = 622
*
* # trans properties : 4
*
* any_t = 14643
* CtrlCanControl = 50
* CCoupGagnant = 11875
* CCoupGagnantUtile = 11439
*/

/*
* Properties for node : System1FCtrlVR1F3I
* # state properties : 7
*
* any_s = 1448
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 310
* out1 = 648
* out2 = 490
*
* # trans properties : 4
*
* any_t = 13567
* CtrlCanControl = 50
* CCoupGagnant = 11244
* CCoupGagnantUtile = 11244
*/

/*
* Properties for node : System1FCtrlVR1F4I
* # state properties : 7
*
* any_s = 1448
* deadlock = 0
* NC = 0

```

```

* SR = 0
* out0 = 310
* out1 = 648
* out2 = 490
*
* # trans properties : 4
*
* any_t = 13567
* CtrlCanControl = 50
* CCoupGagnant = 11244
* CCoupGagnantUtile = 11244
*/

```

**Avec 2 défaillances (1 point)**

```

/*
* Properties for node : System2FCtrlVR
* # state properties : 7
*
* any_s = 12825
* deadlock = 0
* NC = 4529
* SR = 4529
* out0 = 3732
* out1 = 4401
* out2 = 4692
*
* # trans properties : 4
*
* any_t = 487710
* CtrlCanControl = 117
* CCoupGagnant = 84707
* CCoupGagnantUtile = 55556
*/

/*
* Properties for node : System2FCtrlVR2F1I
* # state properties : 7
*
* any_s = 1160
* deadlock = 403
* NC = 0
* SR = 403
* out0 = 264
* out1 = 604
* out2 = 292
*
* # trans properties : 4
*
* any_t = 11862
* CtrlCanControl = 72
* CCoupGagnant = 5817
* CCoupGagnantUtile = 4284
*/

/*
* Properties for node : System2FCtrlVR2F2I

```



```

* # state properties : 7
*
* any_s = 174
* deadlock = 12
* NC = 0
* SR = 12
* out0 = 148
* out1 = 26
* out2 = 0
*
* # trans properties : 4
*
* any_t = 2229
* CtrlCanControl = 26
* CCoupGagnant = 1890
* CCoupGagnantUtile = 1890
*/

/*
* Properties for node : System2FCtrlVR2F3I
* # state properties : 7
*
* any_s = 150
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 148
* out1 = 2
* out2 = 0
*
* # trans properties : 4
*
* any_t = 2007
* CtrlCanControl = 26
* CCoupGagnant = 1782
* CCoupGagnantUtile = 1782
*/

/*
* Properties for node : System2FCtrlVR2F4I
* # state properties : 7
*
* any_s = 150
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 148
* out1 = 2
* out2 = 0
*
* # trans properties : 4
*
* any_t = 2007
* CtrlCanControl = 26
* CCoupGagnant = 1782
* CCoupGagnantUtile = 1782
*/

```

**Avec 3 défaillances (1 point)**

```
/*
 * Properties for node : System3FCtrlVR
 * # state properties : 7
 *
 * any_s = 13030
 * deadlock = 0
 * NC = 4595
 * SR = 4595
 * out0 = 3796
 * out1 = 4470
 * out2 = 4764
 *
 * # trans properties : 4
 *
 * any_t = 554686
 * CtrlCanControl = 125
 * CCoupGagnant = 86894
 * CCoupGagnantUtile = 57743
 */

/*
 * Properties for node : System3FCtrlVR3F1I
 * # state properties : 7
 *
 * any_s = 1108
 * deadlock = 490
 * NC = 0
 * SR = 490
 * out0 = 240
 * out1 = 596
 * out2 = 272
 *
 * # trans properties : 4
 *
 * any_t = 11406
 * CtrlCanControl = 64
 * CCoupGagnant = 3375
 * CCoupGagnantUtile = 3375
 */

/*
 * Properties for node : System3FCtrlVR3F2I
 * # state properties : 7
 *
 * any_s = 128
 * deadlock = 0
 * NC = 0
 * SR = 0
 * out0 = 128
 * out1 = 0
 * out2 = 0
 *
 * # trans properties : 4
 *
```

```

* any_t = 1920
* CtrlCanControl = 27
* CCoupGagnant = 1728
* CCoupGagnantUtile = 1728
*/

/*
* Properties for node : System3FCtrlVR3F3I
* # state properties : 7
*
* any_s = 128
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 128
* out1 = 0
* out2 = 0
*
* # trans properties : 4
*
* any_t = 1920
* CtrlCanControl = 27
* CCoupGagnant = 1728
* CCoupGagnantUtile = 1728
*/

/*
* Properties for node : System3FCtrlVR3F4I
* # state properties : 7
*
* any_s = 128
* deadlock = 0
* NC = 0
* SR = 0
* out0 = 128
* out1 = 0
* out2 = 0
*
* # trans properties : 4
*
* any_t = 1920
* CtrlCanControl = 27
* CCoupGagnant = 1728
* CCoupGagnantUtile = 1728
*/

```

## 2.5 Conclusion (2 points)

On observe que les itérations sur le contrôleur de base permettent de réduire le nombre de situations redoutées et d'améliorer les contrôleurs en réduisant les transitions possible, mais dès lors que le nombre de défaillances est trop élevé, les contrôleurs obtenus de manière automatique ne permet que l'arrêt du système dès son lancement pour éviter les situations de blocage.

Notre choix final portera sur le System3FCtrlVR car il permet d'éviter à la fois les situation redoutées tels que le blocage du système et les niveaux critiques. De plus il nous permet de maximiser le débit de la vanne aval en utilisant les priorités d'Altatica. Le but est de privilégier l'augmentation du débit de la vanne aval par rapport aux autres actions, et privilégier la stagnation du débit par rapport à sa

diminution. Une autre alternative aurait été de réparer la vanne mais les contraintes du cahier des charges ne le permettent pas.