

Sureté de fonctionnement

Compte rendu de TP

ISN 2A

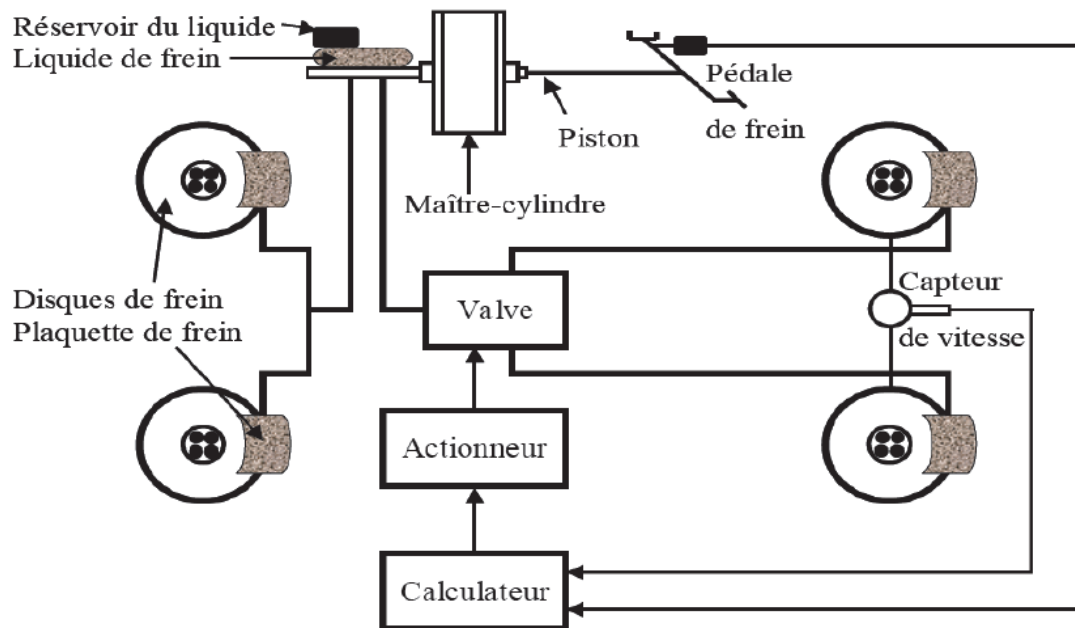
Mokhtari Mohamed

Wiat Jean-Baptiste

Objectif de TP

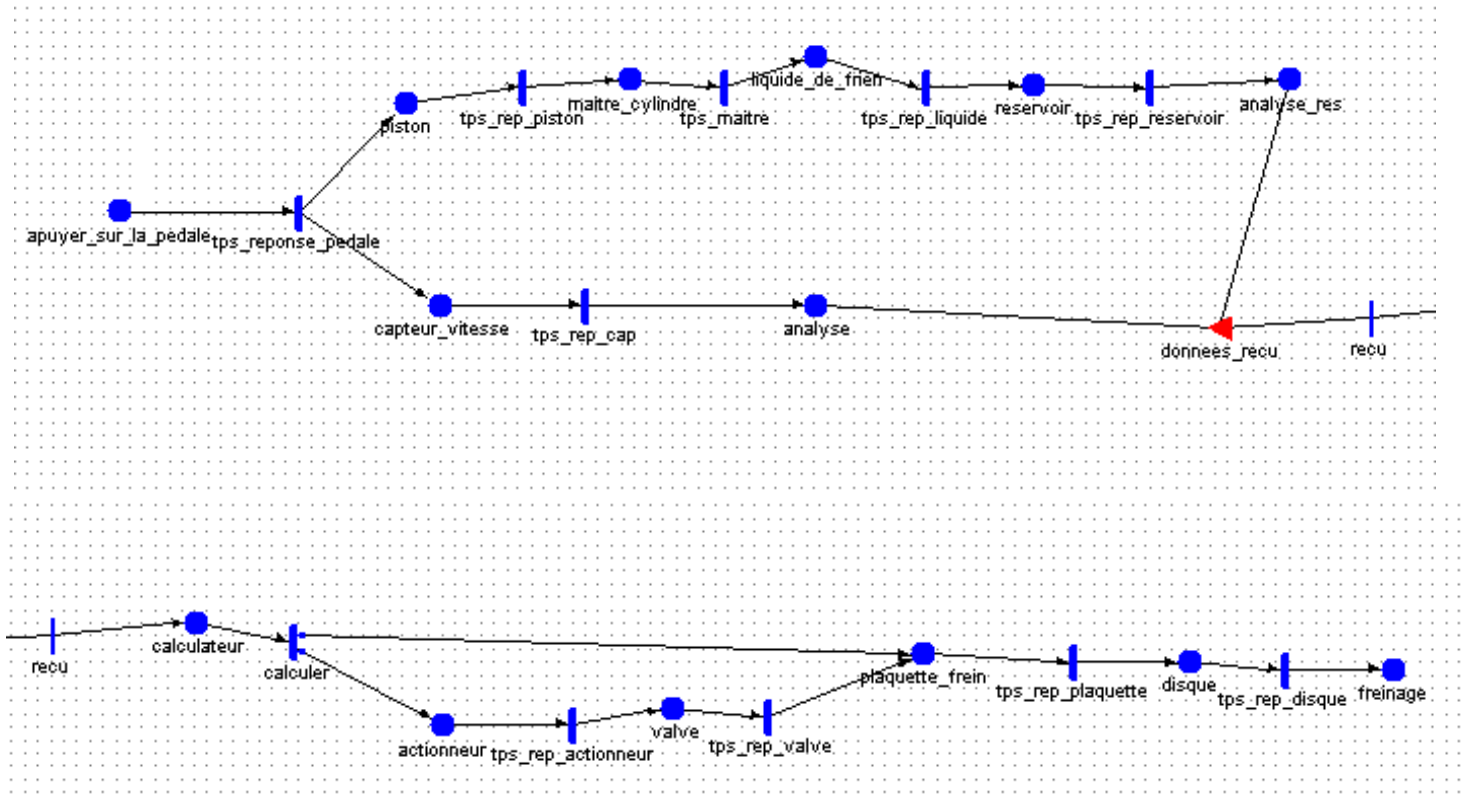
Le but de cet ensemble de TP est de modéliser un système de freinage avec et sans ABS. Pour pouvoir modéliser ce système nous utilisons le logiciel Möbius.

Le système que nous allons modéliser est défini de la manière suivante :



I. Diagramme de fiabilité et de dysfonctionnement :

Pour pouvoir étudier le système de freinage nous allons établir sur le logiciel Möbius le diagramme de fiabilité. Pour cela nous utilisons le schéma ci-dessus ainsi que le tableau des paramètres des distributions du modèle fonctionnel fournissant le temps de réponse de chaque composant. De même le cahier des charges nous impose la fréquence d'utilisation d'un mode de freinage ou l'autre.



Pour le modèle de dysfonctionnement il faut compléter le schéma ci-dessous à l'aide du tableau de défaillance.

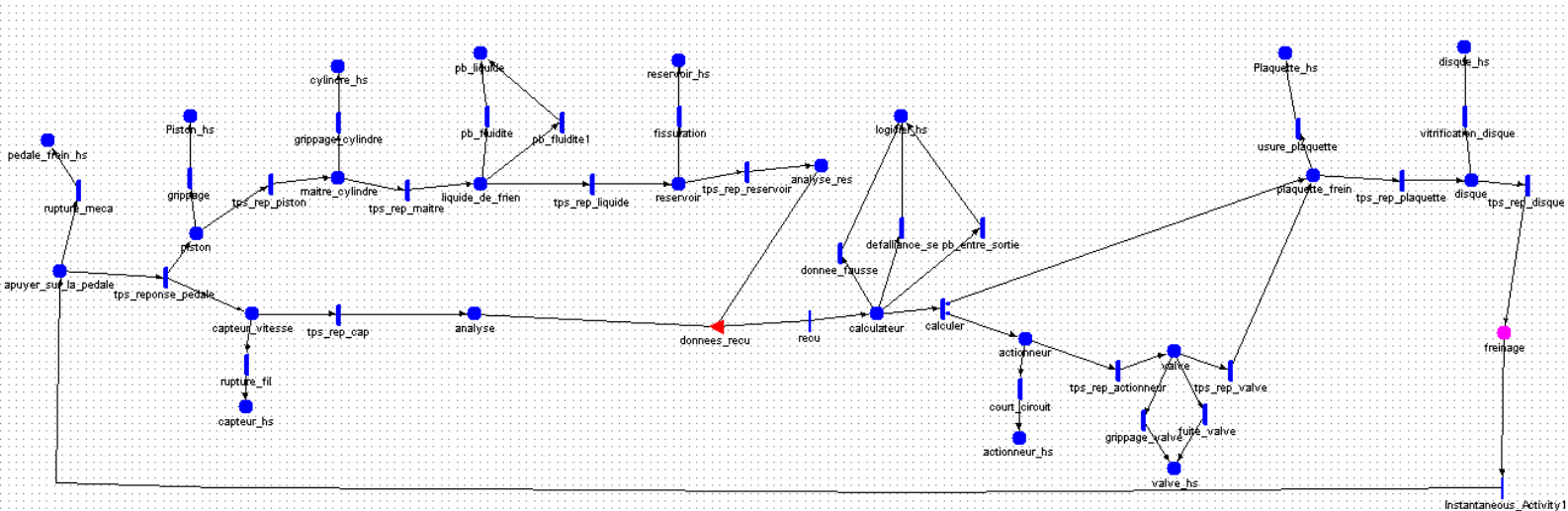


Schéma SAN 'ABS_ dysfonctionnement' modélisant le dysfonctionnement

Le système est fiable lorsqu'il n'y a pas de jeton dans les places suivantes:

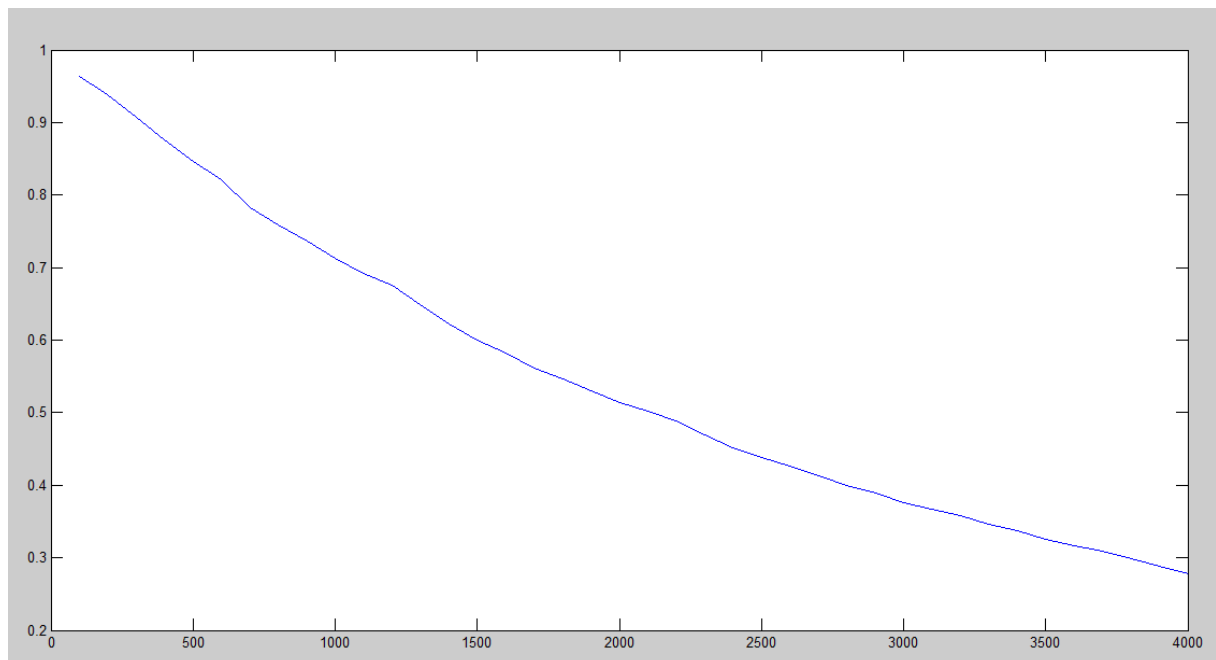
Piston_hs, pedale_frein_hs, pb_liquide, cylindre_hs, reservoir_hs, capteur_hs, actionneur_hs, valve_hs, Plaquette_hs, disque_hs et logiciel_hs.

Modéliser dans la rubrique reward sous Möbius :

Reward Function

```
if (ABS_disfonctionnel->pedale_frein_hs->Mark()==0 &&  
    ABS_disfonctionnel->Piston_hs->Mark()==0 &&  
    ABS_disfonctionnel->pb_liquide->Mark()==0 &&  
    ABS_disfonctionnel->cylindre_hs->Mark()==0 &&  
    ABS_disfonctionnel->reservoir_hs->Mark()==0 &&  
    ABS_disfonctionnel->capteur_hs->Mark()==0 &&  
    ABS_disfonctionnel->actionneur_hs->Mark()==0 &&  
    ABS_disfonctionnel->valve_hs->Mark()==0 &&  
    ABS_disfonctionnel->Plaquette_hs->Mark()==0 &&  
    ABS_disfonctionnel->disque_hs->Mark()==0 &&  
    ABS_disfonctionnel->logiciel_hs->Mark()==0)  
{  
    return 1;  
}
```

Grace au logiciel Matlab nous traçons la courbe de fiabilité :



On remarque une diminution de la fiabilité aux cours du temps.

II Etude du système avec $T=2000$ u.t

Le nombre de cycles de sollicitations s'obtient en créant un reward et en ajoutant 'return 1'. il retourne 1 lorsque la transition Instantaneous_Activity1 est franchie.

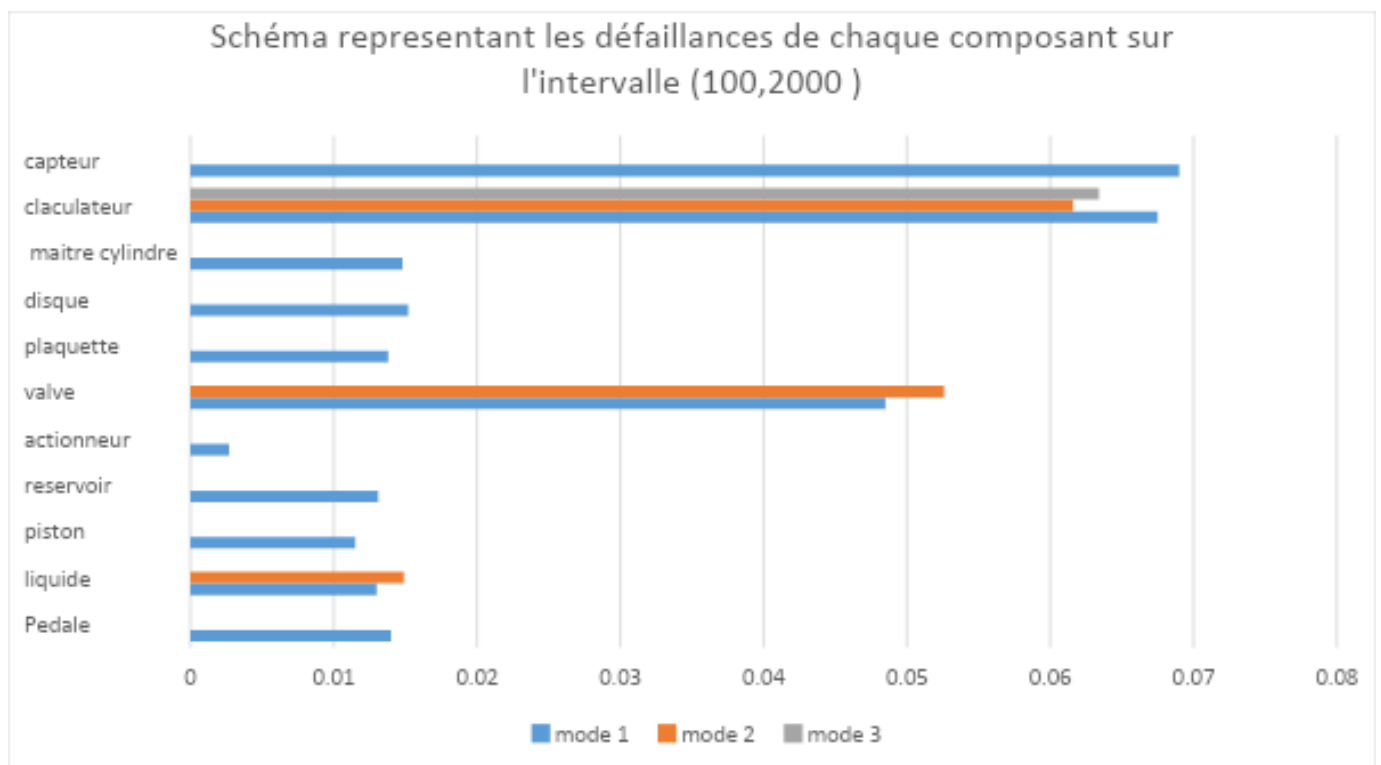
Nous obtenons ainsi 13.53 cycle.

De même pour avoir le nombre Le taux de freinage traditionnel et le taux de freinage avec ABS nous créons un reward sur les transitions et nous obtenons 0.6965 avec ABS et 13.215 sans ABS. On constate que la somme des deux taux n'est pas égale au nombre de cycles car nous n'avons pris pas en compte toutes les défaillances.

De même le cahier des charges est respecté car le taux de freinage avec ABS est de 5% et de 95 % sans ABS.

Maintenant, nous allons voir la contribution de chaque composant dans la défaillance du système.

Pour cela nous créons un Reward 'contribution_element' contenant ainsi 15 variables représentant le franchissement d'une transition impliquant une défaillance du système.



De ce graphique on montre que le calculateur, le capteur et la valve sont les éléments les moins fiables alors que l'actionneur est le plus fiable.

III Etude du système avec $T=4000$ u.t

Nous reprenons les méthodes de résolution de la partie 2. Cependant nous simulerons ces études sur intervalle de temps de (100 ; 4000).

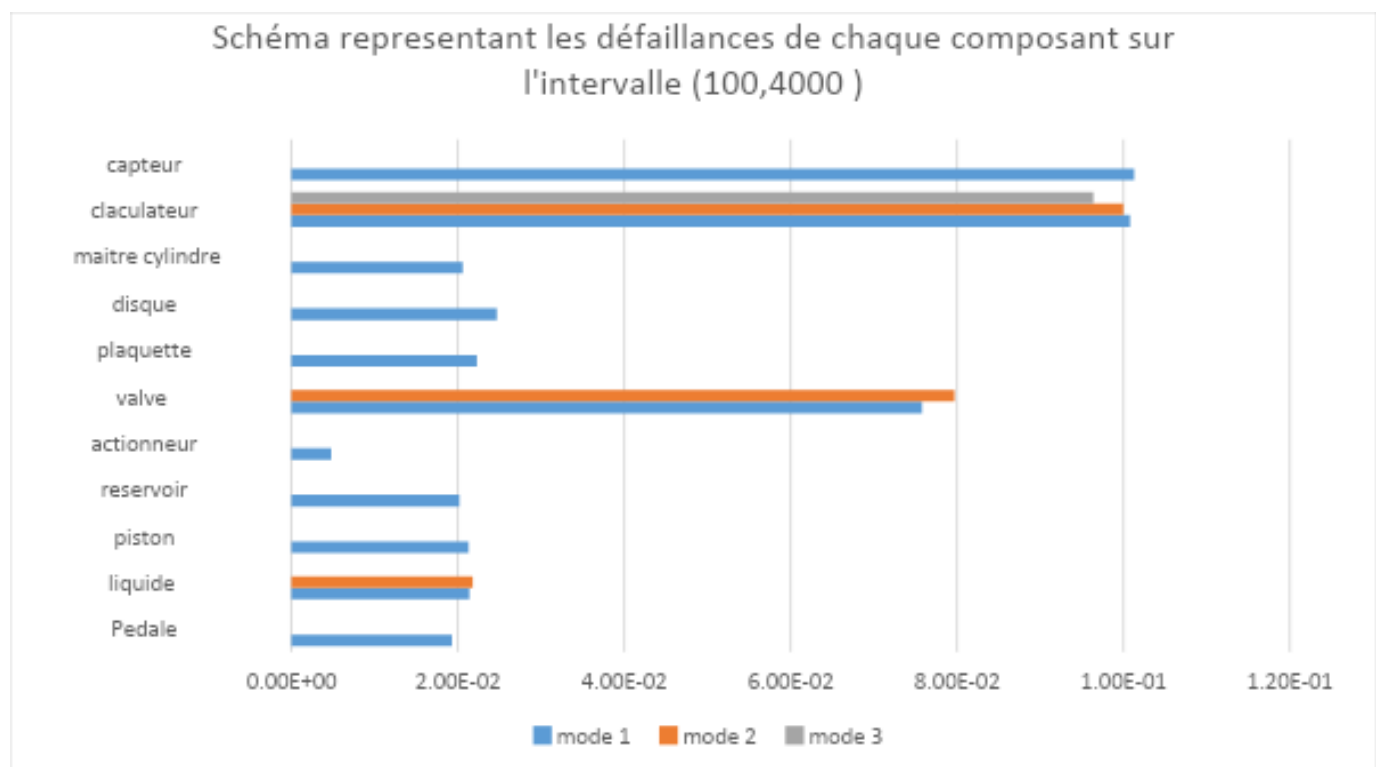
Pour cet intervalle nous remarquons que le cycle de sollicitations du frein est de 19.681.

Vérifions si la propriété suivante est vraie :

il y a 95% de chance d'avoir un freinage traditionnel et 5% d'avoir un freinage ABS. Lors de cette simulation nous obtenons :

18.45 avec un freinage sans ABS soit 95.2 % et 0.943 avec freinage avec ABS soit 4.8%. La propriété reste vérifiée.

De même regardons la contribution de chaque élément :



Comme la partie 2, on constate que le calculateur, le capteur et la valve sont les moins fiables cependant on constate aussi une augmentation de la probabilité de défaillance.

A l'aide de la loi uniforme de paramètre (5,7) nous allons modéliser la réparation du logiciel.

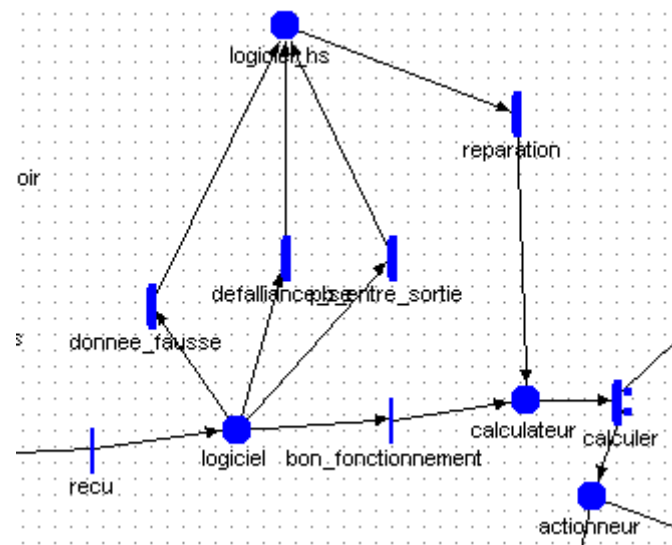
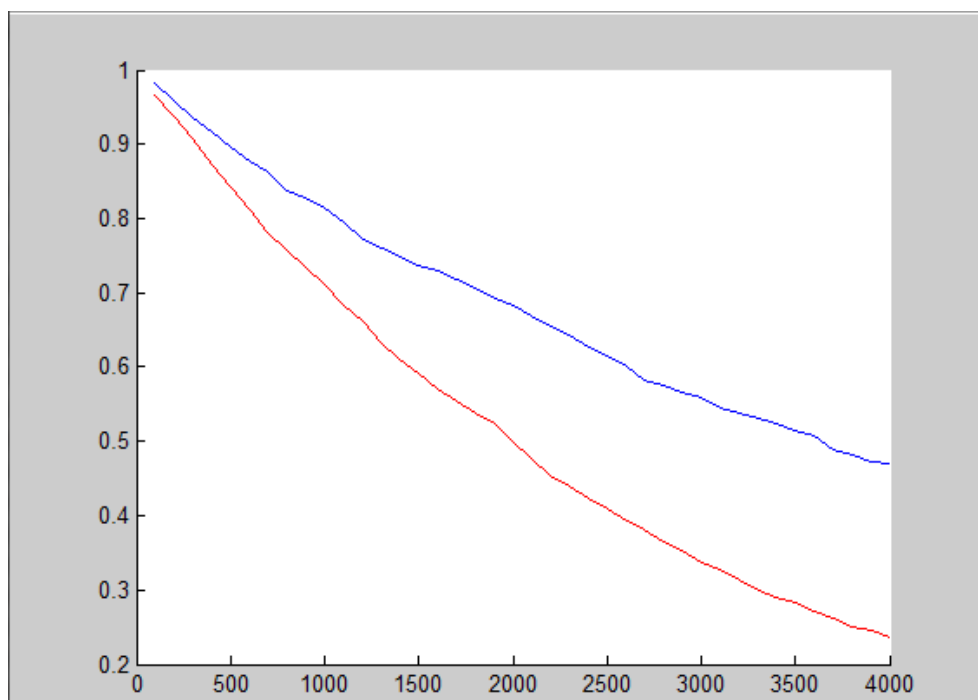


Schéma SAN 'ABS_reparation' incluant la réparation du logiciel

Nous traçons ainsi la fiabilité : (En rouge la courbe de fiabilité sans réparation et en bleu la courbe avec réparation.)



On remarque une amélioration de la fiabilité lorsque nous rajoutons une réparation.

De même l'erreur entre les deux courbes augmente aux cours du temps :

