Realizzazione di un ambiente di Fault Injection per applicazione ridondata

Progetto per il corso di *Programmazione di Sistema*

Carlo Migliaccio Federico Pretini Alessandro Scavone Mattia Viglino

Corso di Laurea Magistrale in Ingegneria Informatica
Politecnico di Torino

Gennaio 2025
Anno Accademico 2024/2025

Fault Injection Environment per Applicazione ridondata

1. Introduzione
2. Irrobustimento codice
   - Creazione del tipo `Hardened<T>`
   - Refactoring degli algoritmi di base
3. Fault Injection Environment
   - Fault List Manager
   - Injector
   - Analizer

Fault Injection Environment per Applicazione ridondata

Prova uso listings

...di Alex Scavonx

Il tipo `Hardened` è definito come segue:

```
#[derive(Clone,Copy)]
pub struct Hardened<T>{
    cp1: T,
    cp2: T,
}
```

Inoltre è implementato come segue:

```
impl<T> Hardened<T>{
    ... //Implementation
}

```

Implementazione tratto

```
impl<T> Sub for Hardened<T>
where T:Sub<Output=T>+PartialEq+Eq+Debug+Copy+Clone{
    type Output=Result<Hardened<T>,IncoherenceError>;
    fn sub(self, rhs: Self) -> Self::Output {
        if self.incoherent() || rhs.incoherent(){
            return Err(IncoherenceError::Generic)
        }
        Ok(Self{
            cp1: self.cp1 - rhs.cp1,
            cp2: self.cp2 - rhs.cp2,
        })
    }
}

```

Tipo IncoherenceError

```
#[derive(Error, Debug, Clone)]
pub enum IncoherenceError{
    #[error("IncoherenceError::AssignFail: assignment failed")]
    AssignFail,
    #[error("IncoherenceError::AddFail: due to incoherence add failed")]
    AddFail,
    #[error("IncoherenceError::MulFail: due to incoherence mul failed")]
    MulFail,
    #[error("IncoherenceError::Generic: generic incoherence error")]
    Generic
}

```

Introduzione all'argomento

Blocco

Esempio di utilizzo di blocchi e elenchi numerati

1. Primo item
2. Secondo item

ì

Tre regole

Sottotitolo

Data redundancy

Tre **regole fondamentali** per l'irrobustimento del codice:

1. Ogni operazione di lettura deve essere preceduta da un controllo di consistenza;
2. Ogni scrittura deve essere eseguita sulle due copie

Fault Injection Environment per Applicazione ridondata

Sample frame title

This is some text in the first frame. This is some text in the first frame. This is some text in the first frame.

Sample frame title

This is some text in the first frame. This is some text in the first frame. This is some text in the first frame.
This is an example of an highlighted text

Trasformata di Laplace

La **Trasformata di Laplace** è una trasformata integrale dal dominio $t \in \mathbb{R}$ al dominio $s \in \mathbb{C}$. Riportiamo per semplicità di seguito la sua definizione.

$$\mathcal{L}\{f(t)\}(s) = \int_0^{+\infty} f(t)e^{-st}dt \tag{1}$$

Definition (**State Space Representation**)

Un sistema lineare tempo invariante (LTI) può avere nello spazio di stato la seguente rappresentazione.

$$\begin{aligned} x(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{2}$$

Example

Questo è un esempio

Importante!

Questo è un alert block

Fault Injection Environment per Applicazione ridondata

Regression form

Definition (Feasible Parameter Set)

The **The Feasible Parameter Set (FPS)** is the set of parameters θ which are consistent with the a-priori and a-posteriori information.

$$\begin{aligned}\mathcal{D}_\theta = \{\theta \in \mathbb{R}^p : \tilde{y}(k) = &-\theta_1 y(k-1) + -\theta_2 y(k-2) + \theta_3 u(k) \\ &+ \theta_4 u(k-1) + \theta_5 u(k-2) + e(k),\ k = 3, ..., H \\ &|e(k)| \leq \Delta_e,\ k = 1, ..., H\}\end{aligned} \tag{3}$$

Under the assumption of **Equation Error (EE)** noise structure the computation of the PUIs becomes a couple of LP problems.

$$PUI_{\theta_j} = [\underline{\theta}_j, \overline{\theta}_j], \tag{4}$$

$$\underline{\theta}_j = \min_{\theta \in \mathcal{D}_\theta} \theta_j, \quad \overline{\theta}_j = \max_{\theta \in \mathcal{D}_\theta} \theta_j \tag{5}$$

Fault Injection Environment per Applicazione ridondata

Fault Injection Environment per Applicazione ridondata

1. Introduzione
2. Irrobustimento codice
   - Creazione del tipo `Hardened<T>`
   - Refactoring degli algoritmi di base
3. Fault Injection Environment
   - Fault List Manager
   - Injector
   - Analizer

Fault Injection Environment per Applicazione ridondata

1. Introduzione
2. Irrobustimento codice
   - Creazione del tipo `Hardened<T>`
   - Refactoring degli algoritmi di base
3. Fault Injection Environment
   - Fault List Manager
   - Injector
   - Analizer

Fault Injection Environment per Applicazione ridondata

1. Introduzione
2. Irrobustimento codice
   - Creazione del tipo `Hardened<T>`
   - Refactoring degli algoritmi di base
3. Fault Injection Environment
   - Fault List Manager
   - Injector
   - Analizer

Fault Injection Environment per Applicazione ridondata

Sample frame title

This is some text in the first frame. This is some text in the first frame. This is some text in the first frame.