

Kurzanleitung: ParameterSweepSequential Simulationsskript

15.07.2024

Inhaltsverzeichnis

1	Setup	3
2	Skriptaufruf	3
3	Skriptkonfiguration	3
3.1	Benutzerdefinierte Einstellungen	3
3.2	Benutzerdefinierte Operationen	4
4	Skriptablauf	4
5	Anpassungen	4
6	Beispielausgabe	5
7	Beispiel-Verzeichnisstruktur	5
8	Hinweise	5
9	Fehlerbehebung	5
10	Anpassung für andere Modelle	6
11	Zusammenfassung	6

Im Rahmen der Test- und Validierungsarbeiten wurde auch eine Parametrisierungsschnittstelle entwickelt. Jenes PLECS-Simulationsskript führt einen mehrdimensionalen Parameter-Sweep durch und speichert die zeitlich definierten Scopedaten in einer CSV-Datei. Die Parameter können frei gewählt werden. Mit Hilfe der Scope-Simulationsdaten können die Ergebnisse analysiert und visualisiert werden, um ein optimales Wertevektor der Eingaben für ein gegebenes Kriterium zu finden.

Diese Anleitung beschreibt, wie das PLECS Simulationsskript `ParameterSweepSequential` verwendet wird, um Parameter-Sweeps durchzuführen und die Ergebnisse zu analysieren.

1 Setup

1. **Modelldatei:** Sicherstellen, dass die PLECS-Modell-Datei (`pv_lut.plecs`) mit dem enthaltenen Skript `ParameterSweepSequential` vorhanden ist.
2. **Ausgabeverzeichnis:** Das Skript speichert Ausgabedateien (CSV und BMP) im angegebenen Ausgabeverzeichnis. Schreibrechte für dieses Verzeichnis sicherstellen.

2 Skriptaufruf

1. **PLECS öffnen** und das Simulationsmodell (`pv_lut`) laden.
2. **Das Skript starten:**
 - Gehe zu `Simulation -> Simulation Scripts` im Menü.
 - Alternativ: `Strg + Shift + T` drücken.
3. **Das Skript `ParameterSweepSequential` auswählen und ausführen.**

3 Skriptkonfiguration

3.1 Benutzerdefinierte Einstellungen

1. **Pfadangaben:**

```
modelPath = 'pv_lut';  
% Pfad zum PLECS-Modell  
scopePath = 'Scope';  
% Pfad zum Scope innerhalb des Modells  
outPath = '../out/automated_testing';  
% Pfad zum Speichern der Ausgabedateien
```

2. **Parameter-Sweep:** Die zu variierenden Parameter und ihre Bereiche definieren:

```
paramNames = { 'fPWM', 'fN' };  
% Namen der Parameter
```

```
paramRanges = 40000:1000:41000, 500:100:600; % Bereiche für jeden Parameter
```

3. **Signal- und Analyse-Einstellungen:** Das zu messende Signal und das Analysezeitintervall definieren:

```
signal = 8;  
% Signalnummer im Scope  
timeInterval = [0.75, 0.8];  
% Zeitintervall in Sekunden
```

4. **Ausgabedateien:** Angeben, ob BMP-Dateien exportiert werden sollen:

```
bmp = true;  
% BMP-Dateien exportieren, wenn true
```

3.2 Benutzerdefinierte Operationen

Die Funktion `customOperations` anpassen, um zusätzliche Operationen auf den gemessenen Scope-Daten durchzuführen. Die bereitgestellte Funktion berechnet den Durchschnitt des Signals über das angegebene Zeitintervall und gibt die Ergebnisse aus.

4 Skriptablauf

1. **Initialisierung:**

- Konsolenausgabe in Octave löschen.
- Sicherstellen, dass das Ausgabe-Verzeichnis (`outPath`) existiert.

2. **Parameterkombinationen generieren:**

- Alle Kombinationen der Parameterwerte erstellen.

3. **Parameter-Sweep durchführen:**

- Jede Parameterkombination wird gesetzt und das Modell wird simuliert.
- Die Scope-Daten werden abgerufen und als CSV und optional als BMP exportiert.
- Die Daten werden in einem strukturierten Format für die weitere Analyse gespeichert.

4. **Benutzerdefinierte Operationen:**

- Zusätzliche Berechnungen auf den gemessenen Scope-Daten durchführen.
- Ergebnisse werden in der Konsole ausgegeben.

5 Anpassungen

- **Pfadangaben ändern:** `modelPath`, `scopePath` und `outPath` an die eigenen Verzeichnisse anpassen.
- **Parameterbereiche anpassen:** `paramNames` und `paramRanges` entsprechend den zu untersuchenden Parametern anpassen.
- **Analysezeitraum ändern:** `timeInterval` an das gewünschte Zeitintervall anpassen.

6 Beispielausgabe

Das Skript gibt die Ergebnisse des Parameter-Sweeps in der Konsole aus. Es zeigt die durchschnittlichen Signalwerte für jede Parameterkombination an und hebt das beste und schlechteste Ergebnis hervor.

Parameter sweep results:

fPWM	fN	Average signal
40000	500	0.123
40000	600	0.456
...		

Best average signal:

fPWM	fN	Average signal
41000	600	0.789

Worst average signal:

fPWM	fN	Average signal
40000	500	0.123

Die Ergebnisse werden auch in den angegebenen Dateien (`outPath`) gespeichert.

7 Beispiel-Verzeichnisstruktur

```
../out/automated_testing/  
I  
I pv_lut_fPWM_40000_fN_500.csv  
I pv_lut_fPWM_40000_fN_500.bmp  
I pv_lut_fPWM_40000_fN_600.csv  
I pv_lut_fPWM_40000_fN_600.bmp  
L ...
```

8 Hinweise

- Sicherstellen, dass das im `outPath` angegebene Verzeichnis existiert oder das Skript versucht, es zu erstellen.
- Das Skript löscht die Octave-Konsole vor Beginn der Simulation.
- Der Scope-Pfad wird aus dem Modellpfad und dem angegebenen Scope-Namen abgeleitet.

9 Fehlerbehebung

- **Dateipfade:** Sicherstellen, dass die angegebenen Pfade für das Modell, das Scope und das Ausgabeverzeichnis korrekt sind.

- **Parameterbereiche:** Sicherstellen, dass die Parameterbereiche korrekt angegeben sind und innerhalb des gültigen Bereichs für das Modell liegen.

Falls Probleme auftreten, die Ausgaben des Skripts auf Fehler überprüfen und die Konfigurationen entsprechend anpassen.

10 Anpassung für andere Modelle

Durch Anpassen der Parameter und Pfade kann das Skript auch in andere PLECS-Modelle eingefügt werden. Einfach die Werte von `modelPath`, `scopePath`, `paramNames` und `paramRanges` ändern, um das Skript an das spezifische Modell anzupassen.

11 Zusammenfassung

Dieses Skript automatisiert die Durchführung von Parameter-Sweeps in PLECS und die Analyse der Ergebnisse. Durch Anpassung der benutzerdefinierten Einstellungen kann das Skript für verschiedene Szenarien und Modelle verwendet werden.