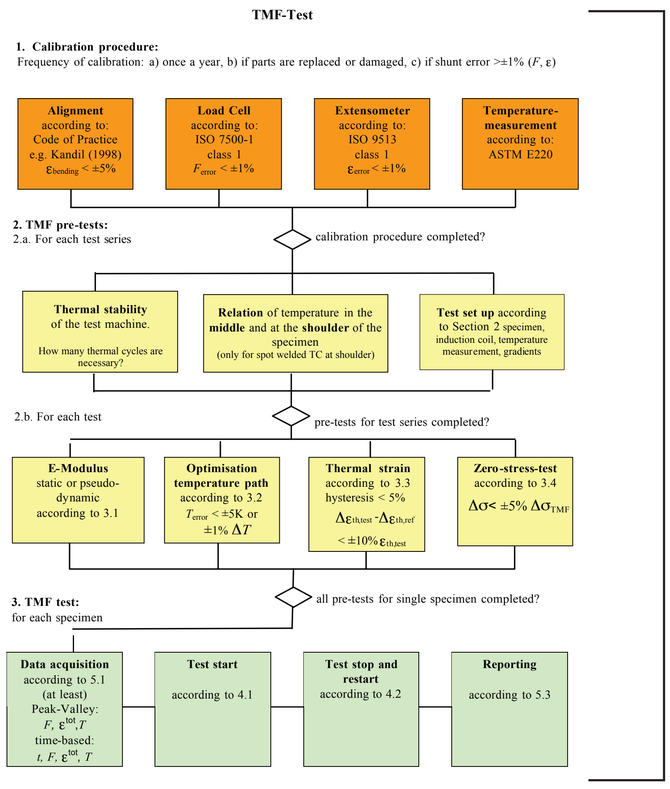
Software Evolution to Support Advanced Mechanical Testing methods: A Thermo-Mechanical Fatigue (TMF) Testing Example

FAJFROWSKI Michel, LESER Christoph

In an attempt to create a standard for TMF testing, a code of practice has been created by a consortium of international partners. The intent of the TMF code of practice is to ensure the consistent characterization of materials subjected to thermal and mechanical loads simultaneously. One of the difficulties inherent in this type of test is that the total strain is the sum of both a thermal strain and a mechanical strain. This complexity can be overcome with new generation controllers, software platforms and advanced techniques that allow the generation of temperature profiles in parallel with load and strain control.

This poster demonstrates how advanced MTS algorithms and techniques can be applied and modified to meet TMF code of practice requirements.

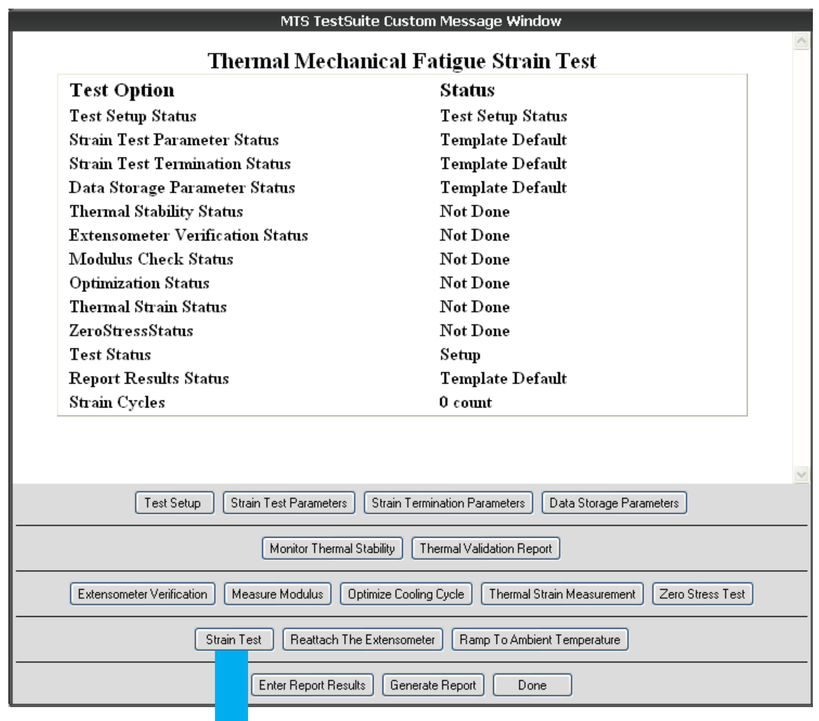
The TMF Code of Practice



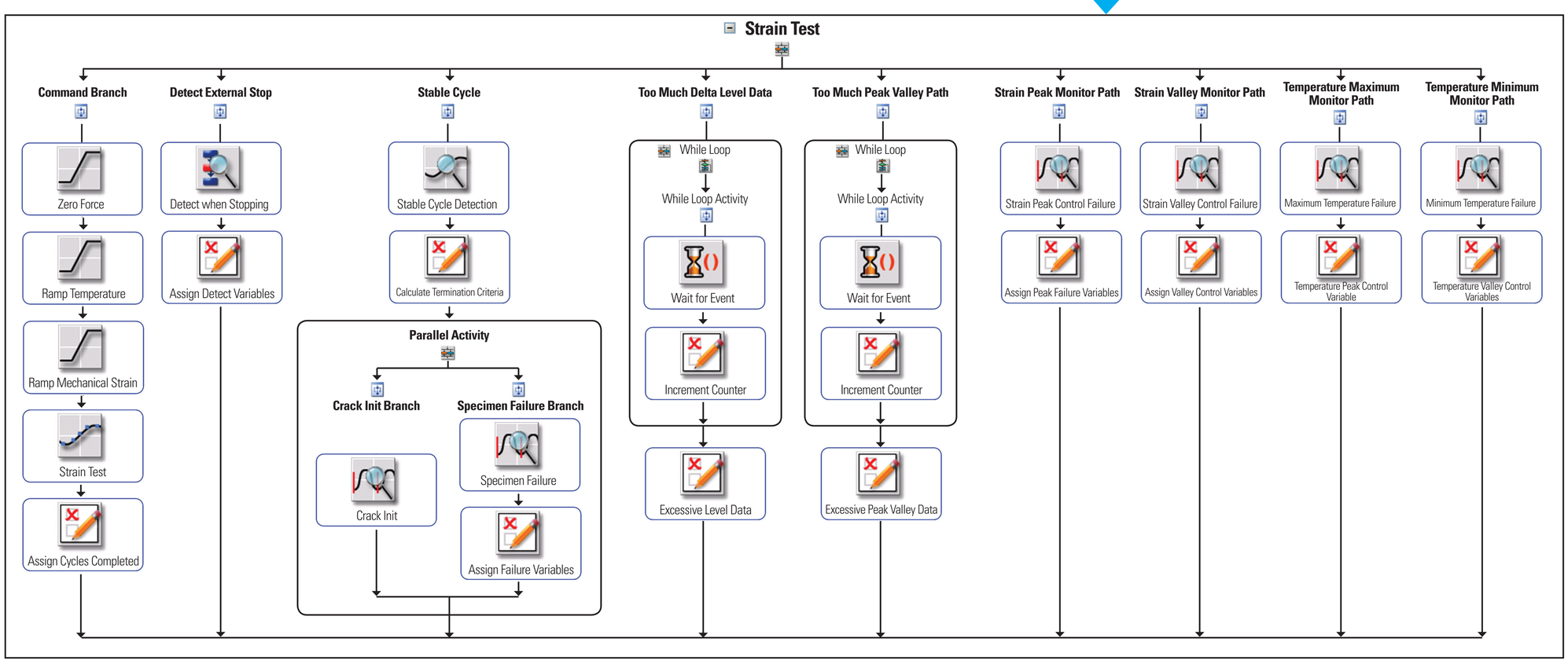
Each task from the code of practice is generated in the template by a button from the command panel

Each button is defined by an open program block, modifiable by the user (strain test button in this example)

The MTS Command Panel of the TMF Template



The strain block is composed of different activities to command algorithms based on force, strain, temperature and to simultaneously control any failure or change of these parameters.



Algorithm example: thermal strain

The code of practice specifies that the thermal strain, obtained from the load free thermal strain measurement, must be compensated. This compensation can be temperature or time based. To achieve this, MTS has developed a specific technique to calculate the thermal strain curve. This technique uses a “polynomial array”that allows one to specify the power of the polynomial. The strain temperature curve (denominated “calculatedthermalstraincurve”) is then fitted with this polynomial, which uses the “coeflist” argument.

CalculatedThermalStrainCurve: PolynomialArray(TemperatureArrayThermalStrain, CoefList)

CoefList: PolynomialFit(ThermalStrainTotal, ThermalTemperatureTotal, 2)

Open source programming language

MTS has implemented an open source programming language, Python, which makes it possible to adjust and create any kind of calculation through function generation. In the following example, the polynomial array calculation is accessible and can be modified:

```
#Function definition
def PolynomialArray(DataArray, coef):

    length = len(DataArray)
    retValue = [0.0]*length

    for i in range (length):
        retValue[i] = Polynomial(DataArray[i], CoefList)
         return retValue
```