

MTS MACHINES FOR DYNAMIC AND STATIC TESTS, NEW FUNCTIONALITY IN SYSTEMS AND SOFTWARE FOR RESEARCH APPLICATIONS.

Jacek Palkie, Spectro-Lab

ul. Warszawska 100/102, Łomianki, Poland, E-mail: jacek.palkie@spectro-lab.pl

1. Trends in test laboratories

- Less people in the lab, more pressure on operator and test engineer
- More tests on the same system
- Several devices around one test
- Test machines get connected to IT infrastructure
- Tests become more complex, include complex calculations

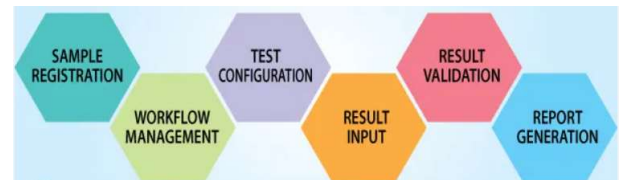
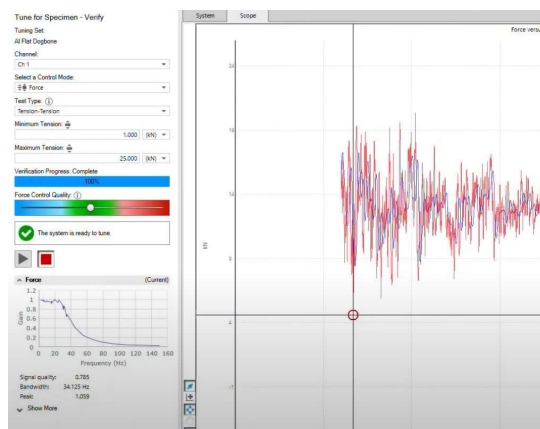
3. MTS Controller for Static Tests

Supports:

- Modbus RTU
- Eurotherm EI-Bysync
- Universal RS-232 ASCII interface

4. LIMS - Laboratory Information Management System

2. Fatigue tests – Autotuning PID



5. Tests with Complex Data Acquisition

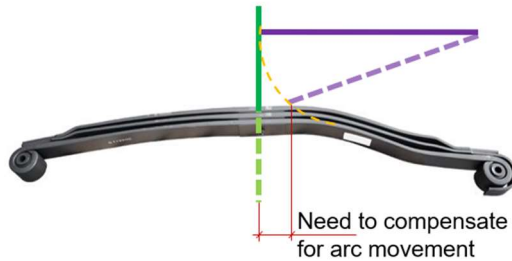
Numerous Transducers

vs.

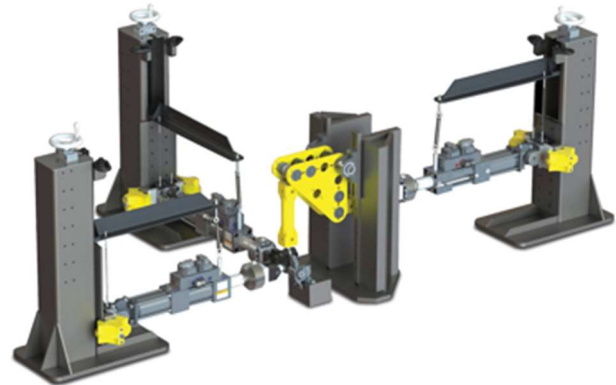
Test Rig Control

6. Matrix Control for Correct Loading

Vertical and Lateral channels kinematically cross-coupled.



8. Modular components to build custom test rigs



7. IronPython for complex calculations

```

Function Name: CoordinatesTransformationWinApp Usage: CoordinatesTransformationWinApp
Category: Test Definition Description:
Python Code:
1 from clr import AddReferenceToFilePath as addref
2 path = "C:\\Users\\nasilbul\\Documents\\VMathNet.Numerics\\lib\\net40\\VMathNet.Numerics.dll"
3 addref(path)
4 import VMathNet.Numerics.LinearAlgebra as la
5 from System import Array as sys_array
6
7 #Function definition
8 def CoordinatesTransformationWinApp():
9     #print AngleBetweenLines(Front_Eye_X, Front_Eye_Z, Rear_Eye_X, Rear_Eye_Z, 100, 0, 0, 0)
10    print "Calc coordinates"
11    def CoordTransform(x1, z1, x0, z0, alpha):
12        # Translate to coordinate system with origin in x0, z0
13        x_t = x1 - x0
14        z_t = z1 - z0
15        x_new = x_t*cos(alpha) + z_t*sin(alpha)
16        z_new = -x_t*sin(alpha) + z_t*cos(alpha)
17        #print " "
18        print "Ross_Point_X" + repr(Ross_Point_X)
19        print "x_t" + repr(x_t)
20        print "x0" + repr(x0)
21        print "x_t" + repr(x_t)
22        print "z_t" + repr(z_t)
23        print "x_new" + repr(x_new)
24        print "z_new" + repr(z_new)
25
26    return x_new
27    def CoordTransform2(x1, z1, x0, z0, alpha):
28        # Translate to coordinate system with origin in x0, z0
29        x_t = x1 - x0
30        z_t = z1 - z0
31        x_new = x_t*cos(alpha) + z_t*sin(alpha)
32        z_new = -x_t*sin(alpha) + z_t*cos(alpha)
33        #print " "
34        print "x_t"
35        print "z_t"
36        print "x_new"
37        print "z_new"
38        print " "
39        print cos(phi)
40        print sin(phi)
41        return z_new
42
43    # print "Alpha" + repr(AngleBetweenLines(Front_Eye_X, Front_Eye_Z, Rear_Eye_X, Rear_Eye_Z, 0, 0, 100, 0))
44
45    Ross_Point_X_GC.SetValue(CoordTransform(Ross_Point_X, Ross_Point_Z, Front_Eye_X, Front_Eye_Z, AngleBetweenLines(Front_Eye_X, Front_Eye_Z, Rear_Eye_X, Rear_Eye_Z, 100, 0, 0, 0)))
46    Ross_Point_Z_GC.SetValue(CoordTransform2(Ross_Point_X, Ross_Point_Z, Front_Eye_X, Front_Eye_Z, AngleBetweenLines(Front_Eye_X, Front_Eye_Z, Rear_Eye_X, Rear_Eye_Z, 100, 0, 0, 0)))
47
48    x_D = x_F1 + h_a*sin(phi1)
49    z_D = z_F1 + h_a*cos(phi1)
50
51    x_C = x_D - (h_a - h_b)*cos(phi1)
52    z_C = z_D - (h_a - h_b)*sin(phi1)
53    print "x_F1" + repr(x_F1)
54    print "z_F1" + repr(z_F1)
55    print "x_C" + repr(x_C)
56    print "z_C" + repr(z_C)
57
58    #h_E = x_C + h_b*sin(phi1)
59    #z_E = z_C + h_b*cos(phi1)
60
61    vb.Bracketed C = x_C - (h_b*cos(phi1))
62    vb.Bracketed C = x_C - (h_b*cos(phi1))
63
64    Validation successful.
65
66    Variable View
67
68    OK Cancel
    
```

Youtube lectures
"Test Rig Design"

