Usersubs for Model Validation

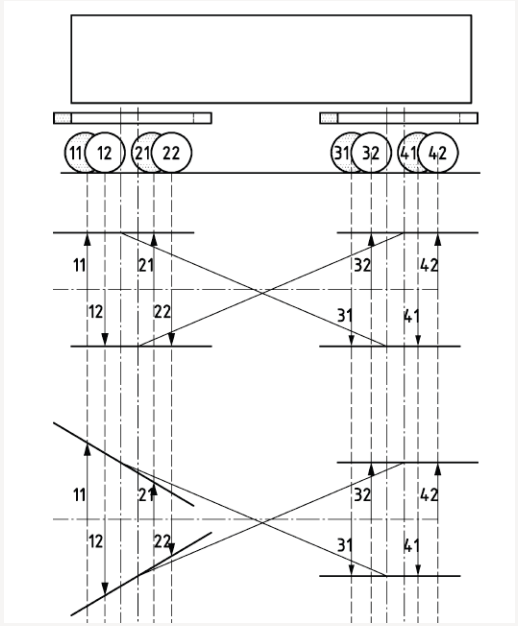
16th July 2020

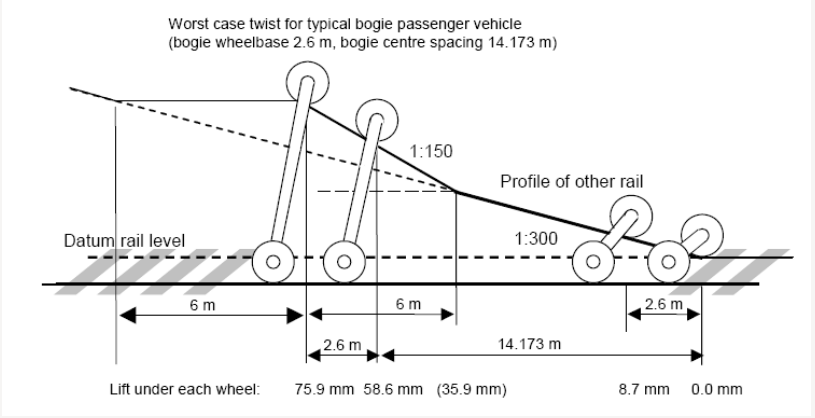
Philip Rogers, Technical Director, Dynamics, Gauging & Testing

Quasi-static model matching

- Model validation is an important stage of producing Vampire vehicle models
- Often includes matching to quasi-static tests:
  - *Wheel unloading (ΔQ/Q)*
  - *Bogie rotation (X-factor)*
  - *Sway tests*

- Most simply done using a *STATIC analysis





SNC·LAVALIN

ATKINS
Member of the SNC-Lavalin Group

*STATIC analysis

- Static analysis works by finding equilibrium at every step
- Works well for more linear models
  - *Typically passenger vehicles*
- Does not always work well with:
  - *Friction*
  - *Abrupt changes in stiffness*
- Typically works less well with freight or on-track machine models
- Does not include velocity dependent effects, such as yaw dampers

*STATIC	
STEPS	1000
AL0	0.7525
DQ	0.00001
STAGE 1	
T01T	0.000
T01Z	0.000
T02T	0.000
T02Z	0.000
T03T	28.908
T03Z	21.750
T04T	37.218
T04Z	28.000

SNC·LAVALIN ATKINS Member of the SNC-Lavalin Group

*TRANSIENT analysis with User Subroutine (Usersub)

- Usersubs allow users to write additional analysis code, which runs during a Transient analysis
- Code written in FORTRAN or C
- Allows users to:
  - *Model the behaviour of active and other novel suspensions*
  - *Implement simplified creep laws to model specific wheel/rail behaviour*
  - *Model specific vehicle conditions including worn or failed components*
  - *Calculate more complex outputs during the simulation*
  - *Accurately simulate acceptance tests*

Usersubs for quasi-static tests

- Makes use of “VTEUModifyTrackPos()” or “VTEUModifyTrackPosName()”
- Allows the user to change track position at any wheelset during the run
- Can simulate ΔQ/Q, X-factor or sway inputs
- Different wheelsets can see different inputs
- Use *USER section in run file to specify information, for example:
  - *Which wheelsets to lift*
  - *What values of maximum lift to apply*
- First method is to vary inputs depending on elapsed time in the run
- Example next slide…

Usersubs for quasi-static test – method 1 example (extract)

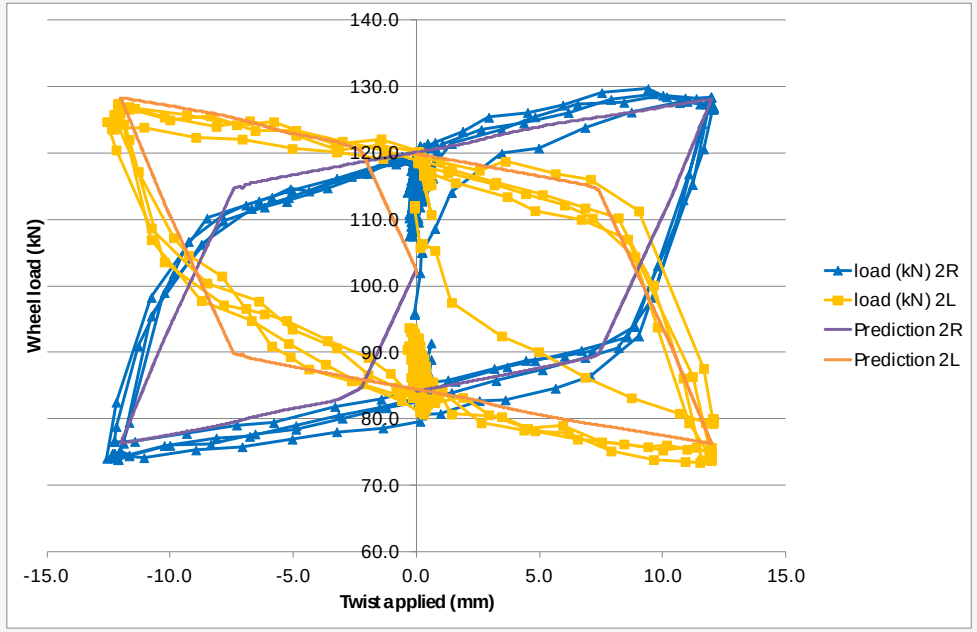
```
    IF ((DIST .GT. DIST1) .AND. (DIST .LE. DIST2)) THEN
C Calculate the XL input based slowly increasing input between s = DIST1 and s = DIST2...
        FACTOR = (DIST-DIST1)/(DIST2-DIST1)

C TTP is XL over gauge - in radians...
        TTP1 = MAXTTP1*FACTOR

C Vertical is XL over 2...
        ZTP1 = MAXZTP1*FACTOR

C Apply the movements to Wheelsets...
        CALL VTEUModifyTrackPos(IWHL1,0.0,ZTP1, TTP1,0.0,0.0,IERR)
    ENDIF
```

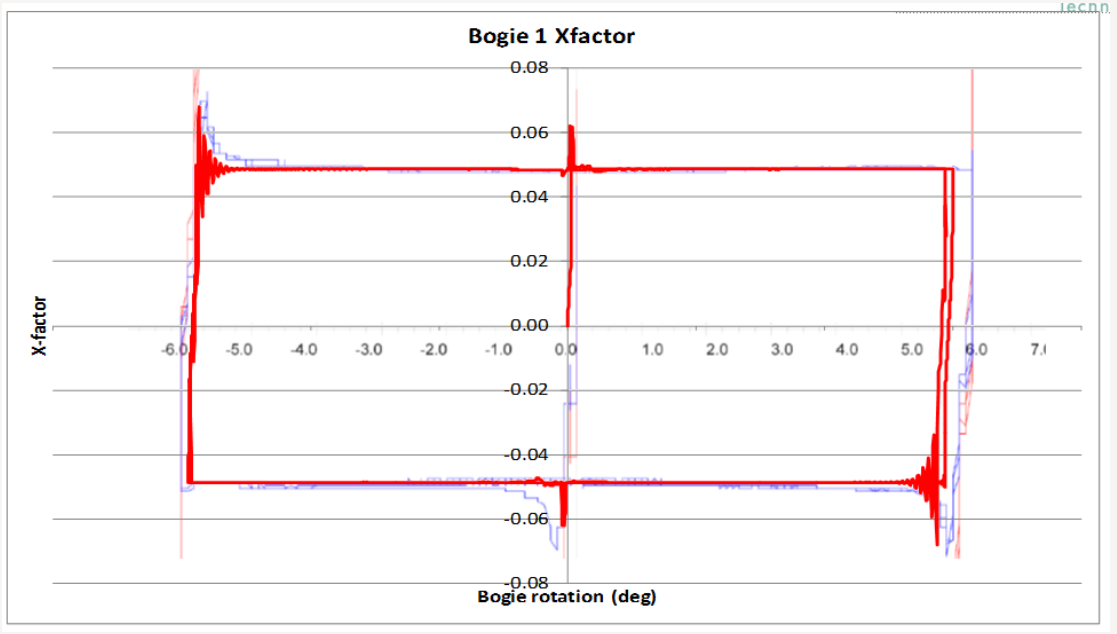
SNC·LAVALIN

ATKINS
Member of the SNC-Lavalin Group

Usersubs for quasi-static test – method 1 limitations (1)

- Step based Usersub method works well when test inputs are applied as per spec:

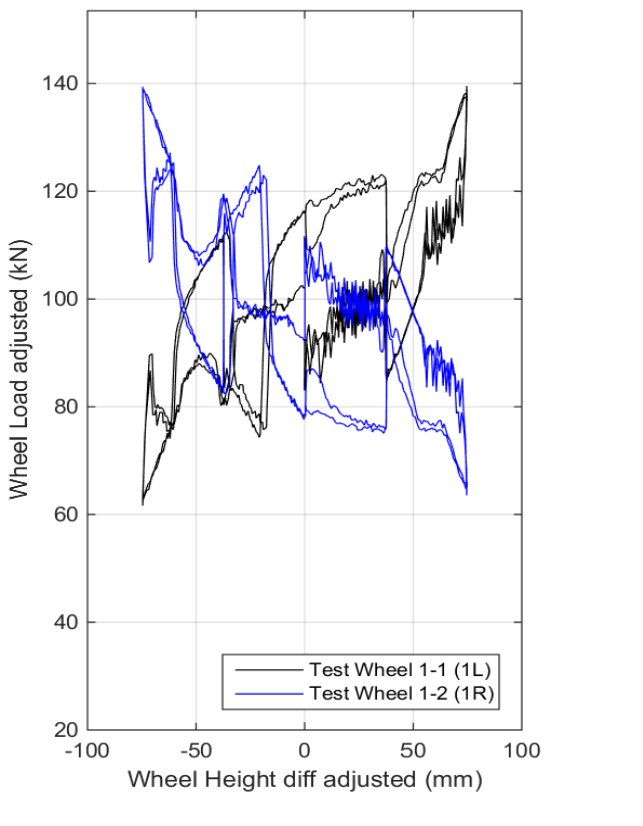


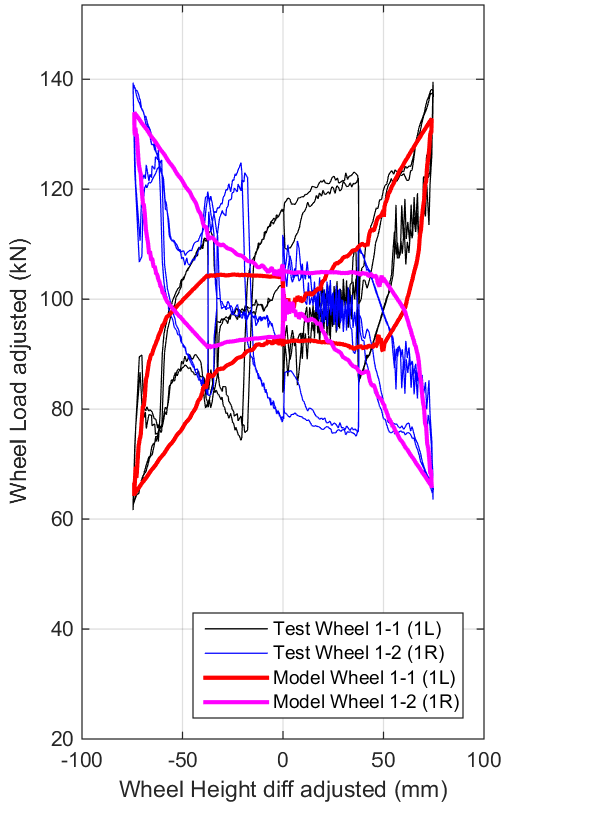


- Not very flexible, as Usersub code must be changed for a different input

Usersubs for quasi-static test – method 1 limitations (2)

- Doesn’t work well on ‘messy’ test results:

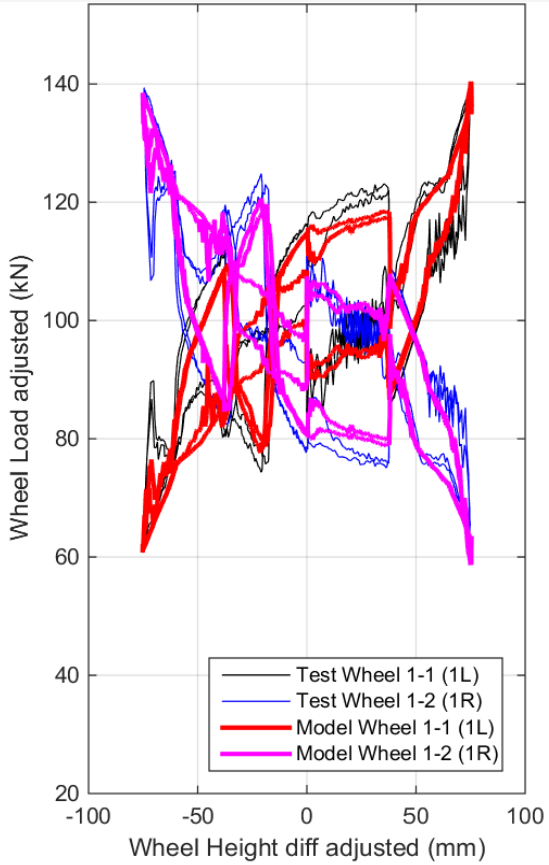




SNC·LAVALIN ATKINS Member of the SNC-Lavalin Group

Usersubs for quasi-static test – method 2

- More flexible Usersub approach
- Uses force file as an input file to replicate real test
  - *With displacements rather than forces*
  - *Avoids writing code to read an additional input file*
- Uses VTEUReadForceFile() and VTEUGetForce()
- VTEUModifyTrackPos() used as for method 1 to apply the inputs
- Allows real test displacements to be conveniently reproduced
  - *Including 'messy' test results*
- Including ΔQ/Q and X-factor tests



SNC·LAVALIN

ATKINS
Member of the SNC-Lavalin Group

Usersubs for quasi-static test – method 2 example (1)

- Displacement inputs using forcing file (fixed format .dat version)
- Actually a time based file, but we are reading it as though distance based

FORCE

DIST	LIFTWS1L	LIFTWS1R	LIFTWS2L	LIFTWS2R	LIFTWS3L	LIFTWS3R	(Distance in m, LIFT in mm
0	0.0	0.0	0.0	0.0	0.0	0.0	
100	6.74	-6.74	0.0	0.0	-6.74	6.74	
200	0.0	0.0	0.0	0.0	0.0	0.0	

Usersubs for quasi-static test – method 2 example (2)

- In USERSTART (at the start of the analysis):

```
CALL VTEUReadForceFile(LIFTSBG1, LEN(LIFTSBG1), I, I, IERR)
```

- In USERSUB (called every timestep):

```
LIFT1L = VTEUGetForce(1,DIST,IERR)
LIFT1R = VTEUGetForce(2,DIST,IERR)

IF (IERR .NE. 0) GOTO 9999

LIFT1L = LIFT1L*0.000001
LIFT1R = LIFT1L*0.000001
TTP1 = (LIFT1L-LIFT1R)/CONTACT
ZTP1 = -(LIFT1R+LIFT1L)/2
```

```
C Apply the movements to Wheelsets...

CALL VTEUModifyTrackPos( NWHL1, 0.0, ZTP1, TTP1, 0.0, 0.0, IERR)
```

Future steps?

- Possible new Vampire development:
  - *Provide Usersub “ModifyTrackPos” functionality without coding*
  - *Could use new input file type*
  - *Could use transient analysis without wheel/rail contact*

Q & A

SNC·LAVALIN

ATKINS
Member of the SNC-Lavalin Group

Thank you

SNC·LAVALIN

ATKINS
Member of the SNC-Lavalin Group

Our values are the essence of our company's identity.
They represent how we act, speak and behave together,
and how we engage with our clients and stakeholders.

SAFETY	*We put safety at the heart of everything we do, to safeguard people, assets and the environment.*
INTEGRITY	*We do the right thing, no matter what, and are accountable for our actions.*
COLLABORATION	*We work together and embrace each other's unique contribution to deliver amazing results for all.*
INNOVATION	*We redefine engineering by thinking boldly, proudly and differently.*