



VAMPIRE Pro

Presented by Peter Klauser

DeltaRail

VAMPIRE Pro Package

- **Railway dynamic simulation package**
 - **Includes a comprehensive range of railway processors**
 - **A range of solvers**
 - Eigenvalue
 - Non-Linear Transient
 - Quasi-Static Curving
 - Non-Linear Static
- **Detailed documentation**
 - **Manual**
 - **Quick Start Guide**
 - **Advanced Tutorials**

VAMPIRE Pro Customers

- **Worldwide customers**
 - Europe, Americas, Asia, Australia
- **A diverse customer base**
 - Train manufacturers, locomotives, passenger, freight, light rail
 - Infrastructure owners, heavy and light rail
 - Consultants
 - Research

VAMPIRE Pro Highlights

➤ Modeling capabilities

- Shear spring element with interaction between vertical and shear loads
- Air spring element with modeling of reservoir and surge pipe plus shear and moment response
- Contact geometry including yaw angle effects

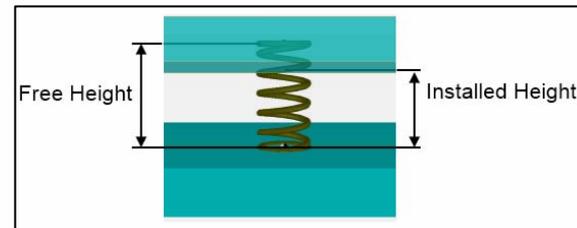
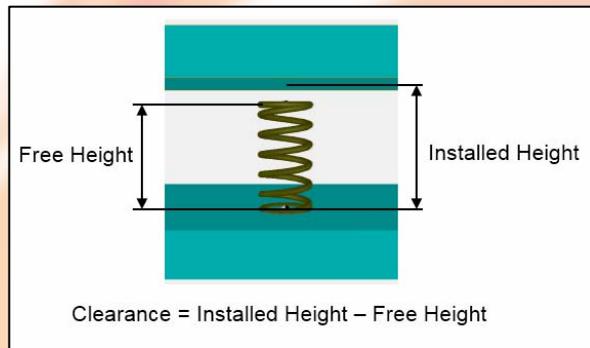
➤ Productivity

- Fully parameterized models
- Powerful process automation capabilities
- User subroutine feature allowing integration of user source code

Recent Developments

► Suspension Element Free Length

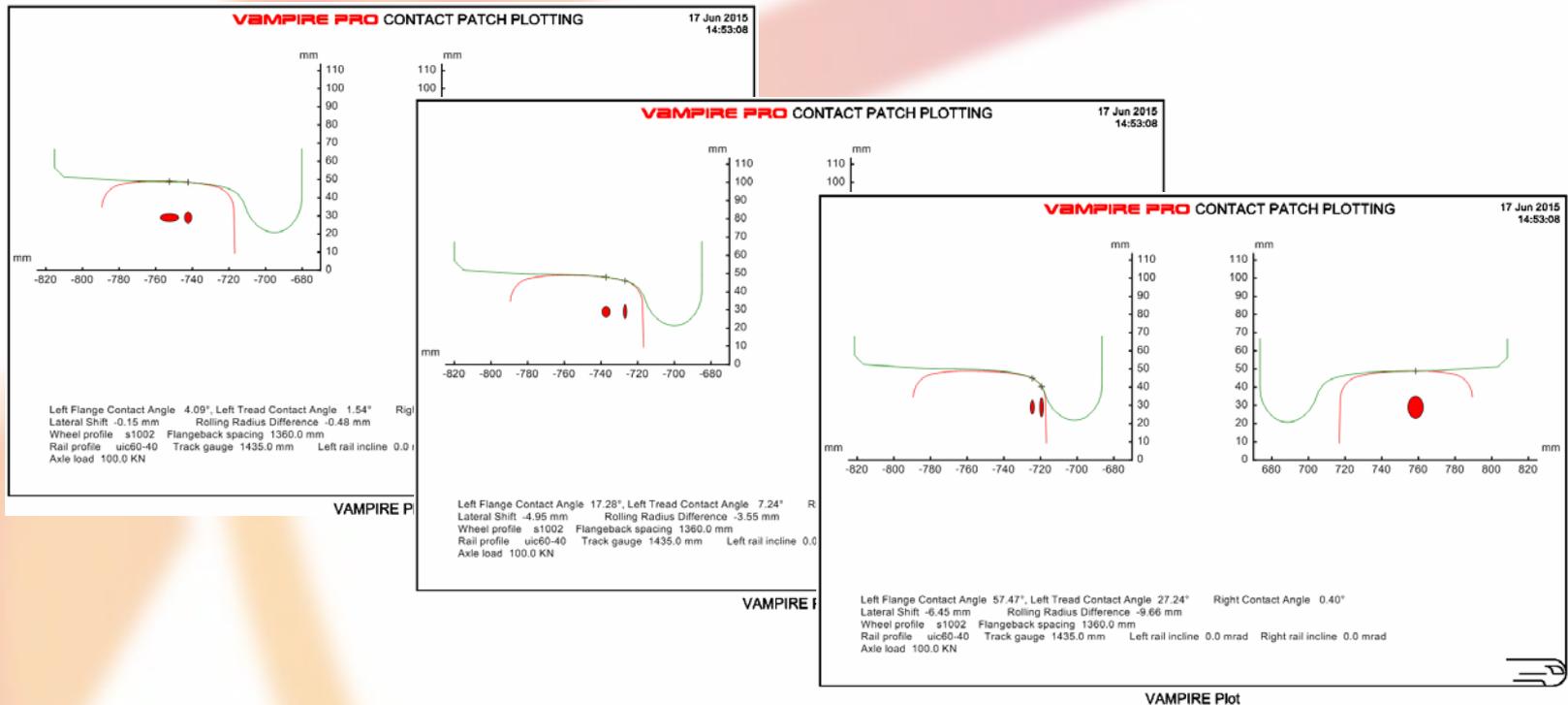
- Allow shear elements to unload during simulation
- Allow elements to start in clearance



Recent Developments

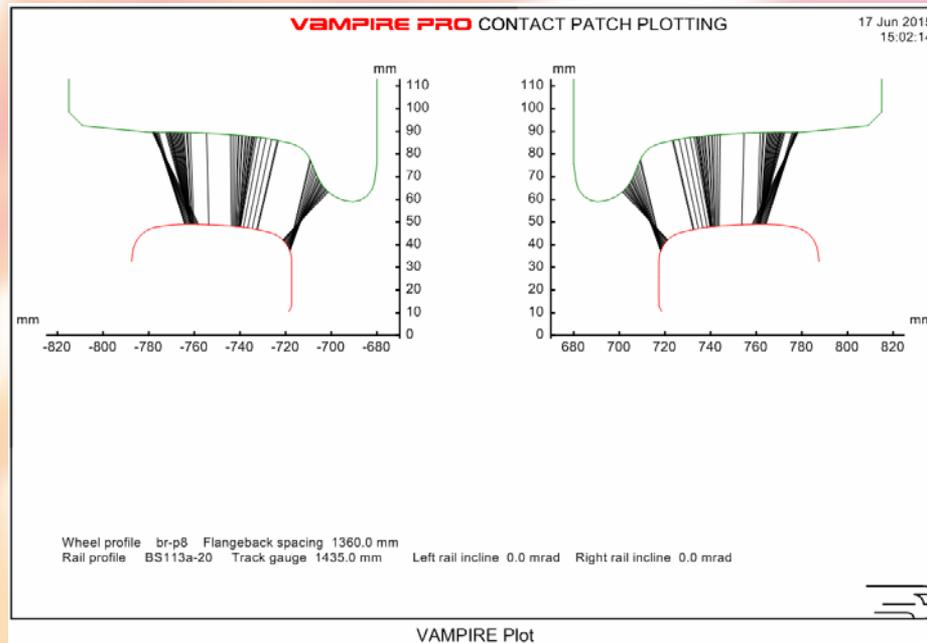
► Multiple Two Point Contact Zones

- Two point contact regions on tread and flange



Recent Developments

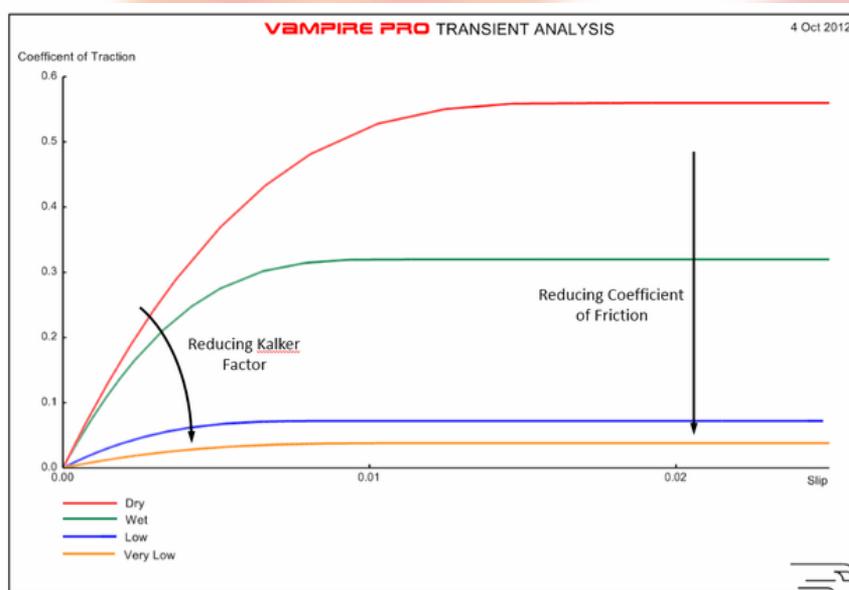
► Wheel Rail Contact Pre Processor Enhancements



- Contact point identification

Recent Developments

- Set Creep Properties per Wheel
 - Coefficient of friction/Kalker factors



- Properties per contact state, single point, two point
- Left and right wheel tread, flange, and flange back

Recent Developments

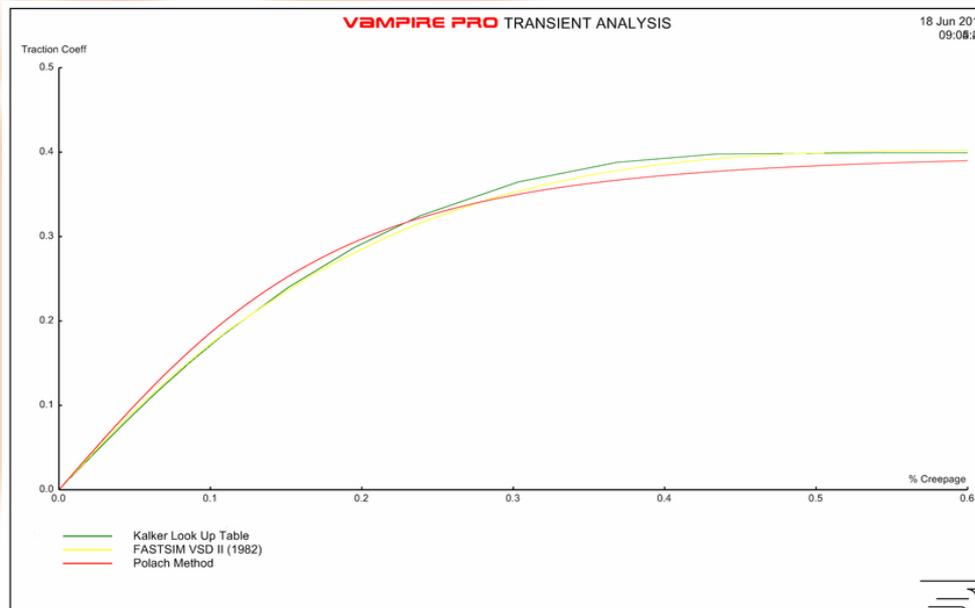
► User Subroutine Enhancements

▪ User creep laws

- Fastsim\Polach\User defined methods

▪ Wheel rail contact functions

- float VTEUWheelsetFloatValue()
- int VTEUSetCreepProperties()



User Subroutine Simplified Creep Laws

```
C Declare constants
G = 8.4E10
PI = 4.0E0*ATAN(1.0E0)

C Initialise outputs
F1 = 0.0
F2 = 0.0
M3 = 0.0

C Get inputs from Vampire
IERR = VTEUGetContactProperties(IW,CW,CP,a,b,Q,mu,V1,V2,V3)

C Calculate coefficients C11, C22, C23
C
AoS = a/b
CALL KALKER_COEFFS(AoS,C11,C22,C23)

C Calculate local F11
C
F11 = G*C11*a*b

C Polach Simplified Creep Law
C
MI = 0.0
V2C = V2

IF(ABS(V2+V3*a) .GT. ABS(V2C)) THEN
V2C=V2+V3*a
ENDIF

VV = (V1*V1 + V2C*V2C)**0.5

IF(VV .NE. 0.0) THEN
S = (V1*V1 + V2*V2)**0.5
IF(S .NE. 0.0) THEN
CJ = ((C11*V1/S)**2.0 + (C22*V2/S)**2.0)**0.5
EP = pi*G*a*b*CJ*VV/(4.0*Q*mu)
MI = (EP/(1.0+EP*EP)+ATAN(EP))*2.0*mu/pi
ENDIF
ENDIF
```

Future Release

➤ User Kalker Tables

- Bespoke Kalker tables for specific adhesion conditions
- Maintains speed of simulation

➤ Static Override

- Override suspension static loads and wheel loads
- Simulate uneven suspensions
 - Twisted frames
 - Incorrectly packed suspensions
 - Broken springs

➤ Enhanced Pre and Post Processors

- Improved vehicle checking
- Enhanced curving diagrams

➤ Current Trends

- **Increasing simulation of non-design fault conditions**
 - Broken springs
 - Incorrectly shimmed suspensions
 - Derailment analysis
- **More detailed modelling requirements**
 - Wheel rail contact
 - Specific adhesion conditions
- **Significantly increased number of simulations**
 - Required for vehicle acceptance
 - General investigations
- **Increased demand on calculation automation**
 - Related to increase in simulations
 - Capturing a repeatable calculation process

Maintaining for the Future

➤ Maintain for the Future

- Ease of use
- Speed of simulation
- Calculation process automation
- Modeling flexibility
- Expert support