Implementation of the wire compensator in MAD-X, xsuite and pymask

P. Bélanger, G. Sterbini, A. Poyet, G. Iadarola, D. Kaltchev
Motivation

- Unify the tools used in the study of wire compensators (BBCW)
  - More easily validated
  - Ease the learning curve
  - Usable by different groups (e.g. Machine Protection)

- Make use of the most recent tools developed (xsuite, lhcmask, pymask):
  - Nicely maintained and constantly improved
  - Modular
Long-Range Beam-Beam Effect

- Close to the interaction points, the beams share a common beam pipe.
- Electromagnetic interaction between the two beams limits the minimum crossing angle achievable.
- The kick on a particle from one beam due to the EM forces from the other beam is comparable to a current-carrying wire:

\[
\frac{dp_r}{dt} = e \frac{\mu_0 I}{2\pi r} \frac{c}{\beta_r} \left[ 1 \mp \beta_r^2 \right] \left( 1 - e^{-\frac{r^2}{2\sigma^2}} \right)
\]

A. Poyet: https://cds.cern.ch/record/2773329
Wire Compensation

- Locally compensate the integrated effect of LRBB interactions
- Wires around IP1 (vertically) and IP5 (horizontally)
Wire Compensation: Benefits

- Facilitate operational conditions
- For HL-LHC: allows to reduce crossing angle, leading to an increase in luminosity

A. Poyet: https://cds.cern.ch/record/2773329
MAD-X Implementation: beambeam

- **BEAMBEAM element:**
  - Model a current-carrying wire by using an additional beam
  - Equivalent charge leading to the desired current is computed

```plaintext
LHC_C = 26659;
gamma_r = Energy/PMASS + 1;
beta_r = SQRT(1-1/gamma_r^2);
N_p := I_w3*LHC_C/(Q ELECT*CLIGHT)/((1+beta_r^2)/beta_r);

! Effective bb charge (counter-rotating beam -> twice bb interactions over a length L_w)
  effCharge := 2*N_p/LHC_C*L_w;

! charge is set negative to attract for positive current and repulse for negative current
wire_bb : beambeam, charge := -effCharge,
Xma := r_w*COS(theta_w),
Yma := r_w*SIN(theta_w),
sigx = 1e-6,
sigy = 1e-6,
width = 1,
BBDIR = -1;
```

(Clickable image)
MAD-X Implementation: multipole

- **MULTIPOLE element:**
  - Any field can be expressed a sum of multipoles:
  
  \[ B_y + iB_x = \sum_{n=0}^{\infty} (b_n + ia_n) \frac{(x + iy)^n}{n!} \]

  - The case of a wire can be represented by a sum over all multipoles
  - Using the pre-defined multipole element of MAD-X:

  \[ K_{N_n} + iK_{S_n} = -\frac{\mu_0(IL)}{2\pi \cdot B \rho} \frac{n!}{(r_w e^{i\phi_w})^{n+1}} \]

- **Note:** kicker is added for dipolar effect
MAD-X Implementation: wire element

- WIRE element:
  - Implemented by T. Persson
  - Based on SixTrack model, A. Patapenka
    https://indico.cern.ch/event/456856/contributions/1968799/attachments/1196420/1738901/Wire_model.pdf
  - Allows to directly specify the current and position of the wire
  - The physical length can differ from the integrated length (fringe fields)

```plaintext
! Common to all wires
L_w = 1.3;
r_w = 8e-3;
theta_w = 0;

! phys_extend -> infty to approach an infinitely long wire
! integrated over the physical length, L_w
int_window = 1;
phys_extend = 10000;
wire_tobias : wire, current := I_w1,
L     = 0,
L_phy := phys_extend*L_w,
L_int := int_window*L_w,
Xma    := r_w*COS(theta_w),
Yma    := r_w*SIN(theta_w);
```

(Clickable image)
Validation of the implementation: TWISS

- Different tests were performed to validate the behaviour of the wire element (https://gitlab.cern.ch/pbelange/abp001.git)
- The WIRE element is now part of the newest MAD-X release! (5.08.00)
- On the right is a comparison of the 3 types of wire (beambeam, multipole, wire) acting in a simple FODO cell
Validation of the implementation: TRACK

- Only single turn kick was compared…
- More comparisons to come when stability studies are conducted!
Implementation in xsuite

- Same physics from SixTrack and used in the MAD-X implementation
- xsuite can now import MAD-X sequences with WIRE elements!
- See V 0.9.4 https://github.com/xsuite/xtrack
- Again, more comparisons to come with stability studies

```python
import xtrack as xt
import xpart as xp

# Particles from ducktrack
particles = xp.Particles(_context=ctx,**dkt_particle.to_dict())

# Wire class
wire = xt.Wire(_context = ctx,
                wire_L_phy = 1.3,
                wire_L_int = 1.3,
                wire_current=250,
                wire_xma = -8e-3,
                wire_yma = -10e-3)

# Tracking
wire.track(particles)
```
Practical case with pymask

- For operational studies of both Run 3 and HL-LHC
- The general approach is:
  1. Replace (seqedit) markers from layout database with wires
  2. Place wires far away from the beam
  3. After luminosity leveling: align the wires
  4. Create Q-feedforward knob links with a pair of quads to cancel tune shift
  5. User can then play around with the knobs and study wire compensation
Q-feedforward

- The quadrupolar contribution of a wire induces a significant tune shift.
- To simultaneously cancel the tune shift in both planes, 2 quadrupoles can be trimmed.

**Note:** In the “2 wire” configuration (2 wires placed symmetrically around the beam), the odd multipoles are cancelled and the even ones are doubled. Recall:

\[
K_{N_n} + iK_{S_n} = -\frac{\mu_0(IL)}{2\pi \cdot B\rho} \frac{n!}{(r_w e^{i\phi_w})^{n+1}}
\]

- Tune shift in x (similar for y):
- With 2 quads, tune shift is cancelled if:

\[
\begin{bmatrix}
K_1 \\
K_2
\end{bmatrix} = 
\begin{bmatrix}
\beta_{1,x} & \beta_{2,x} \\
-\beta_{1,y} & -\beta_{2,y}
\end{bmatrix}^{-1}
\begin{bmatrix}
-\beta_{w,x} \\
\beta_{w,y}
\end{bmatrix}
\left(\frac{\mu_0 \cos(2\phi_w)}{2\pi \cdot B\rho}\right) \left(\frac{I_w}{r_w^2}\right) = 0
\]
Q-feedforward

- Yet another validation of the implementation
- Analytic calculation is only an approximation: beta functions are perturbed
- Was already tested in 2020 with beambeam element as a wire (and quads trimming was done numerically -> better tune shift cancellation)

A. Poyet: 
https://indico.cern.ch/event/927406/contributions/3899418/attachments/2060462/3456063/MPP_RUN3_FF_190620_v2.pdf
Conclusion

- General implementation of wire compensator in MAD-X in three different ways:
  - beambeam, multipole, wire
- Validation of all three implementations with themselves and against analytic calculations (TWISS, TRACK)
- Implementation of wire compensator in xsuite
- Installed wires in pymask for operational studies
- Tested Q-feedforward successfully, with analytic computation of knobs parameters
- Simulation tools are now ready for future BBWC studies!
Outlook

- Need to start stability studies
- In particular: measuring diffusion from the beam core:
  - Compare good and bad DA conditions
  - Define proper observable to quantify diffusion
- Study realistic scenarios for Run 3 and HL-LHC
  - Study tolerance for misalignment, wire imperfection, etc.
  - N.B.: multipolar expansion might be interesting for this
- Compare benefits of wire vs simply using a subset of its multipole components
- Reconstruct position of the wire from effects on the beam (self-alignment)
- Contribute to MDs and/or commissioning in the months to come
- Eventually need to submit thesis proposal for my PhD
Thank you!