

Crab Cavity and RF Multipoles Implementation in MADX and SixTrack

HL-LHC Task 2.3 General Meeting

Javier Barranco García, Riccardo De Maria, Rogelio Tomas

Introduction

- Crab cavities (CC) and RF multipoles (RF mult) are implemented in MADX and SixTrack. However with different conventions for historical reasons.
- So far simulations with CCs (and RF mult) were mainly done with SixTrack (DA studies) and by **experts**.
- However, if a new user (for example collimation group, etc.) wants to include CCs in their simulations, there are few issues regarding **sign convention** and **units** to be taken into account.
- In this talk we'll review the current implementation of CC and RF mult in both codes.

CC Beam Dynamics (Reminder)

The Hamiltonian of a CC reads,

$$H_{crab} = \frac{qV}{P_0} \sin \left(\phi_s + \frac{\omega z}{c} \right) x$$

The transverse kick applied is **maximum for $z=|z_{\max}|$** and zero at the center of the bunch. The convention used here implies $z>0$ for the head of the bunch and for $V>0$ then $\Delta p_x < 0$ which is needed to ensure head on collision for a positive crossing angle.

$$\Delta p_x = -\frac{\partial H_{crab}}{\partial x} = -\frac{qV}{P_0} \sin \left(\phi_s + \frac{\omega z}{c} \right)$$

In order to **preserve symplecticity** a kick in energy is needed.

$$\Delta p_z = -\frac{\partial H_{crab}}{\partial z} = -\frac{qV}{P_0} \cos \left(\phi_s + \frac{\omega z}{c} \right) \frac{\omega}{c} x$$

Similar for a vertical CC, only changing x by y .

CC in MADX

- CC element implemented in MADX for **tracking purposes only** in 2010. Twiss and other functionalities not yet committed to the repository. However if needed one can use the RF dipole with the proper phase as crab cavity, which is supported by Twiss and MADX tracking (although not tested).
- An example from the MADX manual,

```
BEAM, PARTICLE=PROTON, ENERGY=7000.0;  
CAVITY: CRABCAVITY, L=10.0[m], VOLT=5.0[MV], LAG=0.0[2pi], FREQ=400[MHz],  
rv1=0, rv2=50, rv3=1000, rv4=50, rph1=100, rph2=500,lagf=0.125[2pi], TILT=0[rad];
```

- TILT convention: a positive angle is a clockwise rotation. A $+\pi/2$ rotation means going from a negative horizontal kick to a negative vertical kick. In MADX it3 is taken into account but in SixTrack it is used to change type between hor/ver CCs only if it $+\pi/2$.
- No trace of the BV flag, i.e. it should be used only for Beam 1.
- Additional Tracking options: Voltage ramp (up to 4 turns), phase change (2 steps).
- CC kick implementation in MADX,

$$px = \text{track}(2,\text{itrack}) + vrf * \mathbf{sin}(\text{phirf} - \text{omega} * \text{track}(5,\text{itrack}))$$

$$pt = \text{track}(6,\text{itrack}) - \text{omega} * vrf * \text{track}(1,\text{itrack}) * \mathbf{cos}(\text{phirf} - \text{omega} * \text{track}(5,\text{itrack}))$$

CC Conversion MADX-SixTrack

```
static void
```

```
att_crabcavity(struct c6t_element* el)
```

```
{
```

```
double lag = el->value[5]; [2pi]
```

```
double tilt = el->value[12]; [rad]
```

```
if (fabs(tilt - M_PI/2) < eps_9)
```

```
    el->out_1 = -23; Vert CC
```

```
else
```

```
    el->out_1 = 23; Hor CC
```

```
if (cavall_flag == 0)
```

```
{
```

```
    el->out_2 = total_voltage;
```

```
    strcpy(el->name, "CAV");
```

```
}
```

```
else el->out_2 = el->value[1]; [MV]
```

```
el->out_3 = el->value[4]; // freq = // not used [MHz]
```

```
if (lag < -0.5) lag += 1.;
```

```
else if (lag > 0.5) lag -= 1.;
```

```
el->out_4 = 360. * lag; [deg]!!! SixTrack input is [rad]!!
```

```
}
```

$\left| \text{tilt} - \frac{\pi}{2} \right| < 10^{-9}$ Just a flag to determine hor/vert!
Not explained in the MADX manual.

RF multipoles in MADX

- Implemented by A. Latina.
- MADX-SixTrack conversion: change LAG units but not the sign and the shift. SixTrack accepts up to octupolar component (elements ± 26 , ± 27 and ± 28). Tilt seems to be taken into account in MADX (according to Andrea).

```
Label:  RFMULTIPOLE, VOLT=real[MV], LAG=real[2pi],
        HARMON=integer,  FREQ=real[MHz],
        LRAD=real,  TILT=real[rad],
        KNL:={k0n1, k1n1, k2n1, ... }, ! Normal coefficients
        KSL:={k0s1, k1s1, k2s1, ... }, ! Skew coefficients
        PNL:={p0n,  p1n,  p2n,  ... }, ! Normal phases [2pi]
        PSL:={p0s,  p1s,  p2s,  ... }; ! Skew phases [2pi]
```

$$H = \left\{ -\frac{1}{k_{\text{RF}}} \frac{qV_{\text{RF}}}{p_s c} \cos(\vartheta_{\text{RF}} - k_{\text{RF}}z) + \dots \right. \\ \left. \dots + \sum_{n=0}^N \frac{1}{(n+1)!} \Re \left[(K_{N,n}L \cos(\vartheta_n - k_{\text{RF}}z) + iK_{S,n}L \cos(\varphi_n - k_{\text{RF}}z)) (x + iy)^{n+1} \right] \right\} \delta \left(s - \frac{L}{2} \right).$$

RF multipoles in MADX

Expanding the terms in $\mathcal{S}_{\mathcal{K}}$ and $\mathcal{C}_{\mathcal{K}}$, the kick reads:

$$\Delta p_x = -\frac{\partial H}{\partial x} = -\sum_{n=0}^N \frac{1}{n!} \Re [(K_{N,n} L \cos(\vartheta_n - k_{\text{RF}} z) + i K_{S,n} L \cos(\varphi_n - k_{\text{RF}} z)) (x + iy)^n],$$

$$\Delta p_y = -\frac{\partial H}{\partial y} = \sum_{n=0}^N \frac{1}{n!} \Im [(K_{N,n} L \cos(\vartheta_n - k_{\text{RF}} z) + i K_{S,n} L \cos(\varphi_n - k_{\text{RF}} z)) (x + iy)^n],$$

$$\Delta p_t = -\frac{\partial H}{\partial z} = \frac{qV_{\text{RF}}}{p_s c} \sin(\vartheta_{\text{RF}} - k_{\text{RF}} z) - k_{\text{RF}} \sum_{n=0}^N \frac{1}{(n+1)!} \Re [(K_{N,n} L \sin(\vartheta_n - k_{\text{RF}} z) + i K_{S,n} L \sin(\varphi_n - k_{\text{RF}} z)) (x + iy)^{n+1}].$$

Courtesy A. Latina

Installing CCs in MADX

- The optics experts define the parameters and install the CCs in a particular lattice (only HLLHC so far) in two steps.
- **First:** for a given optics the voltage is calculated by installing the CCs first as TKICKER and the kick is matched to force $x=0$ for a $1\sigma_z$ particle at the IP (the MADX macros for this are in the slhc_sequence.madx file /toolkit/rematch_crabs.madx)
- Each of the optics in /afs/cern.ch/eng/lhc/optics/SLHCV3.1b have these kicks already hard-coded (AHCRA_B_L1B1, AHCRA_B_R1B1, AHCRA_B_L5B1, AHCRA_B_R5B1,...)
- **Second:** the TKICKER are replaced by CCs with voltage calculated with the CC parameters and the kick previously matched..
- These scripts are not yet publicly released. Shall we use the RF dipole instead? (See Riccardo's talk).
- From a practical point of view, it will be just enough for a given HLLHC lattice to call the CC installation and later “switch on” the cavities by setting for example, on_crab1 :=1.
- Conversion into SixTrack format is transparent as long as the $\varphi_{CC} = 0$.

Installing RF mult in MADX

- There exist MADX macros to install the RF mult at the location of the CCs. Although not publicly release yet.
- However there is a different sign convention for the phase in MADX and SixTrack, so the RF mult should be “enabled” after all the matchings. For the DA studies with different CC prototypes the MADX scripts were used just as mean to install the RF mult to be translated into SixTrack, i.e. no physics were done in MADX.

CC in SixTrack

- CC format in fort.2 SixTrack lattice definition:

Crab Cavity

Format: name type Voltage Frequency Phase

Name: May contain up to sixteen characters

Type: Type identifier is +23 and -23 for horizontal and vertical crab cavities respectively.

Voltage: Crab Cavity voltage [MV].

Frequency: Crab Cavity frequency [MHz].

Phase: Phase of the excitation in radians

- Example of horizontal CCs in HLLHC fort.2 lattice,

		Voltage[MV]	Frequency[MHz]
acrabcca.l5b1	23	3.819290017e+00	4.007896026e+02
0.000000000e+00	0.000000000e+00	0.000000000e+00	0.000000000e+00
Phase[rad]			
acrabccc.r5b1	23	4.155008313e+00	4.007896026e+02
0.000000000e+00	0.000000000e+00	0.000000000e+00	0.000000000e+00

It should be zero for crabbing, so transparent for the conversion bug.

CC in SixTrack

- CC transverse and energy kick as implemented in SixTrack.

`yv(xory, j) = yv(xory, j) -`

`crabamp * sin(((sigmv(j)/clight) * crabfreq) * 2d0) * pi + crabph(ix))`

`dpsv(j) = dpsv(j) -`

`(((((crabamp * crabfreq) * 2d0) * pi) / clight) * xv(xory, j)) * COS(((sigmv(j) / clight) * crabfreq) * 2d0) * pi + crabph(ix))) * c1m3`

RF multipoles in SixTrack

- Up to octupolar component (normal and skew).
- For crabbing \rightarrow phase= $\pi/2$. Different wrt CC in SixTrack!

Normal Quadrupole

$$\Delta x' = -b_2 x \cos\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,quad}}\right)$$

$$\Delta y' = b_2 y \cos\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,quad}}\right)$$

$$\Delta \delta = \frac{b_2}{2} (x^2 - y^2) \sin\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,quad}}\right) \frac{\omega}{c}$$

Normal Sextupole

$$\Delta x' = -b_3 (x^2 - y^2) \cos\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,sext}}\right)$$

$$\Delta y' = 2b_3 xy \cos\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,sext}}\right)$$

$$\Delta \delta = \frac{b_3}{3} (x^3 - 3xy^2) \sin\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,sext}}\right) \frac{\omega}{c}$$

Normal Octupole

$$\Delta x' = -b_4 (x^3 - 3xy^2) \cos\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,oct}}\right)$$

$$\Delta y' = b_4 (3x^2y - y^3) \cos\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,oct}}\right)$$

$$\Delta \delta = \frac{b_4}{4} (x^4 - 6x^2y^2 + y^4) \sin\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,oct}}\right) \frac{\omega}{c}$$

Skew Quadrupole

$$\Delta x' = -b_2 y \cos\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,quad}}\right)$$

$$\Delta y' = -b_2 x \cos\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,quad}}\right)$$

$$\Delta \delta = b_2 xy \sin\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,quad}}\right) \frac{\omega}{c}$$

Skew Sextupole

$$\Delta x' = -2b_3 xy \cos\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,sext}}\right)$$

$$\Delta y' = b_3 (y^2 - x^2) \cos\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,sext}}\right)$$

$$\Delta \delta = -\frac{b_3}{3} (y^3 - 3yx^2) \sin\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,sext}}\right) \frac{\omega}{c}$$

Skew Octupole

$$\Delta x' = -b_4 (y^3 + 3x^2y) \cos\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,oct}}\right)$$

$$\Delta y' = -b_4 (3y^2x - x^3) \cos\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,oct}}\right)$$

$$\Delta \delta = b_4 (x^3y - y^3x) \sin\left(\frac{\omega z}{c} + \phi_s + \phi_{\text{RF,oct}}\right) \frac{\omega}{c}$$

Implementation (in)consistency

- As said before, the different elements in the two codes have different implementations for historical reasons.

$$\text{CRAB MADX} \quad \rightarrow \Delta p_x \sim -V \sin(\Phi - \omega z)$$

$$\text{CRAB Sixt} \quad \rightarrow \Delta p_x \sim -V \sin(\omega z + \Phi)$$

$$\text{RF Mult Sixt} \quad \rightarrow \Delta p_x \sim -V \cos(\omega z + \Phi)$$

$$\text{RF Mult MADX} \quad \rightarrow \Delta p_x \sim -V \cos(\Phi - \omega z)$$

- For SixTrack simulations this has not been an issue so far.
- We will standardize the codes to avoid possible sign/units errors.