BEAM DYNAMICS ON-LINE SIMULATION

A. Novokhatski, T. Weiland, M. Krassilnikov, W. Koch, TEMF, TU-Darmstadt, Germany
P. Castro, DESY, Hamburg, Germany

Abstract

The V-code development has been motivated by the necessity of a powerful tool for the beam dynamics on-line simulation. Based on the Ensemble model [1], the V-code provides the possibility of fast and efficient beam dynamics simulation. Such a tool is as important for the accelerator commissioning as for the operating machine. The main principles of the code are the complete accelerator simulation and the immediate data on-line exchange with a given accelerator control system. The structure of accelerators and the features of the Ensemble model determine the choice of the object model for the code elaboration. The V-code has been realized within the object-oriented ideology on the C++ platform, and presents an accelerator as a sequence of beam line elements. Such approach gives the essential advantage in further code extension, as the code can be improved continuously by introducing new element classes. Containing definite accelerator components (e.g. a RF-photoinjector, a quadrupole, a cavity, etc.), these classes can implement new features for beam dynamics, like space charge, wake fields, and others. The present beam line database has been developed for the TESLA Test Facility (TTF DESY) accelerator component description. The results of the on-line beam dynamics simulation for TTF are presented.

1 INTRODUCTION

While the approach to beam dynamics based on "macro" particles considers only "macro" particle centre coordinates in phase space, the Ensemble model [1] takes into account also motion of particles inside the ensemble. To imply this, besides the centre position an ensemble involves also correlations in phase space, that gives the possibility of significant reduction of particles number for the simulation of beam dynamics in accelerators.

2 ENSEMBLE MODEL

The Ensemble model is a self-consistent one, derived from Vlasov equation for the distribution function f of particles density in phase space $(\vec{r}, \vec{p})$

$$\frac{df}{d\tau} = \frac{\partial f}{\partial \tau} + \frac{\partial f}{\partial \vec{r}}\frac{\vec{p}}{\gamma} + \frac{\partial f}{\partial \vec{p}}\frac{\vec{F}}{mc^2} = 0, \tag{1}$$

where $\tau = ct$, $\vec{p} = \dfrac{\vec{P}}{mc}$ is the normalized momentum and $\gamma = \dfrac{E}{mc^2}\sqrt{1+\vec{p}\vec{p}}$ is the normalized energy. The force $\vec{F}$ acts on the particle $\dfrac{d\vec{p}}{d\tau} = \dfrac{\vec{F}}{mc^2}$. For linear forces it is possible to built the compact beam model which takes into account moments of the distribution function $f(\tau, \vec{r}, \vec{p})$ of first and second order. To introduce the nonlinear force effects it is necessary to take into account moments of higher orders, which leads to increasing of ensemble variables number.

2.1 Moments and Matrices

Using the normalizing condition for the distribution function

$$\int f(\tau, \vec{r}, \vec{p})\, d\vec{r}\, d\vec{p} = 1, \tag{2}$$

the first order moments are

$$\langle \vec{R} \rangle = \int \vec{R}\, f(\tau, \vec{r}, \vec{p})\, d\vec{r}\, d\vec{p}, \tag{3}$$

where $\vec{R} = \{\vec{X}, \vec{P}\} = \{x, y, z, p_x, p_y, p_z\}$. The second order moments are given by

$$M_{\xi\nu} = \langle \Delta\xi\, \Delta\nu \rangle = \int \Delta\xi\, \Delta\nu\, f(\tau, \vec{r}, \vec{p})\, d\vec{r}\, d\vec{p}, \tag{4}$$

where $\Delta\xi = \xi - \langle\xi\rangle$ and $\xi, \nu = x, y, z, p_x, p_y, p_z$. The second order moments form a symmetric matrix $\hat{\mathbb{M}}$:

$$\hat{\mathbb{M}} = \begin{pmatrix} \hat{\mathbb{S}} & \hat{\mathbb{L}} \\ \hat{\mathbb{L}}^T & \hat{\mathbb{T}} \end{pmatrix}, \tag{5}$$

where $\hat{\mathbb{S}}_{ij} = M_{x_i x_j}, \quad \hat{\mathbb{L}}_{ij} = M_{x_i p_j}, \quad \hat{\mathbb{T}}_{ij} = M_{p_i p_j}$.

The determinant $\det \hat{\mathbb{M}}$, being a square of 6D ellipsoid volume in a phase space, is a square of the 6-dimensional normalized emittance of the ensemble. The equation of the 6D ellipsoid envelope is

$$U(\vec{R}) = 1, \text{where} U(\vec{R}) = (\vec{R} - \langle\vec{R}\rangle)^T \hat{\mathbb{M}}^{-1} (\vec{R} - \langle\vec{R}\rangle). \tag{6}$$

For the Gaussian form the distribution function is given by the formula

$$f(\vec{R}) = \frac{\sqrt{\det \hat{\mathbb{M}}^{-1}}}{(2\pi)^3} \exp\left[-\frac{U(\vec{R})}{2}\right]. \tag{7}$$

For example, in the case of uncoupled transverse and longitudinal motion the longitudinal projection of the distribution function has the form

$$f(z, p_z) = \frac{1}{2\pi\epsilon_z} \exp\left[-\frac{1}{2\epsilon_z^2}\left\{M_{p_z p_z}\Delta z^2 + \right.\right. \\ \left.\left. +2M_{z p_z}\Delta z \Delta p_z + M_{zz}\Delta p_z^2\right\}\right], \tag{8}$$

where $\epsilon_z = \sqrt{M_{zz}M_{p_z p_z} - M_{z p_z}^2}$ is normalized longitudinal emittance.

2.2 *Time Equations*

After average procedure the Vlasov equation (1) gives

$$\frac{\partial\langle\mu\rangle}{\partial\tau} = \langle\frac{\partial\mu}{\partial\vec{r}}\frac{\vec{p}}{\gamma}\rangle + \langle\frac{\partial\mu}{\partial\vec{p}}\frac{\vec{F}}{mc^2}\rangle, \qquad (9)$$

if one assumes applied forces $\vec{F}$ and any ensemble parameter μ satisfying the condition (which corresponds to the full emittance invariance)

$$\langle\mu\,\frac{\partial\vec{F}}{\partial\vec{p}}\rangle = 0. \qquad (10)$$

Linear applied forces can be represented

$$\frac{\vec{F}}{mc^2} = \vec{\mathbb{F}}(\langle\vec{R}\rangle) + \hat{\mathbb{F}}^X\cdot\Delta\vec{X} + \hat{\mathbb{F}}^P\cdot\Delta\vec{P}, \qquad (11)$$

where

$$\hat{\mathbb{F}}^X_{ij} = \frac{1}{mc^2}\left.\frac{\partial\vec{F}_i}{\partial X_j}\right|_{\langle\vec{R}\rangle}, \quad \hat{\mathbb{F}}^P_{ij} = \frac{1}{mc^2}\left.\frac{\partial\vec{F}_i}{\partial P_j}\right|_{\langle\vec{R}\rangle}. \qquad (12)$$

It should be noted that (11) includes external forces from beam line devices as well as "internal" effects, for example, the space charge forces.

When energy spread is small enough, $1/\gamma$ can be expanded

$$\frac{1}{\gamma} = \frac{1}{\gamma_m} - \frac{\sum\limits_n\left[\langle p_n\Delta p_n + \frac{1}{2}\left(\Delta p_n^2 - M_{p_np_n}\right)\right]}{\gamma_m^3}, \qquad (13)$$

with γ_m to be the average value of normalized beam energy

$$\gamma_m = \sqrt{1 + \sum_n\left[\langle p_n\rangle^2 + M_{p_np_n}\right]}. \qquad (14)$$

Applying the time equation (9) for all the moments of first and second order for linear forces (11) and using the expansion (13) one can obtain the time equations for the vector $\vec{R}$ and matrices $\hat{\mathbb{S}}$, $\hat{\mathbb{L}}$ and $\hat{\mathbb{T}}$:

$$\begin{aligned}
\frac{\partial\vec{P}}{\partial\tau} &= \vec{\mathbb{F}},\\
\frac{\partial\vec{X}}{\partial\tau} &= \hat{\mathbb{W}}\vec{P},\\
\frac{\partial\hat{\mathbb{S}}}{\partial\tau} &= \hat{\mathbb{V}}\hat{\mathbb{L}}^T + \hat{\mathbb{L}}\hat{\mathbb{V}}\\
\frac{\partial\hat{\mathbb{L}}}{\partial\tau} &= \hat{\mathbb{V}}\hat{\mathbb{T}} + \hat{\mathbb{S}}\left(\hat{\mathbb{F}}^X\right)^T + \hat{\mathbb{L}}\left(\hat{\mathbb{F}}^P\right)^T\\
\frac{\partial\hat{\mathbb{T}}}{\partial\tau} &= \hat{\mathbb{F}}^X\hat{\mathbb{L}} + \hat{\mathbb{F}}^P\hat{\mathbb{T}} + \left(\hat{\mathbb{F}}^X\hat{\mathbb{L}} + \hat{\mathbb{F}}^P\hat{\mathbb{T}}\right)^T,
\end{aligned} \qquad (15)$$

where

$$\begin{aligned}
\hat{\mathbb{W}}_{ij} &= \frac{1}{\gamma_m}\left\{\delta_{ij} - \frac{1}{\gamma_m^2}\hat{\mathbb{T}}_{ij}\right\},\\
\hat{\mathbb{V}}_{ij} &= \frac{1}{\gamma_m}\left\{\delta_{ij} - \frac{\langle p_i\rangle\langle p_j\rangle}{\gamma_m^2}\right\}.
\end{aligned} \qquad (16)$$

3 V-CODE IMPLEMENTATION

The V-code based on the Ensemble model (15) deals directly with actual data from accelerator control system. Implementation of applied forces (11) is determined by specified beamline devices. The application integrates beamline data hierarchy, data acquisition, user interface and calculation. The code could be used for on-line simulation or for off-line analysis. The V-code has been realized within the object-oriented ideology on the C++ platform.

3.1 *Beamline Data Hierarchy*

The *BeamLine* object is the main in the V-code data hierarchy. The other beam line entities are *SubLine* and *BeamLineElement (BLE)*. Figure 1 illustrates the relationships between these entities.

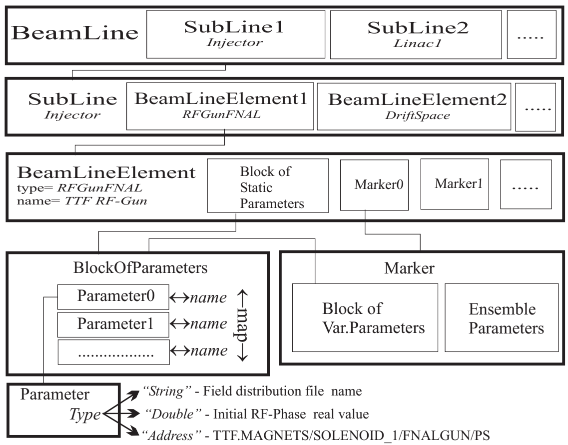

Figure 1: V-Code data hierarchy.

To make beam line element description more universal and flexible the *Parameter* class has been developed. The object of this class consists of either a string information (like a file name for the field distribution in the beam line device), or a real value (like a field amplitude or initial RF-phase), or an address (which is actual in a data acquisition from accelerator control system). It is realized through multiple constructor. The *BlockOfParameters* class consists of a map of *Parameter*s keyed by parameter name. Besides the *BlockOfStaticParameters* the *BLE* contains also a set of *Marker*s which are used for the simulation results recording. The *Marker* object contains all the ensemble parameters $(\vec{R}, \hat{\mathbb{M}})$ after the *BLE* and the *BlockOfVariableParameters* for the actual simulation parameters storage.

3.2 *On-Line Data Acquisition*

Compatibility of *BLE* parameters with given accelerator devices parameters provides with the possibility of on-line collection data from the accelerator control system and beam dynamics simulation with actual beam line parameters. The *On-Line Manager* (Fig. 2) distributes the collected data from control system among the *BLE*s. Mainly these parameters are values of currents from power supplies. The *On-Line Manager* also implements the exchange parameters averaging in time which is synchronized with diagnostics (BPM) average procedure.

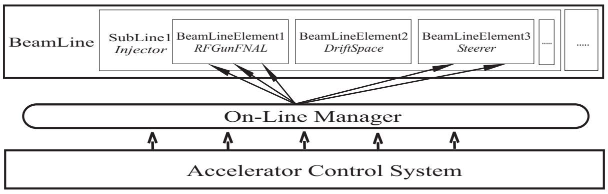

Figure 2: On-line data acquisition.

3.3 User Interface

The V-Code is supplied by graphical user interface, which main parts are the beam line design tool, the current simulation display and the results display facilities. The application is able to read an accelerator layout from script files and provides with an interactive editing of devices parameters. The on-line dialog serves for actual parameters values updating. Example of V-code user interface is presented in Fig. 3.

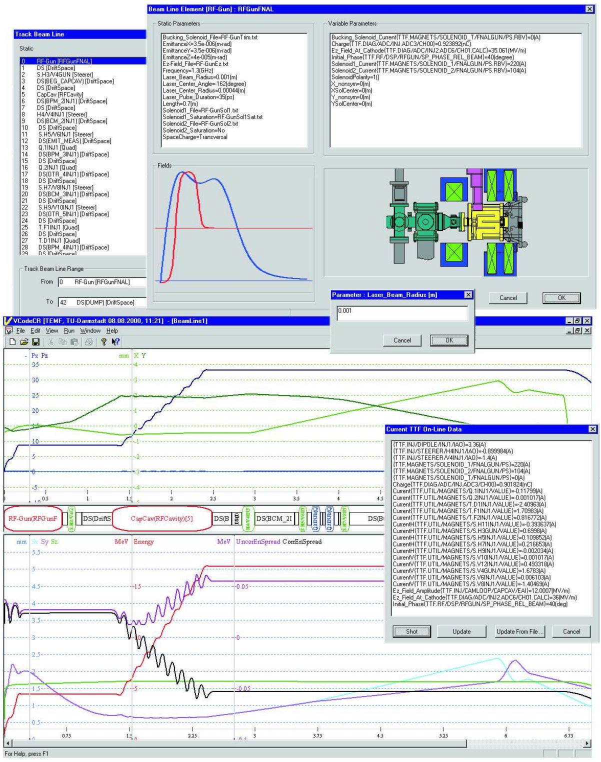

Figure 3: V-code user interface.

3.4 Solver

The solver uses the equations (15) for the time evolution of ensemble parameters. Specified beam line elements are responsible for the contributions to vector $\vec{\mathbb{F}}$ and matrices $\hat{\mathbb{F}}^X$, $\hat{\mathbb{F}}^P$. On-line simulation opportunity caused the *Pre-Run* procedure. The goal of the *Pre-Run* procedure is *BLE* configuration for the given beam line device. If the *Pre-Run* procedure is successful the *Run* procedure will start. This procedure is based on *Piece* class. This pure virtual class inherits real classes for the beam line device implementation. In accordance with *BeamLine* content *Run* creates correspondent *Piece*-derived object, which implements peculiarities of applied forces and boundary conditions. When the ensemble crosses the actual *BLE* boundaries (up- or downstream) the *Run* procedure creates a new *Piece*-derived class instead of the existing one. Due to this approach the code is easy to expand by introducing new *Piece*-derived classes. These classes could be embodied through multiple constructors and inheritance.

4 TTF ON-LINE SIMULATION

The TESLA Test Facility injector was chosen for the on-line beam dynamics simulation with the V-code. The beam line consists of the RF-Gun, the booster cavity, the set of steerers for the trajectory correction, the quadrupoles and the bending magnet with analysis area (Fig. 4).

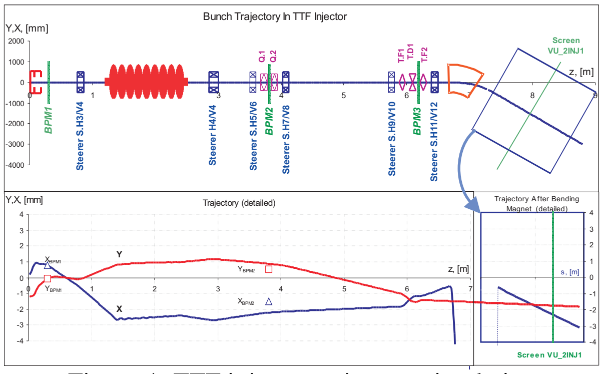

Figure 4: TTF injector trajectory simulation.

In order to reach consistency between measured data and calculation the comparison strategy "trajectory → sizes → emittances" was suggested. This approach corresponds to the ensemble moments order increasing. The beam centre position in phase space (the first order moments) determine the bunch trajectory. The trajectory diagnostics is presented by the beam position monitors. The example of trajectory simulation is presented in Fig. 4.

5 CONCLUSIONS

The V-code presented in this paper is a powerful tool for on- and off-line beam dynamics simulation of complete accelerators. Due to the flexible beam line elements implementation, the code has perspectives in already existing beam line elements improving, as well as in new elements design. First results on TTF on-line beam dynamics simulation are presented.

6 ACKNOWLEDGMENTS

The authors appreciate Reinhard Brinkmann, Jorg Rossbach for discussions.

7 REFERENCES

[1] A.Novokhatski and T.Weiland, The Model of Ensembles for the Beam Dynamics Simulation, Proc. of PAC99, p. 2743