Appendix E
TOP Documentation

DESY summer school 2009

by

Martin Nuss
TU Graz

under the supervision of
Dr. Pedro Castro
Machine Physics Group - MPY

summer, 2009
Hamburg, Germany
DESY, A Research Centre of the Helmholtz Association

Contents

1 TOP Documentation

1.1 Basic structure of TOP

TOP (Trajectory Optimization Program) is a fully object oriented simulation developed in Java `http://www.java.com` with the help of Netbeans `http://www.netbeans.org/`.

TOP basically consists of 4 steps no matter which kind of functionality you choose:

1. **Input**
   The program reads the input structure (see sect. 1.2). Using the given input it immediately calculates the settings (for the beamline, plotting, calculations,...) for all required processes and holds them in the "TOWorkingList". The input layout is stored in the "BeamLine" which consists of "Elements".

   After this step all parameters are known for each calculation that has to be done and all elements are placed properly in the beamline (Attention no settings are applied yet to the elements (i.e. all currents are zero, ...)).

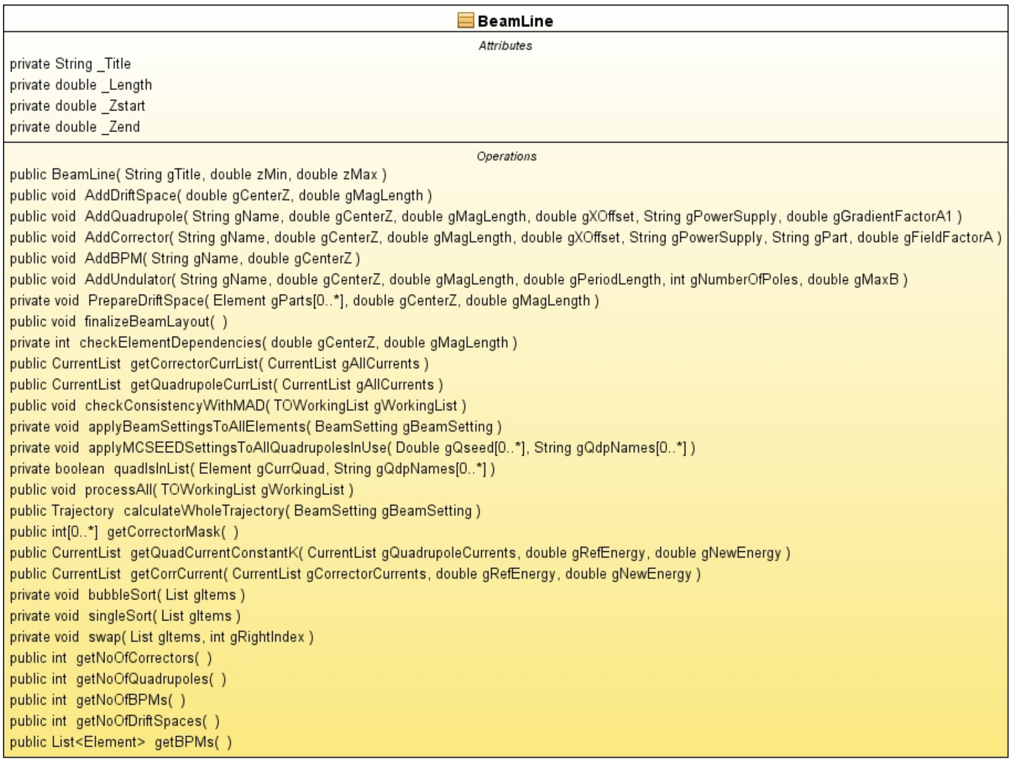
2. **Calculation of Trajectories**
   In a next step the "BeamLine" is used to read each setting stored in the "TOWorkingList" and calculate the appropriate trajectory (which is then also stored in "TOWorkingList").

3. **Additional Calculations**
   Now that all parameters, and trajectories are known the "TOWorkingList" performs all further calculations like calculating the dispersion, optimizing currents, ...

4. **Output**
   In this last step all results are saved automatically to the correct files (see 1.3).

The class diagrams of the four most important classes will be presented in the following:

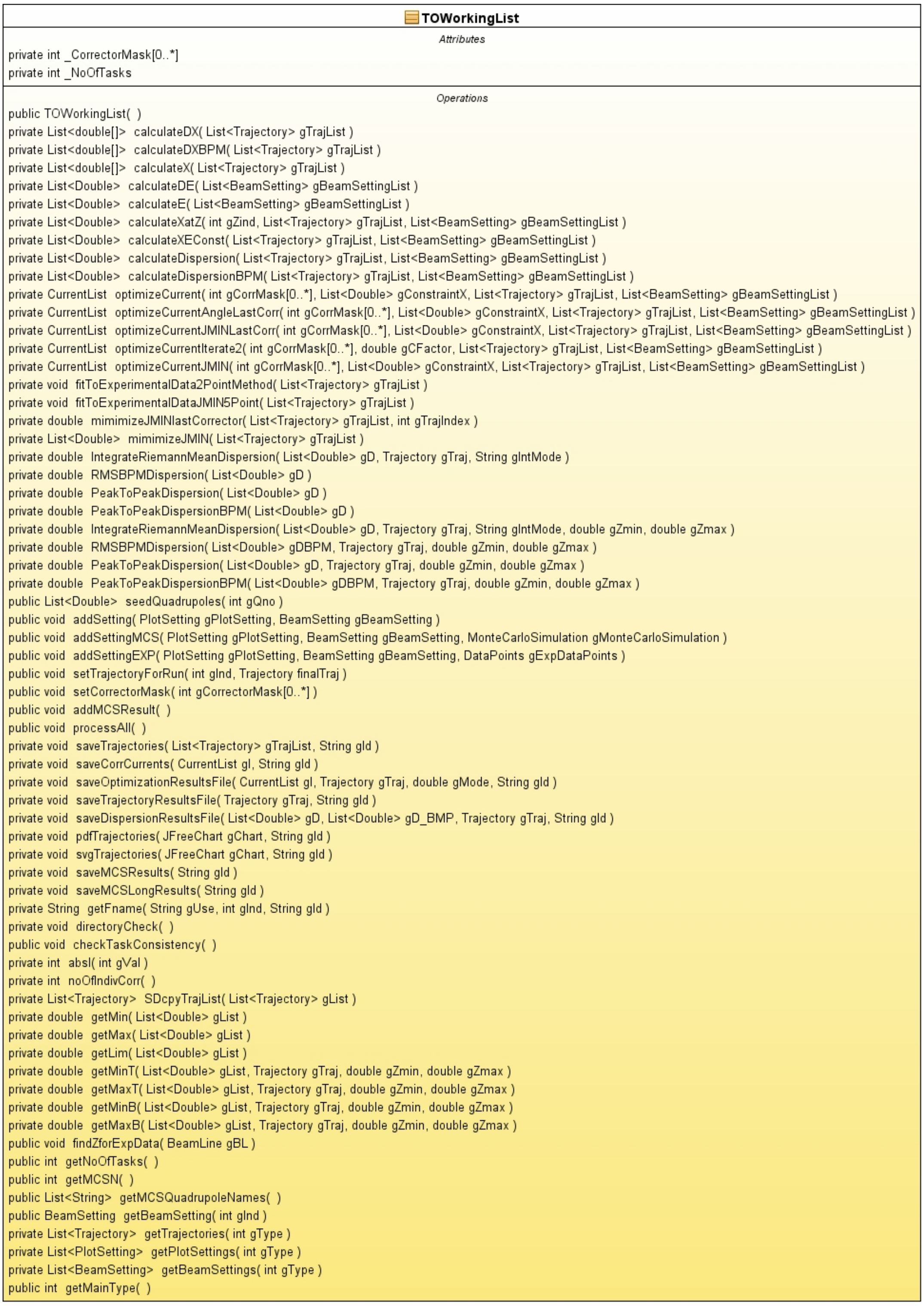
The class diagram for the class BeamLine is shown in figure 1.1. This class is responsible for holding all beamline elements and distributing tasks to them.



```
BeamLine
Attributes
private String _Title
private double _Length
private double _Zstart
private double _Zend
Operations
public BeamLine( String gTitle, double zMin, double zMax )
public void  AddDriftSpace( double gCenterZ, double gMagLength )
public void  AddQuadrupole( String gName, double gCenterZ, double gMagLength, double gXOffset, String gPowerSupply, double gGradientFactorA1 )
public void  AddCorrector( String gName, double gCenterZ, double gMagLength, double gXOffset, String gPowerSupply, String gPart, double gFieldFactorA )
public void  AddBPM( String gName, double gCenterZ )
public void  AddUndulator( String gName, double gCenterZ, double gMagLength, double gPeriodLength, int gNumberOfPoles, double gMaxB )
private void  PrepareDriftSpace( Element gParts[0..*], double gCenterZ, double gMagLength )
public void  finalizeBeamLayout(  )
private int  checkElementDependencies( double gCenterZ, double gMagLength )
public CurrentList  getCorrectorCurrList( CurrentList gAllCurrents )
public CurrentList  getQuadrupoleCurrList( CurrentList gAllCurrents )
public void  checkConsistencyWithMAD( TOWorkingList gWorkingList )
private void  applyBeamSettingsToAllElements( BeamSetting gBeamSetting )
private void  applyMCSEEDSettingsToAllQuadrupolesInUse( Double gQseed[0..*], String gQdpNames[0..*] )
private boolean  quadIsInList( Element gCurrQuad, String gQdpNames[0..*] )
public void  processAll( TOWorkingList gWorkingList )
public Trajectory  calculateWholeTrajectory( BeamSetting gBeamSetting )
public int[0..*]  getCorrectorMask(  )
public CurrentList  getQuadCurrentConstantK( CurrentList gQuadrupoleCurrents, double gRefEnergy, double gNewEnergy )
public CurrentList  getCorrCurrent( CurrentList gCorrectorCurrents, double gRefEnergy, double gNewEnergy )
private void  bubbleSort( List gItems )
private void  singleSort( List gItems )
private void  swap( List gItems, int gRightIndex )
public int  getNoOfCorrectors(  )
public int  getNoOfQuadrupoles(  )
public int  getNoOfBPMs(  )
public int  getNoOfDriftSpaces(  )
public List<Element>  getBPMs(  )
```

Figure 1.1: Class BeamLine

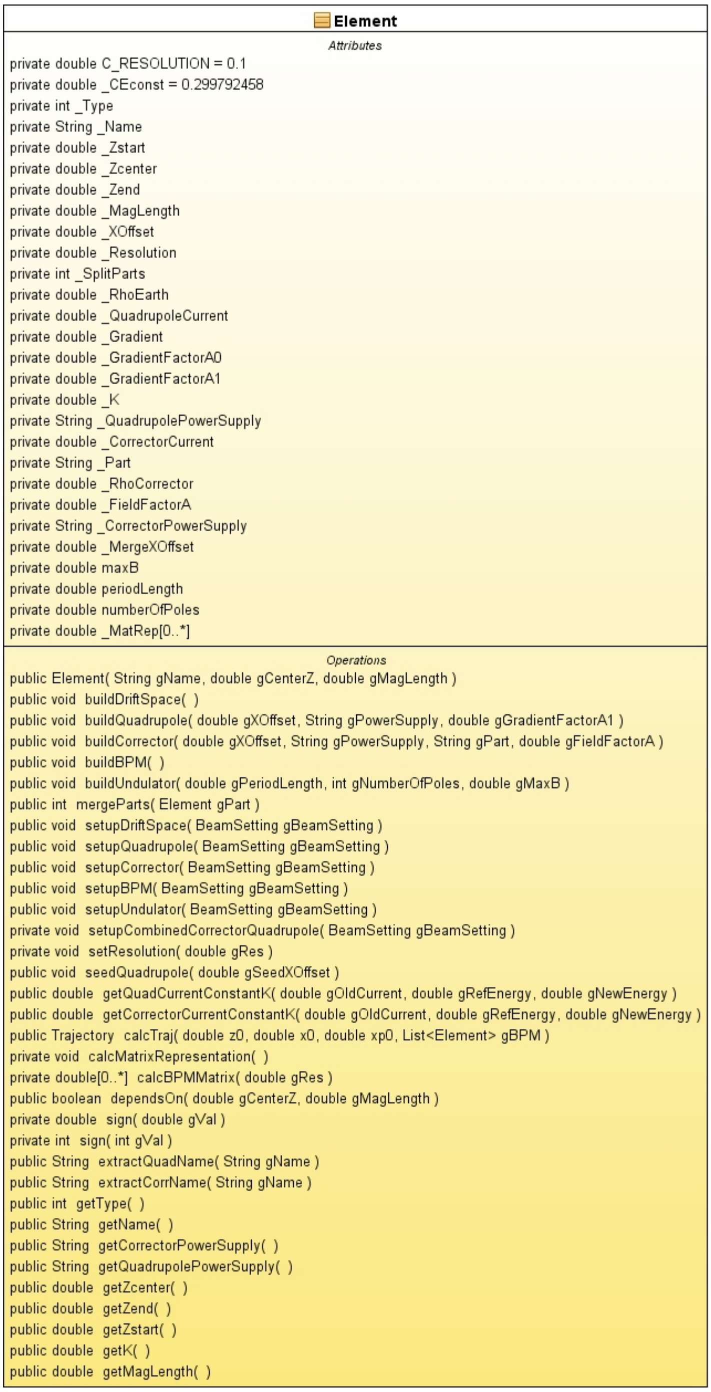
The class diagram for the class TOWorkingList is shown in figure 1.2. This class is responsible for holding all settings, data and tasks for the simulation as well as the results.



```
TOWorkingList
Attributes
private int _CorrectorMask[0..*]
private int _NoOfTasks
Operations
public TOWorkingList( )
private List<double[]>  calculateDX( List<Trajectory> gTrajList )
private List<double[]>  calculateDXBPM( List<Trajectory> gTrajList )
private List<double[]>  calculateX( List<Trajectory> gTrajList )
private List<Double>  calculateDE( List<BeamSetting> gBeamSettingList )
private List<Double>  calculateE( List<BeamSetting> gBeamSettingList )
private List<Double>  calculateXatZ( int gZind, List<Trajectory> gTrajList, List<BeamSetting> gBeamSettingList )
private List<Double>  calculateXEConst( List<Trajectory> gTrajList, List<BeamSetting> gBeamSettingList )
private List<Double>  calculateDispersion( List<Trajectory> gTrajList, List<BeamSetting> gBeamSettingList )
private List<Double>  calculateDispersionBPM( List<Trajectory> gTrajList, List<BeamSetting> gBeamSettingList )
private CurrentList  optimizeCurrent( int gCorrMask[0..*], List<Double> gConstraintX, List<Trajectory> gTrajList, List<BeamSetting> gBeamSettingList )
private CurrentList  optimizeCurrentAngleLastCorr( int gCorrMask[0..*], List<Double> gConstraintX, List<Trajectory> gTrajList, List<BeamSetting> gBeamSettingList )
private CurrentList  optimizeCurrentJMINLastCorr( int gCorrMask[0..*], List<Double> gConstraintX, List<Trajectory> gTrajList, List<BeamSetting> gBeamSettingList )
private CurrentList  optimizeCurrentIterate2( int gCorrMask[0..*], double gCFactor, List<Trajectory> gTrajList, List<BeamSetting> gBeamSettingList )
private CurrentList  optimizeCurrentJMIN( int gCorrMask[0..*], List<Double> gConstraintX, List<Trajectory> gTrajList, List<BeamSetting> gBeamSettingList )
private void  fitToExperimentalData2PointMethod( List<Trajectory> gTrajList )
private void  fitToExperimentalDataJMIN5Point( List<Trajectory> gTrajList )
private double  mimimizeJMINlastCorrector( List<Trajectory> gTrajList, int gTrajIndex )
private List<Double>  mimimizeJMIN( List<Trajectory> gTrajList )
private double  IntegrateRiemannMeanDispersion( List<Double> gD, Trajectory gTraj, String gIntMode )
private double  RMSBPMDispersion( List<Double> gD )
private double  PeakToPeakDispersion( List<Double> gD )
private double  PeakToPeakDispersionBPM( List<Double> gD )
private double  IntegrateRiemannMeanDispersion( List<Double> gD, Trajectory gTraj, String gIntMode, double gZmin, double gZmax )
private double  RMSBPMDispersion( List<Double> gDBPM, Trajectory gTraj, double gZmin, double gZmax )
private double  PeakToPeakDispersion( List<Double> gD, Trajectory gTraj, double gZmin, double gZmax )
private double  PeakToPeakDispersionBPM( List<Double> gDBPM, Trajectory gTraj, double gZmin, double gZmax )
public List<Double>  seedQuadrupoles( int gQno )
public void  addSetting( PlotSetting gPlotSetting, BeamSetting gBeamSetting )
public void  addSettingMCS( PlotSetting gPlotSetting, BeamSetting gBeamSetting, MonteCarloSimulation gMonteCarloSimulation )
public void  addSettingEXP( PlotSetting gPlotSetting, BeamSetting gBeamSetting, DataPoints gExpDataPoints )
public void  setTrajectoryForRun( int gInd, Trajectory finalTraj )
public void  setCorrectorMask( int gCorrectorMask[0..*] )
public void  addMCSResult( )
public void  processAll( )
private void  saveTrajectories( List<Trajectory> gTrajList, String gId )
private void  saveCorrCurrents( CurrentList gI, String gId )
private void  saveOptimizationResultsFile( CurrentList gI, Trajectory gTraj, double gMode, String gId )
private void  saveTrajectoryResultsFile( Trajectory gTraj, String gId )
private void  saveDispersionResultsFile( List<Double> gD, List<Double> gD_BMP, Trajectory gTraj, String gId )
private void  pdfTrajectories( JFreeChart gChart, String gId )
private void  svgTrajectories( JFreeChart gChart, String gId )
private void  saveMCSResults( String gId )
private void  saveMCSLongResults( String gId )
private String  getFname( String gUse, int gInd, String gId )
private void  directoryCheck( )
public void  checkTaskConsistency( )
private int  absI( int gVal )
private int  noOfIndivCorr( )
private List<Trajectory>  SDcpyTrajList( List<Trajectory> gList )
private double  getMin( List<Double> gList )
private double  getMax( List<Double> gList )
private double  getLim( List<Double> gList )
private double  getMinT( List<Double> gList, Trajectory gTraj, double gZmin, double gZmax )
private double  getMaxT( List<Double> gList, Trajectory gTraj, double gZmin, double gZmax )
private double  getMinB( List<Double> gList, Trajectory gTraj, double gZmin, double gZmax )
private double  getMaxB( List<Double> gList, Trajectory gTraj, double gZmin, double gZmax )
public void  findZforExpData( BeamLine gBL )
public int  getNoOfTasks( )
public int  getMCSN( )
public List<String>  getMCSQuadrupoleNames( )
public BeamSetting  getBeamSetting( int gInd )
private List<Trajectory>  getTrajectories( int gType )
private List<PlotSetting>  getPlotSettings( int gType )
private List<BeamSetting>  getBeamSettings( int gType )
public int  getMainType( )
```

Figure 1.2: Class TOWorkingList

The class diagram for the class Element is shown in figure 1.3. This class represents an element of the beamline.



Element

Attributes

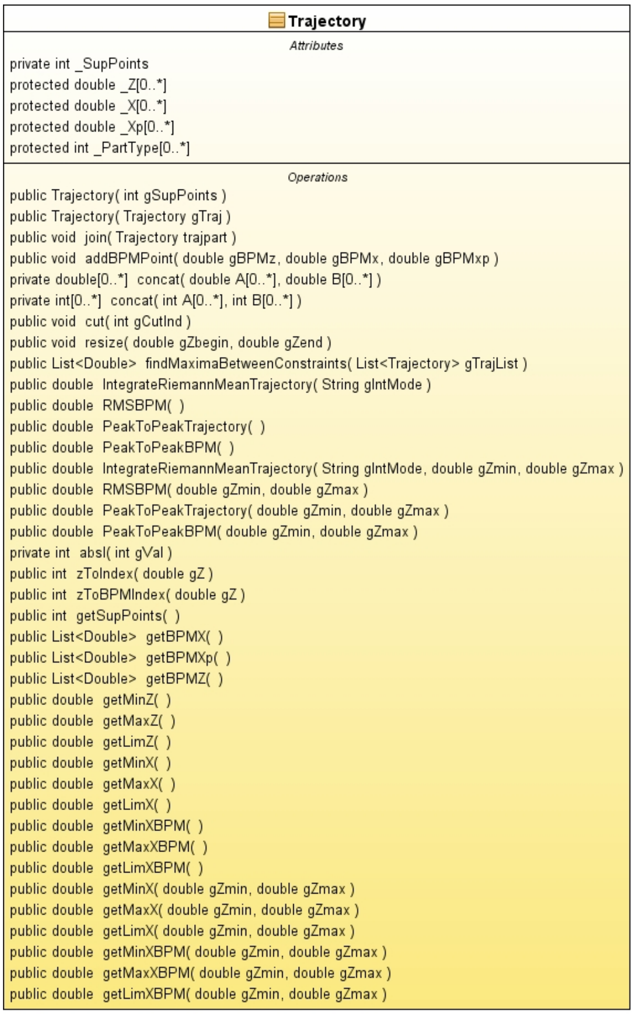
private double C_RESOLUTION = 0.1
private double _CEconst = 0.299792458
private int _Type
private String _Name
private double _Zstart
private double _Zcenter
private double _Zend
private double _MagLength
private double _XOffset
private double _Resolution
private int _SplitParts
private double _RhoEarth
private double _QuadrupoleCurrent
private double _Gradient
private double _GradientFactorA0
private double _GradientFactorA1
private double _K
private String _QuadrupolePowerSupply
private double _CorrectorCurrent
private String _Part
private double _RhoCorrector
private double _FieldFactorA
private String _CorrectorPowerSupply
private double _MergeXOffset
private double maxB
private double periodLength
private double numberOfPoles
private double _MatRep[0..*]

Operations

public Element(String gName, double gCenterZ, double gMagLength)
public void buildDriftSpace( )
public void buildQuadrupole(double gXOffset, String gPowerSupply, double gGradientFactorA1)
public void buildCorrector(double gXOffset, String gPowerSupply, String gPart, double gFieldFactorA)
public void buildBPM( )
public void buildUndulator(double gPeriodLength, int gNumberOfPoles, double gMaxB)
public int mergeParts(Element gPart)
public void setupDriftSpace(BeamSetting gBeamSetting)
public void setupQuadrupole(BeamSetting gBeamSetting)
public void setupCorrector(BeamSetting gBeamSetting)
public void setupBPM(BeamSetting gBeamSetting)
public void setupUndulator(BeamSetting gBeamSetting)
private void setupCombinedCorrectorQuadrupole(BeamSetting gBeamSetting)
private void setResolution(double gRes)
public void seedQuadrupole(double gSeedXOffset)
public double getQuadCurrentConstantK(double gOldCurrent, double gRefEnergy, double gNewEnergy)
public double getCorrectorCurrentConstantK(double gOldCurrent, double gRefEnergy, double gNewEnergy)
public Trajectory calcTraj(double z0, double x0, double xp0, List<Element> gBPM)
private void calcMatrixRepresentation( )
private double[0..*] calcBPMMatrix(double gRes)
public boolean dependsOn(double gCenterZ, double gMagLength)
private double sign(double gVal)
private int sign(int gVal)
public String extractQuadName(String gName)
public String extractCorrName(String gName)
public int getType( )
public String getName( )
public String getCorrectorPowerSupply( )
public String getQuadrupolePowerSupply( )
public double getZcenter( )
public double getZend( )
public double getZstart( )
public double getK( )
public double getMagLength( )

Figure 1.3: Class Element

The class diagram for the class Trajectory is shown in figure 1.4. This class holds the calculated trajectory data and is able to do some calculations on it.



```
Trajectory
Attributes
private int _SupPoints
protected double _Z[0..*]
protected double _X[0..*]
protected double _Xp[0..*]
protected int _PartType[0..*]
Operations
public Trajectory( int gSupPoints )
public Trajectory( Trajectory gTraj )
public void  join( Trajectory trajpart )
public void  addBPMPoint( double gBPMz, double gBPMx, double gBPMxp )
private double[0..*]  concat( double A[0..*], double B[0..*] )
private int[0..*]  concat( int A[0..*], int B[0..*] )
public void  cut( int gCutInd )
public void  resize( double gZbegin, double gZend )
public List<Double>  findMaximaBetweenConstraints( List<Trajectory> gTrajList )
public double  IntegrateRiemannMeanTrajectory( String gIntMode )
public double  RMSBPM(  )
public double  PeakToPeakTrajectory(  )
public double  PeakToPeakBPM(  )
public double  IntegrateRiemannMeanTrajectory( String gIntMode, double gZmin, double gZmax )
public double  RMSBPM( double gZmin, double gZmax )
public double  PeakToPeakTrajectory( double gZmin, double gZmax )
public double  PeakToPeakBPM( double gZmin, double gZmax )
private int  absI( int gVal )
public int  zToIndex( double gZ )
public int  zToBPMIndex( double gZ )
public int  getSupPoints(  )
public List<Double>  getBPMX(  )
public List<Double>  getBPMXp(  )
public List<Double>  getBPMZ(  )
public double  getMinZ(  )
public double  getMaxZ(  )
public double  getLimZ(  )
public double  getMinX(  )
public double  getMaxX(  )
public double  getLimX(  )
public double  getMinXBPM(  )
public double  getMaxXBPM(  )
public double  getLimXBPM(  )
public double  getMinX( double gZmin, double gZmax )
public double  getMaxX( double gZmin, double gZmax )
public double  getLimX( double gZmin, double gZmax )
public double  getMinXBPM( double gZmin, double gZmax )
public double  getMaxXBPM( double gZmin, double gZmax )
public double  getLimXBPM( double gZmin, double gZmax )
```

Figure 1.4: Class Trajectory

1.2 XML Input for TOP

The input of the simulation consists of four xml files. One specifies the layout of the beamline, one applies given settings to this beamline, one tells the simulation what simulations are to be run and the fourth one is to specify some external data points (like experimentally measured data). These four files are structured in the following way:

- *A) CONTROL_*.xml*
- *B) LAYOUT_*.xml*
- *C) SETTINGS_*.xml*
- *D) DATA_*.xml*

A) CONTROL_*.xml

The CONTROL_*.xml are used to tell the simulation what to do and which layout and settings to use. These files have to have the following basic structure, where exactly one layoutFile, settingsFile and plot tag are required:

Listing 1.1: CONTROL_*.xml

```
<input>
        <layoutFile fileName="name and location of LAYOUT_*.xml" />
        <settingsFile fileName="name and location of SETTINGS_*.xml" />
        <plot batchTime="10000">
        <!-- Put here the plot commands or calculations you want to
            perform on the specified Beamline! -->
        </plot>
</input>
```

The node **plot** has the following attributes:

- *batchTime*: The time in ms (integer) until a timer closes the simulation in batch processing. Put it to zero or leave it away to not set the timer.

The following plot commands and calculations are possible (only one per control-file):

1. Plot the Trajectory, $f(z) = x$

```
<CHARTplot zmin="122.0" zmax="235.0" x="0.0" xp="0.0"
    magFieldY="-30e-06" switchQuadrupoles="on" stepping="0.01"
    title="experimental setup" screen="y" raw="y" pdf="y"
    svg="y" plotNo="1" />
```

This creates a plot of the trajectory: horizontal beam position x VS longitudinal beam position z.
It is possible to use several of the **CHARTplot** tags at the same time, then all specified trajectories will be plotted in the same figure.
The parameters of this option are the following:

- *zmin*: The z starting point of the calculation in m.

- *zmax*: The z ending point of the calculation in m.

- *x*: The x starting point of the calculation at zmin in m.

- *xp*: The xp starting point of the calculation at zmin in m.

- *magFieldY*: A given external magnetic field B (i.e. the magnetic field of the Earth) in T.

- *switchQuadrupoles*: This option can be: on or off. It tells the program to set all quadrupole currents to zero.

- *stepping*: The spacing of calculated data points in m.

- *title*: The title of the figure. Be aware that in the case of multiple CHARTplot commands the title of the FIRST CHARTplot will be used for the figure

- *zMinCalc*: Only the section of the beamline in between zMinCalc and zMaxCalc will be used for the calculation of the integrals, RMS values and peak to peak values. z value in m.

- *zMaxCalc*: Only the section of the beamline in between zMinCalc and zMaxCalc will be used for the calculation of the integrals, RMS values and peak to peak values. z value in m.

- *screen*: Flag y or n. Tells the program if the output should be put to the screen.

- *raw*: Flag y or n. Tells the program if the output should be put to raw data format (calculated trajectories).

- *pdf*: Flag y or n. Tells the program if the output should be put to pdf (figure).

- *svg*: Flag y or n. Tells the program if the output should be put to svg (figure).

- *plotNo*: A running (integer) number of the plot.

2. Plot the Trajectory for different Energies at fixed k, $f(z) = x$

```
<CHARTenergyVariation zmin="122.0" zmax="235.0" x="0.0"
    xp="0.0" magFieldY="-30e-06" switchQuadrupoles="on"
    stepping="0.01" title="experimental setup" screen="y"
    raw="y" pdf="y" svg="y" plotNo="1" beamEnergyStart="0.4"
    beamEnergyEnd="1.0" beamEnergySteps="7"/>
```

This creates a plot of the trajectories for many different Energies (The k value stays constant, that means the quadrupole currents are rescaled for each new energy): horizontal beam position x VS longitudinal beam position z.
It is possible to use several of the **CHARTenergyVariation** tags at the same time, then all specified trajectories will be plotted in the same figure. It can also be mixed with CHARTplot and vice versa.
The parameters of this option are the following:

- *zmin*: The z starting point of the calculation in m.

- *zmax*: The z ending point of the calculation in m.

- *x*: The x starting point of the calculation at zmin in m.

- *xp*: The xp starting point of the calculation at zmin in m.

- *magFieldY*: A given external magnetic field B (i.e. the magnetic field of the Earth) in T.

- *switchQuadrupoles*: This option can be: on or off. It tells the program to set all quadrupole currents to zero.

- *stepping*: The spacing of calculated data points in m.

- *title*: The title of the figure. Be aware that in the case of multiple CHARTplot commands the title of the FIRST CHARTplot will be used for the figure

- *screen*: Flag y or n. Tells the program if the output should be put to the screen.

- *raw*: Flag y or n. Tells the program if the output should be put to raw data format (calculated trajectories).

- *pdf*: Flag y or n. Tells the program if the output should be put to pdf (figure).

- *svg*: Flag y or n. Tells the program if the output should be put to svg (figure).

- *plotNo*: A running (integer) number of the plot.

- *beamEnergyStart*: The lowest Energy used for plotting, give in GeV.

- *beamEnergyEnd*: The highest Energy used for plotting, give in GeV.

- *beamEnergySteps*: Number of trajectories plotted in between beamEnergyStart and beamEnergyEnd. In addition to these the Trajectory of the reference energy will be plotted.

3. Plot the distance between trajectories, $f(z) = \frac{dx}{dE}$

```
<CHARTdxOverZ zmin="122.0" zmax="235.0" x="0.0" xp="0.0"
    magFieldY="-30e-06" switchQuadrupoles="on" stepping="0.01"
    title="experimental setup" screen="y" raw="y" pdf="y"
    svg="y" plotNo="1" beamEnergyStart="0.4"
    beamEnergyEnd="1.0" beamEnergySteps="7"/>
```

CHARTdxOverZ creates a plot of the distance between trajectories for many different Energies (The k value stays constant, that means the quadrupole currents are rescaled for each new energy): difference in horizontal beam position divided by difference in energy dx/dE VS longitudinal beam position z.

The parameters of this option are the following:

- *zmin*: The z starting point of the calculation in m.

- *zmax*: The z ending point of the calculation in m.

- *x*: The x starting point of the calculation at zmin in m.

- *xp*: The xp starting point of the calculation at zmin in m.

- *magFieldY*: A given external magnetic field B (i.e. the magnetic field of the Earth) in T.

- *switchQuadrupoles*: This option can be: on or off. It tells the program to set all quadrupole currents to zero.

- *stepping*: The spacing of calculated data points in m.

- *title*: The title of the figure. Be aware that in the case of multiple CHARTplot commands the title of the FIRST CHARTplot will be used for the figure

- *screen*: Flag y or n. Tells the program if the output should be put to the screen.

- *raw*: Flag y or n. Tells the program if the output should be put to raw data format (calculated trajectories).

- *pdf*: Flag y or n. Tells the program if the output should be put to pdf (figure).

- *svg*: Flag y or n. Tells the program if the output should be put to svg (figure).

- *plotNo*: A running (integer) number of the plot.

- *beamEnergyStart*: The lowest Energy used for plotting, give in GeV.

- *beamEnergyEnd*: The highest Energy used for plotting, give in GeV.

- *beamEnergySteps*: Number of trajectories plotted in between beamEnergyStart and beamEnergyEnd. In addition to these the Trajectory of the reference energy will be plotted.

4. Plot the x position of the trajectory and the x position times the energy at a given z position, $f(E) = x|_{z=const.}$ and $f(E) = x \cdot E|_{z=const.}$

```
<CHARTdxOverE zmin="122.0" zmax="235.0" x="0.0" xp="0.0"
    magFieldY="-30e-06" switchQuadrupoles="on" stepping="0.01"
    title="experimental setup" screen="y" raw="y" pdf="y"
    svg="y" plotNo="1" beamEnergyStart="0.4"
    beamEnergyEnd="1.0" beamEnergySteps="7" zPlot="150">
```

CHARTdxOverE creates a plot of the x position and the x position multiplied by the energy as a function of energy. The values are computed for a given z value. The k value stays constant for different energies, that means the quadrupole currents are rescaled for each new energy. Note that: $f(E) = x \cdot E|_{z=const.} = const.$

The parameters of this option are the following:

- *zmin*: The z starting point of the calculation in m.

- *zmax*: The z ending point of the calculation in m.

- *x*: The x starting point of the calculation at zmin in m.

- *xp*: The xp starting point of the calculation at zmin in m.

- *magFieldY*: A given external magnetic field B (i.e. the magnetic field of the Earth) in T.

- *switchQuadrupoles*: This option can be: on or off. It tells the program to set all quadrupole currents to zero.

- *stepping*: The spacing of calculated data points in m.

- *title*: The title of the figure. Be aware that in the case of multiple CHARTplot commands the title of the FIRST CHARTplot will be used for the figure

- *screen*: Flag y or n. Tells the program if the output should be put to the screen.

- *raw*: Flag y or n. Tells the program if the output should be put to raw data format (calculated trajectories).

- *pdf*: Flag y or n. Tells the program if the output should be put to pdf (figure).

- *svg*: Flag y or n. Tells the program if the output should be put to svg (figure).

- *plotNo*: A running (integer) number of the plot.

- *beamEnergyStart*: The lowest Energy used for plotting, give in GeV.

- *beamEnergyEnd*: The highest Energy used for plotting, give in GeV.

- *beamEnergySteps*: Number of trajectories plotted in between beamEnergyStart and beamEnergyEnd. In addition to these the Trajectory of the reference energy will be plotted.

- *zPlot*: The constant z value, the plot will be created for in m.

5. Plot the x position times the Energy as a function of z, $f(z) = x \cdot E$

```
<CorrectorSlope zmin="122.0" zmax="235.0" x="0.0" xp="0.0"
    magFieldY="-30e-06" switchQuadrupoles="on" stepping="0.01"
    title="experimental setup" screen="y" raw="y" pdf="y"
    svg="y" plotNo="1" beamEnergyStart="0.4"
    beamEnergyEnd="1.0" beamEnergySteps="3"/>
```

CorrectorSlope creates a plot of the x position multiplied by the energy (constant value for given z) as a function of z. The k value stays constant for different energies, that means the quadrupole currents are rescaled for each new energy.

The parameters of this option are the following:

- *zmin*: The z starting point of the calculation in m.

- *zmax*: The z ending point of the calculation in m.

- *x*: The x starting point of the calculation at zmin in m.

- *xp*: The xp starting point of the calculation at zmin in m.

- *magFieldY*: A given external magnetic field B (i.e. the magnetic field of the Earth) in T.

- *switchQuadrupoles*: This option can be: on or off. It tells the program to set all quadrupole currents to zero.

- *stepping*: The spacing of calculated data points in m.

- *title*: The title of the figure. Be aware that in the case of multiple CHARTplot commands the title of the FIRST CHARTplot will be used for the figure

- *screen*: Flag y or n. Tells the program if the output should be put to the screen.

- *raw*: Flag y or n. Tells the program if the output should be put to raw data format (calculated trajectories).

- *pdf*: Flag y or n. Tells the program if the output should be put to pdf (figure).

- *svg*: Flag y or n. Tells the program if the output should be put to svg (figure).

- *plotNo*: A running (integer) number of the plot.

- *beamEnergyStart*: The lowest Energy used for plotting, give in GeV.

- *beamEnergyEnd*: The highest Energy used for plotting, give in GeV.

- *beamEnergySteps*: Number of trajectories plotted in between beamEnergyStart and beamEnergyEnd. In addition to these the Trajectory of the reference energy will be plotted.

- *zPlot*: The constant z value, the plot will be created for in m.

6. Optimize the corrector current to achieve the best trajectory

```
<optimizeCurrent zmin="122.5" zmax="235.0" x="0.0" xp="0.0"
    magFieldY="-30.0e-6" switchQuadrupoles="on"
    stepping="0.01" title="Optimize Corrector Currents Q on"
    screen="y" raw="y" pdf="y" svg="y" plotNo="1"
    optimizationMode="0.0"/>
```

optimizeCurrent tries to find the best possible currents for the correctors to compensate the Earth's magnetic field. Different strategies are applied. The found solution will be plotted.

The parameters of this option are the following:

- *zmin*: The z starting point of the calculation in m.

- *zmax*: The z ending point of the calculation in m.

- *x*: The x starting point of the calculation at zmin in m.

- *xp*: The xp starting point of the calculation at zmin in m.

- *magFieldY*: A given external magnetic field B (i.e. the magnetic field of the Earth) in T.

- *switchQuadrupoles*: This option can be: on or off. It tells the program to set all quadrupole currents to zero.

- *stepping*: The spacing of calculated data points in m.
- *title*: The title of the figure. Be aware that in the case of multiple CHARTplot commands the title of the FIRST CHARTplot will be used for the figure
- *screen*: Flag y or n. Tells the program if the output should be put to the screen.
- *raw*: Flag y or n. Tells the program if the output should be put to raw data format (calculated trajectories).
- *pdf*: Flag y or n. Tells the program if the output should be put to pdf (figure).
- *svg*: Flag y or n. Tells the program if the output should be put to svg (figure).
- *plotNo*: A running (integer) number of the plot.
- *optimizationMode*: a double number that specifies the optimization algorithm used.
  - a number > 0.0: the 2 iteration maximum constraint approach will be used with JMinuit for the last corrector
  - a number $= 0.0$: the one iteration method will be used with JMinuit for the last corrector
  - a number $= 777.0$: the whole trajectory will be optimized using JMinuit

7. Plot the dispersion of the beamline, $D(z) = \frac{dx}{\frac{dE}{E_0}}$

```
<DispersionRelation zmin="122.0" zmax="235.0" x="0.0"
    xp="0.0" magFieldY="-30e-06" switchQuadrupoles="on"
    stepping="0.01" title="experimental setup" screen="y"
    raw="y" pdf="y" svg="y" plotNo="1"
    beamEnergyDelta="0.0001"/>
```

DispersionRelation plot the dispersion for the beamline using a give absolute energy increment to simulate infinitesimal increase. The method converges very fast to the values for infinitesimal calculation.

The parameters of this option are the following:

- *zmin*: The z starting point of the calculation in m.
- *zmax*: The z ending point of the calculation in m.
- *x*: The x starting point of the calculation at zmin in m.
- *xp*: The xp starting point of the calculation at zmin in m.
- *magFieldY*: A given external magnetic field B (i.e. the magnetic field of the Earth) in T.

- *switchQuadrupoles*: This option can be: on or off. It tells the program to set all quadrupole currents to zero.

- *stepping*: The spacing of calculated data points in m.

- *title*: The title of the figure. Be aware that in the case of multiple CHARTplot commands the title of the FIRST CHARTplot will be used for the figure

- *zMinCalc*: Only the section of the beamline in between zMinCalc and zMaxCalc will be used for the calculation of the integrals, RMS values and peak to peak values. z value in m.

- *zMaxCalc*: Only the section of the beamline in between zMinCalc and zMaxCalc will be used for the calculation of the integrals, RMS values and peak to peak values. z value in m.

- *screen*: Flag y or n. Tells the program if the output should be put to the screen.

- *raw*: Flag y or n. Tells the program if the output should be put to raw data format (calculated trajectories).

- *pdf*: Flag y or n. Tells the program if the output should be put to pdf (figure).

- *svg*: Flag y or n. Tells the program if the output should be put to svg (figure).

- *plotNo*: A running (integer) number of the plot.

- *beamEnergyDelta*: Energy increment in GeV. 0.0001 should be sufficient.

8. Monte Carlo Simulation of the effects of unproperly placed quadrupoles on the trajectory and the dispersion

```
<MonteCarloQuadrupole zmin="122.0" zmax="235.0" x="0.0"
    xp="0.0" magFieldY="0.0" switchQuadrupoles="on"
    stepping="0.01" title="Monte Carlo Simulation for D -
    Sigma dependence - 1.0 BOFF - ECOL/TCOL" screen="y"
    raw="y" pdf="y" svg="y" plotNo="1" N="1000"
    approxQuadYQuality="0.0010" randMode="gaussian"
    zMinCalc="203.43225" zMaxCalc="233.42675"
    beamEnergyDelta="0.0001" usedQuadrupoles="Q3ECOL Q4ECOL
    Q5ECOL Q2TCOL Q8TCOL Q9TCOL" beamEnergyFix="0.0"/>
```

MonteCarloQuadrupole simulates the effects on the trajectory and the dispersion for not perfectly aligned quadrupoles in the beamline. A histogram and data of: the quadratic integral of the trajectory, the RMS at the BPMs, the peak to peak trajectory, the peak to peak at BPM, the quadratic integral of the dispersion, the dispersion RMS at the BPMs, the peak to peak dispersion and the peak to peak dispersion at BPM is available.

The parameters of this option are the following:

- *zmin*: The z starting point of the calculation in m.

- *zmax*: The z ending point of the calculation in m.
- *x*: The x starting point of the calculation at zmin in m.
- *xp*: The xp starting point of the calculation at zmin in m.
- *magFieldY*: A given external magnetic field B (i.e. the magnetic field of the Earth) in T.
- *switchQuadrupoles*: This option can be: on or off. It tells the program to set all quadrupole currents to zero.
- *stepping*: The spacing of calculated data points in m.
- *title*: The title of the figure. Be aware that in the case of multiple CHARTplot commands the title of the FIRST CHARTplot will be used for the figure
- *screen*: Flag y or n. Tells the program if the output should be put to the screen.
- *raw*: Flag y or n. Tells the program if the output should be put to raw data format (calculated trajectories).
- *pdf*: Flag y or n. Tells the program if the output should be put to pdf (figure).
- *svg*: Flag y or n. Tells the program if the output should be put to svg (figure).
- *plotNo*: A running (integer) number of the plot.
- *N*: Number of Monte Carlo runs.
- *approxQuadYQuality*: The sigma of the Gaussian mode or cutoff of the uniform mode in m.
- *randMode*: either Gaussian or uniform.
- *zMinCalc*: Only the section of the beamline in between zMinCalc and zMaxCalc will be used for the calculation of the integrals, RMS values and peak to peak values. z value in m.
- *zMaxCalc*: Only the section of the beamline in between zMinCalc and zMaxCalc will be used for the calculation of the integrals, RMS values and peak to peak values. z value in m.
- *beamEnergyDelta*: Energy increment in GeV. 0.0001 should be sufficient. Used for the calculation of the dispersion.
- *usedQuadrupoles*: A list of the names of quadrupoles which should be included in the Monte Carlo process.
- *beamEnergyFix*: It is possible to run for a different energy than specified in the settings. The k value stays constant for different energies, that means the quadrupole currents are rescaled for the new energy.

9. Plot the trajectory and measurements to compare them

```
<CompareWithExperimentalData zmin="122.0" zmax="235.0"
    x="0.0" xp="0.0" magFieldY="-30e-06"
    switchQuadrupoles="on" stepping="0.01" title="Comparison
    Simulation - Experimental Data" screen="y" raw="y" pdf="y"
    svg="y" plotNo="1"
    dataFileName="\BEAMLINE\DATA\DATA_FLASH_8966.xml"/>
```

CompareWithExperimentalData fits the simulation to two of the measured data points to be able to compare the simulation and real data.

The parameters of this option are the following:

- *zmin*: The z starting point of the calculation in m.
- *zmax*: The z ending point of the calculation in m.
- *x*: The x starting point of the calculation at zmin in m.
- *xp*: The xp starting point of the calculation at zmin in m.
- *magFieldY*: A given external magnetic field B (i.e. the magnetic field of the Earth) in T.
- *switchQuadrupoles*: This option can be: on or off. It tells the program to set all quadrupole currents to zero.
- *stepping*: The spacing of calculated data points in m.
- *title*: The title of the figure. Be aware that in the case of multiple CHARTplot commands the title of the FIRST CHARTplot will be used for the figure
- *screen*: Flag y or n. Tells the program if the output should be put to the screen.
- *raw*: Flag y or n. Tells the program if the output should be put to raw data format (calculated trajectories).
- *pdf*: Flag y or n. Tells the program if the output should be put to pdf (figure).
- *svg*: Flag y or n. Tells the program if the output should be put to svg (figure).
- *plotNo*: A running (integer) number of the plot.
- *dataFileName*: Filename of the measured data.

*B) LAYOUT_ *.xml*

The LAYOUT_*.xml is used to specify the elements of the beamline:

Listing 1.2: LAYOUT_*.xml

```
<input>
  <layout title="FLASH" zMin="122.5" zMax="235.0">
  <!-- Undulators -->
  <!-- Quadrupoles -->
  <!-- Correctors / Steerers -->
  <!-- Beam Position Monitors -->
  </layout>
</input>
```

The node **layout** has the following attributes:

- *title*: Name of the layout/accelerator.
- *zMin*: Longitudinal starting position of this layout (in m).
- *zMax*: Longitudinal ending position of this layout (in m).

The layout has to be designed carefully. It is not possible to have overlapping elements (except for merged quadrupole-correctors (see below) or BPMs).

The beamline can consist of the following elements:

1. *Quadrupole*, $f(z) = x$

   ```
   <quadrupole name="Q4SUND3" centerZ="180.198"
       magLength="0.2768" gradientFactorA1="0.128"
       powerSupply="Q5SUND1" xOffset="0.001"/>
   ```

   This creates a quadrupole at the specified position.
   The parameters of the quadrupole are the following:

   - *name*: The name (unique) of the element.
   - *centerZ*: The position of the center of the element in m on the longitudinal axes.
   - *magLength*: The effective length of the element in m.
   - *gradientFactorA1*: Factor for the gradient of the quadrupole: $gradient[\frac{T}{m}] = a0[\frac{T}{m}] + a1[\frac{T}{mA}] * I[A]$.
   - *powerSupply*: The name of the element's power supply. Do not specify if the element has its own supply.
   - *xOffset*: The offset in x direction in m.

2. *Corrector*

   ```
   <corrector plane="h" name="H9ACC6" centerZ="122.809"
       magLength="0.185" fieldFactorA="7.1E-4"
       powerSupply="H10ACC6"/>
   ```

This creates a corrector at the specified position. It is possible to merge quadrupoles and correctors if they have the same centerZ and the same magLength.
The parameters of the corrector are the following:

- *plane*: The plane the corrector is used for: "h" or "v".
- *name*: The name (unique) of the element.
- *centerZ*: The position of the center of the element in m on the longitudinal axes.
- *magLength*: The effective length of the element in m.
- *fieldFactorA*: Factor for the field of the corrector: $B[T] = a[\frac{T}{A}] * I[A]$.
- *powerSupply*: The name of the element's power supply. Do not specify if the element has its own supply.

3. *Undulator*

```
<undulator name="UND0" centerZ="205.6845" maxB="0.486"
    periodLength="0.0273" numberOfPoles="330"/>
```

This creates a undulator at the specified position.
The parameters of the undulator are the following:

- *name*: The name (unique) of the element.
- *centerZ*: The position of the center of the element in m on the longitudinal axes.
- *maxB*: The B field of the dipoles in T.
- *periodLength*: The length of one undulator period in m.
- *numberOfPoles*: The number of poles of the undulator. (Note: $magLength = \frac{1}{2} periodLength \cdot numberOfPoles$)

4. *Beam position monitor (BPM)*

```
<BPM name="9ACC6" centerZ="122.56"/>
```

This creates a beam position monitor at the specified position. The BPM is tho only element which can be at any position of the beamline with respect to the other elements (i.e. it can be in an undulator, quadrupole, steerer or in between). The length of the BPM is approximately 0.
The parameters of the BPM are the following:

- *name*: The name (unique) of the element.
- *centerZ*: The position of the center of the element in m on the longitudinal axes.

*C) SETTINGS_ *.xml*

The SETTINGS_*.xml is used to specify the current setting of the beamline (beam energy and power):

Listing 1.3: SETTINGS_*.xml

```
<input>
  <settings title="FLASH" beamEnergy="1.0">
    <!-- put here the power settings -->
  </settings>
</input>
```

The node **settings** has the following attributes:

- *title*: Name of the layout/accelerator.
- *beamEnergy*: Energy of the beam in GeV.

The settings you can specify are:

Powersupply

```
    <powerSupply name="Q9ACC6" current="20.791304"/>
```

This creates a power supply with a given name and current.

- *name*: The name (unique) of the power supply.
- *current*: The current of the power supply in A.

*D) DATA_ *.xml*

The DATA_*.xml is used to include experimentally measured data points in the simulation:

Listing 1.4: DATA_*.xml

```
<input>
        <data file="2008-11-21T174454">
                <!-- specify data -->
        </data>
</input>
```

The node **data** has the following attributes:

- *file*: Name of the measurement.

You can add measured data points:
Point

```
    <point name="6MATCH" x="0.00273479" />
```

This creates a data point for a given BPM.

- *name*: The name (unique) of a BPM.

- x: The measured x position in m.

1.3 Output produced by TOP

As TOP provides lots of different functionality the variety of produced output is large. All output however is saved to `C:\\RESULTS\`. TOP takes care of the directory structure itself. A naming convention for all files is applied using the filename of the input control file, the date and the type of output file.

In this results folder several sub folders provide the actual output these are:

- **CURRENTS**
  Files provided here contain the calculated optimized corrector currents in suitable xml format for use in a SETTINGS_*.xml file.

- **DISPRESULTS**
  Files provided here contain the calculated values for characteristic parameters of the dispersion D in the specified region of the beamline.

- **MONTECARLORESULTS**
  There are two types of files in this folder. One contains the characteristic parameters calculated in the MCS. The same file with "LONG" in the filename contains all data produced by the MCS for further use in other programs.

- **OPTRESULTS**
  Files in this section contain all output of a current-optimization run, like characteristic parameters (attention here they are calculated for the whole beamline!), currents and coordinates at the BPMs.

- **PDF**
  Contains the images of the produced plot as .pdf file.

- **RAW**
  Contains the raw data of all calculated trajectories (z, x, x', type) for use in other programs.

- **SVG**
  Contains the images of the produced plot as .svg file.

- **TRAJRESULTS**
  Files provided here contain the calculated values for characteristic parameters of the trajectory x in the specified region of the beamline and the coordinates at the BPMs.

- **DXVALUES**
  The calculated data points for the plot delta x - z.

- **DXRMS**
  The RMS of all lines calculated in an delta x - z run. May be multiple entries if this was a energy variation run (one RMS for each energy calculated).

1.4 Libraries needed by TOP

TOP needs several libraries to run which are:

- **JFreeChart**
  JFreeChart is used for plotting.
  Version used: 1.0.13
  Homepage: `http://www.jfree.org/jfreechart/`
  Available at: `http://sourceforge.net/projects/jfreechart/files/`

- **JCommon**
  JCommon is needed by JFreeChart.
  Version used: 1.0.16
  Homepage: `http://www.jfree.org/jcommon/`
  Available at: `http://sourceforge.net/projects/jfreechart/files/3.%20JCommon/`

- **iText**
  iText is used to convert charts produced by JFreeChart to pdf.
  Version used: 2.1.7
  Homepage: `http://www.lowagie.com/iText/`
  Available at: `http://www.lowagie.com/iText/download.html`

- **Batik**
  Batik is used to convert charts produced by JFreeChart to svg.
  Version used: 1.7
  Homepage: `http://xmlgraphics.apache.org/batik/`
  Available at: `http://xmlgraphics.apache.org/batik/download.cgi`

- **JMinuit**
  JMinuit is used for optimization (minimization) of functions.
  Version used: 1.0
  Homepage: `http://java.freehep.org/`
  Available at: `http://java.freehep.org/maven2/org/freehep/freehep-jminuit/1.0/`