LhARA linear optics documentation

A.N. Other

1 Introduction

Introduction to the documentation!

2 Coordinate systems

2.1 Laboratory coordinate system

The origin of the LhARA coordinate system, the "laboratory coordinate system" or "laboratory reference frame", is at the position of the laser focus at the position of the laser-target interaction. The z axis is horizontal and points along the nominal capture axis, pointing in the downstream direction, i.e. away from the target. The y axis points vertically upwards, and the x axis completes a right-handed orthogonal coordinate system.

Phase-space coordinates as well as vector and scalar quantities referred to the laboratory coordinate system will be written in lower case. Unit vectors along the x, y and z axes are $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ respectively. The position of the reference particle, its momentum and energy are described as functions of the distance it has travelled from the origin of coordinates to its current position. The distance travelled is defined to be s, making the position, $\boldsymbol{r}_r$, momentum, $\boldsymbol{p}_p$, and energy, ϵ_r, of the reference particle at position s:

$$
\begin{aligned}
\boldsymbol{r}_r &= \boldsymbol{r}(s)\,; \\
\boldsymbol{p}_r &= \boldsymbol{p}(s)\,; \text{and} \\
\epsilon_r &= \epsilon(s)\,.
\end{aligned}
$$

The time, t, at which the reference particle is at s is also a function of s:

$$
t = t(s) = \frac{p_r}{\epsilon_r c}\,;
$$

where $p_r = |\boldsymbol{p}_r|$ and c is the speed of light.

2.2 Reference particle local coordinate system

A coordinate system defined relative to the position of the reference particle, the "reference particle local coordinaate" (RPLC) system, may be defined using the direction in which the particle is travelling. The position of the particle defines the origin of the RPLC system, see figure **??**.

Phase-space coordinates as well as vector and scalar quantities referred to the laboratory coordinate system will be written in capital letters. The tangent to the reference particle trajectory at s defines the Z axis with unit vector $\boldsymbol{K}$. The presence of local electric or magnetic fields may cause the reference particle's trajectory to curve. In the neighbourhood of the particle, the curved trajectory may be described in terms of an arc of a cicle. The X axis (with unit vector $\boldsymbol{I}$) is then taken to be in the direction pointing towards the centre of the circle. The third coordinate axis, Y, is defined to complete the right-handed orthogonal coordinate system; the unit vector along the Y axis being given by $\boldsymbol{J} = \boldsymbol{K} \times \boldsymbol{I}$.

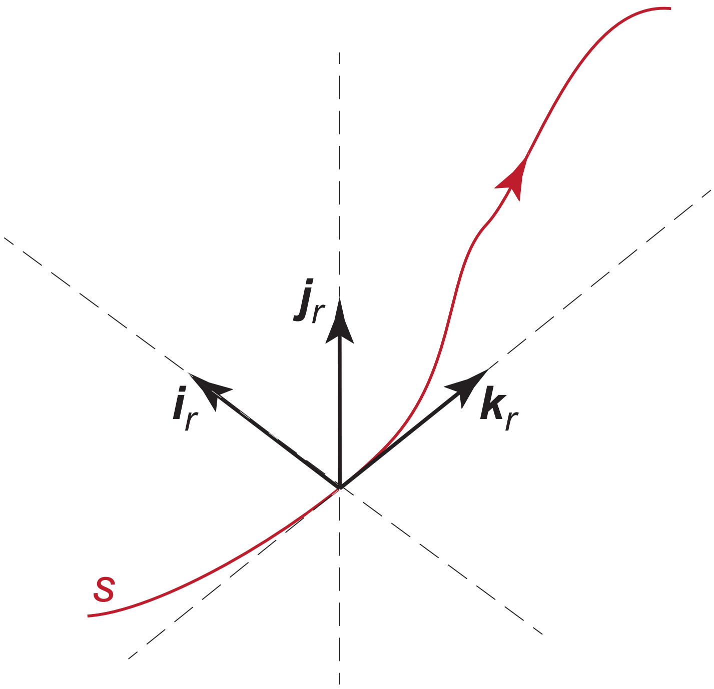

Figure 1: Reference particle local coordinate system. The trajectory of the reference particle is shown as the red line. The distance the reference particle has travelled, measured from the origin of coordinates in the laboratory frame, is labelled s. The origin of the RPLCs is coincident with the position of the reference particle. The directions of unit vectors along each of three righthanded, orthogonal coordinate axes are shown as black arrows labelled $\boldsymbol{i}_r$, $\boldsymbol{j}_r$, and $\boldsymbol{j}_r$.

The trajectory of the reference particle will be a straight line as it traverses a drift space or when the particle's energy is increased (or decreased) through an electric field applied parallel to its direction of motion. In such cases the RPLC axes are taken to coincide with either the laboratory coordinate system or the RPLC system defined at the exit of the beam-line element that preceded the drift space or accelating element.

2.3 Transforming to and from reference particle local coordinates to laboratory coordinates

In RPLC, the trajectory of the reference particle, $\boldsymbol{R}_r$, is:

$$\boldsymbol{R}_r(s) = \boldsymbol{0}\,. \tag{1}$$

The position of a test particle in the RPLC frame ($\boldsymbol{R}_t$) is described with reference to the position of the reference particle. In the laboratory frame, the position of the test particle is:

$$\boldsymbol{R}_t(s) = \boldsymbol{R}(s) + \boldsymbol{\delta r}(s)\,; \tag{2}$$

where:

$$\boldsymbol{\delta r}(s) = \underline{\underline{R}}(s)\boldsymbol{R}_t(s)\,; \tag{3}$$

where $\underline{\underline{R}}(s)$ is a rotation matrix that takes the RPLCs at s to the laboratory frame coordinates.

In the laboratory frame, the unit vectors $\boldsymbol{I}$, $\boldsymbol{J}$ and $\boldsymbol{K}$ may be written:

$$\begin{aligned}
\boldsymbol{i}_r &= \begin{pmatrix} i_{rx} \\ i_{ry} \\ i_{rz} \end{pmatrix} \\
\boldsymbol{j}_r &= \begin{pmatrix} j_{rx} \\ j_{ry} \\ j_{rz} \end{pmatrix} \\
\boldsymbol{k}_r &= \begin{pmatrix} k_{rx} \\ k_{ry} \\ k_{rz} \end{pmatrix}
\end{aligned}$$

The rotation matrix, $\underline{\underline{R}}$, may now be written:

$$\underline{\underline{R}}(s) = \begin{bmatrix} i_{rx} & j_{rx} & k_{rx} \\ i_{ry} & j_{ry} & k_{ry} \\ i_{rz} & j_{rz} & k_{rz} \end{bmatrix} \tag{4}$$

References

Introduction

This document summarises the steps needed to set-up and run LhARA_Beamline linear optics simulation of the LhARA beamine. A summary of the tasks that LhARA_Beamline software suite performs will be documented in due course. LhARA_Beamline has been developed in python; python 3 is assumed.

Getting the code

LhARA_Beamline is maintained using the GitHub version-control system. The latest release can be downloaded from the . . . should we moce to the LhARA repository ... its git, but, bespoke to CCAP/LhARA.

Dependencies and required packages

LhARA_Beamline requires the following packages:

- Python modules: `scipy`, `matplotlib`, `pandas`, and `iminuit`;
- CERN programme library: `pyroot` (which may be installed using the standard `root` installers, see the documentation at https://root.cern/install/).

It may be convenient to run LhARA_Beamline in a "virtual environment". To set this up, after updating your python installation to python 3.9.2, and installing root, execute the following commands:

1. `python3 -m venv --system-site-packages venv`
   - This creates the director `venv` that contains files related to the virtual environment.
2. `source venv/bin/activate`
3. `python -m pip install pandas scipy matplotlib iminuit`

To exit from the virtual environment, execute the command `deactivate`.
The command `source venv/bin/activate` places you back into the virtual environment.

Unpacking the code, directories, and running the tests

After downloading the package from GitHub, or cloning the repositiry, you will find a "`README.md`" file which provides some orientation and instructions to run the code. In particular, a `bash` script "`startup.bash`" is provided which:

- Sets the "`LhARAOpticsPATH`" environment variable so that the files that hold constants etc. required by the code can be located; and
- Adds "`01-Code`" (see below) to the PYTHONPATH. The scripts in "02-Tests" (see below) may then be run with the command "python 02-Tests/$<$ filename $>$.py".

Below the top directory, the directory structure in which the code is presented is:

`01-Code`: contains the python implementation as a series of modules. Each module contains a single class or a related set of methods.

`02-Tests`: contains self-contained test scripts that run the various methods and simulation packages defined in the code directory.

`11-Parameters`: contains the parameter set used in `02-Tests/RunSimulation.py` to generate muon decays in the production straight.

The instruction in the `README.md` file should be followed to set up and run the code.

Running the code

The file in 02-Tests/RunSimulation.py - will run the code.

The file **RunSimulation.py** contains:

- The definition of csv input file to control the running of the Simulation; and
- The call to the Simulation class with; the number of events to generate; the central energy to generate; and the filenames.