Detector Geometry in the BLAST Reconstruction and Monte Carlo

A.Maschinot

September 11, 2002

1 Introduction

This paper shall establish the conventions and define the variables used to completely position and orient any of the six types of detectors used in the BLAST reconstruction and Monte Carlo libraries. In addition, rotation identities and conventions will also be defined; such concepts are of fundamental importance in positioning and orienting the detectors in the BLAST setup.

2 Rotation Conventions

In order to avoid confusion, we shall, first and foremost, explicitly state the rotation conventions to be used in the BLAST reconstruction and Monte Carlo. These conventions should *always* be followed. Straying from them will only cause confusion.

1. We shall only use right-handed coordinate systems.
2. All rotations shall be active rotations. By definition, an *active* rotation is one in which the vector in question is rotated by an angle. (On the other hand, a *passive* rotation, which we shall *not* use, is one in which the coordinate system, not the vector, is rotated through the angle in the opposite direction.)
3. All rotations shall be in a right-handed-corresponding-to-positive- angle sense.

3 Defining the Six Degrees of Freedom

Now that we have specified our rotation conventions, we can go on to talk about what is required in order to position and orient a detector in the BLAST coordinate system.

In general, a coordinate system (hereafter referred to as the "daughter" coordinate system) existing within another coordinate system (hereafter referred to as the "mother" coordinate system) possesses six degrees of freedom. All six degrees of freedom need to be specified in order to position and orient completely the daughter system within the mother one.

For clarity, a distinction between the terms "positioning" and "orienting" as used in this paper should be pointed out here. The term "positioning" refers to the act of specifying those degrees of freedom (out of the initial six) which completely specify the coordinates of the daughter system's origin within the mother system. Similarly, the term "orienting" refers to the act of specifying those degrees of freedom (once again, out of the initial six) which completely specify the daughter system's axes positions *about the daughter system's origin*. These two processes (positioning and orienting) are independent of one another, and they nicely form a logical division of the six initial degrees of freedom.

In order to position a daughter system's origin within a mother system, three variables need to be specified. The three most common examples of the three positioning variables are 1) the x, y, and z Cartesian coordinates, 2) the r, θ, and ϕ shperical coordinates, and 3) the ρ, θ, and z cylindrical coordinates. For our purposes, the set of three spherical basis positioning variables shall be the most useful. For completeness, these spherical variables along well their ranges are defined below:

Name	Description	Range
r	Vector Magnitude	$0 \leq r \leq \infty$
θ	Polar Angle	$0 \leq \theta \leq \pi$
ϕ	Azimuthal Angle	$0 \leq \phi < 2\pi$

In terms of rotations (which will be extremely important later), we can see that choosing a spherical basis is equivalent to the following sequential operations:

1. Rotate the mother's (unit-length) $+z$-axis about the mother's $+y$-axis by the angle θ. Call the resulting vector $+z'$.

2. Rotate the $+z'$ (unit) vector about the mother's $+z$-axis by the angle ϕ. Call the resulting vector $+z''$.

3. Advance along the $+z''$ (unit) vector by a length r.

Now that we have decided on a procedure for positioning a daughter system's origin within a mother coordinate system, we shall now move on to the problem of orienting a daughter system's coordinate axes about its center. In order to do so, we first must set up a reference coordinate system, *centered also at the daughter system's origin*, which we will use to define the daughter system's orientation. The easiest choice for such a reference system is to simply choose the mother system's axes. As such, we "pick" up a copy of the mother system's axes and translate it to the origin of the daughter system.

Using this choice for a reference system, we can now define the three orientation variables. To do so, we imagine that the daughter's axes originally coincide with those of the reference system. First, we rotate the daughter system's axes about the reference system's $+y$-axis until the daughter system's $+z$-axis lies at the same polar angle (as measured from the reference system's $+z$-axis) as that of the final daughter system's $+z$-axis. The angle through which we rotated (about the reference system's $+y$-axis) we shall denote by α. Additionally, denote this (intermediate) set of daughter axes with a subscript α, as in $x \to x_\alpha$, $y \to y_\alpha$, and $z \to z_\alpha$.

The second rotation to be performed rotates the α-subscripted axes about the reference $+z$-axis until the $+z_\alpha$-axis coincides with the final desired $+z$-axis. This angle through which we rotated (about the reference system's $+z$-axis) we shall denote by β. Additionally, denote this intermediate set of axes with a β subscript, as in $x_\alpha \to x_\beta$, $y_\alpha \to y_\beta$, and $z_\alpha \to z_\beta$. Note, as previously stated, that, after the β rotation, the $+z_\beta$-axis coincides with the final desired $+z$-axis.

The third (and final) rotation to be performed is one rotating the $+x_\beta$-axis about the $+z\beta$-axis (or, equivalently, about the final desired $+z$-axis) until it coincides with the final desired $+x$-axis. This angle through which we rotate (about the $+z\beta$-axis) we shall denote by γ. At this point, we have finished with our orientation; the rotated β axes now all coincide with the desired set of axes.

We have thus succeeded in defining three orientation angles (α, β, and γ) along with three direction vectors (the reference $+y$-axis, the reference $+z$-axis, and the $+z\beta$-axis) which, when applied in subsequent order, will rotate a set of axes from the reference (i.e. mother) axes to the desired axes. For completeness, the orientation angles along with their ranges are given below:

Name	Description	Range
α	Rotation about reference $+y$-axis	$0 \le \alpha \le \pi$
β	Rotation about reference $+z$-axis	$0 \le \beta < 2\pi$
γ	Rotation about $+z_\beta$ axis	$0 \le \gamma < 2\pi$

Note that there is a one-to-one correspondence between the spherical polar positioning angle, θ, and the first orientation angle, α, as well as a one-to-one correspondence between the spherical azimuthal angle, ϕ, and the second orientation angle, β. These correspondences were, in fact, the reason these orientation angles were chosen. Note, too, that the directions about which the α and β rotations occur are coordinate axes and, as such, can be represented fairly simply mathematically.

However, the γ rotation introduces some complexity. It does not occur about one of the coordinate axes; instead is occurs about the (intermediate) $+z_\beta$-axis. This leads to the question of how exactly can such a rotation be expressed mathematically. The answer shall be derived in the next section.

4 Equation of a General Rotation

Knowing the equations governing a general rotation is required in order to use our choice of orientation angles. As such, in this section, we shall derive the general equations governing the rotation of an *arbitrary* vector about an *arbitrary* (nonzero) direction vector by an *arbitrary* angle. Remember that, according to our conventions, all rotations are to be active ones and have a right-handed-corresponding-to-positive-angle sense[1].

Consider a vector, $\vec{v}$, existing in a right-handed (Cartesian) coordinate system, with its tail located at the coordinate system's origin. In addition, assume that we have a (unit length) direction vector, $\hat{d}$, whose tail is also located at the origin of the coordinate system. We would like to derive the set of equations that rotate $\vec{v}$ about $\hat{d}$ by an angle ω. We shall denote the resulting rotated vector by $\vec{v}\,'$.

First note that $\vec{v}$ can be decomposed into a vector, $\vec{v}_d$, parallel to $\hat{d}$, and a vector, $\vec{v}_n$, normal to $\hat{d}$. Since $\hat{d}$ is assumed to be a unit vector, we can express these decomposed vectors as:

$$\vec{v}_d = \hat{d}\left(\hat{d} \bullet \vec{v}\right) \tag{1}$$

$$\vec{v}_n = \vec{v} - \hat{d}\left(\hat{d} \bullet \vec{v}\right) \tag{2}$$

Since $\vec{v}_d$ is parallel to $\hat{d}$, it will not change when $\vec{v}$ is rotated. As for $\vec{v}_n$, note that its magnitude is given simply by:

$$v_n = \left|\hat{d} \times \vec{v}\right| \tag{3}$$

As such, the normalized portion of the component of $\vec{v}$ normal to $\hat{d}$ has the form of:

$$\hat{v}_n = \frac{\vec{v} - \hat{d}\left(\hat{d} \bullet \vec{v}\right)}{\left|\hat{d} \times \vec{v}\right|} \tag{4}$$

Note, also, that if we further define a "transverse" (unit) vector, $\hat{v}_t$, according to:

$$\hat{v}_t = \frac{\hat{d} \times \vec{v}}{\left|\hat{d} \times \vec{v}\right|} \tag{5}$$

then it follows that $\hat{v}_n$, $\hat{v}_t$, and $\hat{d}$ form a right-handed orthonormal coordinate system with a direct correspondence to a Cartesian coordinate system:

$$\hat{x} \to \hat{v}_n \quad , \quad \hat{y} \to \hat{v}_t \quad , \quad \hat{z} \to \hat{d} \tag{6}$$

Using this coordinate system, our problem is seen to have become a problem of calculating the equations for rotating a vector, $\vec{v}_n$, about the $\hat{d}$ axis by an angle ω. From any book dealing with rotations, we thus get:

$$v_n' = v_n \cos\omega - v_t \sin\omega \tag{7}$$

$$v_t' = v_n \sin\omega + v_t \cos\omega \tag{8}$$

[1]In this paper, a vector shall be represented with an arrow over it, as in $\vec{a}$. A normalized vector will instead wear a hat, as in $\hat{a}$. The magnitude of a vector shall be denoted as the vector without an arrow or hat, as in $|\vec{a}| = a$.

In our case, initially we have $v_n = \left|\hat{d} \times \vec{v}\right|$ and $v_t = 0$. Thus:

$$v_n' = \left|\hat{d} \times \vec{v}\right| \cos\omega \tag{9}$$

$$v_t' = \left|\hat{d} \times \vec{v}\right| \sin\omega \tag{10}$$

Adding together 1) plus the product of 4) and 9) plus the product of 5) and 10) thus gives the final rotated vector, $\vec{v}\,'$:

$$\vec{v}\,' = \hat{d}\left(\hat{d} \bullet \vec{v}\right) + \left(\vec{v} - \hat{d}\left(\hat{d} \bullet \vec{v}\right)\right)\cos\omega + \left(\hat{d} \times \vec{v}\right)\sin\omega \tag{11}$$

This is the general (vector) equation for the rotation of a vector, $\vec{v}$, about a (unit length) directional vector, $\hat{d}$, by an angle ω in an (active) right-handed sense into a new vector, $\vec{v}\,'$.

In explicit component form, where:

$$\vec{v} = v_x\hat{x} + v_y\hat{y} + v_z\hat{z} \tag{12}$$

$$\hat{d} = d_x\hat{x} + d_y\hat{y} + d_z\hat{z} \tag{13}$$

the general rotation equations take on the following form:

$$v_x' = v_x\cos\omega + d_x\left(\hat{d} \bullet \vec{v}\right)(1 - \cos\omega) + (d_y v_z - d_z v_y)\sin\omega \tag{14}$$

$$v_y' = v_y\cos\omega + d_y\left(\hat{d} \bullet \vec{v}\right)(1 - \cos\omega) + (d_z v_x - d_x v_z)\sin\omega \tag{15}$$

$$v_z' = v_z\cos\omega + d_z\left(\hat{d} \bullet \vec{v}\right)(1 - \cos\omega) + (d_x v_y - d_y v_x)\sin\omega \tag{16}$$

If matrix form, these equations become:

$$\begin{aligned} v_x' &= \left(d_x^2(1 - \cos\omega) + \cos\omega\right) v_x \\ &\quad + \left(d_x d_y(1 - \cos\omega) - d_z\sin\omega\right) v_y \\ &\quad + \left(d_x d_z(1 - \cos\omega) + d_y\sin\omega\right) v_z \end{aligned} \tag{17}$$

$$\begin{aligned} v_y' &= \left(d_x d_y(1 - \cos\omega) + d_z\sin\omega\right) v_x \\ &\quad + \left(d_y^2(1 - \cos\omega) + \cos\omega\right) v_y \\ &\quad + \left(d_y d_z(1 - \cos\omega) - d_x\sin\omega\right) v_z \end{aligned} \tag{18}$$

$$\begin{aligned} v_z' &= \left(d_x d_z(1 - \cos\omega) - d_y\sin\omega\right) v_x \\ &\quad + \left(d_y d_z(1 - \cos\omega) + d_x\sin\omega\right) v_y \\ &\quad + \left(d_z^2(1 - \cos\omega) + \cos\omega\right) v_z \end{aligned} \tag{19}$$

As an example, consider a rotation about the coordinate system's y axis by an angle α. In this case, $\hat{d} = \hat{y}$. Thus, from the general rotation equations, we get:

$$v'_x = v_x \cos\alpha + v_z \sin\alpha \tag{20}$$

$$v'_y = v_y \tag{21}$$

$$v'_z = -v_x \sin\alpha + v_z \cos\alpha \tag{22}$$

which is just what we would expect. Similarly, for a rotation about the coordinate system's z axis by an angle β, we have $\hat{d} = \hat{z}$ and thus:

$$v'_x = v_x \cos\beta - v_y \sin\beta \tag{23}$$

$$v'_y = v_x \sin\beta + v_y \cos\beta \tag{24}$$

$$v'_z = v_z \tag{25}$$

which is also what we would expect.

It will be noted that the above two derived sets of equations correspond to those of the respective spherical polar and azimuthal angles. However, the equations for the more-complicated γ angle rotation (about the $+z_\beta$-axis) still need to be defined. Note that, after applying subsequently the α and β rotations, the $+z_\beta$-axis has the following form:

$$\hat{z}_\beta = \hat{x} \sin\alpha \cos\beta + \hat{y} \sin\alpha \sin\beta + \hat{z} \cos\alpha \tag{26}$$

Substituting $\hat{z}_\beta$ for $\hat{d}$ in the above equations thus gives the answer we are after:

$$\begin{aligned} v'_x \;=\; & \left(\cos\gamma + \sin^2\alpha \cos^2\beta \,(1 - \cos\gamma)\right) v_x \\ & + \left(\sin^2\alpha \sin\beta \cos\beta \,(1 - \cos\gamma) - \cos\alpha \sin\gamma\right) v_y \\ & + \left(\sin\alpha \cos\alpha \cos\beta \,(1 - \cos\gamma) + \sin\alpha \sin\beta \sin\gamma\right) v_z \end{aligned} \tag{27}$$

$$\begin{aligned} v'_y \;=\; & \left(\sin^2\alpha \sin\beta \cos\beta \,(1 - \cos\gamma) + \cos\alpha \sin\gamma\right) v_x \\ & + \left(\cos\gamma + \sin^2\alpha \sin^2\beta \,(1 - \cos\gamma)\right) v_y \\ & + \left(\sin\alpha \cos\alpha \sin\beta \,(1 - \cos\gamma) - \sin\alpha \cos\beta \sin\gamma\right) v_z \end{aligned} \tag{28}$$

$$\begin{aligned} v'_z \;=\; & \left(\sin\alpha \cos\alpha \cos\beta \,(1 - \cos\gamma) - \sin\alpha \sin\beta \sin\gamma\right) v_x \\ & + \left(\sin\alpha \cos\alpha \sin\beta \,(1 - \cos\gamma) + \sin\alpha \cos\beta \sin\gamma\right) v_y \\ & + \left(\cos\gamma + \cos^2\alpha \,(1 - \cos\gamma)\right) v_z \end{aligned} \tag{29}$$

5 Positioning and Orienting The Detectors in the BLAST Setup

Having defined variables appropriate for positioning and orienting a daughter set of axes within a mother set of axes, we shall now consider a generic detector (the daughter system) within the BLAST setup (the mother system).

First off, before any positioning or orienting can take place, we must define four things:

1. the center of the BLAST system
2. the axes of the BLAST system
3. the center of the detector system
4. the axes of the detector system

The center of the BLAST system we shall always take to be the center of the target cell (the target cell is a cylinder whose inside gas cavity is $40cm$ in length).

The BLAST $+z$-axis (written as $+z_{BLAST}$) is defined to point in the direction of beam electron flow; this direction is also the direction along which the target cell runs. The BLAST $+y$-axis (written as $+y_{BLAST}$) points vertically upward (i.e. perpendicular to and away from the floor of the BLAST setup). The BLAST $+x$-axis (written as $+x_{BLAST}$) points in the remaining direction so as to form a right-handed coordinate system with the $+y_{BLAST}$ and the $+z_{BLAST}$ axes; the $+x_{BLAST}$ axis points, in general, away from the counting bay and towards the 10-ton door.

Having defined all relevant aspects of the BLAST system, we now move onto the (generic) detector system. At first glance, defining the center of a detector system may seem trivial. However, some complications immediately arise. First off, all six types of detectors in BLAST (i.e. Cerenkov counters, lead glass calorimeters, neutron counters, recoil detectors, scintillators, and wire chambers) have radically different shapes, so, right away, we know that one encompassing definition for a (generic) detector's center will not be possible. Additionally, the detectors have "extensions" to them that, while technically not active regions of a detector, should still be included in their definition (e.g. the photomultiplier tubes on either side of a scintillator, the aluminum frame surrounding the wires in a wire chamber, and the encasing surrounding the Cerenkov counters). Such an inclusion simplifies matters in that it shall only be required to position and orient the detectors center and axes as opposed to the detector *along with* the extensions; with the previously-mentioned inclusions in the detector definition, the extensions are automatically positioned and oriented when the detector's center and axes are.

I have decided on a definition of the center for each of the detector types. Those definitions shall be included within the BLAST geometry file, "blast.geom"; as such, I shall not reproduce them here[2].

In addition to each detector's center, its axes also need to be defined. Once again, these definitions are provided within in the "blast.geom" file; similar to the detectors' centers, each detector type's axes' definitions are different from one another.

With such detector aspects defined, we can now describe the process for converting a vector, $\vec{v}$, initially expressed in a detector's frame into a vector

[2] The exact definitions of each detector type's center and axes are not relevant to this paper. What *is* relevant is that each detector has an existing definition for its center and axes and that these definitions are not the same for the various detector types.

expressed in the BLAST frame; such a process shall be particularly useful for wire chamber hit positioning.

The process is as below:

1. Rotate the vector, $\vec{v}$, through an angle of $-\gamma$ about the detector's $+z$-axis. Call the resulting vector $\vec{v}_\gamma$.

2. Rotate the vector, $\vec{v}_\gamma$, through an angle of $-\beta$ about the BLAST $+z$-axis. Call the resulting vector $\vec{v}_\beta$.

3. Rotate the vector, $\vec{v}_\beta$, through an angle of $-\alpha$ about the BLAST $+y$-axis. Call the resulting vector $\vec{v}_\alpha$.

4. Add to $\vec{v}_\alpha$ the vector corresponding to the detector's center, $\vec{v}_{DET}$, as expressed in the BLAST system (this vector is spherically expressed within the variables r, θ, and ϕ):

$$\vec{v}_{BLAST} = \vec{v}_\alpha + \vec{v}_{DET} \tag{30}$$

   $\vec{v}_{BLAST}$ is the vector $\vec{v}$ expressed in the BLAST system.

6 How the BLAST Software Incorporates This Geometry

Due to radical new detector geometry definitions, the old BLAST geometry software files (i.e. the data file "blast.geom"; the GEANT files "xx_geom.f", "xx_sets.f", "xx_hits.f", "xx_digi.f", where the "xx" stands for "cc", "lg", "nc", "rc", "sc", or "wc"; and the reconstruction geometry read-in files "TBLGeometry.h" and "TBLGeometry.cc" as well as any other reconstruction files using the data within these) had to be discarded.

The Monte Carlo replacement files will be worked on and finished by myself, as I am probably the one most familiar with GEANT's use of geometry.

Additionally, I shall work on and finish the new TBLGeometry.* files along with the new "blast.geom", the writing of which goes hand-in-hand with the writing of the Monte Carlo TBLGeometry.* files.

However, at that point, nearly all of the other old reconstruction classes will need to be updated for the new geometry. This job is clearly too big for one person. Others should decide on and divy out the remaining classes to be updated.

Actually, most of the new TBLGeometry.* files as well as the new "blast.geom" file have already been written and checked into CVS. A description of the new geometry classes (in the reconstruction) is below:

1. The six (general) position and orientation variable names (i.e. r, θ, ϕ, α, β, and γ, which are common to all the detectors) as well as their "Get"ter and "Set"ter functions are defined within the class TBLGeomGeneral.

2. The variables specific to each of the six detector types (and thus different from detector to detector) are defined within the corresponding 'TBLGeomXX class, where "XX" stands for "CC", "LG", "NC", "RD", "SC", or "WC". Each of these six classes inherits from TBLGeomGeneral. In this way both sets of variable names (i.e. those common to all the detectors and those specific to each detector) are logically divided. Additionally, *all* implementation specific to any of the detectors shall be included within the corresponding TBLGeomXX class file. Once again, this forms a nice division of each detector's implementation in case modifications need to occur later.

3. The TBLGeometry class does NOT inherit from any of the TBLGeomXX classes; instead, it includes object arrays for each of the detector types. It also contains appropriate "Get"ter functions which will enable other classes to access the geometry information contained within this class[3].

4. I have also included a TBLVector class which will allow for vector manipulation. Such manipulation shall be extremely important for positioning, orienting, rotating and translating detector vectors.

7 Conclusions

Much further work still needs to be done, but I believe that a solid geometrical basis has finally been set. All interested parties (i.e. those who will eventually update the old reconstruction classes) should take a look at the new "blast.geom" file. Unless I hear otherwise, the variables and definitions contained within that file shall become the standard on which the detectors' geoemtries are defined. If a detector does not have an explanation within "blast.geom", it means that I do not fully understand the geometry of that detector. If you (the reader) are an "expert" in that detector, please come talk to me about its geometry so that I can implement it.

[3] A quick look at the class TBLGeomGeneral or any of the TBLGeomXX files will show that all member variables are declared private or protected. A strongly regimented process (via good code) of obtaining geometry variables only via the "Get"ter functions will allow for easy modications later on...