
Subject: Conversion of track parameters
Posted by [StefanoSpataro](#) on Mon, 13 Jul 2009 09:54:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,
let's assume that I have one track represented by a parabola (FairTrackParP), and I want to convert it into a helix (FairTrackParH). What should I do exactly?

I have seen there is a class FairGeaneUtil doing conversions, but it is not using FairTrackPar objects, and for me SC SD Pt MARS are just understandable acronyms.

Is there some function doing the conversion in a simple way?

Subject: Re: Conversion of track parameters
Posted by [Anonymous Poster](#) on Mon, 13 Jul 2009 10:20:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

dont know if it will help a lot, but it's a start:

SC = FairTrackParH
SD = FairTrackParP
MARS = master reference system \vec{x} & \vec{p}

The names parabola and helix are questionable and you can ignore them for any practical purposes. What the real difference between SD and SC is, that SD comes from a fixed geometrical plane (the D stands for detector), whereas SC is defined in a plane perpendicular to the track.

Cheers, Christian

Subject: Re: Conversion of track parameters
Posted by [StefanoSpataro](#) on Mon, 13 Jul 2009 11:33:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks,
even if those functions do not solve my problem.
I have decided to use the following constructor, which should do the job:

```
FairTrackParP *parabula....  
FairTrackParH *helix = new FairTrackParH(TVector3(parabula->GetX(), parabula->GetY(),  
parabula->GetZ()),  
TVector3(parabula->GetPx(), parabula->GetPy(), parabula->GetPz()),  
TVector3(parabula->GetDX(), parabula->GetDY(), parabula->GetDZ()),  
TVector3(parabula->GetDPx(), parabula->GetDPy(), parabula->GetDPz()),  
parabula->GetQ());
```

which is not so... stylish, but it should work. Maybe one should implement a constructor from a generic FairTrackPar, such as:

```
FairTrackParP::FairTrackParP(const FairTrackPar* xxx);  
FairTrackParH::FairTrackParH(const FairTrackPar* xxx);
```

to have something easier to write.

Subject: Re: Conversion of track parameters
Posted by [Anonymous Poster](#) on Mon, 13 Jul 2009 12:06:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

well you lose all correlations that way... So it is not just an elegance issue...

Christian

Subject: Re: Conversion of track parameters
Posted by [Stefano Spataro](#) on Mon, 13 Jul 2009 12:33:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ok,
I lose error correlation, but for me the function:

```
FairGeaneutile::FromSCToSD(Double_t PC[3], Double_t RC[15], Double_t H[3], Double_t CH,  
    Double_t DJ[3], Double_t DK[3],  
    Int_t &IERR, Double_t &SPU,  
    Double_t* PD, Double_t* RD);
```

is almost impossible to understand and to use.

We need a function which moves easily from FairTrackParP to FairtrackParH and vice versa. I supposed it was already implemented, but it seems not yet.

Could this kind of function be implemented? I think it should be not so difficult, at least for somebody who knows what those variables mean and how to use them.

Subject: Re: Conversion of track parameters
Posted by [Alberto Rotondi](#) on Mon, 13 Jul 2009 14:01:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Stefano,

I wrote the routines that change references. The systems are defined in:
<http://www.pv.infn.it/~rotondi/tracking.pdf>
and explained in the routine comments.

If you want to avoid the direct use of these, I will think about with Lia and we will try to solve.

As a last remarks, remember that the helix and parabola labels are meaningless: the tracking is always made by using helix pieces!
The SC system is the transverse system, with momentum perp to the reference plane (is called helix); the SD system has the reference plane coincident with that of the detector (it is called for misterious reasons parabola)

best regards
Alberto

Subject: Re: Conversion of track parameters
Posted by [Stefano Spataro](#) on Tue, 14 Jul 2009 07:55:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks,
the mathematics is clear, but I think a direct implementation inside the classes could avoid the developer doing wrong assignments (such as, in the previous case, diagonalize the covariance matrix).

About the labels, I am just trying to do track extrapolation based on helix assumption, without geane (faster). Therefore I need the helix parameters, such as lambda, phi, etc etc. The track parameters are stored inside the PndTrack as FairTrackParP, and the conversion between these two parametrizations is missing.

regards

Subject: Re: Conversion of track parameters
Posted by [Anonymous Poster](#) on Tue, 14 Jul 2009 08:57:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

so, I guess the real question here is: Who can provide a proper set of constructors for the FairTrackParams? I cant do it. Who can?

Cheers, Christian

Subject: Re: Conversion of track parameters

Posted by [Lia Lavezzi](#) on Tue, 14 Jul 2009 12:03:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi all,

I can create two new constructors to make the conversion from parabola to helix (and reverse), i.e.:

```
FairTrackParP(FairTrackParH *helix)
```

and

```
FairTrackParH(FairTrackParP *parab)
```

If it is ok, I can do it!

Ciao,

Lia.

Subject: Re: Conversion of track parameters

Posted by [Stefano Spataro](#) on Tue, 14 Jul 2009 12:23:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

That would be nice!

Subject: Re: Conversion of track parameters

Posted by [Stefano Spataro](#) on Tue, 14 Jul 2009 13:10:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

By the way,

the conversion is very important, even because inside PndTrack the track parametrization is using FairTrackParP, while in FairgeanePro the propagation to volume is not implemented using FairTrackParP but only with FairTrackParH.

This means that in order to propagate one track to volume one has to convert the param from P to H and I think this is not so nice.

Subject: Re: Conversion of track parameters

Posted by [Lia Lavezzi](#) on Wed, 15 Jul 2009 14:42:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

after testing the changes I uploaded to svn the new versions of the FairTrackPar* classes.

1) In FairTrackParP/H, there are two new constructors:

```
FairTrackParP(FairTrackParH *helix, TVector3 dj, TVector3 dk, Int_t &ierr)
```

and

```
FairTrackParH(FairTrackParP *parab, Int_t &ierr)
```

which allow the user to create a parabola from an helix, by setting also the unit vector which span the plane, (and an helix from a parabola) without directly calling FairGeaneUtil functions.

The transformation is not always possible, the problematic cases are flagged in FairGeaneUtil by the IERR flag, so I kept it also here and I printed an error message.

2) In FairTrackPar, Stefano, I added your two functions to easily get momentum and position in MARS (master reference system) as a 3-vector.

Obviously if you find that something does not work as it should, please tell me!

Ciao,
Lia.
