Track reconstruction at the ILC: the ILD tracking software

Gaede, F. (DESY) *et al*

18 October 2013

The research leading to these results has received funding from the European Commission under the FP7 Research Infrastructures project AIDA, grant agreement no. 262025.

This work is part of AIDA Work Package **2: Common software tools**.

Track reconstruction at the ILC: the ILD tracking software

Frank Gaede[a], Steven Aplin[a], Robin Glattauer[b], Christoph Rosemann[a] and Georgios Voutsinas[a]

[a] Deutsches-Elektronen-Synchrotron, DESY, Notkestr.85, 22607 Hamburg, Germany
[b] Institut für Hochenergiephysik, HEPHY, Nikolsdorfer Gasse 18, 1050 Wien, Austria

E-mail: frank.gaede@desy.de

Abstract. One of the key requirements for Higgs physics at the International Linear Collider ILC is excellent track reconstruction with very good momentum and impact parameter resolution. ILD is one of the two detector concepts at the ILC. Its central tracking system comprises of an outer Si-tracker, a highly granular TPC, an intermediate silicon tracker and a pixel vertex detector, and it is complemented by silicon tracking disks in the forward direction. Large hit densities from beam induced coherent electron-positron pairs at the ILC pose an additional challenge to the pattern recognition algorithms. We present the recently developed new ILD tracking software, the pattern recognition algorithms that are using clustering techniques, Cellular Automatons and Kalman filter based track extrapolation. The performance of the ILD tracking system is evaluated using a detailed simulation including dead material, gaps and imperfections.

1. Introduction

Higgs physics precision measurements are among the highlights of the physics program at the International Linear Collider ILC [1] and pose stringent requirements on the detector performance. The precise reconstruction of the Higgs boson from the recoiling Z boson in the so called Higgs-Strahlung process, where the Z decays into a lepton pair, demands excellent momentum resolution of $\sigma_{1/p_t} = 2 \times 10^{-5} \oplus 1 \times 10^{-3}/(p_t \sin\theta)$. The measurement of Higgs branching ratios calls for an unprecedented level of performance for jet-flavor tagging. This is driven by the impact parameter resolution, which is expected to be $\sigma_{r\phi} = 5\ \mu\mathrm{m} \oplus \frac{10}{p[\mathrm{GeV}]\sin^{3/2}\theta}\ \mu\mathrm{m}$. In this paper we describe the ILD tracking software that has recently been developed in order to study the performance of the ILD detector [2] - one of two detector concepts planned for the ILC. We start with a brief overview on the tracking sub-detectors, followed by a description of the simulation and digitization tools for track reconstruction and focus then on the pattern recognition algorithms to then conclude with the results on the performance and an outlook on future work.

1.1. The ILD detector

The ILD detector concept is optimized for particle flow (PFA) based reconstruction, which aims at reconstructing every single particle produced in the event. It has been shown that with PFA one can reach excellent jet energy resolution of $3 - 4$ % for 100 GeV jets, allowing for the discrimination of W and Z boson di-jets [2]. The electromagnetic and hadronic calorimeters with their high granularity provide the imaging capabilities needed for PFA. They are surrounded by the coil and return yoke that create the 3.5 Tesla solenoidal field for the tracking system.

The ILD tracking system comprises of the following sub-detectors:

- **VTX**: a Si-pixel detector with a barrel geometry, consisting of 3 double layers or optionally 5 single layers for excellent momentum and impact parameter resolution
- **SIT**: the intermediate Si-strip barrel detector with two double layers with shallow stereo angle bridging the gap between VTX and TPC
- **TPC**: the large time projection chamber that provides 220 space points for a central track allowing for dE/dx-based particle ID and excellent pattern recognition
- **SET**: the Si-strip barrel detector surrounding the TPC with one double layer for improved momentum resolution and time tagging
- **FTD**: two Si-pixel disks followed by five Si-strip disks providing low angle tracking coverage

The tracking sub-detectors are shown in Fig. 1 and some key parameters are summarized in Tab. 1

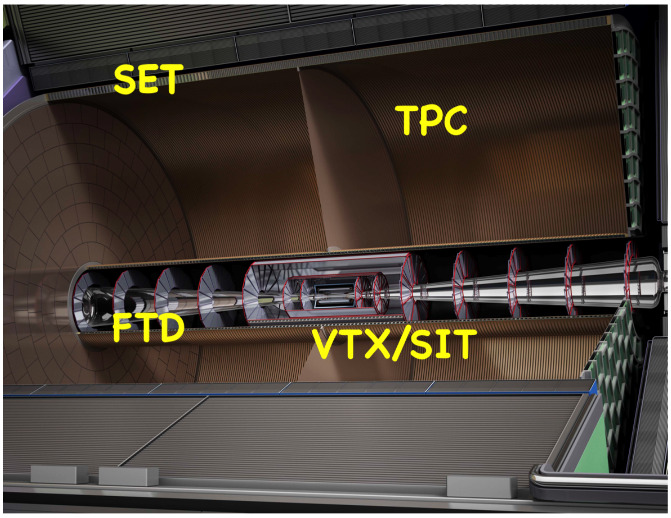

Figure 1. Tracking system of the ILD detector - see Tab. 1 for details on the individual sub-detectors

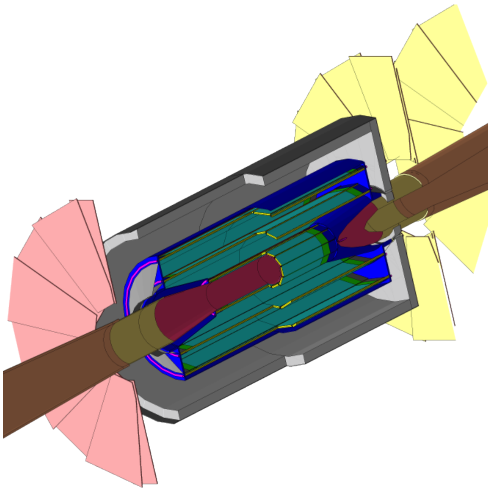

Figure 2. The inner tracking detectors in the detailed simulation model: VTX, SIT and FTD

2. ILD tracking software

The ILD tracking software is developed in the context of the *iLCSoft* software framework with the *Mokka* [3] simulation, the *Marlin* [4] application framework and the *LCIO* [5] event data model and persistency package as the core tools. The *Track* class in LCIO holds several *TrackStates* with fitted track parameters at points of interest, typically at the IP, the first and last hit and the face of the calorimeter. ILD uses a perigee track parametrization with track curvature Ω, impact parameters d_0 and z_0 and direction parameters ϕ_0 and $\tan(\lambda)$ [6].

Table 1. The ILD tracking detectors and their key parameters [2].

detector	geometry	description	single point resolution
VTX	$r_{in} = 16$ mm $r_{out} = 60$ mm $z = 125$ mm	3 double layers Si-pixel sensors	$\sigma_{r\phi,z} = 2.8\mu$m (layer 1) $\sigma_{r\phi,z} = 6.0\mu$m (layer 2) $\sigma_{r\phi,z} = 4.0\mu$m (layers 3-6)
SIT	$r_{in} = 153$ mm $r_{out} = 300$ mm $z = 644$ mm	2 double layers Si-strip sensors	$\sigma_{\alpha_z} = 7.0\mu$m $\alpha_z = \pm 7.0^\circ$ (angle with z-axis)
SET	$r = 1811$ mm $z = 2300$ mm	1 double layer Si-strip sensors	$\sigma_{\alpha_z} = 7.0\mu$m $\alpha_z = \pm 7.0^\circ$ (angle with z-axis)
FTD$_{pixel}$	$z_{min} = 230$ mm $z_{max} = 371$ mm	2 disks Si-pixel sensors	$\sigma_r = 3.0\mu$m $\sigma_{r_\perp} = 3.0\mu$m
FTD$_{strip}$	$z_{min} = 644$ mm $z_{max} = 2249$ mm	5 disks - double Si-strip sensors	$\sigma_{\alpha_r} = 7.0\mu$m $\alpha_r = \pm 5.0^\circ$ (angle with radial direction)
TPC	$r_{in} = 330$ mm $r_{out} = 1808$ mm $z = 2350$ mm	MPGD readout > 220 layers 1 x 6 mm^2 pads	$\sigma^2_{r\phi} = (50^2 + 900^2 \sin^2\phi +$ $((25^2/22) \times (4T/B)^2 \sin\theta)\,(z/\mathrm{cm}))\,\mu\mathrm{m}^2$ $\sigma^2_z = (400^2 + 80^2 \times (z/\mathrm{cm}))\,\mu\mathrm{m}^2$ where ϕ and θ are the azimuthal and polar angle of the track direction

2.1. Simulation and Digitization

In order to realistically study the tracking performance of ILD, it is crucial to have a simulation model with a realistic geometry description and material budget. In the modular GEANT4 [7] program *Mokka* every sub-detector is described in a so called *geometry driver class.* These classes are maintained by experts from the R&D groups and kept in close consistency with the engineering models. Fig. 2 shows a snapshot of the inner tracking region of the simulation model. The VTX detector is modeled with detailed support material holding individual sensors and electronics surrounded by a cryostat. Both the SIT and SET consist of individual sensors on a representative amount of support material. The FTD disks are built from Si-petals on a space frame support structure. The estimated material budget for power and readout cables has been averaged into Al-cylinders and cones in the inner tracking system. The TPC is modeled with the correct gas mixture, field cage material budget and a conservative estimate for the end plate support structure, electronics and cooling pipes. The overall material budget in the ILD simulation model is shown in Fig. 3.

In Mokka the exact positions where the simulated trajectory of the particles cross the sensitive detector surfaces are recorded as *SimTrackerHits* and written to disk. These surfaces may be virtual as in the case of the TPC where they consist of cylinders in the gas volume, defined by the drift field lines meeting the pad row centers. The subsequent digitization of the hits uses the parametrization of the single point resolutions shown in Tab. 1.These have been established by the corresponding R&D groups from test beam measurements with prototypes. In the case of the Si-strip detectors SIT/SET and FTD, one dimensional *TrackerHits* are created at the digitization stage. For the purpose of pattern recognition these 1D hits are combined into 3D space points, thus correctly accounting for ghost hits. The space points are created in the virtual

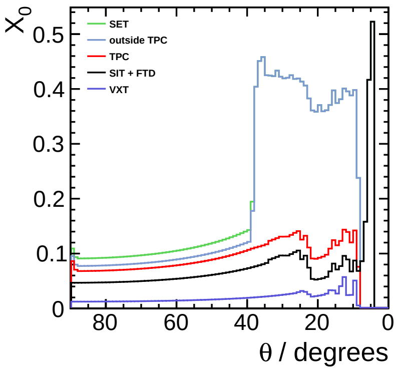

Figure 3. Material distribution in the tracking system in the ILD simulation model as a function of polar angle expressed in integrated radiation lengths X_0.

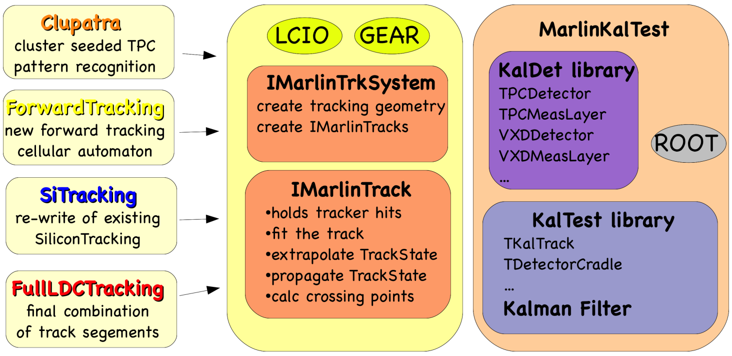

Figure 4. Schematic view of the ILD tracking software design. The Marlin processors *Clupatra, SiliconTracking, ForwardTracking* and *FullLDCTracking* (see sec. 3), are programmed against the *MarlinTrk* interface, which decouples them from the underlying fitting package *KalTest/KalDet*.

middle-plane between the double layers at the crossing of the projection of the individual 1D-straight line measurements - as seen from the interaction point. For the final track fits again the 1D hits are used in order to avoid possible biases due to the curvature of the track.

2.2. Tracking software design

The track reconstruction software has recently been rewritten in C++ in order to replace the older software based on FORTRAN code from LEP experiments which had become increasingly hard to maintain and develop further. A modular and flexible design of the software was a key requirement, with the eventual goal to have as much as possible detector and framework independent tracking tools that can be re-used by other HEP groups. Modularity is achieved naturally in the Marlin framework, where tasks are organized in so called *Processors* that communicate only through the transient LCIO event. Flexibility is provided through the introduction of an abstract interface *MarlinTrk*, used for track extrapolation, propagation and fitting. This interface separates the pattern recognition code from the details of the actual fitting program. As a default implementation for *MarlinTrk* a standard Kalman-Filter tool for perigee parameters *KalTest/KalDet* [8] has been chosen. This general design, as used for the current implementation of the ILD tracking software, is shown in Fig. 4.

3. Pattern recognition algorithms

Finding and reconstructing charged particle tracks in ILD is split up into standalone track finding in the TPC, in the VTX and SIT and in the FTD, followed by a final process of merging compatible track segments. The corresponding software modules are called *Clupatra, SiliconTracking, ForwardTracking* and *FullLDCTracking* respectively. The underlying algorithms are described in the following.

3.1. SiliconTracking

The *SiliconTracking* algorithm is used to find tracks in the two inner barrel detectors VTX and SIT. It starts with a *brute force* seed-triplet search in fixed solid angle sectors in given sets of seed-layer-triplets. A helix χ^2-fit is applied to the seed triplets, followed by a road search in layers not used in the seeding. Leftover hits are assigned to the seed tracks, ordered with ascending χ^2/ndf. A refit with the Kalman filter in *MarlinTrk* is applied to the final Si-tracks.

3.2. Clupatra

In *Clupatra*, track finding in the TPC is performed with an outside-in seed finding, followed by a road search based on the Kalman filter and track extrapolation. The track seeds are created in a fixed number (e.g. 15) of pad rows with a simple nearest neighbor clustering algorithm. If the seed clusters found are sufficiently isolated they are used to initialize a Kalman track that is extended inwards by adding the most compatible hit, based on a χ^2-criterion. The seed search is then repeated in consecutive ranges of pad rows going inwards, in order to find forward tracks and those with lower p_t. After this first step, only close by neighboring tracks that are not separated along the full range of the TPC will not have been found. These are identified as clusters with a number of hits per pad row that is close to an integer number i. It is straight forward to split these clusters into i tracks based on topology. Track segments from curling particles are merged, using a coarse circle matching criterion that allows for energy loss. In a final step, split tracks that might occur in dense jets due to joining of close by *TrackerHits*, are merged using the Kalman filter. The algorithm is quite robust, also against removal of hits in a clean up process for removal of background from coherent e^+e^--pairs. This background gives rise to micro curlers that spread out along the z-axis with an $r\phi$-extent of one or two pads and can thus be rather easily removed based on topology [9].

3.3. ForwardTracking

The algorithm for standalone forward track reconstruction in the FTD is based on Cellular Automatons and Hopfield networks [10]. The Cellular Automaton, originally developed for modeling biological systems, is used for pattern recognition by identifying short track segments in consecutive layers with cells and applying a quality criterion, consistent with charged particle tracks, as a cell state. In an iterative procedure cells (track segments) are compared to neighbor cells (those that share one hit and have the same state). If certain consistency criteria are fulfilled the state value of the the cell is increased. After some iterations no changes will occur and one is left with a, possibly large, set of track candidates. As some of the candidates will typically share common hits, an arbitration procedure for selecting the best consistent set of tracks is needed. This is done with a Hopfield Neural Network, where every Kalman-fitted track candidate is assigned to a neuron. Every neuron/track is connected to every other track with a weight and has an activation function based on track quality and the sum of the connected weights. A dynamic update procedure decreases weights between inconsistent tracks and increases those of good consistent tracks. When only very small changes occur, a set of tracks with large activation functions is taken to be the final set of tracks.

3.4. FullLDCTracking

The final combination of the tracks that were found in the standalone algorithms described above is done in a processor called *FullLDCTracking*. In this step the tracks are merged based on consistency of their track states, transformed to position and momentum at the IP. Optionally leftover hits are assigned to the resulting combined tracks, in particular those in the SET detector. After a final refit with the Kalman filter the tracks are written to disk, preserving the pointers to the original tracks segments.

4. Performance of ILD tracking

The track reconstruction efficiency of the algorithms described above is evaluated in simulated high multiplicity $t\bar{t} \rightarrow$ events in the presence of beam background, at $\sqrt{s} = 500\,\mathrm{GeV}$ and $1\,\mathrm{TeV}$

respectively. The efficiency is shown as a function of momentum and polar angle in Figs. 5 and 6.

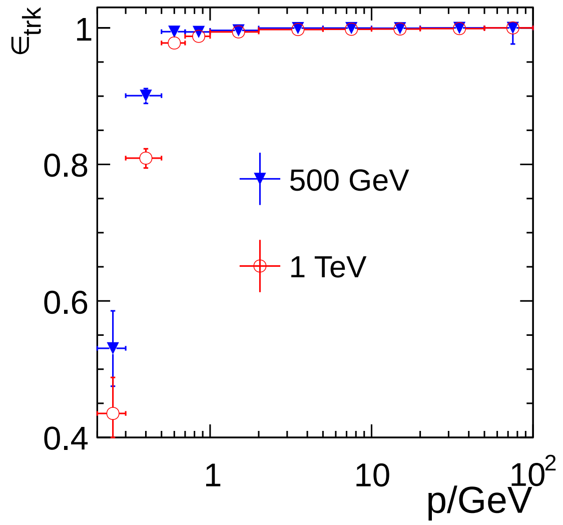

Figure 5. Tracking Efficiency for $t\bar{t} \to 6$ jets at 500 GeV and 1 TeV versus momentum in the presence of beam background.

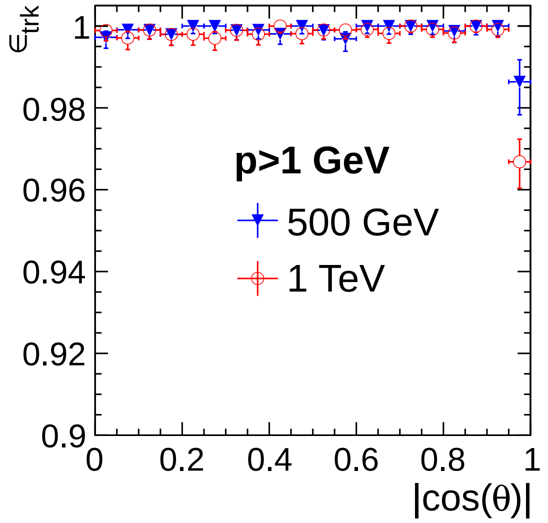

Figure 6. Tracking Efficiency for $t\bar{t} \to 6$ jets at 500 GeV and 1 TeV versus $|cos(\theta)|$ for particles with $p > 1$GeV in the presence of beam background

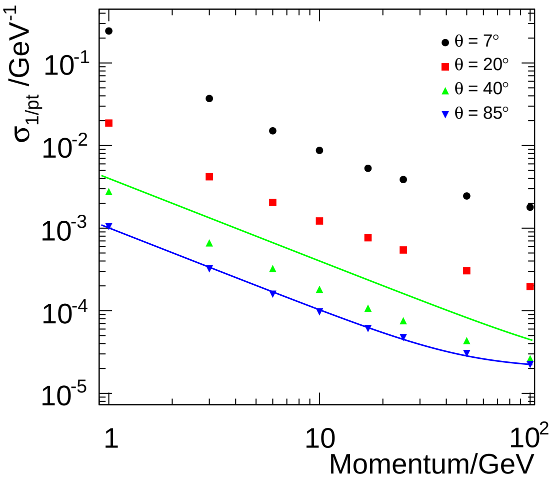

Figure 7. Transverse momentum resolution for single muon events versus p. The lines show the performance goal mentioned in the text.

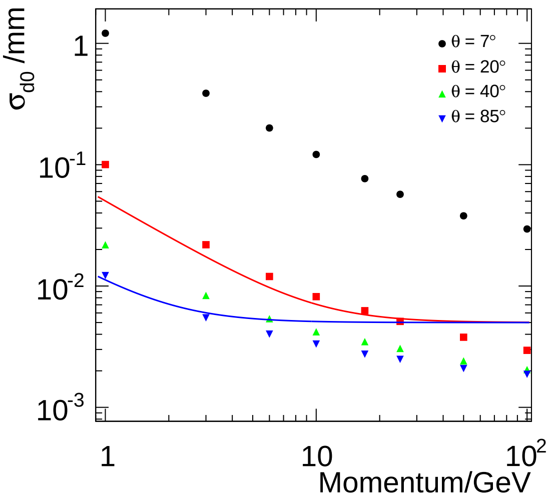

Figure 8. Impact parameter resolution for single muon events versus p. The lines show the performance goal mentioned in the text.

The efficiencies are computed for Monte Carlo particles that stem from a region of 10 cm around the IP with $p_t > 100$ MeV and $|\cos(\theta)| < 0.99$, excluding decays in flight and requiring at least 90 % purity. Hits from coherent pair production are overlaid for the corresponding number of bunch crossings that result from the foreseen readout times of the detectors. For the

VTX detector these readout times range from 10 to 100 μs and result in O(10^6) tracker hits.

For the combined tracking system, the track reconstruction efficiency is on average 99.7 % for tracks with momenta greater than 1 GeV across the entire polar angle range, and it is larger than 99.8 % for $|\cos(\theta)| < 0.95$. The design goal for the transverse momentum resolution of ILD, based on the Higgs recoil mass measurement, is $\sigma_{1/p_t} = 2 \times 10^{-5} \oplus 1 \times 10^{-3}/(p_t \sin\theta)$ and that for the impact parameter resolution is $\sigma_{r\phi} = 5\ \mu\mathrm{m} \oplus \frac{10}{p[\mathrm{GeV}]\sin^{3/2}\theta}\ \mu\mathrm{m}$. Both resolutions are evaluated with single muon events at a number of fixed polar angles θ. They are shown in Figs. 7 and 8 together with the given parametrization, demonstrating that the design goals for ILD tracking has been achieved.

5. Conclusion

The new tracking software for ILD described above, shows a very high tracking efficiency of 99.7 % for tracks with momenta greater than 1 GeV across the entire polar angle range. The excellent momentum and impact parameter resolution required of the ILD tracking system is shown to be reachable, based on a detailed simulation including dead material, gaps and imperfections. The observed smaller efficiency for lower momenta tracks, which do not leave a significant number of hits in the TPC, can be improved by improving the standalone tracking in the central Si-detectors. Improving the pattern recognition efficiency in this region by adapting the Cellular Automaton approach is the focus of ongoing work. In parallel, work is ongoing to further improve the software design and restructure the code in a way that makes the algorithms and tools available to other experimental groups in a generic tracking toolkit. The next step in this direction will be to adapt the DD4hep [11] detector description toolkit as the source of geometry information used in the tracking.

Acknowledgments

Part of this work was supported by the AIDA project, WP2: *Common Software Tools* (FP7 Research Infrastructures, grant no 262025). The authors would like to thank Keisuke Fujii for providing his *KalTest* tool for the iLCSoft framework and his continuous support and very helpful discussions during the development of the tracking software.

References

[1] Chris Adolphsen, Maura Barone, Barry Barish, Karsten Buesser, Philip Burrows, et al. The International Linear Collider Technical Design Report - Volume 3.II: Accelerator Baseline Design. 2013.

[2] Ties Behnke, James E. Brau, Philip N. Burrows, Juan Fuster, Michael Peskin, et al. The International Linear Collider Technical Design Report - Volume 4: Detectors. 2013.

[3] P. Mora de Freitas and H. Videau. Detector simulation with MOKKA / GEANT4: Present and future. pages 623–627, 2002. LC-TOOL-2003-010.

[4] F. Gaede. Marlin and LCCD: Software tools for the ILC. *Nucl.Instrum.Meth.*, A559:177–180, 2006.

[5] Frank Gaede, Ties Behnke, Norman Graf, and Tony Johnson. LCIO: A Persistency framework for linear collider simulation studies. *International Conference on Computing in High Energy and Nuclear Physics (CHEP 2003), La Jolla, CA, 2003, proceedings.*, C0303241:TUKT001, 2003. SLAC-PUB-9992, CHEP-2003-TUKT001.

[6] Thomas Kraemer. Track Parameters in LCIO. 2006. LC-DET-2006-004.

[7] S. Agostinelli et al. GEANT4: A Simulation toolkit. *Nucl.Instrum.Meth.*, A506:250–303, 2003. SLAC-PUB-9350, FERMILAB-PUB-03-339.

[8] Keisuke Fujii. Extended Kalman Filter. http://www-jlc.kek.jp/subg/offl/kaltest.

[9] Toshinori Abe et al. The International Large Detector: Letter of Intent. 2010. FERMILAB-LOI-2010-03, FERMILAB-PUB-09-682-E, DESY-2009-87, KEK-REPORT-2009-6.

[10] Robin Glattauer, Rudolf Fruhwirth, Jakob Lettenbichler, and Winfried Mitaroff. Forward Tracking in the ILD Detector. 2012.

[11] Frank Gaede Markus Frank and Pere Mato. DD4hep: A Detector Description Toolkit for High Energy Physics Experiments. *International Conference on Computing in High Energy and Nuclear Physics (CHEP 2013), Amsterdam, 2013.* these proceedings.