

CONTROLLING ELECTRON BEAMS IN THE International Linear Collider

BY GLEN WHITE, UNIVERSITY OF LONDON

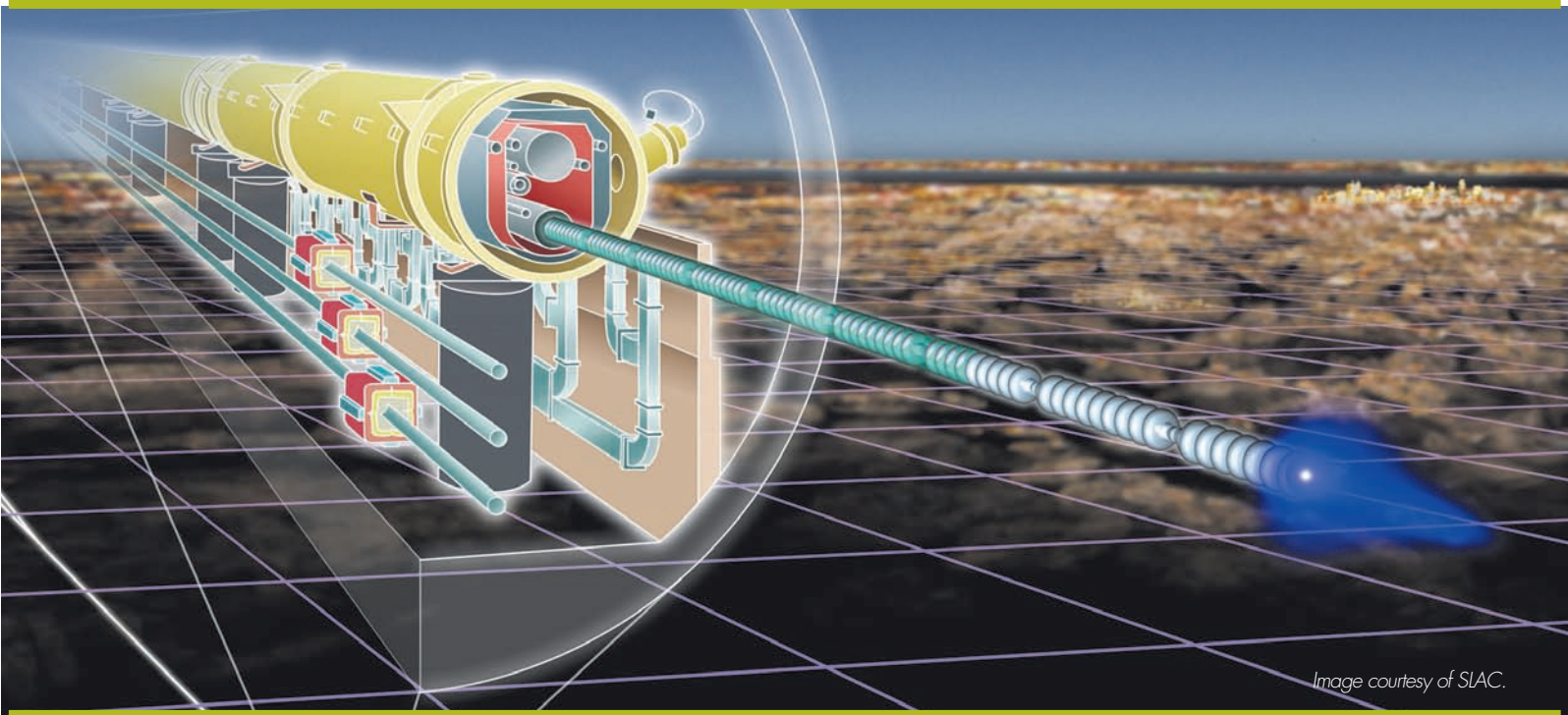


Image courtesy of SLAC.

How was the universe really formed? What is the nature of gravity? What is dark matter? Do extra dimensions exist?

The International Linear Collider (ILC) promises to bring physicists closer to answering these fundamental questions.

This 15-year design effort, driven by the work of scientists and engineers at universities and laboratories around the world, could produce an operational collider as early as 2016.

The ILC is a highly complex machine with 20,000 accelerating structures. It consists of two 20-kilometer-long linear accelerators that accelerate beams of electrons and positrons toward each other, producing collision energies of up to 1,000 giga-electron volts.

At less than five nanometers thick and traveling near the speed of light, the

particle beams in the ILC are so thin that the smallest disturbance can cause them to be misaligned. As a result, the ILC design must account for the natural movement of the Earth—from earthquakes and the tidal pull of the moon to ground motion caused by trains and vehicles.

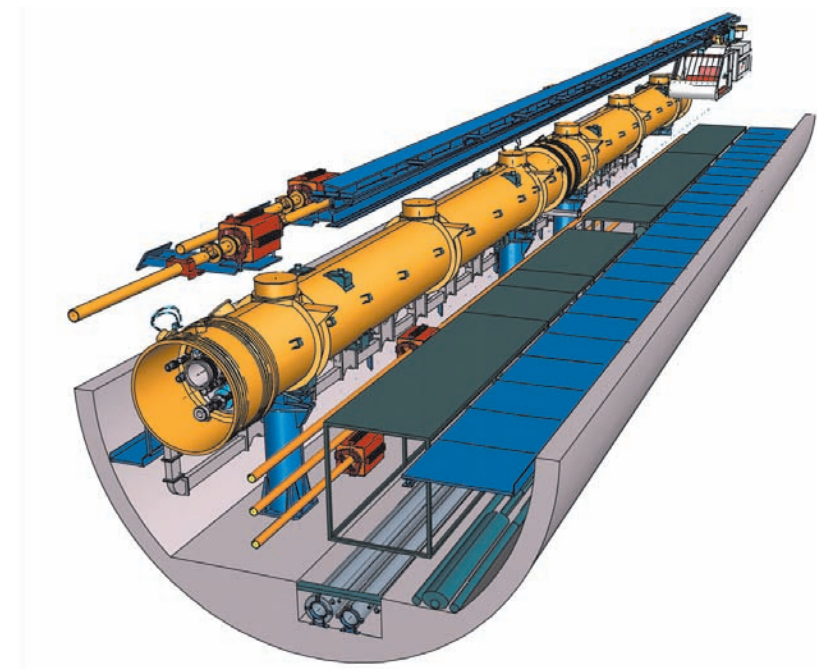
Glen White, accelerator physicist at Queen Mary, University of London, designs and develops the feedback systems that keep the beams in alignment through the collider. In this article, White explains how he uses multiple simulations and high-throughput computing to rapidly test tuning, alignment, and feedback algorithms on realistic models.

Delivering the particle beam to the interaction point while maintaining a small beam size requires very accurate alignment and tuning algorithms in a distributed feedback system. We use MATLAB® and Simulink to design the control system, model the entire ILC, and simulate the way the particle beams travel through the accelerator and interact with tens of thousands of magnetic elements.

When Beams Collide

The particle beams are about a thousand times thinner than a human hair. Delivering such small beams accurately is a daunting technical challenge that places huge demands on the alignment and operational tolerances of all the machine's components.

In the ILC, beams will collide in bunches of 20 billion particles, with trains of 3,000 bunches arriving at the collision point every 300 nanoseconds. The first bunches in the train typically miss because of ground motion. The collision rate depends heavily on the shape of the colliding bunches, and the entire process is highly nonlinear. Our task was to develop the very fast electronics and beam-correction systems needed to align the beams and ensure the highest



The tunnel for the proposed ILC is a straight concrete tube that runs at a depth of 10 – 30 meters below the ground and has an inner diameter of about 5 meters. Image courtesy DESY Hamburg.

possible particle collision rate—a task never needed or encountered before on any accelerator project.

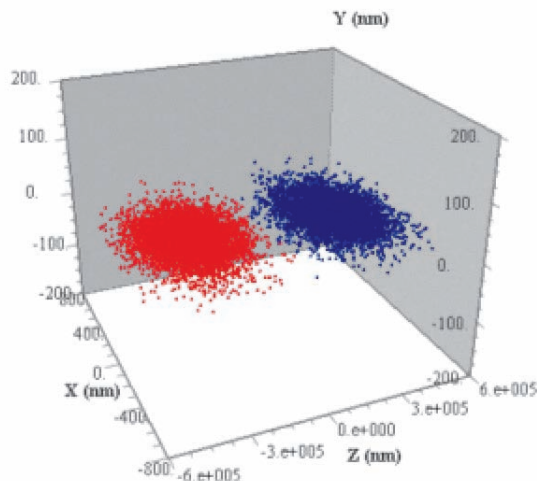
Modeling the Entire ILC System

In the 15 years since the project began, researchers worldwide have created a number of models for different linear collider com-

ponents, including the systems that produce the beams, the sections that accelerate them, and the components that prepare the beams for collision. Some were written in MATLAB, but many were written in other languages, including C and C++.

Before designing the beam-alignment algorithms, we needed to combine these individual models and simulate the entire system. This was the first time that all the models had been brought together in this way to create a complete model—and again, the task was unique to the ILC.

My colleague, Peter Tannenbaum, developed a set of core particle-tracking algorithms in MATLAB that we use to simulate the transport of the beam through the ILC. A single structured cell array contains all the information about the machine, such as the strength of individual magnets. MATLAB routines act on this array to provide control functions that, for example, change the magnet's strength. At a lower level, we use MEX-files to incorporate C routines that perform computationally intensive actions, such as transporting the beam particles through the elements.



Simulation of the beam collision process.

Running Multiple Simulations of the Collision Process

We use Simulink to run simulations that track the beams through all the accelerator systems. A single simulation, called a seed, transports 600 particle bunches, each comprising 80,000 particles representing an actual bunch, through 20 kilometers of beamline components twice. Each simulation incorporates all possible ground motion effects.

We run multiple Monte Carlo simulations to model random accelerator imperfections and various tolerances. To arrive at an optimal design, we must then repeat each set of simulations for different machine configurations. On a modern CPU, simulating a single seed requires two to three days of processing time. We typically evaluate 100 seeds per machine configuration.

Accelerating the Simulation

Tracking thousands of bunches and millions of particles through hundreds of components is an extremely demanding simulation task. To handle the volume we needed an environment that would enable us to simulate many seeds in parallel while keeping track of all the results and corresponding parameter sets.

Using our distributed computing environment we can evaluate all 100 seeds for a particular machine configuration in the same time that it would take to complete one seed on a single CPU.

We used Distributed Computing Toolbox to set up this environment on the high-throughput cluster at Queen Mary. In this 348-processor system, a daemon process picks up new jobs that researchers add to a

database. Distributed Computing Toolbox runs each seed of the new job on a separate processor in the cluster through a Maui scheduler and portable batch system. The results and simulation parameters are automatically stored in the database.

Using this environment we can evaluate all 100 seeds for a particular machine configuration in the same time that it would take to complete one seed on a single CPU.

Testing Algorithms on Real-World Instruments and Devices

I have begun testing our algorithms on the Accelerator Test Facility (ATF) in Tsukuba, Japan. The ATF is a real particle accelerator that determines the trajectory of the beams and measures their positions as they go past. I use a Lyrtech signal processing board, based on a VirtexII Xilinx FPGA chip, to process beam position signals from devices in the ATF. I program the FPGA logic through Xilinx System Generator and a blockset provided by Lyrtech for their processing board. I can then use Simulink to test the feedback model in the ILC and program the same logic into the Lyrtech board for the

testing in a real test accelerator with the actual beam. Communication to the Lyrtech processing board, its ADCs and DACs and its onboard FPGA is done through Real-Time Workshop® via an onboard Texas Instruments DSP chip.

We will be running these tests for the next couple of months, using

Instrument Control Toolbox to control signal generators and continuously acquire beam position and hardware diagnostic information from high-bandwidth Tektronix oscilloscopes and other equipment.



The high-throughput cluster at Queen Mary.

We will then analyze the data in MATLAB, and use the results to further refine our control algorithms.

Looking Forward to an Operational ILC

Later this year, we expect to release a reference design report that will serve as a build blueprint for the ILC. Like my fellow particle physicists around the world, I anticipate the start of an international funding agreement and a site selection process—and to having the first beam in an operational ILC by the second half of the next decade. ◀◀

RESOURCES

- ▶ **International Linear Collider**
www.linearcollider.org
- ▶ **Distributed Computing Toolbox**
www.mathworks.com/res/distribtb

