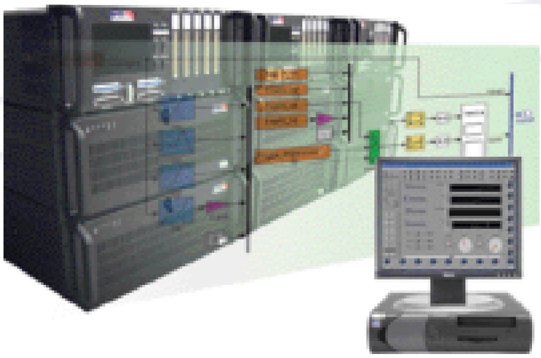
OPAL·RT

RT-LAB
Distributed Real-Time Power

Model-Based Design and Virtual Prototyping
Control Prototyping and Testing
Embedded Control
Data Logging



Product Description
Feature Details
Typical Applications

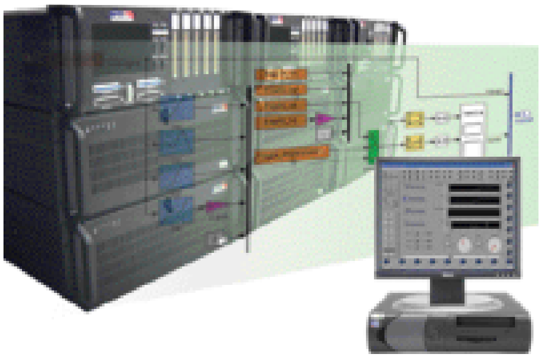
IMAGINATION
TO
REAL-TIME

OPAL-RT

www.opal-rt.com

RT-LAB Distributed Real-Time Power

Model-Based Design and Virtual Prototyping
Control Prototyping and Testing
Embedded Control
Data Logging



RT-LAB™ is the distributed real-time platform that is revolutionizing the design process for engineering systems. Through its openness, it has the flexibility to be applied to the most complex simulation and control problem, whether it is for real-time hardware-in-the-loop applications or for dramatically speeding up model execution. Its scalability provides a low-risk entry point for any real-time engineering application, allowing the developer to add compute-power where and when needed.

Handling Complexity and Simplicity

RT-LAB is an industrial grade, open, scalable real-time platform for engineers who need to use mathematical models of dynamic systems built using MATLAB/Simulink or MATRIXx/SystemBuild for simulation, control testing and related applications.

Since these models can be very complex, it is often not possible to run them in real-time on a single processor. RT-LAB provides tools for running simulations of highly complex models on a network of distributed run-time targets, communicating via ultra low-latency technologies, in order to achieve the required performance.

And, RT-LAB's scalability goes both ways. Its modular design enables the delivery of economical systems by supplying only the modules needed for the application in order to minimize computational requirements and meet end-user price targets. This is essential for high-volume embedded applications.

Key Features

Real-Time Model Development

- Fully integrated with MATLAB/Simulink/Stateflow/SimPowerSystems/Real Time Workshop, and MATRIXx/SystemBuild/AutoCode
- Specialized Blockset to prepare the model for distributed processing, inter-node communication and signal I/O
- "Six Clicks to Real-Time" User interface
- Comprehensive API for developing your own on-line application, including tools for LabVIEW, C++, MATLAB, Visual Basic, TestStand, Python and 3D Virtual Reality Tools
- Supports models from CarSim/TruckSim, GT-Power, AMESim, Dymola, as well as legacy models in FORTRAN 90/95

Run-time Features

- Distributed, synchronized execution across multiple processors, including multiple CPU motherboards
- Shared Memory, FireWire, SignalWire or InfiniBand interprocessor communication
- Integrated signal visualization and control panel
- Dynamic Signal Tracing - select any signal for monitoring during run-time
- On-line Parameter Editor - change parameters on-the-fly (manually or via parameter files)
- Extensive I/O card support - over 80 devices supported
- Choice of RTOS: QNX, RedHawk Linux (or Windows for soft real-time)

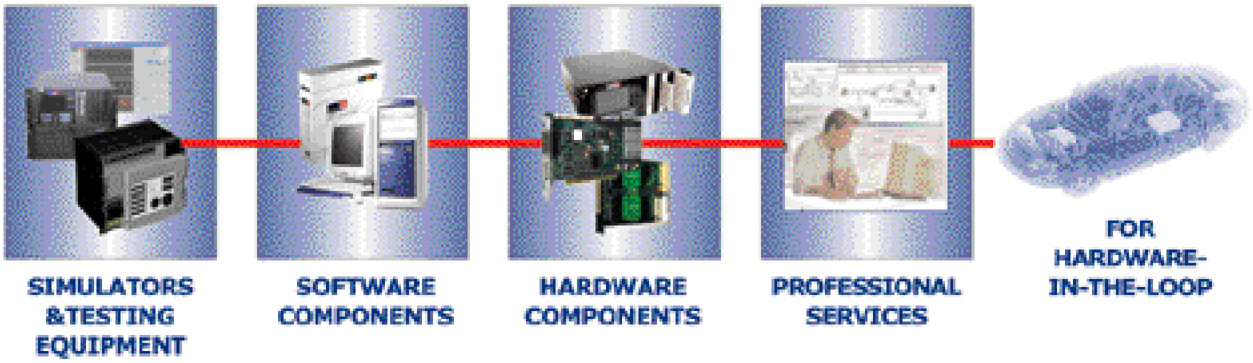
Performance Features

- XHP Mode - high speed mode to give closed-loop cycle times below 10 microseconds
- Optimized Hard-Real-Time Scheduler - high performance, low jitter

Off-the-Shelf Technologies

RT-LAB is the first fully scalable simulation and control package that allows you to separate models for execution in parallel on a network of Pentium-based targets - standard desktop PCs, PC/104s or on SMP (symmetric multi-processor) servers.

Driven by the demands of a mass market, users take advantage of rapid advancements in a wide range of readily available technologies, as well as relatively low costs. RT-LAB uses standard Ethernet and FireWire (IEEE 1394) communications, and an extensive range of ISA, PCI, PXI and PCMCIA analog and digital I/O boards.



Hard Real-Time Performance

Through many years of research, RT-LAB delivers the best hard-real-time performance on any PC-based platform, and makes sure that parallel execution does not subtly alter model behavior, introduce real-time glitches or cause deadlocks. Also with its new XHP mode, RT-LAB can achieve closed-loop cycle times down below 10 microseconds (100 kHz), with jitter in the nanosecond range.

Choice of Real-Time Operating System

RT-LAB is the only real-time simulation framework that offers you a choice of two high-performance Real-Time Operating Systems (RTOS). RT-LAB is available for QNX - the world's best-established RTOS for mission-critical engineering applications - and RedHawk Linux - the premier real-time version of the popular open-source Linux operating system - from Concurrent Computer Corp. RT-LAB also offers Windows as a pseudo-real-time RTOS.

Rapid Productivity

RT-LAB provides many usability features to allow you to develop your real-time solution in minimal time. Its tight integration with Simulink and SystemBuild means you only need to design your model and incorporate our real-time specific blocks and, if necessary, our special fixed-step solvers to prepare it for distribution and execution on the target processor or processors. RT-LAB handles the rest with an easy-to-use interface that takes your model to real-time simulation with a few mouse clicks.

RT-LAB provides support for several specialized high-fidelity modeling tools that allow you to create a system model, incorporate it into Simulink, and run it in real-time. See the separate section on third party modeling tools currently supported.

Finally, to complete your application, RT-LAB's API tools allow you readily connect to on-line visualization tools such as LabVIEW, WorldUp and Altia without programming. It also allows you to connect to your own applications written in MATLAB, Python, Visual Basic and C++.

Reducing Risk

RT-LAB will take you from Simulink or SystemBuild dynamic models to real-time with hardware-in-the-loop, in a very short time, at a low cost. Through its scalability and use of off-the-shelf components, your initial investment will be very low, you will be free to add computation power as your requirements grow. RT-LAB's performance and usability features will ensure you avoid the computational and development problems usually associated with real-time engineering applications, allowing you to deliver the optimal solution, on time and within budget.

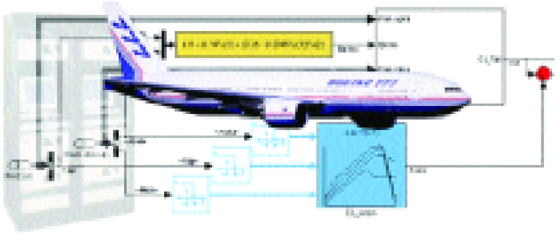
RT-LAB Typical Applications

High-Fidelity Plant Simulation
Control System Prototyping
Embedded Data Acquisition and Control

RT-LAB is used in many different applications, from vehicle ECU development and test, to rocket propulsion controllers. Most RT-LAB applications fall into the following categories:

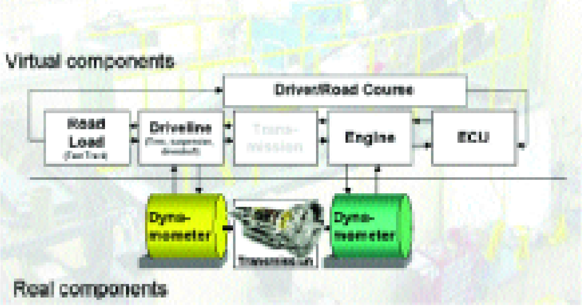
High-Fidelity Plant Simulation

Real-Time Simulation for System Integration



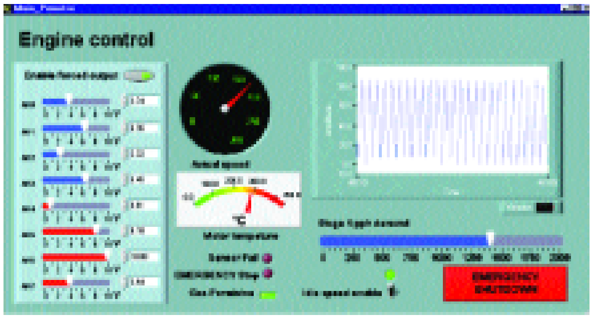
For many projects, the objective is to integrate the components of a complete system. For example, an aircraft flight control system is made up of many components that have to interface with each other. The most effective way to perform system integration is to test the components together in a virtual environment. That allows components to be tricked into thinking they are in their native environment. Realistic inputs like the ones produced by a pilot will propagate throughout the system.

Physical Component Testing with Virtual Systems



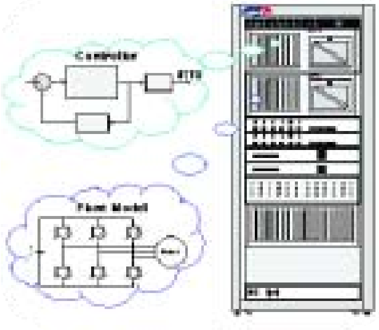
There is a growing trend towards testing new engineering components in a dynamic test cell that simulates the behavior of the system in which the component will be installed. For example, a vehicle transmission can be installed in a virtual vehicle test cell that behaves like a real test vehicle. In this way the transmission manufacturer saves on the cost of a test vehicle, does not need to rely on the stamina of the driver and the tests are entirely repeatable.

Operator Training



Since the simulation of a system has been developed for the design stage, RT-LAB provides a very economical platform for training operators of the system. RT-LAB's API allows the model to be connected to the real operator interface in order to provide a close-to-reality environment in which the operator can learn procedures and deal with faults in the system without causing damage or injury.

Controller Development

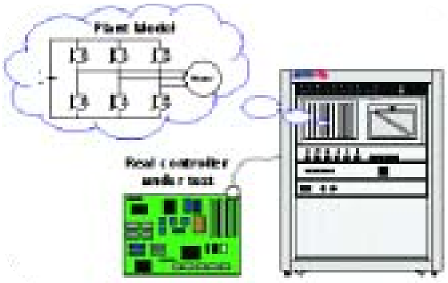


The first stage in any control development project is to better understand the system that needs to be controlled – the plant. This is normally achieved through the use of the many system-modeling tools available, including Simulink and SystemBuild. While a great deal can be accomplished by analyzing the plant model in non-real-time, it is when the plant model runs in real-time that a great deal of insight into the system’s behavior can be determined.

Once a high fidelity model of the plant has been achieved, the controller can be developed and even prototyped and tested with the plant model.

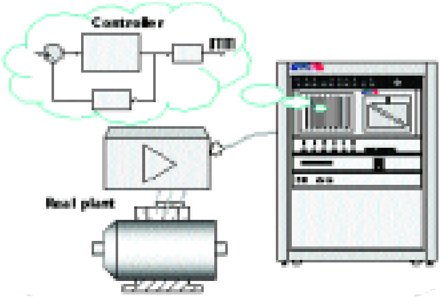
Control System Prototyping

Prototype Controller Connected to Virtual Plant



Once a controller has been developed, it needs to be tested. This could be done with the physical plant, assuming it has been built, but the risk of damaging the plant can be considerable. Connecting the prototype controller to a high-fidelity simulation of the plant has many advantages beyond the safety of the plant. It also means that the controller can go through a series of automated tests unattended without increasing the risk of damage to the plant even further.

Control Testing with Real Plant



As part of the design process, the controller has to be tested with the real plant. RT-LAB can simulate the controller by running the control software and connecting to the plant. In fact RT-LAB provides an excellent environment in which the controller can be switched readily from controlling the model to controlling the plant and back again.

Hardware-in-the-Loop with Real Controller

Once the controller has gone into production, instead of verifying the controllers on a real plant, they can be readily tested using the same plant simulator that was used in the prototyping stage, allowing rigorous, clean, repeatable, automated tests to be carried out.

Embedded Data Acquisition and Control

Controller on RT-LAB Embedded Target

Once the controller has been developed and fully tested, in many cases it makes sense to simply deploy the controller on the platform it was developed. RT-LAB has several tools to help you convert the prototype controller into a robust, secure embedded system that can be connected to the real plant and all the other peripheral systems it needs to communicate with.

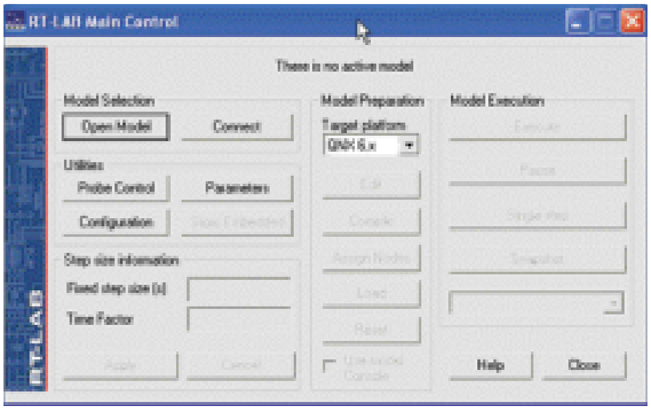
High-Speed Signal Capture and Data Logging

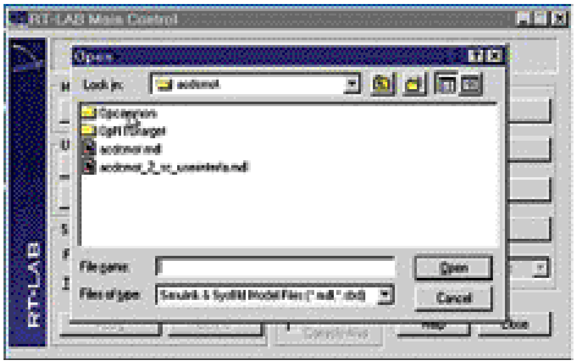
RT-LAB's unique high-speed data logging and event capture features allow data to be acquired and streamed to disk on the target system, as well as capturing and time-stamping events, making an ideal tool for data logging applications. Added to this, its embedded support allows you to create standalone systems for unattended or remote operation.

Six-Steps to Real-Time

Schematic Model to
Real-Time Simulation

The RT-LAB interface is a dialog box that takes you through the workflow in order to compile, distribute and execute the model. The process can be described in six steps.

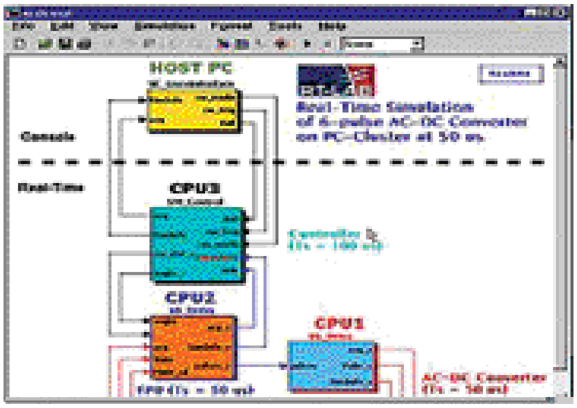




Step 1

Open Model

The first step is to **select the model** you wish to run in real-time. RT-LAB currently supports models developed in Simulink and SystemBuild.



Step 2

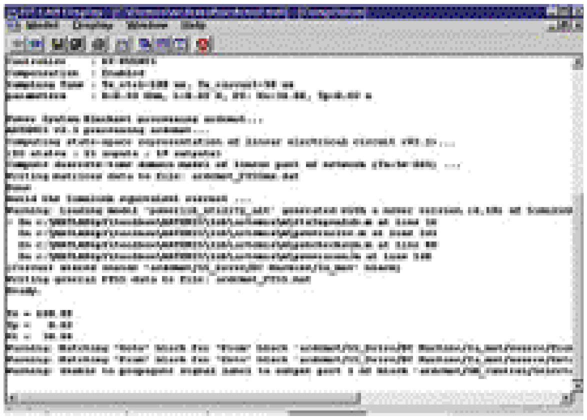
Edit

RT-LAB provides tools for **separating the model into subsystems** for execution on each target processor (or "node"), including a specialized block-set that allows you to incorporate many advanced real-time features into the model.

Step 3

Compile

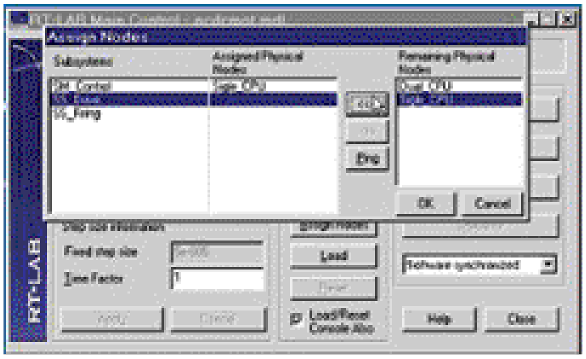
Using the appropriate code generator (Real-Time Workshop or AutoCode) and RTOS (QNX or RedHawk Linux), **each component of the model is converted to an executable**, optimized for hard-real-time execution.



Step 4

Assign Nodes

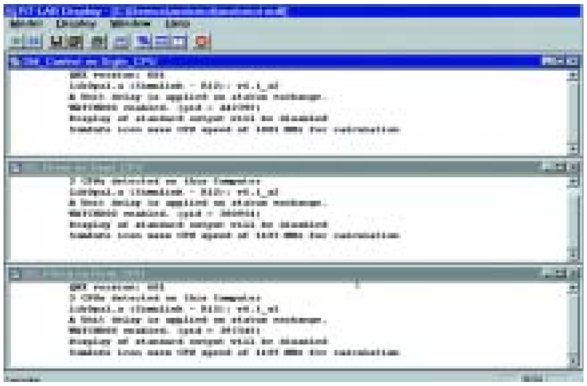
RT-LAB searches the network for appropriate nodes and allows the user to **assign each component** executable to the selected nodes.



Step 5

Load

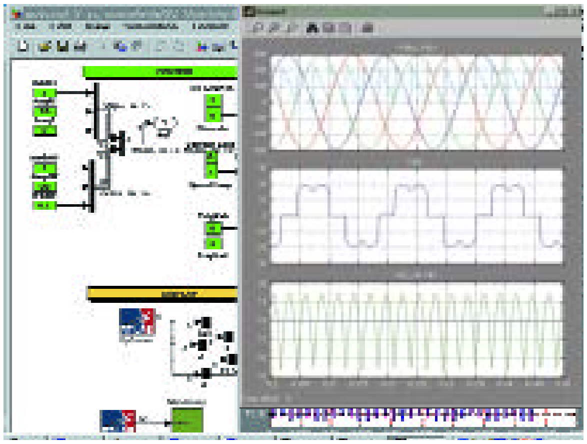
The user instructs RT-LAB to **load the executable** into the target nodes' memory, initialize the models and communications, and synchronize the models so that the simulation runs exactly in step over the separate nodes.



Step 6

Execute

A final click on the Execute button **starts the simulation**. Data from the model is directed to the user via a special subsystem, called the console. Here you can view the signals being generated, and change parameters on-line to the simulation. You can also use the extensive API tools to build your own user interface.



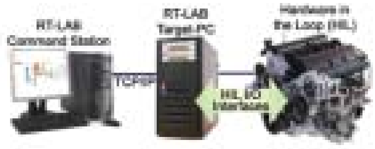
Using RT-LAB, you can very easily run a demanding model in real-time by distributing the computational load across several PC-based processors. Using the six steps outlined you only need to define how the model will be separated, and RT-LAB automatically handles the rest.

RT-LAB Features
Distributed Computation

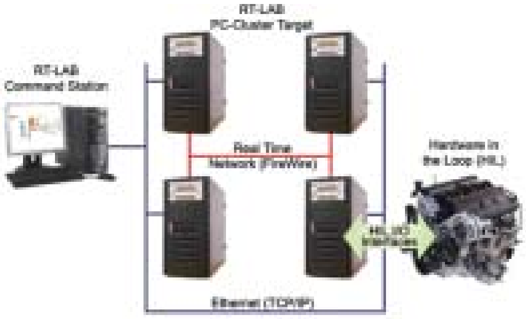
The main uniqueness of **RT-LAB** is its capacity to execute large models over several parallel processors. By doing this using standard, off-the-shelf technology, RT-LAB brings many benefits apart from raw computation power. It means you get a scalable, flexible real-time platform at an affordable cost.

Distributed Target Execution

RT-LAB provides tools for easy separation of the system model into subsystem models that can be executed on parallel target processors (standard PCs running either the QNX Real-Time operating system or RedHawk Linux). In this way, if you need to run a model in real-time that cannot be run on a single processor, RT-LAB provides a means of sharing the load over several processors.

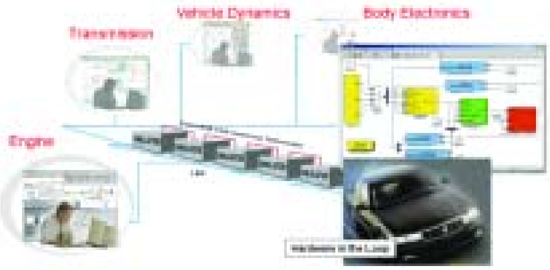


At execution time, RT-LAB provides seamless support for inter-processor communication, using any combination of UDP/IP and Shared Memory, all readily available technologies for low-latency communication of data between the target processors. The user can also interact with the simulation in real-time from the Host station using UDP/IP.



Distributed User Stations

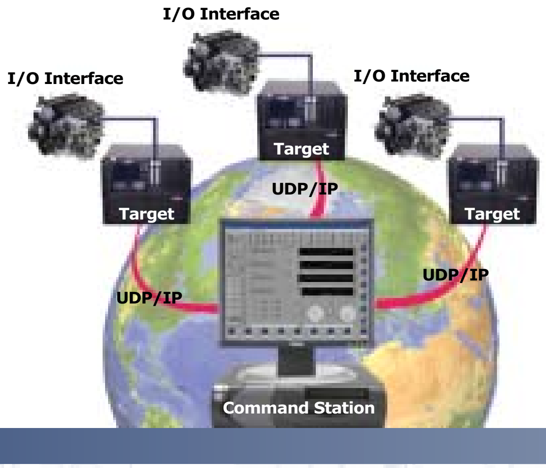
Because RT-LAB allows subsystem simulation on parallel targets, it is an ideal platform for group collaboration on large simulation projects. Each group can focus on developing the simulation for their own subsystem, testing it in real-time on their own simulator. Then it can be readily incorporated into a simulation of the complete system, where each group contributes their subsystem model, communicating input and output data through the real-time network.



Users can also use RT-LAB to analyze data from the whole simulation in order to study how their subsystem interacts with the other subsystems, and even change parameters in order to optimize the subsystem design.

Remote Operation Over the Internet

Since RT-LAB is using UDP/IP communications protocols to communicate between the host stations and the target systems, you can communicate with target systems anywhere in the world using the Internet. This makes it simple to share resources in, say, a controls laboratory, with external collaborators.



It also allows remote operation of real-time systems, such as a controller in a remote plant. Control engineers can change parameters and even complete control strategies from one office while analyzing the effect from a number of remote stations.

RT-LAB Features
Connectivity

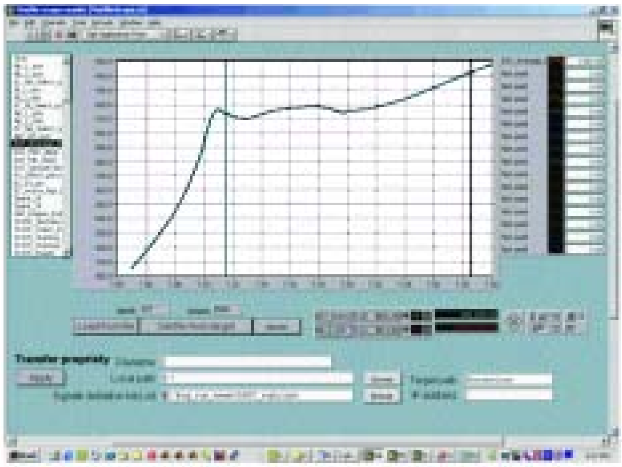
RT-LAB's fully featured, fully documented Application Programming Interface (API) allows solution developers to rapidly integrate their own applications into the real-time system. This is ideal for skilled programmers, but RT-LAB also provides tools that ease the connection between the real-time simulation and other applications running on the Host computer for non-programmers.

Specifically, there are two application areas where these tools will have significant impact: the development of LabVIEW user interfaces and the support for scripting languages, Visual Basic and Python.

Development of LabVIEW User Interfaces

The RT-LAB LabVIEW API Tools will allow you to easily develop a direct link between the real-time model and LabVIEW, without writing any program code. These tools include special, fully documented SubVIs to help you build LabVIEW GUIs very quickly. These SubVIs provide not only data paths between the model and the GUI but also error handling and status signals in order to increase the robustness of the system.

The RT-LAB API also allows LabVIEW to access any signals while the simulation is running, using names to reference them. This is particularly useful if you wish to create a user interface that provides the user with a choice of signals to display.



Finally, RT-LAB includes an "off-line" version of the API that allows you to develop and test a LabVIEW interface using Simulink in interpreted mode. This means that you do not need to have a copy of Real Time Workshop on the development computer and you don't need to compile each time you make a change in the model in order to test it. You simply run the model in Simulink in interpreted mode and the system, including the LabVIEW interface, will behave exactly like the final compiled version, although not in real-time.

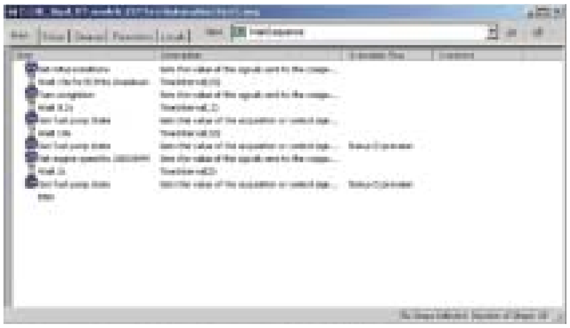
Test Automation with Python

Python is a cross-platform, open-source scripting language that is growing in popularity, particularly within technical applications.

Its syntax is very close to m-script, which has made it very popular among MATLAB users. It is object-oriented, with classes and threads etc., and allows users to automate applications on any platform. The RT-LAB API allows users to configure RT-LAB models and automate test runs using the Python language. Also, because Python is multi-threaded, you can interface to multiple concurrent models, running on several RT-LAB target processors. This means you can coordinate many different tests, even have data flow from one test platform to another, from a single operator station.

Python scripts can also run directly on the target, in real time. Running Python script in real-time will allow you to run test case that requires hard real-time with the flexibility of an high level language. The Python script can be modified and reloaded using the new automatic file transfer function without the need for resetting or recompiling the model.

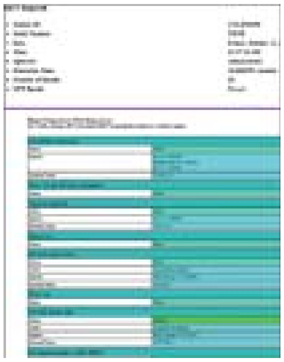
Test Automation with TestStand



TestStand is a ready-to-run test management environment for automating your test and validation systems. It is designed to maximize test-code reuse through built-in language adapters and simplify maintenance by using a modular fully customizable test-system framework. TestStand provides a way for you to see how to code using your preferred language/tool.

- A test sequence consists of three parts:
  - Set up
  - Main
  - Clean up

  For RT-LAB, the Set up and Clean up parts are usually the same for each test case, but the Main is where your test logic resides

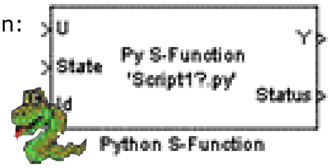


- When you execute the sequence, the RT-LAB model will be loaded and executed.

  At the end of a run, an analysis using m-script or python script could be done. TestStand will then display the report including a graph which is saved as an HTML file.

Python S-Function

Python S-functions can be used to create custom Simulink blocks for use with RT-LAB. An S-function:



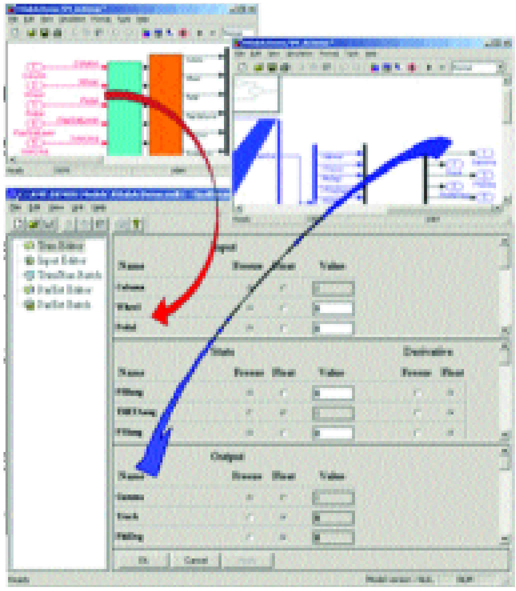
- Enables you to change your script in real-time without having to reset or recompile your model.
- Can be used to perform sequencing of tasks.
- Can be used as a signal generator.

The Python S-Function block provides access to Python S-functions from a Simulink block diagram. The S-function must be coded in Python and be saved as an external Python module.

Automatic File Transfer Between Host & Target

RT-LAB can make use of many data files for logging, signal generation and changing parameters. Until now, the process of getting these files from, or sending the files to the target system was a manual process that had to be carried out before and after a simulation run.

RT-LAB DINAMO for Simplifying Aeronautical Simulations



At the center of most aeronautical model-based design initiatives are Simulink 6-DOF simulations. These simulations must be used for analysis and experimentation, while being deployed on desktops and test rigs. DINAMO facilitates 6-DOF model deployments by adding features commonly required by aeronautical engineers to RT-LAB.

DINAMO solves the vexing problem of how to trim RTW generated simulations. It automatically configures an RT-LAB 6-DOF simulation for trimming. Then, through its graphic interface, allows the definition of trim types, specification of trim conditions and setting of model parameters. The result is a greatly simplified trim implementation with sufficient flexibility for desktop analysis and test rig initialization.

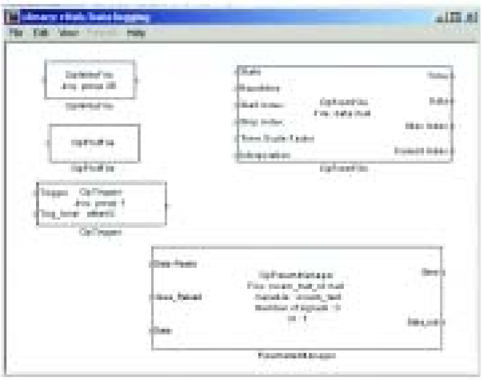
DINAMO includes tools for flight mechanics analysis or test rig stimulus. Tabular input files may be created, trims and inputs sequenced and batches run in real-time or faster.

DINAMO also provides accelerated flight test matching of 6-DOF models. A simulation can be associated to flight test data and sequenced to run in a batch. Parameter estimation algorithms resident in DINAMO interface to the RT-LAB simulation for fast-as-possible optimizing.

Support for 3D Virtual Reality Visualization Tools

RT-LAB allows you to use 3D Virtual Reality tools, such as the MATLAB Virtual Reality Toolbox, Altia or X-Plane, to create stunning visualizations of your model in 3D. These tools, along with RT-LAB, help you to view the behavior of the model, and interact with it, in real-time.

RT-LAB Features
Data I/O and Logging



RT-LAB extensive block-set for acquisition, transfer and logging of data covers the full range of features required by real-time systems.

Real-Time Acquisition for Visualization and Data Logging

RT-LAB provides unique data visualization and logging features that make it the most complete and convenient real-time systems design environment available. Real-Time visualization and logging is a crucial element of the system and considerable energy has been put into the implementation of features that do not limit online access to model signals.

Extensive I/O Board Support

RT-LAB provides support for the largest range of ISA/PC104/PCI/PXI/PCMCIA data I/O hardware for QNX and RedHawk linux, with drivers for over 75 commercial devices. From CANbus to 1553 protocols, through analog and digital I/O for hardware-in-the-loop (HIL) interfaces, to high speed OHCI and IEEE-1394 (or FireWire) real-time communications, RT-LAB provides high-performances drivers for the most demanding applications.

Support for Opal-RT I/O Boards

RT-LAB provides support for the multi-function PCI and PC-104 boards developed by Opal-RT, as well as the SignalWire protocol for data acquisition and transfer and communication between systems. Specifically designed for Real-time applications, Opal-RT I/O products feature common analog and digital I/O functionalities, as well as advanced features such as signal generation and capture.

Triggered Subsystems

RT-LAB supports Triggered Subsystems, which allows you to set up specific tasks to be activated EXACTLY when triggered, not at the next time step after the trigger. This is particularly useful where you need, say, a control sequence to be executed when triggered, or when synchronizing the model to rotating machinery.

Hardware Synchronization and Triggering

RT-LAB provides several I/O drivers for synchronizing the clocks of multiple RT-LAB systems, or for hardware triggering data acquisition or transfer. These features are offered with the blockset for boards such as the Sensoray s626, the National Instruments ESeries or 6602 boards, and the Opal-RT OP5100 series. National Instruments and Opal-RT boards share the same RTSI bus architecture and connector for transmitting the synchronization signal from one board to another.

Frame-Based Data Acquisition

RT-LAB uses powerful new data-acquisition options that take full advantage of the input buffering, burst mode, DMA, and interrupt capabilities of the latest generation on ADC cards. You can configure high-speed acquisition, either continuously or during periods of interest. Best of all, there is no increase in the time-step since the ADC card can acquire data at the same time that the main processor is computing your model.

Ideal for very-high-bandwidth applications where conventional acquisition would overwhelm bandwidth and storage subsystems, RT-LAB data acquisition features help you acquire all the data you need, and only the data you need. Moreover, for the ultimate in flexibility, you can combine windowed data acquisition and triggered subsystems to create powerful event-based systems.

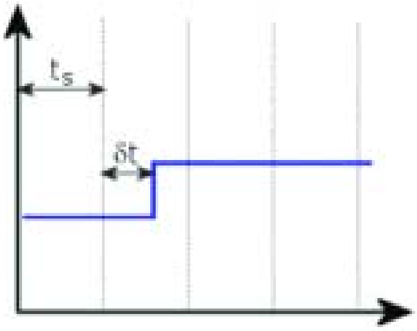
Advances Features for Waveform Generation

RT-LAB I/O block-set provides signal generation features, with characteristic and performances to suit a broad range of application needs.

RT-LAB also allows for generation of transitions on digital output lines at arbitrary times during calculation steps for very accurate triggering of external electronics and PWM control. The RT-LAB event generator driver for the Opal-RT OP5100 boards provides a 10ns precision, and a user-definable number of transitions per calculation step on up to 32 individual I/O lines.

Opal-RT I/O products also support application-specific advanced signal generation features such as the reference pulse generation for pattern generation in automotive applications.

Advanced Feature for Capture of Digital Waveforms



RT-LAB can detect input events that occur between computation time steps and compensate any calculations in order to remove errors that would be introduced if the event were only detected at the next time step after the event occurred.

While the simulation is running at a regular timestep (ts), an event may occur between two timesteps (dt). If the simulation waits until the next timestep to update the model, a numerical error will be introduced into the simulation. Using features of Opal-RT OP5000 boards, RT-LAB can accommodate this time difference and update the model appropriately.

SignalWire Data Transfer

RT-LAB supports the SignalWire protocol for data transmission and reception between RT-LAB systems, or between RT-LAB systems and I/O boards. SignalWire-based I/O boards can be daisy-chained on the same communication link, thus allowing for a large expansion of the I/O capabilities of a single target computer. The SignalWire link is allows you to install I/O boards remotely from the computing node and at a convenient location near the electronic system that receives or generates the data.

Generic Shared Memory Driver

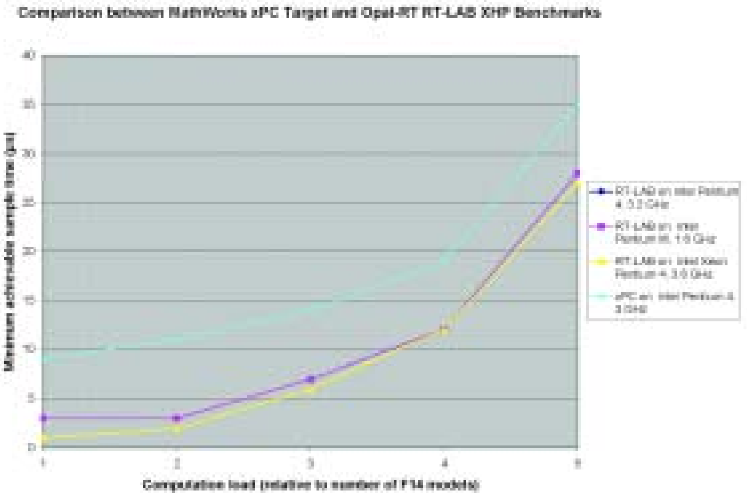
RT-LAB transparently supports the use of Shared Memory for very fast communications between two target processors in a distributed simulation, for example, embedded controllers or other processors that are not part of the simulation.

RT-LAB also includes special Shared Memory blocks that allow the user to configure shared memory and reflective memory devices, These are useful when the user wishes to read and write data between devices.

RT-LAB Features
Real-Time Performance

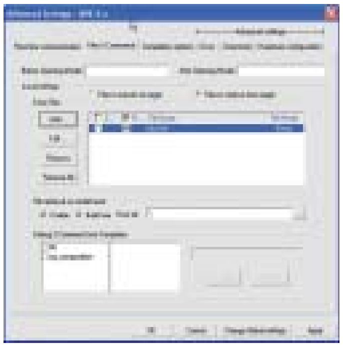
RT-LAB gives you the speed you need for computation of real-time models.

XHP Mode



RT-LAB XHP (eXtra High Performance) mode allows very fast computation of the real-time model on the target system. This has allowed our customers to simulate complex systems over distributed processors, with analog and digital I/O, at cycle times below 10 microseconds!

RT-LAB XHP mode slashes overhead, letting you use the full power of your computer for the first time. A constant challenge for developers of real-time simulations is achieving accurate, high fidelity responses for increasingly complex models. Even when the signals in a hardware-in-the-loop (HIL) system need only be updated in the 100s of microseconds time frame, the model may need to be computed many times between each major time step in order to maintain numerical accuracy. The XHP mode in RT-LAB brings an unprecedented 400% performance boost from normal mode! This, by far, out-performs any other real-time system, and is particularly useful for modeling electrical systems, such as drive controls and power electronics.



Multi-Rate XHP Mode

RT-LAB can run multi-rate models in normal and XHP modes. This allows unprecedented flexibility in configuring multi-rate models, while maximizing computation power.

Software-Synchronized Mode

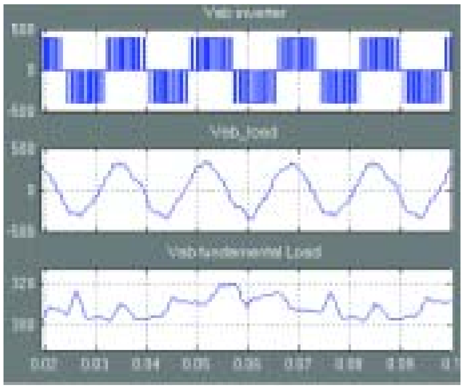
RT-LAB for Windows has a "Software-Synchronized" mode, allowing real-time simulation using Windows XP OS. Real-time simulations can be run at 100 Hz with only 1.5 ms of jitter, enabling low rate simulations, like aircraft or spacecraft dynamics without an RTOS, yet still benefit from many of the features of RT-LAB.

Optimized Scheduler

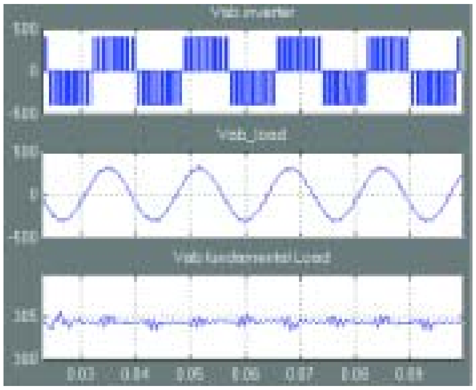
Within a time step, the system is doing more than computing the dynamic model. It also does administrative tasks, such as reading and writing I/O, updating the system clock, scheduling tasks, logging data, and handling communications. This restricts the amount of time available within a frame to compute the model values limiting the size of model that can be computed on a single processor. OPAL-RT has reduced this overhead to a few percent of raw hardware performance without losing functionality, thereby increasing the capacity to compute more complex models.

ARTEMIS: Special Solvers for Converting SimPowerSystem* Circuits to Real-Time

ARTEMIS (Advanced Real-Time ElectroMechanical Simulator) For High Precision, Real-Time Simulation of ElectroMechanical and Electric Systems



Standard Fixed-Step Solver



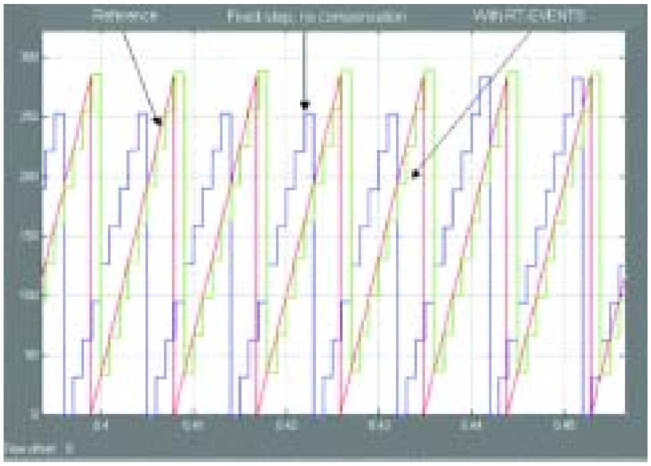
with ARTEMIS

For users of The MathWorks' SimPowerSystems Blockset (formerly known as the Power System Blockset) who need to accelerate their power system models or simulate them in real-time, ARTEMIS provides enhanced algorithms that ensure reliable, accurate, and fast fixed step-length computations, essential for high fidelity, high-performance simulations. Unlike SimPowerSystems on its own, ARTEMIS was designed from the ground up to support real-time implementations of power systems simulations, dramatically improving computation speed while preserving accuracy.

RT-EVENTS Blockset: Time Compensation for Mixed-Mode Simulation

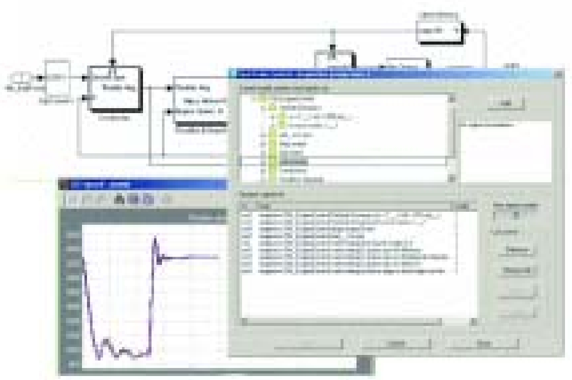
The RT-Events Blockset is an add-on that works with MATLAB Simulink™ to improve the efficiency and accuracy of continuous-time and discrete-time systems simulations whose dynamics are affected by discrete events. RT-Events relies on a compensated discrete-time simulation method that features:

- Improved accuracy for mixed-mode systems
- Compensation for the errors introduced when events occur between samples
- Fast simulation of cyclic event-driven systems
- Non iterative, uses fixed time-step algorithm
- Faster simulation than variable-step algorithms
- Support for distributed real-time simulation
- Suitable for hard real-time applications such as hardware-in-the-loop or embedded simulations
- Compatible with commercial code generators



*Formerly known as the Power System Blockset

RT-LAB Features Usability

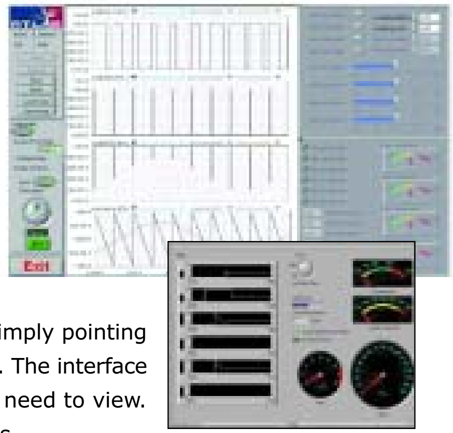


RT-LAB usability features help to develop real-time applications faster based on customer feedback.

Dynamic Signal Tracing

RT-LAB's on-the-fly dynamic signal tracing significantly improves the way you can build and test simulations, either inter actively or through custom front-end applications using the RT-LAB API.

Instead of having to define which signals need to be displayed or logged from the simulation before compiling the model, the Dynamic Signal Tracing function in RT-LAB allows you to access to the value and name of any signal in the model at run-time. This saves a significant amount of work, particularly when developing and debugging the model. You can create the model in Simulink without worrying about what signals you are likely to require at runtime in order to visualize their behaviour because you can access any signal in the model at run time through the "Probe Control" dialog panel in the RT-LAB Main panel, or through the RT-LAB API. Then access the signal through the API and or by clicking on the “Dynamic Signal Output” option in the RT-LAB OpComm Block.



Integrated Signal Visualization and Control Panel

RT-LAB includes a generic graphical user interface, built in LabVIEW, that allows the user to select and view signals as plots or gauges while the simulation is running, and interact with the simulation by changing parameters and input signals through sliders and buttons. It also allows you to build your custom panels effortlessly as it does not require you to do any LabVIEW coding. The application itself is a standalone executable for Windows.

While executing your RT-LAB models, you can map any signal of model parameter by simply pointing a signal in the signal tree and click on a panel component such as a slider, knob or scope. The interface extracts the signal names from the model so it is very easy to identify the signals you need to view. On the other hand, you can rename your signals with aliases to better suit your needs.

Online Access to Parameters and Parameter Portability

Many users make extensive use of RT-LAB's ability to change model parameters on the fly through its on-line Parameter Editor. Not only can parameters be changed interactively, files of parameter lists can be created and loaded into the editor in order to change many parameters in one step. However, because it is possible to change the structure of the model after a parameter file has been created, sometimes conflicts arise because a parameter may have been moved.

RT-LAB includes a Parameter Conflict Manager, which will highlight to the user when a conflict occurs and provides search tools and an easy-to-use interface to help the user resolve these conflicts. This means that files of parameters can be shared not only by different versions of the same model, but also by different models.

Snapshot Features

The snapshot feature allows a user to record the current state of a simulation and to restore it at a later stage. When running lengthy simulations, one can use the snapshot feature to "freeze" a simulation at a given time. The whole state of the system, including inputs and integrator values is saved, so that the simulation can be restarted from this freezing point instead of from the beginning. The snapshot has two modes of activation:

- Manual activation
- Triggered, when a signal in the simulation reaches a given threshold.

Compilation by Subsystem

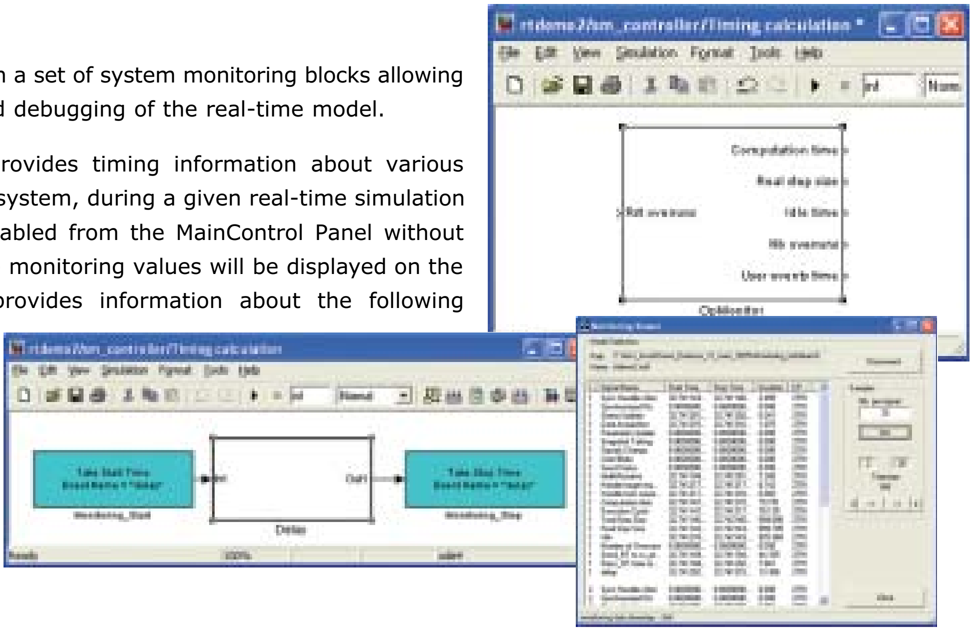
RT-LAB helps you save time compiling large models! Yes, it's a fact: compiling large models can take a long time, but you don't have to recompile the whole model just because you have changed one of the subsystems. RT-LAB allows you to select and compile just the subsystem you're working on, instead of the whole model, saving you significant amounts of time.

Tools for Monitoring Performance

The RT-LAB library comes equipped with a set of system monitoring blocks allowing for online performance assessment and debugging of the real-time model.

The OpMonitor block, for instance, provides timing information about various operations that are performed in a subsystem, during a given real-time simulation timestep. Monitoring could also be enabled from the MainControl Panel without adding and recompiling the model. The monitoring values will be displayed on the Monitoring ViewerPanel. The block provides information about the following characteristics, among other things:

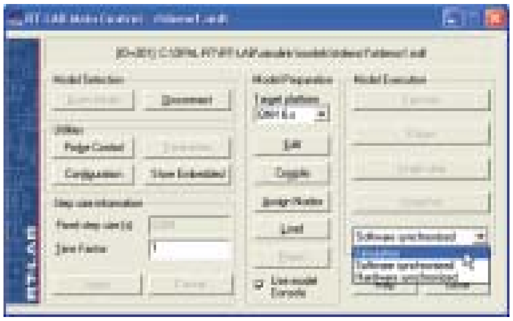
- Model computation time
- Real Step Size (computing, overhead and synchronization)
- Idle time
- Number of overruns
- User time



Available Simulation Modes

RT-LAB provides different simulation modes, depending on the application and hardware/software configuration of the system:

- Simulation "as fast as possible", with no real-time constraints for running fast simulations or simulation batches e.g. for Monte-Carlo optimization
- Software synchronized, using the CPU clock as a software timer
- Hardware synchronized, using a hardware timer card



Automatic MATLAB Version Detection

This allows legacy models to be automatically supported. You do not need to manually select the appropriate runtime libraries according to the version of MATLAB that was used to create the model, RT-LAB ships with all versions of the libraries and will automatically select the correct version.

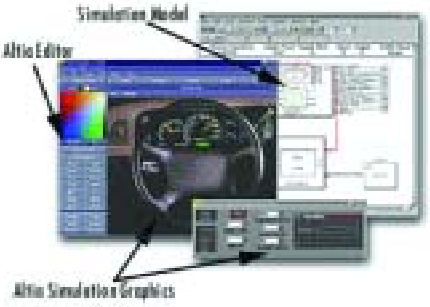
It also allows models from different versions to be used together, eliminating the need to recompile and test older models.

MATLAB Versions Supported: 6.0 (R12), 6.1 (R12.1), 6.5 (R13), 7.x(R14.x)
Please Note: RT-LAB will always support the latest version of MATLAB.

Integrated Online Documentation

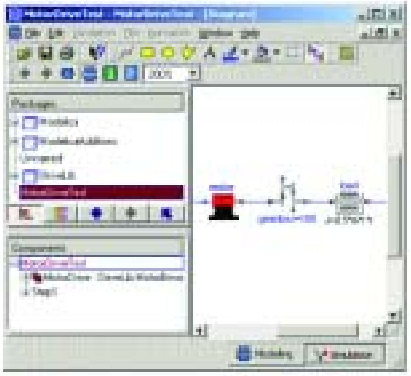
- Complements the User Manual
- Comprehensive context-sensitive help
- Complete Documentation online
- Example models fully documented to ease learning how to use RT-LAB

RT-LAB Third-Party Products



Altia

Altia provides a suite of Simulation Graphics software that enables engineers to build photo-realistic interactive front ends to their RT-LAB simulations.

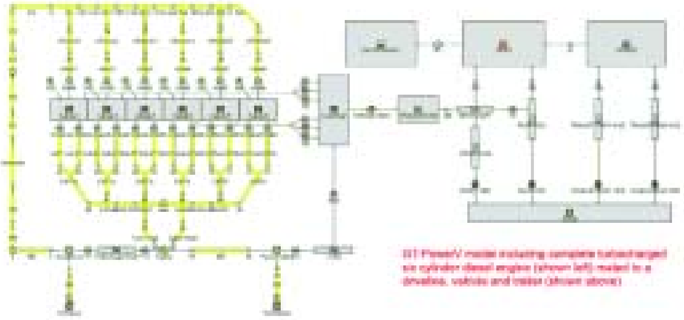


Dynasim, Dymola

Dynasim has developed Dymola for multi-engineering modeling and simulationfor use within automotive, aerospace, robotics, process and other applications. Dymola allows simulation of the dynamic behavior and complex interactions between, for example, mechanical, electrical, thermodynamic, hydraulic and control systems.

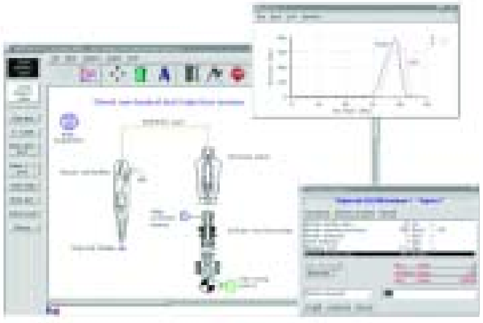
RT-LAB has been successfully tested with Dymola models to demonstrate that they can be run in real-time.

Gamma Technologies, Inc. GT-Power


GT-PowerV model including complete turbocharged six cylinder diesel engine (shown left) mated to a driveline, vehicle and trailer (shown above)

GT-POWER is the industry-standard engine simulation tool used by all leading engine and vehicle makers and suppliers. It is the industry's most comprehensive and flexible simulation tool, suitable for application to a wide variety of design issues. Its usefulness is further enhanced by integration with 3rd party products such as STAR-CD, Fluent, Simulink, MATRIXx, MS/Excel, Isight and Bistro.

GT-POWER is specifically designed for steady state and transient simulations, and can be used for analysis of engine/powertrain control. It is applicable to all types of I.C. engines, and it provides the user with many components to model any advanced concept. GT-Power is available as a standalone tool or coupled with GT-Drive as GT-PowerV. RT-LAB has been integrated with GT-Power to produce a real-time simulations of systems modeled in the GT-Power environment.

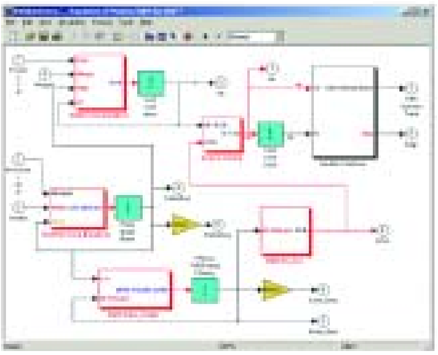


Imagine Software, AMESim

AMESim stands for Adaptive Modeling Environment for Simulation. It is one of a growing number of application-oriented software with libraries made with recognizable icons and sub-models, designed to address the requirements of engineers in industries such as automotive, aerospace or hydraulics.

RT-LAB now supports models generated in AMESim for high-performance real-time execution.

The MathWorks Inc, MATLAB/Simulink/Real-Time Workshop



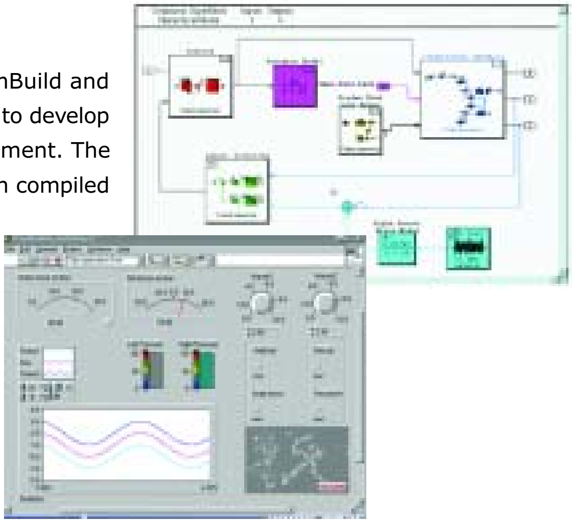
RT-LAB is tightly integrated with the Simulink environment, allowing the user to use the same environment used to develop the model for preparation and separation for real-time execution over distributed targets. RT-LAB provides a special blockset to prepare the model for distributed processing, inter-node communication and signal I/O.

Once the model has been prepared, RT-LAB uses Real-Time Workshop to convert the separated models into C code for compilation as subsystem simulations on each target processor.

Mechanical Simulation Corp., CarSim and TruckSim

CarSim and TruckSim are software packages for simulating and analyzing the way road vehicles, both with and without trailers, respond to driver controls on 3D road surfaces. RT-LAB has been integrated with CarSim and TruckSim to allow user to run the vehicle dynamics models in real-time not only to better visualize the vehicle behavior but also for hardware-in-the-loop applications.

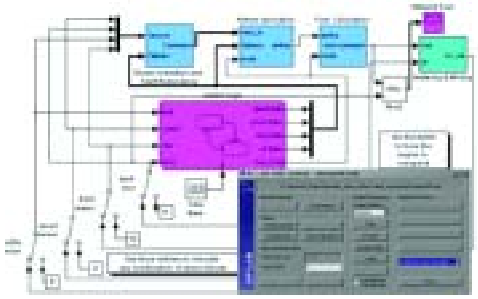
National Instruments, MATRIXx Product Family and LabVIEW



RT-LAB is fully compatible with the MATRIXx product family - Xmath, SystemBuild and AutoCode. Its tight integration with SystemBuild and TestStand allows the user to develop the system model and prepare it for real-time execution in the same environment. The separated subsystem models are converted to C code using AutoCode and then compiled and executed on the distributed targets.

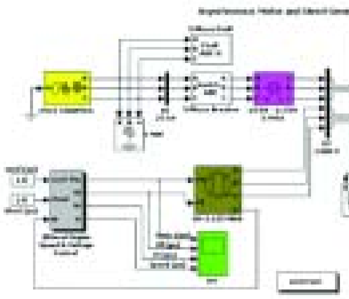
National Instruments LabVIEW delivers a powerful graphical development environment for signal acquisition, measurement analysis, and data presentation, giving you the flexibility of a programming language without the complexity of traditional development tools. LabVIEW allows you to acquire, analyze and present data. RT-LAB's control environment interfaces are created with LabVIEW.

Full Support for Stateflow Libraries



Your design includes state logic? Need real-time? No problem! RT-LAB fully supports Stateflow blocks within Simulink models for real-time implementation (required state flow coder).

SimPowerSystems Blockset for Simulink



The SimPowerSystems Blockset (formerly known as the Power Systems Blockset) has been designed to allow users to create models of power electrical circuits using familiar component blocks inside a Simulink schematic diagram. It has been built for off-line (non-real-time) analysis but, with RT-LAB's specialized model development tools, you can readily convert these models into real-time simulations.

RT-LAB Components

RT-LAB Host Development Tools

Component Descriptions

RT-LAB Development Tools.

- Prepares your Simulink Model for hard-real-time execution on QNX or RedHawk Linux target.

Operating Systems

- Windows NT/2000/XP
- QNX
- RedHawk Linux

Multiple-rate Module.

- True multi-threaded, multi-rate execution of subsystems.
- Allows some subsystems to be updated at faster rates than others.
- Run bigger models on a single processor.

Multiple-processor Module.

- This provides the support for handling communication between two or more target nodes.

RT-Scope - Data Visualization GUI Module

- A standalone LabVIEW panel that incorporates the model's preparation, compilation, execution capabilities of the standard RT-LAB user interface with a simple-to-use data visualization

RT-LAB API Tools

- Provides a software interface to third party applications. Several tools have been developed to simplify the link through the API to:
- NI TestStand
- NI LabVIEW
- MATLAB M scripts
- Python scripting language
- Visual Basic
- C/C++ applications

Key ✓ = Standard Component O = Optional component Available	**RT-LAB Solo Host System** Basic Single Host Single Target System Educational Use Only	**RT-LAB Professional Host System** Full access to all development tools for industrial and research applications	**RT-LAB Host Runtime System** Provides access to the simulation only via the host PC. Ideal for deployed applications that require operator interaction
Floating License	O	O	✓
Multiple Target Support	-	✓	O
Model Separation	-	✓	-
Compilation on Target	✓	✓	-
Data Display and Logging	✓	✓	✓
API Tools	O	O	O
RT-Scope	✓	✓	✓
Manual Set and CD-Rom	O	✓	O
Maintenance Contract	O	✓*	O
RT-EVENTS	-	O	O
ARTEMIS	-	O	O
DINAMO	-	O	O
Test Manager	-	O	O

*1st Year Only

Licensing Options

Node-locked license (standard)
Locked to one computer.
Floating license (optional)
Allows access to RT-LAB development tools on any PC on the LAN connected to one or more targets.

Visualization and Analysis Tools

DINAMO

A specialized simulation sequencing tool for aeronautical modeling and testing. Includes rapid trim computation, and parameter estimation for RT-LAB simulations.

Third-party Modeling Tools

RT-LAB supports the following high-fidelity modeling tools:

MATLAB/Simulink
MATRIXx/SystemBuild
CarSim
TruckSim
AMESim
Modelica/DYMOLA

This list is constantly being updated. Please check regularly for new modeling tools, or ask us about supporting the tool(s) that you use.

RT-LAB Target Software

Component Descriptions

Hard Real-Time Scheduler
Provides all timing and scheduling management in order to get high-performance, minimal-jitter real-time execution of the compiled model.

IO Management Module
Provides support for the I/O interface cards for Hardware-in-the-Loop applications as well as data logging. See www.opal-rt.com for supported boards.

XHP Run Mode
E**X**tra **H**igh **P**erformance mode of RT-LAB. Allows the user to bypass most of the RTOS kernel in order to maximize the computational execution of the model.

Embedded Self-boot Module
Allows the real-time system to start up and run without having a Host Station connected. Ideal for embedded control applications.

Multiple-rate Module
True multi-threaded, multi-rate execution of subsystems on the Target Node. Allows some subsystems to be updated at faster rates than others, which means you can run bigger models on a single processor.

Multiple-Processor Support Module
Required for two or more Target Nodes

FireWire, SignalWire and InfiniBand Target-Target Comms. Module
Allows data flow between two or more Targets through a high-speed, low latency FireWire, SignalWire and InfiniBand links. Requires Multi-processor Support Module and FireWire SignalWire and InfiniBand interfaces on each Target Node.

Shared Memory Module
Allows data flow between two or more Target Nodes on a Dual or Quad processor motherboard. Requires Multi-processor Module on each Target.

TCP-IP Host-Target Communication Module
Handles all communication between the Host and Target(s) via UDP/IP link such as LAN, WAN and the Internet.

Real-Time Operating System QNX and RedHawk Linux

QNX is an established RTOS for mission-critical engineering applications.

RedHawk Linux is an industry-standard, POSIX-compliant, real-time version of the open source Linux operating system.

RTOS, Single Processor Support
Engineering RTOS for each Target Node.

RTOS, Multiple Processor Support
Required for systems with two or more Target Nodes.

RTOS, Compilation Tools
Compiler, linker and support libraries required by RT-LAB to update real-time models. Changes to the model structure will require a recompilation in order reflect those changes. This process requires no interaction by the user. Note that model parameters can be changed on-line at runtime and, therefore, do not require a recompilation.

Key ✓ = Standard Component O = Optional component Available	**RT-LAB Solo Target System** Basic Single Host Single Target System Educational Use Only	**RT-LAB Professional Target System** Full access to all development tools for industrial and research applications	**RT-LAB Target Runtime System** For deployed applications that require operator interaction	**RT-LAB Embedded Target Runtime** For deployed applications that are required to run in unattended operations
Hard Real-Time Scheduler	✓	✓	✓	✓
Multirate	✓	✓	✓	✓
Multiple Target Support	-	O	O	-
FireWire Target-Target Comms	-	O	O	-
Shared Memory Communications	-	O	O	-
TCP-IP Host Communication	✓	✓	✓	O
FireWire Target-Host Comms.	O	O	O	O
XHP Mode	-	O	O	-
RTOS (QNX or RedHawk Linux)	✓	✓	✓	✓
RTOS Compiler	✓	✓	-	-
I/O Management	✓	✓	✓	✓
Embedded Self-Boot	O	O	O	✓
Local High-Speed Data Logging	✓	✓	✓	✓
Maintenance Contract	O	✓*	O	O
ARTEMIS	-	O	O	O
RT-EVENTS	-	O	O	O

*1st Year Only

Special Real-Time Solvers for Simulink

RT-EVENTS
Time compensation for mixed-mode continuous and event-driven models. (with or without HIL)

ARTEMIS
Enhances the SimPowerSystems (aka Power Systems) Blockset for Simulink to run power systems models in real-time.
Precomputation of switched circuit topologies
Fixed-step solvers for stiff systems
Non-iterative solving of algebraic loops
Time compensation for discrete events

Printed Documentation, Software CD-ROM

RT-LAB is completely downloadable, including the documentation. Software can be supplied on CD-ROM with a printed manual.

Maintenance & Support

The maintenance and support contract ensures that our customers benefit from the latest enhancements to RT-LAB as they are developed, tested and released.

If you're considering purchasing an Opal-RT product, please be sure to include a Maintenance contract in your budget. It is the surest way of keeping your systems "future-proof".

Support for I/O

Support for Third-Party I/O

OPAL-RT supports several board types such as:

- IP Modules
- Multifunction Boards
- GPIB
- 1553
- CAN
- etc.

The supported third-party I/Os are from various manufacturers like:

- National Instruments
- Acromag/SBS Technologies
- RTD
- Kontron
- DDC
- Connect Tech
- etc.

To see a complete list of third-party I/Os go to:

www.opal-rt.com/productsservices/hardwarecomponents/io_hardware/io_thirdparty/io_index.html

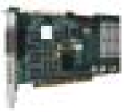
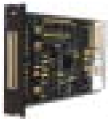
Support for Reconfigurable OP5000 FPGA-Based I/O

High-Speed, reliable signal capture and generation is essential to the precision of many HIL applications. RT-LAB adds to its already extensive I/O capabilities the new OP5000 range, which allows high-speed data acquisition and control output, fully optimized for HIL simulation applications. RT-LAB fully supports this range in order to handle large channel densities at very high update rates. For example, it allows capture and generation of PWM signals at a resolution of 10 ns.

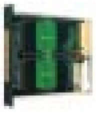
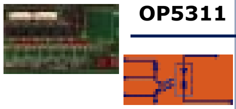
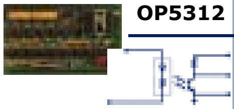
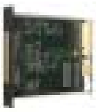
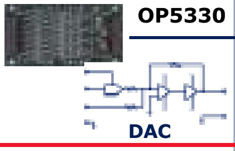
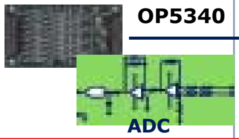
See table on following page.

FPGA-Based Signal Conditioning & I/O Products

OP5100 FPGA RECONFIGURABLE PROCESSING & I/O BOARDS

Product Name	Form Factor	Data link	I/O Channels	Range	PWM	Timer/ Counter	Xilinx Device	Comments
 **OP5110**	**PCI Interface, SignalWire & Digital I/O**						**Virtex II Pro**	
	PCI 3/4 Length	PCI SignalWire (up to 4 ports)	Digital In/Out 128	TTL	Resolution: 10 ns	Yes Clock: 100 MHz	XC2VP7	ADC via OP5330, and DAC via OP5340 (see below) and carrier OP5220 or OP5130
OP5120	**PC/104+ Interface, SignalWire & Digital I/O Boards**						**Virtex II**	
	PC104+	PC104+ SignalWire (1 port)	Digital In/Out 64	TTL	Resolution: 10 ns	Yes Clock: 100 MHz	XC2V1000	ADC and DAC via OP5130, (see I/O Modules below)
 **OP5130**	**SCXI-Format 4U Carrier for OP5330 and OP5340 I/O Modules**						**Virtex II Pro**	
	SCXI	SignalWire (1 port)	Option: Analog In: up to 32 Analog Out: up to 32	Digital: 5V TTL Analog:	Resolution: 10 ns	Yes Clock: 100 MHz	XC2VP7	Includes seating for up to 2 I/O Mezzanine Modules type B (see below)

OP5200 I/O CARRIER BOARDS & OP5300 MEZZANINE I/O MODULES

Product Name	Form Factor	Data link	I/O Channels	Input Range	Output Range	Protection	Requirements	Comments
 **OP5210**	**Carrier for Signal Conditioning Type A Module**							Includes seating for up to 2 Type A Signal Conditioning Modules (see below)
	4U SCXI Cassette	Parallel DIO		N/A	N/A	N/A	OP5100, or 3rd Party Interface	
 **OP5311**	**Digital Input Signal Conditioning Type A Module**							Input voltage selectable by resistor-pack
	Mezzanine Module for OP5210	On-Board Interface	16 Digital In	0-30V	N/A	Opto-isolated	OP5210 Carrier	
 **OP5312**	**Digital Output Signal Conditioning Type A Module**							
	Mezzanine Module for OP5210	On-Board Interface	16 Digital Out	N/A	0-28V	Opto-isolated	OP5210 Carrier	
OP5320	**Differential Amplifier Type A Module**							Input or output mode selected by orientation of the module on the carrier. Gain and offset selectable through resistor packs
	Mezzanine Module for OP5210	On-Board Interface	16 differential channels Analog	+/- 100V	+/- 16V	Short-circuit	OP5210 Carrier	
 **OP5220**	**Carrier for Analog Conversion & Signal Conditioning Type B Module**							Includes seating for up to 2 Type B Signal I/O Modules (see below)
	4U SCXI Cassette	Parallel DIO		N/A	N/A	N/A	OP5100	
 **OP5330**	**Digital to Analog Converter & Output Protection Type B Module**							16 independent DA converters, 1 µs settling time. 16MS/s total throughput
	Mezzanine Module for OP5220	On-Board Interface	Analog Output 16 channels. 16-bit 1 MS/s per channel	5V TTL	Programmable Max: +/-16V Min: +/-100mV	Short-circuit	OP5220 Carrier or OP5130 Carrier	
 **OP5340**	**Analog to Digital Converter & Signal Conditioning Type B Module**							16 independent AD converters, 2 µs settling time. 8MS/s total throughput
	Mezzanine Module for OP5220	On-Board Interface	Analog Input 16 differential channels. 16-bit 500 kS/s per channel	Programmable Max: +/- 16V Min: +/-100mV	5V TTL	+/- 100V	OP5220 Carrier or OP5130 Carrier	
OP5231	**Fiber-Optic I/O Type B Interface**							
	4U SCXI Cassette	Parallel DIO	16 Digital Inputs 16 Digital Outputs	5V TTL	5V TTL	Optical fiber	OP5100, or 3rd Party Interface	

OPAL-RT TECHNOLOGIES INC.
1751 Richardson, suite 2525
Montréal, QC H3K 1G6
Canada

For more information about RT-LAB
and how to receive an evaluation copy,
contact us today!

SALES
Toll Free:
1-877-935-2323 (North America only)
Phone: +1-514-935-2323
Fax: +1-514-935-4994
info@opal-rt.com
www.opal-rt.com