November 13–14, 2024 | Online

How to Verify Satellite Links in the Lab: Orbit-Ready Testing

Colin Telfer, RLS Product Manager, Square Peg Communications

MATLAB EXPO

Agenda

- New age of satellite communications
- Challenges of non-terrestrial radio links
- Simulation and test to the rescue
- Real example of satellite links in the lab

A New Era for Satellite Communications

The space economy is now more accessible than ever

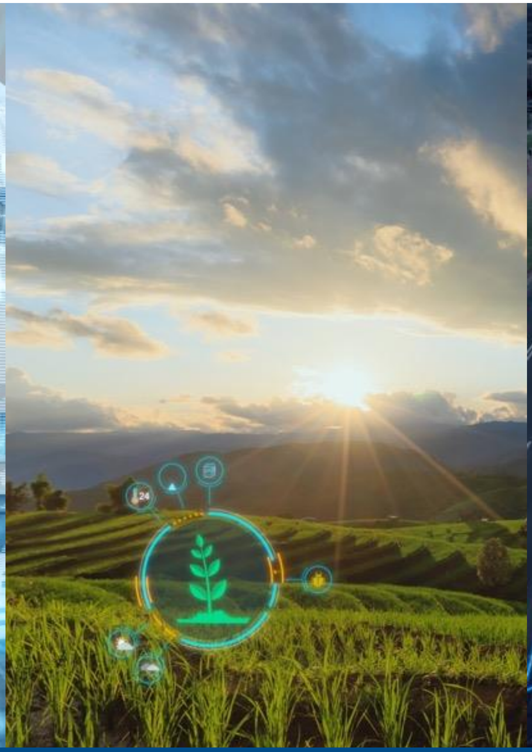
Dramatic drop in launch costs are igniting new SATCOM opportunities

SQUARE PEG COMMUNICATIONS MATLAB EXPO

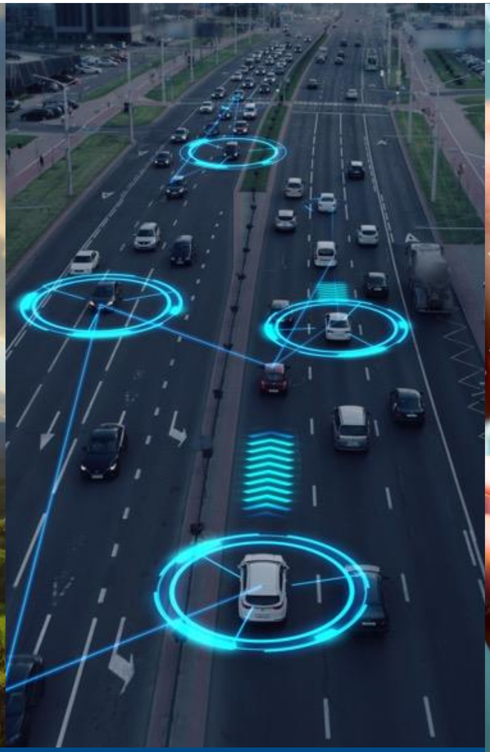
Creating Endless Opportunities for Innovation

Reliable disaster response

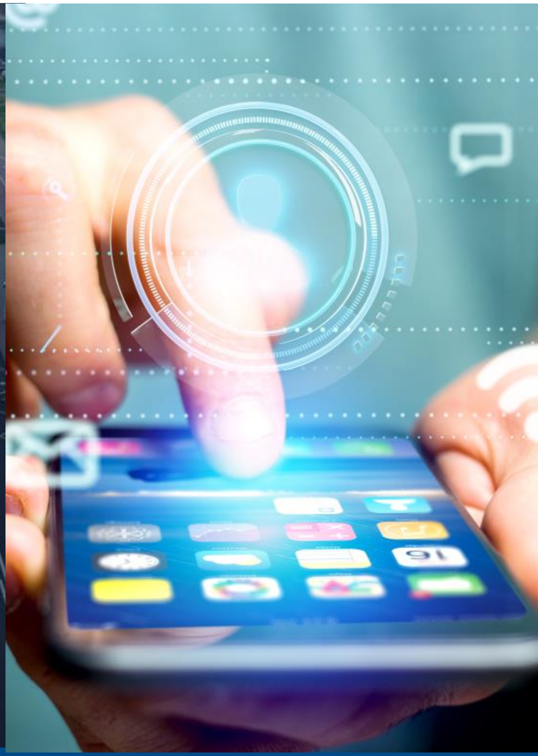
Tracking global assets



Precision farming



Autonomous vehicle connectivity



D2D coverage everywhere

SQUARE PEG COMMUNICATIONS MATLAB EXPO

Turning SATCOM Ideas into Reality is No Easy Task

- Example: direct-to-device (D2D) connectivity
  - Seamless connectivity from satellite to cell phones and existing devices
  - Satellite to fill coverage gaps for cellular mobile
- Demands unique considerations
  - As technology evolves, the need for critical testing grows
  - Satcom complexity requires critical testing to ensure QoS

Complex Technology Requires Rigorous Preparation

As technology evolves, the need for critical testing and simulation increases

- Missteps mean massive costs
- There are no second chances
- Space is unpredictable, and unforgiving
- Rigorous testing is crucial

SQUARE PEG COMMUNICATIONS MATLAB EXPO

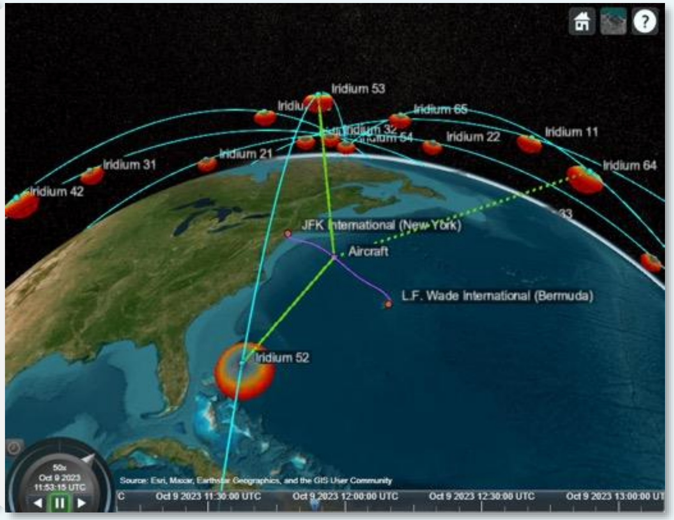
Benefits of Lab Simulation and Link Testing

- Reduce risk
- Decrease cost
- Maximize resources
- Improve schedule

Introducing MathWorks' Satellite Communications Toolbox

Simulate, analyze and test satellite communications systems and links on your workstation

```
% Generate satellite scenario
startTime = datetime(2024,09,12,18,35,0);
stopTime = startTime + minutes(8);
sampleTime_s = 1;
sc = satelliteScenario(startTime,stopTime,sampleTime_s);

% Create satellite object
tleFile = "STARLINK-1019.tle";
sat = satellite(sc,tleFile, "Name","STARLINK-1019", ...
    "OrbitPropagator","sgp4");

% Create Gateway object
gs = groundStation(sc, 47.5487, -52.7184, ...
    "Name","Gateway");

% Create User Terminal object
ut = groundStation(sc, 43.5631, -72.8974, ...
    "Name","User Terminal");

% Configure link parameters
```



RLS-2100 - Ensuring Solutions Are Orbit-Ready

- Bring satellite and terrestrial radio simulation to life, in real time, in the lab
- Solves:
  - ✓ Integration complexities
  - ✓ Equipment challenges
  - ✓ Real time operation for extended periods

RLS-2100 Radio Link Simulator

- 1.2 GHz processing bandwidth
- Physically accurate end-to-end path simulation
- Integrated scenario modelling
- Graphical displays
- Support for all orbits LEO/MEO/GEO/HEO
- Internal GPS, OpenAMIP and ARINC simulators
- Python API

Bringing Satellite Connectivity into the Lab

Use RLS-2100 and MATLAB to test and simulate low Earth orbit (LEO) links in the lab

Modeling LEO Scenario with the Satellite Communications Toolbox

```
% Generate satellite scenario
startTime = datetime(2024,09,12,18,35,0);
stopTime = startTime + minutes(8);
sampleTime_s = 1;
sc = satelliteScenario(startTime,stopTime,sampleTime_s);

% Create satellite object
tleFile = "STARLINK-1019.tle";
sat = satellite(sc,tleFile, "Name","STARLINK-1019", ...
    "OrbitPropagator","sgp4");

% Create Gateway object
gs = groundStation(sc, 47.5487, -52.7184, ...
    "Name","Gateway");

% Create User Terminal object
ut = groundStation(sc, 43.5631, -72.8974, ...
    "Name","User Terminal");
```

Generate scenario

```
% Calculate delay
s_delays_up = latency(sat,gs);
s_delays_dn = latency(sat,ut);

% Derive range
ranges_up = s_delays_up * physconst("LightSpeed");
ranges_dn = s_delays_dn * physconst("LightSpeed");
offsets_s = sampleTime_s * (0:(length(s_delays_up)-1));

% Calculate Doppler shift
fShifts_up = dopplershift(gs,sat,Frequency=fc_up);
fShifts_dn = dopplershift(sat,ut,Frequency=fc_dn);

% Calculate propagation loss
loss_up = fspl(ranges_up, physconst("LightSpeed")/fc_up);
loss_dn = fspl(ranges_dn, physconst("LightSpeed")/fc_dn);
```

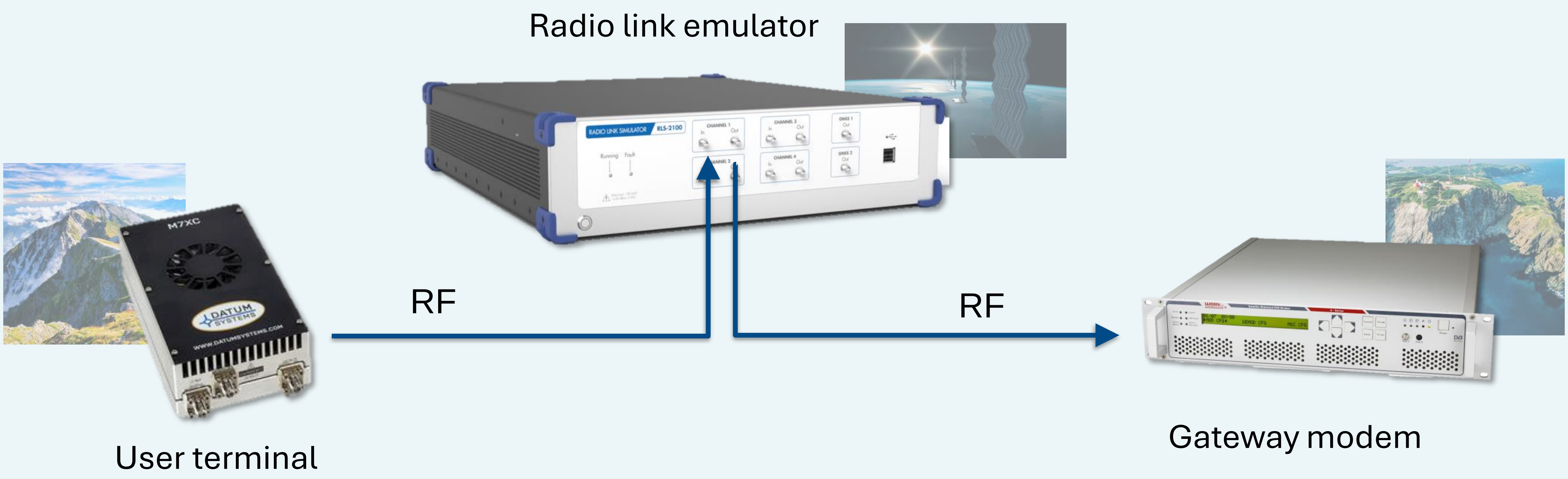
Calculate channel characteristics

```
% Generate ouput profile to RLS-2100
file_header = ["#RLS-2100 Link Parameters; time_into_run", ...
    " delay_s", " doppler_hz", " gain_db"];

file_entries_up = [offsets_s; s_delays_up; fShifts_up; gains_up]';
file_entries_dn = [offsets_s; s_delays_dn; fShifts_dn; gains_dn]';

file_contents_up = [file_header; file_entries_up];
file_contents_dn = [file_header; file_entries_dn];

writematrix(file_contents_up, 'demo1_up.csv');
writematrix(file_contents_dn, 'demo1_dn.csv');
```

Output profile

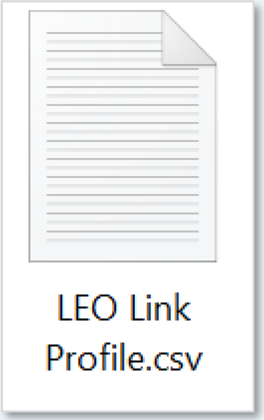
Hardware-in-the-Loop Satellite Link Testing

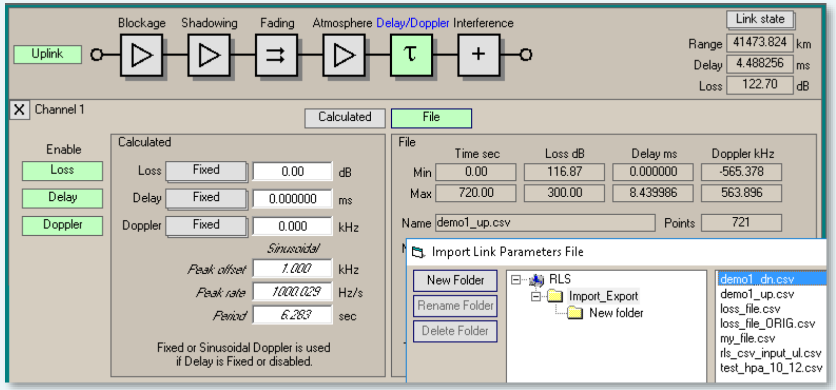
Physical satcom equipment in the lab - connected at RF



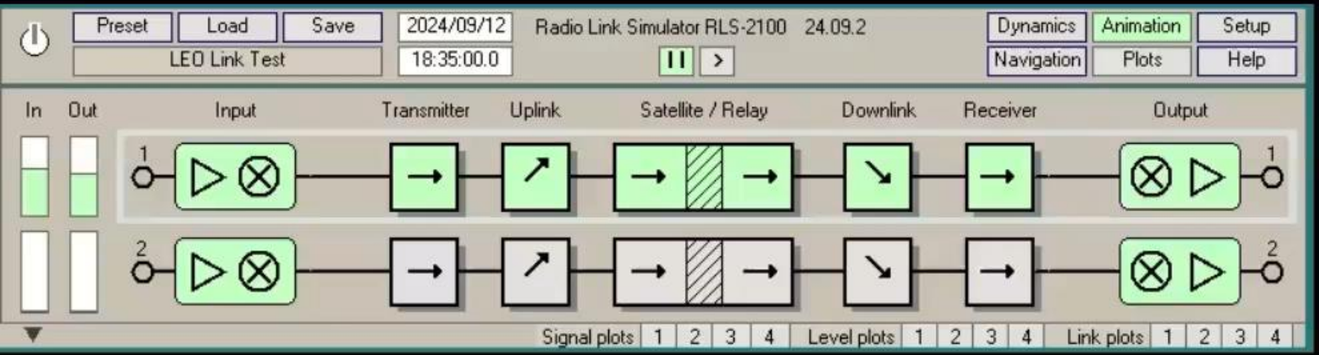
Importing the Profile from the Satellite Communications Toolbox to the RLS-2100

Output from Matlab is compatible with RLS-2100 user profile feature





SQUARE PEG COMMUNICATIONS MATLAB EXPO


Preset
Load
Save
2024/09/12
Radio Link Simulator RLS-2100 24.09.2
Dynamics
Animation
Setup
LEO Link Test
18:35:00.0
Navigation
Plots
Help
In
Out
Input
Transmitter
Uplink
Satellite / Relay
Downlink
Receiver
Output
1
2
Signal plots 1 2 3 4
Level plots 1 2 3 4
Link plots 1 2 3 4


Animation
STARLINK-1019
GS
UE
2024-09-12 18:35:00

Connect with Confidence

Orbit ready applications with

Square Peg Communications'

RLS-2100

and MathWorks' Satellite

Communications Toolbox

SQUARE PEG COMMUNICATIONS MATLAB EXPO

SQUARE PEG COMMUNICATIONS

November 13–14, 2024 | Online

Questions?

Colin Telfer, RLS Product Manager, Square Peg Communications
squarepeg.ca | LinkedIn: @square-peg-communications

MATLAB EXPO

MATLAB EXPO

MathWorks®