

DigiLock 110 MATLAB Control Interface

Overview

This MATLAB package provides complete remote control of the Toptica DigiLock 110 Feedback Controlyzer via TCP/IP connection. It implements all major functionality including:

- **Scan Generation** - Waveform generation for laser scanning
- **PID Controllers** - Two independent PID feedback loops
- **Lock-In Modulation** - For top-of-fringe locking
- **Pound-Drever-Hall (PDH)** - High-frequency modulation for cavity locking
- **AutoLock** - Intelligent automatic locking features
- **Data Acquisition** - Oscilloscope and spectrum analyzer functions

Files Included

1. **DigiLock110.m** - Main class, handles TCP/IP connection
2. **DigiLockScan.m** - Scan waveform generator module
3. **DigiLockPID.m** - PID controller module (for both PID1 and PID2)
4. **DigiLockLockIn.m** - Lock-In modulation/demodulation
5. **DigiLockPDH.m** - Pound-Drever-Hall modulation
6. **DigiLock Helper Modules** - Offset, AutoLock, System, Scope, Spectrum
7. **Examples.m** - Comprehensive usage examples

Installation

1. Download all MATLAB files to a directory
2. Add the directory to your MATLAB path:
`addpath('/path/to/digilock/files');`

Getting Started

Step 1: Find Your DigiLock 110 Connection Info

1. Start the DigiLock Module Server software on your PC
2. Note the **IP Address** and **Port Number** displayed in the interface
3. Typical port number: 5000 (but check your specific setup)

Step 2: Basic Connection

```
% Create DigiLock object  
dl = DigiLock110('192.168.1.100', 5000);
```

```
% Connect  
dl.connect();
```

```
% Check connection  
if dl.isConnected()  
    disp('Connected successfully!');  
end
```

Step 3: Basic Scan

```
% Configure scan parameters  
dl.scan.setType('triangle');  
dl.scan.setFrequency(10); % 10 Hz  
dl.scan.setAmplitude(5.0); % 5V peak-to-peak  
dl.scan.setOutput('SC110');
```

```
% Start scanning  
dl.scan.start();
```

```
% Stop scanning  
dl.scan.stop();
```

Quick Reference

Connection Management

```
dl = DigiLock110(host, port)      % Create object  
dl.connect()                      % Establish connection  
dl.disconnect()                   % Close connection  
dl.reset()                        % Reset to defaults
```

```
status = dl.isConnected()      % Check status
```

Scan Module

```
dl.scan.setType('triangle')    % sine, triangle, square, sawtooth  
dl.scan.setFrequency(10)      % Hz  
dl.scan.setAmplitude(5.0)     % V peak-to-peak  
dl.scan.setOutput('SC110')    % Output channel  
dl.scan.start()              % Start scan  
dl.scan.stop()               % Stop scan
```

PID Controllers (PID1 and PID2)

```
dl.pid1.setInput('MainIn')    % Input channel  
dl.pid1.setOutput('MainOut')  % Output channel  
dl.pid1.setGain(10)           % Overall gain  
dl.pid1.setP(1.0)             % Proportional gain  
dl.pid1.setI(0.5)             % Integral gain  
dl.pid1.setD(0.1)             % Derivative gain  
dl.pid1.setICutoff(100)        % I cutoff freq (PID1 only)  
dl.pid1.setSetpoint(0)         % Setpoint  
dl.pid1.setSign('POS')         % Polarity: POS/NEG  
dl.pid1.setSlope('POS')        % Slope direction  
dl.pid1.setLimits(-5, 5)       % Output limits  
dl.pid1.lock()                % Engage controller  
dl.pid1.unlock()              % Disengage
```

Lock-In Module

```
dl.lockin.setFrequency(100e3)  % 100 kHz  
dl.lockin.setAmplitude(0.01)   % 10 mV pp  
dl.lockin.setOutput('MainOut') % Modulation output  
dl.lockin.setInput('MainIn')   % Signal input  
dl.lockin.setPhase(0)          % Phase (degrees)
```

```
dl.lockin.start()          % Start modulation  
dl.lockin.adjustPhase()    % Auto phase adjust
```

PDH Module

```
dl.pdh.setFrequency(12.5e6)    % 12.5 MHz  
dl.pdh.setAmplitude(0.1)       % 100 mV pp  
dl.pdh.setOutput('MainOut')    % Modulation output  
dl.pdh.setInput('MainIn')      % Signal input  
dl.pdh.setPhase(0)             % Phase (degrees)  
dl.pdh.start()                % Start modulation
```

AutoLock Module

```
dl.autolock.enable(true)        % Enable AutoLock  
dl.autolock.setInput('MainIn')  % Common input  
dl.autolock.setSetpoint(0)      % Setpoint  
dl.autolock.selectControllers(true, true) % PID1, PID2  
dl.autolock.lockToSlope()       % Lock to slope  
dl.autolock.lockToExtremum()    % Lock to peak/valley  
dl.autolock.unlock()           % Release lock
```

System Configuration

```
dl.system.setInputOffset(0)     % Input DC offset  
dl.system.setInputGain(1)       % Input gain  
dl.system.setInvert(false)      % Invert signal  
dl.system.setLowPassFilter(1e6, 2) % LP filter: freq, order  
dl.system.setHighPassFilter(10, 1) % HP filter: freq, order
```

Data Acquisition

```
data = dl.scope.acquire(1, 1000)  % Channel 1, 1000 points  
dl.scope.setChannel(1, 'MainIn')  % Set channel source  
dl.scope.setTimebase(0.01)        % 10 ms/div
```

```
spectrum = dl.spectrum.acquire(1) % Get FFT data  
dl.spectrum.setSpan(1e6) % 1 MHz span
```

Common Locking Scenarios

Scenario 1: Side-of-Fringe Lock

Lock to the slope of an absorption line (e.g., Doppler-free spectroscopy)

```
% Configure PIDs
```

```
dl.pid1.setInput('MainIn');  
dl.pid1.setOutput('MainOut'); % Fast: current modulation  
dl.pid2.setOutput('SC110'); # Slow: piezo  
% Set gains, then lock  
dl.pid2.lock();  
dl.pid1.lock();
```

Scenario 2: Top-of-Fringe Lock (Lock-In)

Lock to peak using frequency modulation

```
% Setup Lock-In
```

```
dl.lockin.configure('Frequency', 100e3, ...  
'Amplitude', 0.01, ...  
'Start', true);
```

```
% Adjust phase, then lock
```

```
dl.autolock.setInput('LIOut');  
dl.autolock.lockToExtremum();
```

Scenario 3: Cavity Lock (PDH)

Lock laser to Fabry-Perot cavity

```
% Setup PDH
```

```
dl.pdh.configure('Frequency', 12.5e6, ...  
'Amplitude', 0.1, ...  
'Start', true);
```

```
% Lock to cavity resonance  
dl.autolock.setInput('PDHOut');  
dl.autolock.lockToExtremum();
```

Important Notes

Command Protocol

The actual TCP/IP command set depends on the DigiLock 110 firmware. This implementation uses a **SCPI-like command structure**:

- Commands: MODULE:PARAMETER value
- Queries: MODULE:PARAMETER?

You may need to adapt commands based on the actual RCI protocol. Refer to:

- The separate "DigiLock 110 RCI Manual" (download from Toptica website)
- Section 6 of the main manual (pages 13-14)
- Port numbers displayed in the DigiLock Module Server

Connection Tips

- Use 'Verbose', true during development to see commands
- Check firewall settings if connection fails
- Verify DigiLock Module Server is running
- Multiple DigiLock units can be controlled by specifying different ports

Best Practices

1. **Always disconnect cleanly:** Use dl.disconnect() or clear dl
2. **Test scan first:** Verify signal before attempting lock
3. **Start conservative:** Use lower gains initially
4. **Monitor stability:** Use scope/spectrum to verify lock quality
5. **Document settings:** Save working parameters for reproducibility

Troubleshooting

Cannot Connect

- Verify IP address and port number in DigiLock Module Server

- Check network connectivity: ping 192.168.1.100
- Ensure DigiLock software is running
- Try connecting with Toptica software first to verify hardware

Commands Not Working

- Enable verbose mode: dl = DigiLock110(host, port, 'Verbose', true)
- Check command responses for errors
- Consult the DigiLock 110 RCI Manual for actual command syntax
- Some features may require specific firmware versions

Lock Unstable

- Reduce PID gains
- Check error signal amplitude and slope
- Verify correct polarity (sign) settings
- Use spectrum analyzer to check for oscillations
- Ensure no saturation of inputs/outputs

Additional Resources

- **Main Manual:** toptica_digilock_manual.pdf (See sections 6-10)
- **RCI Manual:** Available from Toptica website
- **Application Examples:** See section 10 of main manual
- **Toptica Support:** service@toptica.com

License and Disclaimer

This is an unofficial implementation created for convenience. Always refer to official Toptica documentation for accurate specifications and supported features.

The implementation assumes SCPI-like command structure - actual commands may vary depending on firmware version. Test thoroughly and adapt as needed for your specific setup.

Version History

- **v1.0** - Initial implementation with all major modules
- Based on DigiLock 110 Manual M-031 Version 11 (February 2023)