A modular and turn-key remote-access hardware-in-the-loop platform for testing electric motors

Michael-Anthony TEDESCO* and M. Reza EMAMI*

*Aerospace Mechatronics Group
University of Toronto Institute for Aerospace Studies
4925 Dufferin Street
Toronto, Ontario, M3H 5T6, Canada
E-mail: emami@utias.utoronto.ca

Received 14 May 2013

Abstract
This paper details the architecture, hardware and software of a platform for testing electric motors over the Internet that enables users to test multiple physical motors remotely under the specific loading conditions of the real-world application for which a motor is required. The system is divided into three major modules: the Server Software Application, the Target Software Application and the Motor Test Platform. The system is unique as it combines modularity, scalability and deliverability. It is modular, as it is capable of readily testing a variety of electric motors, scalable, as new motors can be quickly added to the system for testing, and is turn-key, easily deployed and installed. The proof-of-concept prototype was developed and examined against benchmark tests to determine its capabilities. The platform was effective as a remote access emulation and evaluation tool.

Keywords: Remote experimentation, Hardware-in-the-loop platform, Motor testing, Load emulation

1. Introduction

Remote experimentation and testing is an emerging technology in both the academia and industry. There have been several remote access experimental setups developed in universities around the globe, including Massachusetts Institute of Technology (del Alamo, et al., 2005), University of California, San Diego (Travelyan, 2004), University of Essex, UK (Hu, et al., 2001), and University of Toronto (Helander and Emami, 2008). The University of Western Australia has made its remotely-operated robotic manipulator open to the public, i.e., anyone is free to register (Elgamal, et al., 2005). In the industry, despite a number of applications, remote access technology is still not fully utilized. Current applications are focused around rapid prototyping (Lan, 2009)(Luo, et al, 2001)(Luo and Tzou, 2004) and manufacturing (Lan, 2009)(Hanwu and Yueming, 2009)(Zheng, et al, 2008). In Luo, et al., (2001) and Luo and Tzou, (2004) live video streaming of the system is included to monitor the progress. Few works appear to harness remote access for industrial testing and design purposes. Ericsson’s Virtual Lab is one such remote access testing system applied to software for the development and testing of Java applets for use on its cell phones prior to the release of the phone (Sony Ericson, 2007).

This paper presents the design of a modular and turn-key hardware-in-the-loop platform for remote-testing electric motors. The torque-speed characteristics are crucial factors when selecting an electric motor for a specific application (Poulin, 1984). In both open-loop and closed-loop (servo) applications, the loading condition on a motor critically affects its dynamic response, efficiency, and power consumption, and the effect varies widely at different speeds. Current performance charts available to customers to assist them in finding a suitable selection are based on limited benchmark testing. Hence, not every possible load profile is actually tested to generate the performance charts. Instead, certain loads are applied to the motor to determine the corresponding performance, and the results of the benchmarking are fitted to generate the torque-speed characteristic curves. Consequent to these restrictions, the existing
motor datasheets, albeit useful to a certain extent, do not provide complete information about the efficiency, repeatability and variability of the motor performance under various load systems. Users often require a more precise reading for the motor performance under the specific conditions it will operate at, as well as the motor’s transient dynamics at start-up or response to dynamic loads, in order to select the most suitable motor for the application. Therefore, there is a need for a system that allows electric motor manufacturers and distributors to offer their customers the ability to test available motors prior to purchase, under the specific dynamic load profiles the motor will be subjected to in its real-world application.

Dynamic load test systems are found mostly in the academic literature and rarely in industrial applications. Although dynamometers have been used first to evaluate combustion engines and more recently to evaluate electric motors (Newton, et al, 1995), they are heavy and bulky and thus unable to emulate fast dynamics. Consequently, they are usually used for large-size motors with static or quasi-static loads and in situations where the high costs of their acquisition, installation, and maintenance justify the extra expense. Several attempts have been made to emulate nonlinear rapidly-changing load profiles using dynamometers through rather complicated feedback control algorithms (Akpolat, et al, 1999). A Hardware-in-the-Loop (HIL) simulation using a dynamometer to model a wind turbine generator through emulating the torque generated by the wind is presented in Li, et al (2006). The development of a Robotic Hardware-in-the-Loop simulation for dynamic load emulation of robot manipulators is detailed in Martin and Emami (2011). The platform simulates a standard 5 degree of freedom industrial robot, with torque motors providing the load on the joint motors of the robot according to torque calculated by a computer simulation. The computer simulation is based on the kinematics and dynamics of the manipulator. It is shown that the robotic HIL approach more closely approximates the robot than a purely computer-based simulation.

There have been several works that bridge the two fields of remote access and hardware-in-the-loop simulation. The University of Michigan and TARDEC connected two distinct HIL systems via the Internet to evaluate automotive powertrain systems (Ersal, et al, 2011). The University of Alaska Fairbanks developed a HIL simulation of a PUMA 560 that was remotely accessible, enabling the remote development of control algorithms (Singaraju, et al, 2006).

None of the existing systems described in the literature are modular, scalable or turn-key. In each of these systems, the HIL platform and remote access software are specific to the implementation, and therefore, not modular. While some of the existing architectures are described as modular, their physical setups are specific to the motors being tested. Furthermore, they are permanent installations, requiring significant set-up time, and a dedicated load module for each motor to be tested. Regarding scalability, the existing literature describes architectures for testing a single motor(Newton, et al, 1995) (Akpolat, et al, 1999) (Li, et al 2006), or multiple motors on fixed stands (Martin and Emami, 2008, 2011) (Singaraju, et al, 2006), where the motors are operating together to simulate the joints of a serial robot, but do not discuss adding new motors, and again, each motor being tested requires its own load module. The architectures described may be scaled to include testing multiple motors as in Martin and Emami (2008, 2011) and Singaraju, et al (2006); however, adding new motors to the system is non-trivial. Finally, deliverability is not discussed in the literature as all are custom experiment to demonstrate the developed concept, and it is further limited as they mostly utilize costly third-party software, such as Matlab/Simulink as in Martin and Emami (2008, 2011) and Singaraju, et al (2006).

The HIL platform presented in this paper allows remote users to apply custom dynamic load profiles based on their own applications, while observing the motor performance on-line from anywhere and at any time. The platform is highly modular; a variety of electric motors within the testing specifications of the platform can easily be added, and the system can be readily scaled up. It is also a turn-key solution. The novelty and originality of the work come from the design of the remote-access platform as a modular, scalable and turn-key solution. The paper outlines the architecture of the new platform, and details the subsystems that enable the platform to meet its multiple objectives. It then describes various experiments that were designed to validate system performance. Next, the results of the experiments are discussed, and some concluding remarks are made in the end.
Nomenclature

\( \tau \): Available output torque of test motor; 
\( \tau_c \): Load motor torque; 
\( \tau_{wm} \): Load motor torque between stator and rotor; 
\( b \): Viscous friction coefficient; 
\( b_c \): Viscous friction coefficient for load motor; 
\( J \): Inertia of load motor; 
\( J_{shaft} \): Polar moment of inertia of load motor shaft 
\( L \): Load motor shaft length 
\( \Phi_{shaft} \): Load motor shaft twist angle 
\( G \): Shear modulus of shaft material 
\( \dot{\theta} \): Angular speed of test motor; 
\( \ddot{\theta} \): Angular acceleration of test motor; 
\( I \): Motor current; 
\( K_T \): Motor torque constant 
\( \Phi \): Total motor flux 
\( B \): Motor flux density 
\( \phi \): Motor phase angle 
\( F \): Prony brake belt tension force; 
\( r \): Radius of prony brake hub; 
\( \sigma \): Torque measurement standard deviation 

DAQ: Data Acquisition; 
GUI: Graphic User Interface; 
HIL: Hardware-in-the-loop;

2. Platform Architecture

The system is divided into three principle subsystems as shown in Fig. 1, namely Server Software Application for enabling remote access and maintaining user accounts, Target Software Application for executing and controlling the test operations, and Motor Test Platform for positioning and coupling the test motors to the load module and applying loads. In order to satisfy the multiple goals of modularity, scalability and deliverability, a number of technical and academic challenges needed to be overcome in each of the three principal subsystems. The motor test platform uses
a load module to emulate the custom load profiles. The server software application enables users to create user accounts and connect to the experiment via the internet, allowing the platform to be remotely accessible. It also controls user access by means of a scheduler, which modifies user’s permission to grant or block access to the experiment. The target software application provides a graphical user interface (GUI) to facilitate the user’s interaction with the motor test platform. The target software application sends the user’s commands to the platform, and records the output generated by the experiment. The entire software solution is bundled for easy deployment from an installation package. This, along with the modularity of physical platform, allows for rapid and easy deployment of new setups, adding new test motors and scaling up the system as required.

2.1 Server Software Application

The Server Software Application is the collection of libraries, webpages and utilities, shown in Fig. 1, that sit on a central administrative server to enable and control remote access to the target application. The structure builds off the work presented by Helander and Emami (2008). The system relies on Windows Server 2008, Active Directory and the .Net framework for creating and modifying user profiles as well as authentication. The system in Helander and Emami (2008), however, was a modular architecture with a specific custom implementation and pre-populated users accounts created beforehand by an administrator. Furthermore, the system in Helander and Emami (2008) is not able to be readily carried over to another domain. In order to make the system a turn-key solution, prospective users need to have the ability to remotely create their own accounts and profiles, and for the system to be deployed on any domain. The module developed in this paper improves on the existing system by removing all installation-specific information from the software, such as computer name, IP addresses, etc., and creating an XML database that is populated with such information automatically on the installation. The new module also contains a web interface that is capable of creating new user accounts on demand. The Account Manager is the collection of webpages that allows new users to register and authenticated users to log onto their personal profile page. The result is a software package that is fully ready to be deployed, from a single installation disc, on any server computer running Windows Server and Terminal Services.

The Scheduler consists of webpages for viewing and booking sessions, the Schedule Library, the Schedule Service, the Schedule Database and the Scheduler utility. At their scheduled time, the Scheduler modifies the user’s privileges in Active Directory to enable access to target PC. The user navigates to the Test Page. This Active Server Page contains frames for a video feed of the test setup, as well as a remote desktop frame to log onto the Target PC. On start-up, the remote desktop session launches the Testing GUI, shown in Fig. 2.

The Management Console utility provides means for the administrator to setup and maintain the remote access software including: users, scheduled sessions, availability of various experiments, as well as add testing platforms. The Data.mdb central database is a Microsoft Access file containing the Schedule Database and Resource Database.

Fig. 2 Graphic User Interface for a) motor selection and test parameters, b) test results
2.2 Target Software Application

Figure 3 shows the Target Application Architecture, which enables the system to be both modular and turn-key, qualities that are missing in the current state-of-the-art brought in the literature (Martin and Emami, 2008, 2011) (Ersal, et al, 2011) (Singaraju, et al, 2006). This architecture is composed of three interconnected subsystems: Target PC, Hardware-in-the-loop Platform and Manufacture Supplied Test Motors, and is the interface between the virtual Server Software Application and the physical Motor Test Platform hardware. The system utilizes some of the techniques described by Helander and Emami (2008) for handling remote access to multiple experiments.

The Target application enables the OEM motor manufacturers to connect their own test motors to the Target PC (via any communication link) and launch their own motor control software automatically when the platform connects to a given motor. This control software is then accessible by the remote user. While only a single test motor is shown in Fig. 3, multiple motors may be connected to the Target PC, limited only by the number of physical connections available. The architecture combines the existing modules (Terminal Services, OEM Motor Manufacturer Software) and hardware (Data Acquisition Board, motors, sensors and controller) with custom modules, such as Target Application, Motor Database, Load Module.

The target PC is the local machine that runs the target software application and interfaces with the motor test platform through a data acquisition (DAQ) board. It also hosts Terminal Services that provide remote desktop connection to the scheduled user, via the Server Software Application described in 2.1.

The user gains access to the experimental setup controls by logging onto the experiment via the Remote Desktop frame on the ASP. On login, the target application launches and presents the user with the Testing GUI. The GUI allows the user to browse the available test motors and select one for testing. Once a motor is selected and ready for testing, the user defines the test parameters; this can be a constant load, a generated torque function, or the user can upload a torque profile and specify the profile sampling rate. The target application accesses the motor testing Dynamic Linked Library, which contains the C# classes and methods for interpreting the user input and sending commands to the test platform via the PCI-2517 DAQ board from Measurement Computing Corp. The target application also collects relevant test information regarding the applied torque and speed via the DAQ board. These experimental results as well as information on position and acceleration calculated using the data are shown on the Results tab and can be saved to the user’s profile for future analysis.

The test motors are controlled via the motor manufacturers’ own motor controller and software, and therefore are not dependant on the test platform software. This is critical to the modularity of the platform. The method of connecting the driver to the computer is not important; connection can be done through USB, Serial, RS232 or even a DAQ board. The only requirement is that there is an executable for the motor controller installed on the computer that
can be called by target application to operate the motor, and that the location of the executable is correctly stored in the Motor Database. The executable is launched when the user selects the motor for testing and the load module is coupled to the test motor.

2.3 Motor Test Platform

To enable remote operation of the physical test platform while maintaining modularity, numerous technical problems needed to be solved. The motor test platform, as illustrated in Fig. 4, is the collection of hardware components that enable performing load emulation and hardware-in-the-loop simulation on a variety of test motors. The Test Motor Platform uses a load module similar to the one described by Martin and Emami (2008, 2011). However, that system was for a fixed, dedicated, load module/test motor configuration for each test motor. Two new subsystems, the Positioning System (2.3.3) and the Automated Coupling (2.3.4), were developed to enable a single load module to engage with and test multiple motors, while maintaining the load module requirements of rigidity and zero-backlash. A simple but effective generic fixture (described in 2.3.2) was designed that allows for a range of new test motors to be easily added to the platform with their motor-specific stand.

![Motor Test Platform](image)

Fig. 4 Motor Test Platform

2.3.1 Load Module

The heart of the motor test platform is the Load Module, which utilizes a torque motor to apply dynamic loads to the test motors. A tachometer and a reaction torque sensor provide information on the position, speed and acceleration of the motor. Ideally, a load emulator should possess intrinsic fast dynamics, and have as little impact on system performance as possible without introducing any further complexities to the system dynamics. It should also have low inertia and be able to generate sufficient torque even at low speeds. To achieve this, the developed platform utilizes a torque (or ring) motor, which satisfies the above-mentioned requirements and has the additional benefits of low power consumption and good heat dissipation. The PSR200 rotary motor from IntelLiDrives Inc. was used for the development of the proof-of-concept prototype, enabling the module to emulate loads up to 45 Nm peak and 17 Nm continuous, with speeds up to 500 rpm. The motor was controlled using the Xenus XSL 230 from Copley Controls. Further, an innovative torque sensing technique was used in this work that requires no complex modeling or indirect calculations. Instead of mounting a rotary torque sensor on the shaft, a reaction torque sensor, the 147 Nm TFF350 from Futek Advanced Sensor Technology Inc. is installed between the load motor case (stator) and its mounting fixture. Consequently, the measured signal is directly proportional to the load motor torque, not the coupled torque between the two motors. By decoupling the torque at the measurement stage, there is no need for approximation or estimation techniques involving actuator parameters to track the load motor torque. The speed is measured by the Series 88 tachometer from Hohner Corp mounted to the output shaft of the load module. The tachometer can measure speeds up
to 300 rpm. The load module is therefore capable of emulating continuous loads up to 17 Nm and peak loads up to 45 Nm at speeds up to 300 rpm.

![Free Body Diagram of Load Emulation](modified from Martin and Emami (2008))

The principle of load emulation used here are based on those presented by Martin and Emami (2008). Figure 5 shows a free body diagram of the load module and test motor. The torque of interest is \( \tau \): the torque applied to the test motor by the torque motor for the load module. This torque is transmitted through the shaft. The shaft twist angle \( \phi_{shaft} \) can be calculated using

\[
\phi_{shaft} = \frac{\tau_p L}{J_{shaft} G}
\]  

(1)

Where \( \tau_p \) is the peak observed torque, \( L \) is the shaft length, \( J_{shaft} \) is the polar moment of inertia of the shaft and \( G \) is the shear modulus of the shaft material, in this case ANSI 1020 steel. The maximum twist angle for the peak observed torque is just 0.035°. The rigid body assumption is consistent with Akpolat, et al (1999) and Martin and Emami (2008). Having such an assumption, the dynamic equation for the test motor can be written as (Toliyat and Kliman, 2004):

\[
\tau = J\ddot{\theta} + b\dot{\theta} + \tau_l
\]  

(2)

where: \( \tau \) is the net torque generated by the test motor after internal friction, inertia and the efficiency of any transmission system are taken accounted for, \( J \) is the inertia of the torque motor, flange and the coupling hubs, \( b \) is the viscous friction coefficient for the load module, \( \tau_l \) is the torque generated by the torque motor, \( \dot{\theta} \) is the angular acceleration of the output shaft, and \( \dot{\theta} \) is the angular speed of the output shaft. Since we are interested in emulating the external torque applied to the test motor, there is no need to take into account any of the losses internal to the test motor.

For a DC motor, such as the one used for the load module, the motor torque is a product of a constant \( K_F \), the rotor current \( I \) and the total flux \( \Phi \) (Hughes and Drury, 2013):

\[
\tau_l = K_F \cdot \Phi \cdot I
\]  

(3)

In the majority of cases of a DC motor, the motor flux is constant. Thus, the relationship simplifies and the torque becomes proportional to the applied current(Hughes and Drury, 2013):

\[
\tau_l = K_F \cdot I
\]  

(4)

If an induction motor is used for the load module, the motor torque is a product of the motor constant \( K_F \), the flux density \( B \), the rotor current \( I \) and the phase angle \( \phi \) (Hughes and Drury, 2013):
\[ \tau_i = K_T \cdot B \cdot I \cdot \cos \phi \]  

(5)

The reaction torque sensor mounted behind the torque motor measures \( \tau_m \): the torque applied between the stator and the rotor. This is equal to the torque applied by the load motor, plus the friction between the stator and rotor:

\[ \tau_m = \tau_i + b_i \dot{\theta} \]  

(6)

where: \( b_i \) is the viscous friction coefficient for the torque motor. The viscous friction coefficient \( b \) in equation has two components: friction from the mounted bearing and friction between the stator and rotor. Since typical coefficients of frictions for self-aligning ball bearing units is 0.001 (Norton, 2006), for design purposes the friction from the bearing can be assumed negligible. Re-arranging Eqs. (2) and (6) will result in the equation for the torque applied by the torque motor as:

\[ \tau_i = \tau - f \dot{\theta} - b_i \dot{\theta} \]  

(7)

The torque applied to the test motor becomes:

\[ \tau = f \dot{\theta} + \tau_m \]  

(8)

### 2.3.2 Test Motor

The Test Motor is the collection of hardware components and peripherals from the motors being tested. Each test motor is mounted onto its own motor bracket, which in turn is mounted to a modular, adjustable positioning fixture. The fixtures can be seen in Figs. 4 & 7 with three identical fixtures supporting three different motors. The modularity and configurability of the motor positioning fixture ensures that the platform does not depend on the size or number of motors to be tested.

### 2.3.3 Load Module Positioning System

The Load Module Positioning System is the hardware mechanism that enables the load module to connect to different test motors. The forward/reverse motion engages/disengages the load module from the test motors, while the lateral motion enables the load module to move to the different test motors. The Positioning System works in conjunction with the Test Motor database to track the current position of the load module and the fixed position of installed test motors. It is the design of both the Positioning system and Test Motor database that ensures platform scalability, where new test motors can be quickly and easily added.

### 2.3.4 Automated Coupling

The Automated Coupling shown in Fig. 6 (a) is the hardware mechanism for engaging the load module with the selected test motor and transmitting torque through their respective shafts. The mechanism is based on the jaw-type coupling. Zero flexibility is ensured by eliminating the elastomer insert. Backlash is eliminated by the tapered teeth, which ensure full contact on both sides when the shafts are pressed together. Shaft concentricity is enforced using a male/female conical locator pair built into the hubs to correct any fine lateral misalignments that remain after initial load module positioning.
Because the test motor may not have encoder feedback, a unique approach was developed to orient the couplings for engagement. The initial orientation is achieved by measuring torque fluctuations caused by magnets located on the outside of the hubs as shown in Fig. 6 (b). By monitoring the fluctuation using the reaction torque sensor, it is possible to know the relative position of the hubs and align them for engagement.

3. Platform Validation

3.1 Benchmark Tests

A number of tests were carried out to ensure that the results achieved using the load module correspond to the performance of the motor under the real-world conditions. This consisted of taking the results from the benchmark tests for torque and speed and recreating them with the load module using the same test motor and driver. The tests included: i) applying a series of constant braking torque to the test motor operating at steady-state, ii) accelerating a variety of inertial load at motor start-up, and iii) accelerating a combination of an inertial load and braking torque at motor start-up. Figure 7 shows a photo of the prototype platform.

The first test consisted of applying a constant braking torque on the motor, evaluating the corresponding steady state speed and repeating for a range of braking loads to generate the torque-speed curve for the motor. The benchmark braking torque was applied using a prony brake, consisting of a tensioned belt wrapped around a hub. The belt tension meter only had a digital display output. The differential tension in the belt was observed and recorded manually. The torque applied to the hub was calculated as:
\[ \tau = (F_1 - F_2) \cdot r \]  

where \( \tau \) is the torque applied to hub in the opposite direction of rotation, \( F_1 \) is the higher of the two tensions, \( F_2 \) is the lower of the two tensions, and \( r \) is the distance from center of hub. For each test condition, the motor was allowed to reach steady state before recording. The tachometer signal, the current monitor and speed monitor from the motor amplifier were sampled at 2000 Hz. These load conditions were emulated on the test platform. The test motor ran with the load module idle for 45 seconds to allow the test motor to reach no-load steady-state speed. The load module then applied the braking torque for 45s.

The second test consisted of recording the speed from the tachometer as the test motor accelerated a known inertial load. For each test condition the motor was started and accelerated to its steady-state speed while recording the tachometer speed. The torque applied by the motor on the inertial load was calculated as:

\[ \tau = J \ddot{\theta} \]

where \( \tau \) is the applied torque, \( J \) is the inertia and \( \ddot{\theta} \) is the angular acceleration. The inertia \( J \) is the sum of the inertia values for all of the components not present during emulation test. The inertia internal to the test motor, such as the motor shaft and the transmission system, are not included in the calculation of \( J \), nor are any friction losses internal to the test motor. For each test case, the tests were repeated 6 times, and the speed time-series was an average of those six.

The speed time-series was filtered using a first-order Butterworth filter with a cutoff frequency of 60 Hz. to reduce the noise. The acceleration time-series was calculated by numerically differentiating the filtered speed time-series. The numerical differentiation amplified the noise in the speed signal. Therefore, the acceleration was also filtered using the same first order Butterworth filter. The main output from benchmark test was the torque profile, a text file of the torques that the test motor applied to the inertial load, with a sampling rate of 2000 Hz.

The final benchmark test involved the evaluation of the start-up characteristics of a combined inertial load and braking torque. The tests were carried out in the same manner as for the inertia tests described previously. The torque resulting from the acceleration of the inertial load was calculated using Eq. (10). Because of the limitations with the belt force sensors, and the qualitative nature of recording the force, it was not practical to get the instantaneous braking force applied by the belt directly. Since the belt tension is measured beforehand and the speed can be recorded directly, the applied friction force was estimated as a function of speed and belt tension. The estimated braking force was added to the inertial torque to generate the torque profile.

3.2 Experimental Results and Verifications

Figure 8 presents the average steady-state speed vs. mean braking torque for a series of different braking torque. The graph compares the torque from the benchmark test to the torque from two different emulated cases: i) the load module manually coupled to the test motor, ii) the load module automatically positioned and coupled. The error bars represent the 2\( \sigma \) (95\%) confidence interval for the 45-second emulated loads series. For example, the standard deviation \( \sigma \) was 0.180 N\( \cdot \)m, for the manually coupled -10.6 N\( \cdot \)m emulated braking load at 1.48 rad/s, giving a 95\% confidence interval of \( \pm 0.36 \) N\( \cdot \)m. Because the torque in the benchmark test was derived from the displayed differential between the observed tension measurements, the error for the benchmark braking load is the maximum and minimum observed values. The largest observed deviation was \( \pm 0.5 \) N\( \cdot \)m.

The results from steady-state torque-speed for the emulated torque agree with those obtained in the benchmark tests for torque in the linear operating region of the test motor. The figure shows the overlapping performance of the test motor. However, while the general motor characteristic is the same, the applied torques for the emulated tests deviate from those for the benchmark tests. The values from the benchmark test were used as the desired torque for the emulations test and, ideally, should match. The difference is likely due to the error in measuring the force/torque in the benchmark tests. Once the motor begins operating in the non-linear region, the results for the emulation deviate from those obtained during the benchmark testing. For the benchmark test, the non-linear region was not evaluated, as it proved impossible to operate the motor at a steady speed in current mode with such high torques using the test setup. The figure shows test motor operating in the non-linear region for the higher torques with the platform, while the torque from the benchmark test still have the motor operating in the linear range. There are two possible explanations
for this behaviour: i) the load module may be applying a significantly higher torque than the torque sensor is measuring, ii) the benchmark test may be overestimating the torque applied by the braking belt for higher belt tensions. It should be noted that while the results deviate with regards to speed, the test motor is very sensitive to changes in applied torque when operating in this region.

![Steady State Torque-Speed Curve](image)

Fig. 8 Steady-state torque-speed characteristics

Figures 9 and 10 present the results for start-up inertia and combined inertia and breaking. The first plot in each figure compares the filtered torque profile from the benchmark test to the filtered torque profile from three different emulated cases: i) the load module manually coupled to the test motor, ii) the load module positioned using the load module positioning system and coupled using the Automated Coupling System, and iii) the load module positioned manually in line with the test motor but coupled using the automated coupling system. The torque profile from the benchmark tests was used as the commanded torque for the emulation tests. The second plot compares the speed from the benchmark test to the speed from the emulated cases.
The results for the emulated start-up generally agree with those from the benchmark tests. The applied torque profiles closely follow the commanded torque profiles during the acceleration stage. In the steady state region, i.e., no acceleration, the torque from the commanded torque for the inertial load test has a mean close to zero, while the combined test has a mean corresponding to the braking load. The applied torque, however, sometimes has a slight bias. It is proposed that the introduction of the torque sensor into the control loop may improve the results. In addition, the applied torque signal in both the inertia only and combined inertia-breaking load oscillates significantly more than the commanded signal during steady state and is likely caused by the discontinuous nature of the torque motor, an effect known as torque ripple (Carlson, et al, 1992). A statistical analysis of the steady state region shows that the standard deviation for the applied torque is 0.224 N\(\cdot\)m. This result agrees with the error bars in figure 8, where the 2\(\sigma\) (95%) confidence interval was ±0.36 N\(\cdot\)m for emulated loads and ±0.5 N\(\cdot\)m for the benchmark loads.

The transient and steady state speed also largely agree with the results from the steady state tests. There were, however, few occasions where the emulated results did not display the smooth acceleration of the benchmark test. Instead, the test motor initially accelerated faster during the emulated inertia test than during the benchmark test, decelerating suddenly, as seen in Fig. 10 (b) for the combination brake-inertia tests. This behaviour is strongest for loads with high constant braking torque. This phenomenon is likely caused by the torque offset introduced to compensate for the torque added by the friction when backdriving the load Module. The mean steady state velocities were very close for both the emulated and benchmark tests, though the pure inertia start-up contains a speed oscillation that is not present in the combination load tests.
Fig. 10 Start-up characteristics for inertial-breaking load: a) torque, b) speed

3.3 Effects from Positioning and Automated Coupling

Comparing the results from the three different emulation cases (manual coupling, automatic positioning and coupling, and manual positioning automatic coupling) in Figs. 9 and 10 reveals that the automatic positioning impacts the emulation. For the start-up inertia tests, as seen in the speed time-series in Fig. 9 (b) the rise time is longer than for the manually coupled emulation, and more erratic. The effect is less dramatic for the combination braking torque and inertial load in Fig. 10. However, the torque time-series shown in Fig. 9 (a) for the start-up tests clearly show that the load module is correctly applying the torque profiles for both manual and automated positioning and coupling. Likewise, for the steady state braking torques shown in Fig. 8, the load module is applying very nearly the same mean torque, but the resulting speed for the test motor is reduced. Careful observation of the entire positioning and coupling process, the likely culprit is the mechanism for correcting lateral misalignment. Forces transmitted from the test motor to the load module via the shaft of the load module may result in residual stresses in the shaft and can cause the circular spline of the harmonic drive to become slightly angled relative to its flexspline. Even such a small angular misalignment of the circular spline relative to the flexspline can lead to a condition known as *dedoidal*, whereby there
is only one deflection of the flexspline per rotation instead of two equal deflections (Zalud, 2008), resulting in high friction loads (Tischler, 2000).

4. Conclusion

A modular architecture for remote evaluation of physical electric motors was developed and implemented, and the potential for hardware-in-the-loop simulation as a means of evaluating motors was illustrated. A review of current literature in the field showed that remote testing for academic purposes is growing, but that little has been done to apply remote testing to the evaluation of commercial products.

The basic structure of the architecture was outlined, presenting the hardware design and implementation of the Motor Test platform, as well as the structure of the software for remotely accessing and operating the platform. The architecture was applied to the emulation of benchmark tests for a particular test motor. A series of tests were carried out to evaluate the basic functionality of the platform – its ability to emulate start-up inertial loads, steady state braking torque and start-up of a combined inertial load and braking torque. The validation of the Motor Test Platform as a proof-of-concept prototype was largely successful, with the platform being able to repeatedly move to and connect to different test motors. Some shortcomings with the current design were identified; in particular the current means of correcting lateral misalignments affected the ability of the system to accurately emulate loads. A system patent for the design has been granted by the United States Patent and Trademark Office (Emami and Tedesco, 2013).

References

Akpolat, Z.H., Asher, G.M. and Clare, J.C., Dynamic emulation of mechanical loads using a vector controlled induction motor-generator set, IEEE Transactions on Industrial Electronics, Vol.46, No.2 (1999), pp.370-379.
Carlson, R., Lajoie-Manzec, M. and Fagundes, J., Analysis of torque ripple due to phase commutation in brushless DC machines, IEEE Transactions on Industrial Applications, Vol.28, No.3 (1992), pp.632-638.
del Alamo, J.A., Bailey, P., Harward, J., Hardison, J., Lerman, S.R. and Long, P.S., The iLab Architecture: Towards a community of Internet accessible laboratories, Sloan-C International Conference on Asynchronous Learning Networks (2005).
Elgamal, A., Fraser, M. and Zonta, D., Webshaker: Live Internet shaker-table experiment for education and research, Computer Applications in Engineering Education, Vol.13, No.1 (2005), pp.99-110.
Emami, M.R. and Tedesco, M.A., System, method and computer program for remotely testing system components over a network, United States patent disclosure 8447554 (2013).
Ersal, T., Brudnak, M., Salvi, A., Stein, J., Filipi, Z. and Fathy, H., Development and model-based transparency analysis of an Internet-distributed hardware-in-the-loop simulation platform, Mechatronics, Vol.21, No.1 (2011), pp.22-29.
Hanwu, H. and Yueming, W., Web-based virtual operating of CNC milling machine tools, Computers in Industry, Vol.60, No.9 (2009), pp.686-697.
Helander, M.G. and Emami, M.R., Engineering eLaboratories: Integration of remote access and eCollaboration, International Journal of Engineering Education, Vol.24, No.3 (2008), pp.466-479.
Hu, H., Yu, L., Tsui, P.W. and Zhou, Q., Internet-based robotic system for teleoperation, Assembly Automation, Vol.21, No.2 (2001), pp.143-151.
Hughes, A. and Drury, B., Electric Motors and Drives, 4th edn (2013), pp.77, 82, 161, Newnes.
Lan, H., Web-based rapid prototyping and manufacturing systems: A review, Computers in Industry, Vol.60, No.9 (2009), pp.643-656.
Li, H., Steuer, M., Shi, K.L., Woodruff, S and Zhang, D, Development of a unified design, test, and research platform for wind energy systems based on hardware-in-the-loop real-time simulation, IEEE Transactions on Industrial Electronics, Vol.53, No.4 (2006), pp.1144-1151.
Luo, R., Tzou, J.H. and Chang, Y., Desktop rapid prototyping system with supervisory control and monitoring through Internet, IEEE/ASME Transactions on Mechatronics, Vol.6, No.4 (2001), pp.399-409.
Luo, R. and Tzou, J.H., The development of an intelligent web-based rapid prototyping manufacturing system, IEEE Transactions on Automation Science and Engineering, Vol.1, No.1 (2004), 4-13.
Martin, A. and Emami, M.R., Dynamic load emulation in hardware-in-the-loop simulation of robot manipulators, IEEE
Transactions on Industrial Electronics, Vol.58, No.7 (2011), pp.2980-2987.
Martin, A. and Emami, M.R., Design and simulation of robot manipulators using a modular hardware-in-the-loop platform.
In: Ceccarelli M. (ed.) Robot manipulators, (2008), pp 347-372, I-Tech Education and Publishing.
Newton, R.W., Betz, R.E. and Penfold, H.B., Emulating dynamic load characteristics using a dynamic dynamometer, Proceedings of the IEEE Conference on Power Electronics and Drive Systems (1995) pp.465-470.
Norton, R., Machine Design, 3rd edn., (2006) Person Prentice Hall.
Poulin, J., Practical considerations in DC motor and amplifier selection, IEEE Transactions on Industrial Applications, Vol.20, No.5 (1984), pp.1130-1140.
Singaraju, T., Turan, A., Gokasan, M. and Bogosyan, S., Hardware-in-the-loop simulation of PUMA 560 via Internet, IEEE Conference on Industrial Electronics (2006), pp. 5426-5432.
Sony Ericsson and Mobile Complete team for virtual lab service, Wireless News (2007).
Tischler, N., Experimental investigation of stiffness control for a robotic manipulator, Dissertation (2000), p.57, University of Toronto.
Toliyat, H.A. and Kliman, G.B., Handbook of electric motors (2004), p.339, Marcel Dekker Inc.
Trevelyan, J., Lessons learned from 10 years experience with remote laboratories, International Conference on Engineering Education and Research (2004).
Zalud, T., Sometimes it pays to be eccentric, Machine Design, Vol.72, No.10 (2008), pp.75-79.
Zheng, L., Yang, X.M., Zheng, Z.H. and Liu, T.I., A Web-based machining parameter selection system for life cycle cost reduction and product quality enhancement, Computers in Industry, Vol.59, No.2-3 (2008), pp.254-261.