Balance board

Introduction

A balance board is commonly used in motor control labs and neurologic clinics. They essentially consist of a set of finely calibrated scales, measuring mechanical forces. The pattern of forces can be used to derive body position, at a high spatial and temporal resolution.

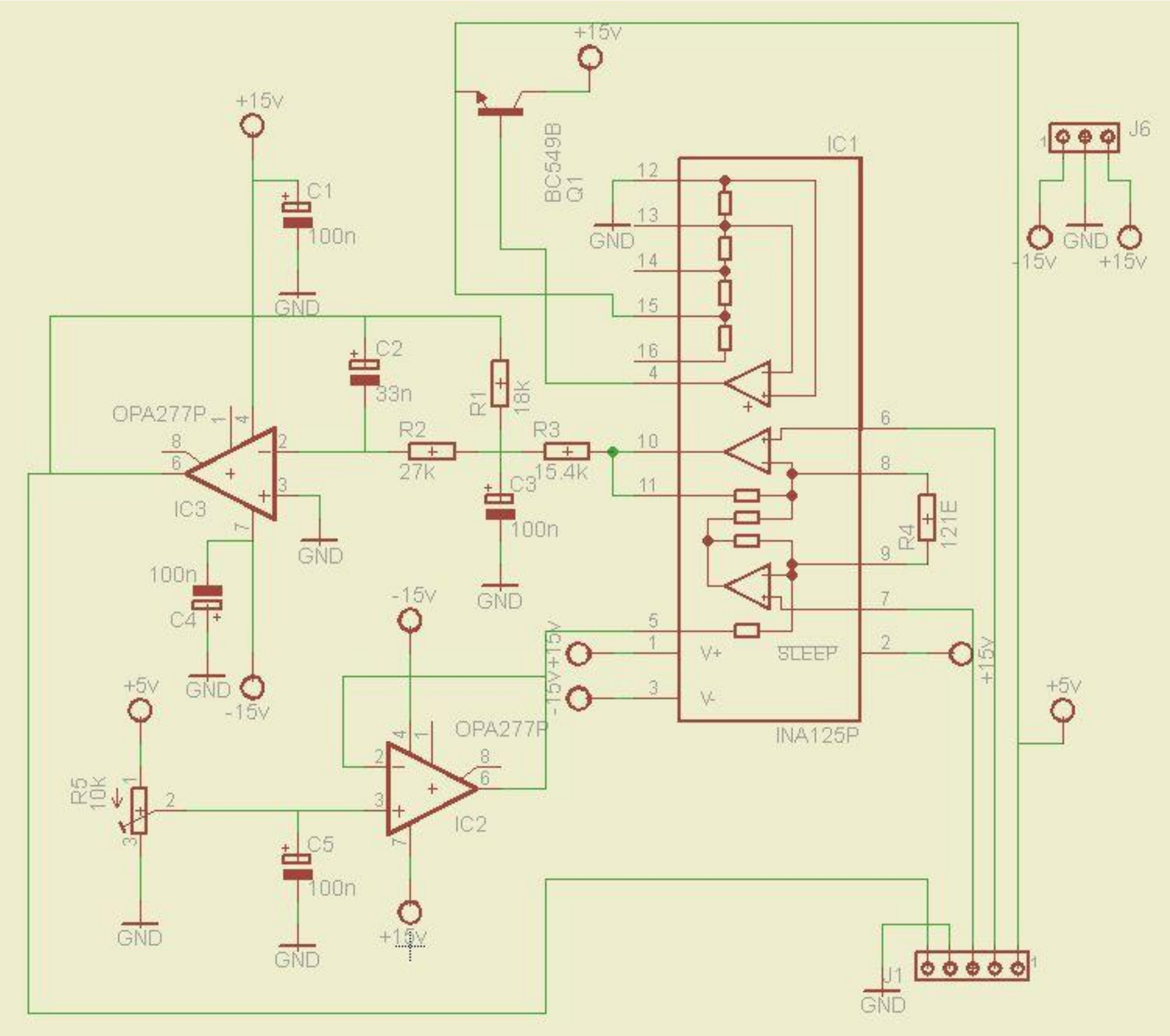
Description

The boards has been specified up to 1M with a sub-millimeter spatial resolution and a 200 hz temporal resolution.

A typical balance board records at a high spatial and temporal resolution. Regarding spatial resolution: a platform can detect changes from a few sub-millimeters (freezing) to at least 500 centimeter (forward or backward steps). The platform records reliably over four vertical forces. Regarding temporal resolution: the signals are typically sampled at 200 Hz. Even though the system allows even higher sampling rates, in practice 100 Hz is sufficient.

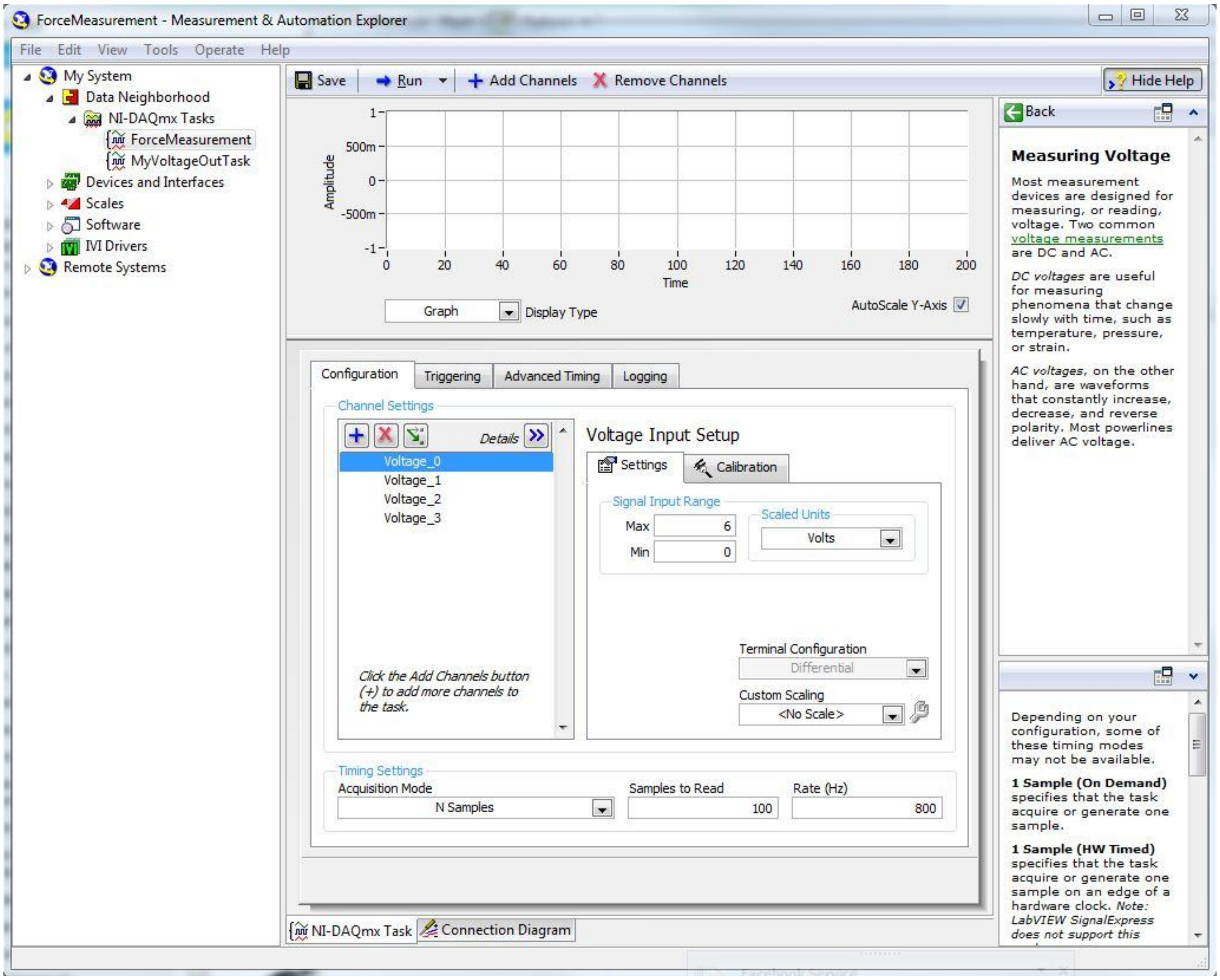
The pressure sensors derive directly from the wii balance board. Each sensor has a maximum pressure of 120Kg. In-house electronics is build to get a clean amplification from the sensors. A National instruments card, USB-6221, takes care of the A/D conversion and connects with usb to a pc.

Technical design



Sensor amplifier

National instruments settings



Channel Settings

Details

Voltage_0
Voltage_1
Voltage_2
Voltage_3

Click the Add Channels button (+) to add more channels to the task.

Voltage Input Setup

Signal Input Range

Max 6

Min 0

Scaled Units: Volts

Terminal Configuration: Differential

Custom Scaling: <No Scale>

Timing Settings

Acquisition Mode	Samples to Read	Rate (Hz)
N Samples	100	800

Measuring Voltage

Most measurement devices are designed for measuring, or reading, voltage. Two common voltage measurements are DC and AC.

DC voltages are useful for measuring phenomena that change slowly with time, such as temperature, pressure, or strain.

AC voltages, on the other hand, are waveforms that constantly increase, decrease, and reverse polarity. Most powerlines deliver AC voltage.

Depending on your configuration, some of these timing modes may not be available.

1 Sample (On Demand) specifies that the task acquire or generate one sample.

1 Sample (HW Timed) specifies that the task acquire or generate one sample on an edge of a hardware clock. *Note: LabVIEW SignalExpress does not support this*

Presentation software

```
sub runtrials_national begin

        # The dio_device will setup NI-DAQmx device number 1 "Dev1"
        dio_device card = new dio_device(ni_dio_device, 1, 0 );
        #int id = card.acquire_analog_input( "MyVoltageOutTask" );
        int id1 = card.acquire_analog_input( "ForceMeasurement,Voltage_0" );
        int id2 = card.acquire_analog_input( "ForceMeasurement,Voltage_1" );
        int id3 = card.acquire_analog_input( "ForceMeasurement,Voltage_2" );
        int id4 = card.acquire_analog_input( "ForceMeasurement,Voltage_3" );
        count_old = response_manager.total_response_count();
        loop
        until false
        begin
                if response_manager.total_response_count() > count_old then
                        count_old = response_manager.total_response_count();
                        calibrate_board = true;
                end;

                message_scale[1] = round(round(card.read_analog( id1, 1000.0 ),6) * 1000.0, 0);
                message_scale[2] = round(round(card.read_analog( id2, 1000.0 ),6) * 1000.0, 0);
                message_scale[3] = round(round(card.read_analog( id3, 1000.0 ),6) * 1000.0, 0);
                message_scale[4] = round(round(card.read_analog( id4, 1000.0 ),6) * 1000.0, 0);

                if calibrate_board then
                        zero_scale_left_up    = message_scale[left_up];
                        zero_scale_left_down  = message_scale[left_down];
                        zero_scale_right_up   = message_scale[right_up];
                        zero_scale_right_down = message_scale[right_down];
                        calibrate_board = false;
                end;

                message_scale[left_up]    = message_scale[left_up] - zero_scale_left_up;
                message_scale[left_down]  = message_scale[left_down] - zero_scale_left_down;
                message_scale[right_up]   = message_scale[right_up] - zero_scale_right_up;
                message_scale[right_down] = message_scale[right_down] - zero_scale_right_down;
                t_scale11.set_caption(string(message_scale[left_up]));
                t_scale11.redraw();
                t_scale22.set_caption(string(message_scale[left_down]));
                t_scale22.redraw();
                t_scale33.set_caption(string(message_scale[right_up]));
                t_scale33.redraw();
                t_scale44.set_caption(string(message_scale[right_down]));
                t_scale44.redraw();

                pos_dot_x = (message_scale[right_up] + message_scale[right_down]) -
message_scale[left_up] + message_scale[left_down]);
                pos_dot_y = (message_scale[left_up] + message_scale[right_up]) -
message_scale[left_down] + message_scale[right_down]);
                p_balance.add_part( balance_pos, (pos_dot_x * 1.0), (pos_dot_y * 1.0));
                t_coord.set_caption(string(pos_dot_x)+","+string(pos_dot_y));
                t_coord.redraw();

                p_balance.present();
                p_balance.remove_part( 8 );
        end;
        card.release_analog_input( id1 );
        card.release_analog_input( id2 );
        card.release_analog_input( id3 );
        card.release_analog_input( id4 );
end;
```