Control System and User Interface for Hydraulic Bike

Project Proposal

TA: Jacob Bryan

ECE 445

February 11, 2015

Table of Contents

1. Introduction 3
1.1 Statement of Purpose 3
1.2 Objectives 3
1.2.1 Goals and Benefits 3
1.2.2 Functions and Features 3
1.3 Competition Overview 4
1.4 Mechanical System Description 4
1.4.1 Hydraulic Circuits to Support 4
2. Design 5
2.1 Block Diagrams 5
2.2 Block Descriptions 5
User Interface Module 5
Control System Module 5
Power Module 6
Sensors Module 6
3. Requirements and Verification 7
Tolerance Analysis 8
4. Cost and Schedule 8
4.1 Cost Analysis 8
4.1.1 Labor 8
4.1.2 Parts 8
4.1.3 Grand Total 9
4.2 Schedule 9

1. Introduction

1.1 Statement of Purpose

Parker-Hannifin, a fluid power systems company, hosts an annual competition for the design of a chainless bicycle. A control system, user interface, and electrical power are needed for UIUC's entry into this competition. Previous control systems for these bicycles used either commercially-available systems1 or discrete logic2. A new and unique approach to this problem is a custom-built control system and user interface. This has the possibility of being tailored to the end users, who are both accessible and on campus.

Automation and control systems are of personal interest, so working on the control system and operator interface for a real product would be a significant personal and professional benefit.

1.2 Objectives

1.2.1 Goals and Benefits

- Electro-mechanically control flow of fluid in hydraulic circuits
- React to user input and give feedback to rider
- Automatic control of electromechanical valves improves bicycle's ease of use (versus manual valves)
- Stay within competition guidelines
- Help create a competitive or winning hydraulic bicycle
- No control failures during competition (happened in 2013-2014)

1.2.2 Functions and Features

- Fail-safe behavior
- Require user interaction to change hydraulic circuits (for safety of rider)
- Interpret sensor data and ensure operation is within safe parameters
- Visually output current hydraulic circuit and accumulator pressures
- Otherwise support functioning of mechanical systems on bicycle

[1] Team using Parker-Hannifin IQAN system:
https://engineering.purdue.edu/ABE/InfoFor/CurrentStudents/SeniorProjects/2012/GeskeLamneckSparenbergEtAl
[2] Team using discrete logic (page 19):
http://deepblue.lib.umich.edu/bitstream/handle/2027.42/86206/ME450?sequence=1

1.3 Competition Overview

The 2014/2015 Chainless Challenge is a competition sponsored by Parker-Hannifin with the goal being to develop a fluid-powered bicycle, preferably without a chain. The competition consists of an endurance race, an efficiency race, and a sprint race. Points are awarded for placing in races as well as for various design features.

1.4 Mechanical System Description

A brief explanation of the mechanical system that is being controlled is necessary to give context to the rest of this document. Fluid power is provided by the rider and is input at the pedals. The fluid power can either be stored in the accumulators or sent directly to the hydraulic motor, as determined by the valves in the manifold. The hydraulic circuits to support this are listed below in section 1.3.1. The accumulators have valves that can either be setup to accept incoming fluid or expel outgoing fluid to the motor. Incoming fluid can either come from the rider (charge paths) or from the motor (regenerative braking path). Figure 1 is an overview of how components are connected.

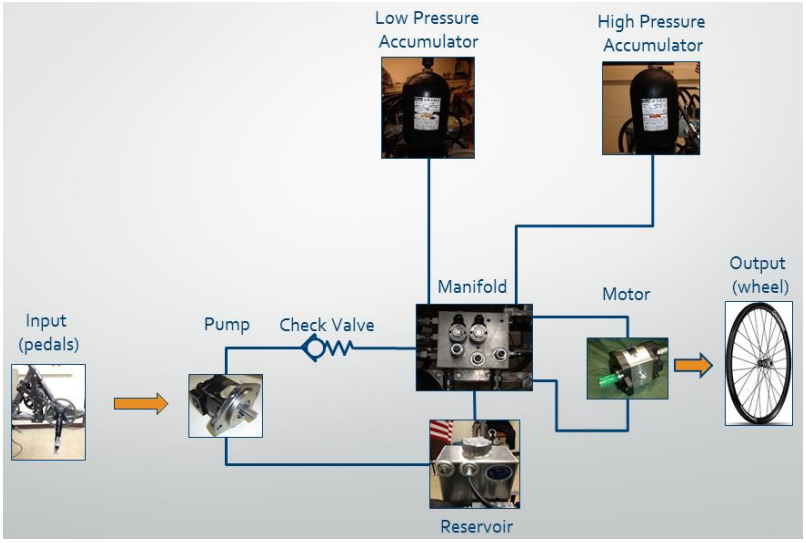

Figure 1. Overview of mechanical components.
Blue lines represent hosing for fluid.

1.4.1 Hydraulic Circuits to Support

There are 7 hydraulic circuits that need to be supported by the electrical system, listed below.

1. Direct Drive Path (Default)
2. Coasting Path
3. Charge Low Pressure Accumulator and Coast
4. Charge Low and High Pressure Accumulators
5. Discharge Low Pressure Accumulator
6. Discharge High Pressure Accumulator
7. Regenerative Braking Path

2. Design

2.1 Block Diagrams

Figure 2 shows the electrical component of the hydraulic bicycle, which consists of three subsystems: control system, user interface, and power management.

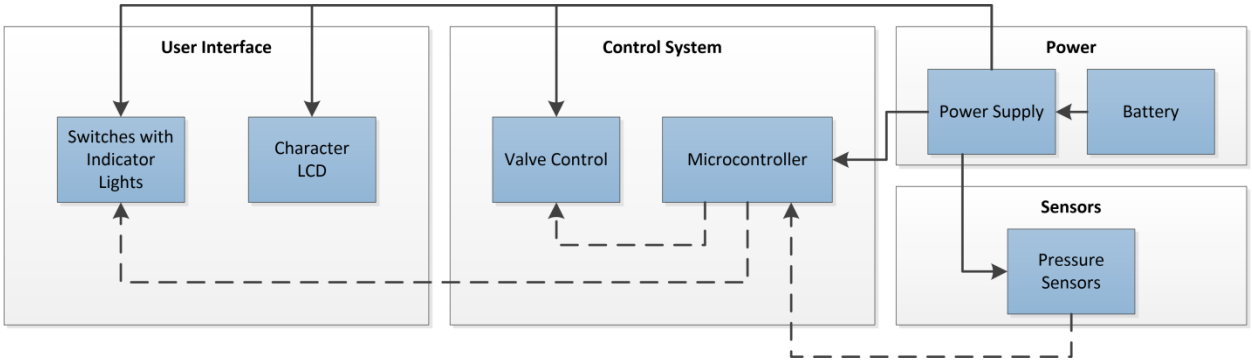

Figure 2. Block diagram of electrical system.
Solid lines indicate power, dashed lines indicate signals.

2.2 Block Descriptions

User Interface Module

The user interface module is the main way to facilitate user interaction and display information to the user. The user needs to be able to select between 6 operating modes, view the operating mode on the Character LCD, and view sensor data on the Character LCD, at the very least. The user interface will be mounted on the handlebars for accessibility and convenience. Future addition of features will most impact the user interface.

2.2.1 Switches with Indicator Lights

Three buttons with indicator lights will be used to select Charge, Discharge, and Regenerative Braking. A 3-position selector switch will select between charging/discharging the low and/or high pressure accumulators when the Charge or Discharge modes are selected. All buttons will be momentary, since if 'toggle' functionality is desired, it can be achieved in software. Switch circuitry will include pull-up resistors. Inputs and outputs will likely be communicated with the microcontroller via an I2C/TWI port expander IC.

2.2.2 Character LCD

The Character LCD block will be composed of at least one 20x4 character LCD and will display the current pressure of both accumulators and text indicating the current state. If a PID controller or similar functionality is later added to the project, the character LCD might also show suggestions for the best mode as based on accumulator pressures, race type, etc. Character LCD(s) will likely communicate with the microcontroller via I2C/TWI using a commercially-available "LCD backpack".

Control System Module

The control system's microcontroller is responsible for sending/receiving all signals to other modules.

2.2.3 Microcontroller

The microcontroller is responsible for reading the button state on the user interface, sending display data to the character LCD(s), controlling the indicator lights, reading the pressure sensors, and turning on/off the valves. For rapid prototying, an ATmega 328P is currently in use since it is supported by the Arduino environment (the 328P is used in the Arduino UNO) and it is available in a 28-pin DIP. The microcontroller is not physically size-constrained and is not very power-constrained, as discussed in the Battery block description. Memory is the biggest constraint, especially if a PID controller with state suggestions is to be implemented and the ASCII text for the LCD is stored. Extra EEPROM or similar non-volatile memory could be added over an I2C/TWI bus if memory became an issue.

2.2.4 Valve Control

The electromechanical valves require 12V at 1.2A in order to actuate. Since an ATmega 328P can only source 200mA across all pins, either transistors or small power relays are required. Power relays were chosen because they effectively isolate the input from the output. Since turning on the valves will create a momentary drop in supply voltage, future separation of the power supplies may be desired. Thus, power relays were chosen. The power relays currently in use require 80mA to turn on, so a transistor is being used to drive them. All outputs for valves will also have indicator LEDs to aide in troubleshooting.

Power Module

Power from the battery will be stepped down to the necessary voltages and currents.

2.2.5 Power Supply

The power supply will step down to 5 VDC for all components except for the valve outputs and will pass the 12V battery voltage to the valves (through the relays). While the valves do not have a polarity, the 5VDC circuitry can be damaged by reversed-polarity, so a protection circuit will be added. A power PMOS is currently being used for this purpose.

2.2.6 Battery

The battery is required to be 12V and will need to power the entire system for 1 day of the competition. While exact power usage is unknown, it has been estimated that 20Ah is the bare minimum, based on 8 hours usage at 2.5 Amps. A lead-acid battery will likely be used due to cost vs capacity vs volume.

Sensors Module

Pressure sensors are currently the only sensors anticipated.

2.2.7 Pressure Sensors

Pressure sensors attached to the accumulators will provide feedback on accumulator pressures. The pressure sensors that have been selected are Parker IQAN-SP pressure sensors, which take 5 VDC input and return an analog voltage from 0 to 4.5 VDC.

3. Requirements and Verification

Block	Requirement	Verification	Points
1. Switches with indicator lights	**a.** Button state is correctly read over I2C **b.** Indicator lights are correctly controlled over I2C	**a.** Connect I2C master to bus and read from the I2C port expander with all buttons unpressed. Press one button at a time, continuing to read the state over I2C. Data should indicate logic LOW when buttons are unpressed and logic HIGH when buttons are pressed. Multiple button presses should be properly registered. **b.** Start with all indicator lights off. Connect I2C master to bus and write to I2C port expander the data required to turn on each indicator light, then turn off each indicator light. Only the specified indicator lights should change state on each write.	15
2. Character LCD	**a.** Character LCD displays ASCII text from microcontroller	**a.** Connect I2C master to bus and send ASCII text to LCD backpack over bus. Ensure ASCII text sent fills character LCD and that all characters are displayed where expected.	5
3. Micro controller	**a.** Microcontroller can communicate with all I2C slaves **b.** Output a low signal below the turn-on voltage of the valve control circuit **c.** Output a high signal above the turn-on voltage of the valve control circuit	**a.** Connect microcontroller to I2C bus as master. Verify microcontroller can perform all I2C tasks mentioned elsewhere in this table. **b.** Program the microcontroller to send a low signal. Check voltage is 25% below threshold voltage of transistor using DMM in parallel with microcontroller. **c.** Program the microcontroller to send a low signal. Check voltage is 25% above the threshold voltage of transistor using DMM in parallel with microcontroller.	30
4. Valve control	**a.** Turn on output given an input above transistor threshold voltage **b.** Turn off output given an input below transistor threshold voltage	**a.** Connect a variable power supply to the input of the circuit. Sweep the voltage from below $V_{threshold}$ to 25% above $V_{threshold}$. Relay should turn on and indicator should light. **b.** Connect a variable power supply to the input of the circuit. Sweep the voltage from above $V_{threshold}$ to 25% below $V_{threshold}$. Relay and indicator should turn off.	20
5. Power supply	**a.** Provide 5 +/- 0.25 V at 2A +/- 0.25A **b.** Provide reverse-polarity protection	**a.** Place DMM in parallel with power supply output. Measure voltage difference – should be between 4.75 and 5.25VDC. Next, add a constant-current load to the power supply output and set it to 2A. Let run for 1 hour. At the end of the hour, if the DMM still is within the voltage requirements, power supply passes. Do not use heatsinks or airflow beyond normal application. **b.** Disconnect power supply output from rest of circuits. Connect battery in reverse polarity. Connect DMM to power supply output and measure voltage – should NOT be less than 0V.	5
6. Battery	**a.** Check battery can last 1 day at event	**a.** Charge battery to full capacity. Discharge battery with constant current load of 2.4 Amps and time how long it takes to discharge to 11V. Must take more than 6 hours. Recharge battery after full discharge to prevent permanent damage.	10
7. Pressure sensors	**a.** Check pressure readings increase as each accumulator is charged **b.** Check pressure readings decrease as each accumulator is charged	**a.** Set assembled and otherwise functional bike to charge desired accumulator. Pedal bike to charge accumulator and read pressure sensor output with microcontroller to verify pressure is increasing. Test other accumulator.[3] **b.** Set assembled and otherwise functional bike to discharge desired accumulator. Read pressure sensor output with microcontroller to verify pressure is decreasing as bike moves forward. Test other accumulator.	15

[3] Max system pressure at accumulators is 3000 PSI, so this is the best available methodology I could think of.

Tolerance Analysis

The reliability of the I2C/TWI bus is the most critical part of the system, both from a safety and a functionality standpoint. If the bus cannot reliably communicate data, the user may lose control over the bike. In this scenario, the user could disconnect power to the control system and the valves would default to the Direct Drive path, but this severely impacts the functionality of the bike as a whole.

4. Cost and Schedule

4.1 Cost Analysis

4.1.1 Labor

Name	Hourly Rate	Total Hours Invested	Total Labor
Iain	$30.00	400	$30,000

4.1.2 Parts

Item	Part Number	Quantity	Cost
Microcontroller	ATMEGA328P-PU	1	$3.74
Power switch	100SP1T1B4M2QE	1	$2.11
Fuse and holder	Unknown	1	$5.00
Reverse polarity transistor	IRF9540PBF	1	$1.72
Capacitors, 10uF	--	2	$0.54
Voltage regulator	LM7805-ACT	1	$0.67
LEDs (red, green, blue)	Cree C503B-XXX	11	$1.65
Status LED transistor	Low current NPN	1	$0.25
Valve control transistors	SS8050DBU	9	$2.61
Resistors	--	23	$3.45
Connectors on PCB, 2pos	OSTOQ023250	14	$4.62
Connectors on PCB, 3pos	OSTOQ033250	2	$0.86
Connectors on wires, 2pos	OSTTJ0231530	14	$11.48
Connectors on wires, 3pos	OSTTJ0331530	2	$2.46
Ceramic resonator	CSTLS16M0X53-B0	1	$0.40
Capacitors, 22pF	--	2	$0.68
Pushbutton switch	B3F-1000	1	$0.35
Diode	1N4004	9	$0.99
Relay	T77S1D10	9	$20.52
Character LCD, 20x4	Adafruit 198	1	$17.95
I2C port expander	MCP23017-E/SP	1	$1.44
I2C LCD backpack	Adafruit 292	1	$10.00
I2C ADC	ADS1015	1	$9.95
Illuminated pushbuttons	Adafruit 1439	4	$6.00
Selector switch	AutomationDirect GCX3370-22	1	$14.50
Battery	UB12220	1	$43.96
Wire	--	1	$10.00
		TOTAL	$177.90

4.1.3 Grand Total

Section	Total
Labor	$30,000
Parts	$167.95
Grand Total	***$30,177.90***

4.2 Schedule

Week	Task	Responsibility
2/9	Test individual subcircuits (done) Finalize proposal Prepare mock design review Finish basic functions prototype	Iain
2/16	Order parts Finalize PCB (and order?) Wire and mount user interface	Iain
2/23	Order PCB, if not already Mount PCB on bike, if ready Design review Bicycle testing	Iain
3/2	Mount PCB on bike Fine tuning of bicycle Create backup set of electronics for competition	Iain
3/9	Fine tuning of bicycle	Iain
3/16	Fine tuning of bicycle	Iain
3/23	Ship bike	Iain
3/30	Individual progress report due 4/1	Iain
4/6	Mock demo – reschedule for earlier competition Competition starts 4/7, ends 4/9	Iain
4/13	Demo	Iain
4/20	Presentation	Iain
4/27		Iain
5/4	Final papers due 5/6 Lab checkout on 5/7	Iain