Technical Report

Subject: Overview of Additions to Aerify Software
Date Issued: 18-JUL-2019
Order: RD001, ATA151 Aerify
Issued By: Graham Waechter
Attachments: *kbl1xxx datasheet.pdf, Roboteq Controllers User Manual v18.pdf*
Distribution: Anthony Khoraych, Dusan Mandic, Luke Osmokrovic

Scope of work

Software development work is expected to require up to 6 hours of developer time. The work is to modify and add functionality to an existing stand-alone executable in Labview RTE 2017.

- Software functionality to be created by the software developer:
  - Serial communication between NI USB DAQ and two Roboteq motor controllers (RS-232)
  - Implementation of low-level closed loop PID control scheme of motors based on speed feedback from motor controllers
  - Implementation of high-level closed loop PID control scheme of one motor speed based on pressure feedback (existing I^2C input)
- Software documentation required of the software developer
  - Connection and wiring instructions for USB DAQ to motor controller(s) interface
  - Instructions for configuration of USB DAQ in NI MAX if additional steps are required
- Supporting documentation to be supplied by ATA to the software developer
  - Functional description below
  - Functional diagram of PID loop
  - Motor controller datasheet
  - Motor controller user manual

Serial communication interface

The desired communication interface is between the existing NI DAQ (NI USB 6001) and the two motor controllers (Roboteq KBL 1660). The default pinout of the motor controller's DB-15 connector is shown in the attached "kbl1xxx datasheet" on page 6. The software developer must provide instructions/diagrams on the physical connection required between the motor controllers and the USB DAQ so ATA can perform the required wiring. If the existing DAQ (NI USB 6001) is inadequate for this communication, the software developer must notify ATA as soon as this is known and recommend an appropriate device.

Further information on the Serial Operation of the motor controller can be found in Section 14 of the attached "Roboteq Controllers User Manual v18". Section 19 of the same document is a reference of all the commands and syntax for the motor controller.

Low-level PID Control of Speed

The low-level PID loop to is to control individually control the speed of two pumps (one motor controller for each) based on the speed feedback from the motor controllers. A flowchart of the both levels of control is shown below. The low level control of speed adjusts the motor power command “G” based on the motor speed reported by the controller “S”. Note that the relationship between G and S changes with the load on the motor, which is why this low level control loop is necessary. It is not certain if these are the two exact commands required, so it must be possible to change these commands at a later time.

The speed setpoint (“n_set_up” for the upstream pump) is based on a set flow-rate multiplied by a constant. The set flow rate should be configured so that a different value can be used for each motor.

This low level control is applied to each of the two motor controllers. The PID coefficients are hard coded constants.

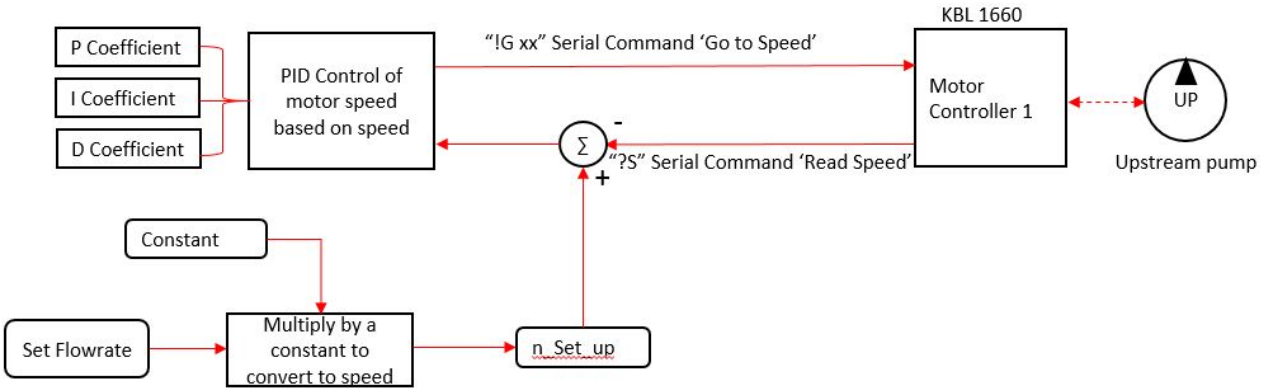
High-level PID Control of Pressure

The higher level control of pressure is applied on top of the low-level speed control of the upstream pump. The higher level control loop sets the flowrate of the upstream pump based on pressure feedback. The pressure used is the pressure read from the Aerify sensor (an existing I^2C input).

The flowrate for the upstream pump that is generated by the high-level control loop is passed to the low-level speed control loop for the upstream pump (a nested control loop). The PID coefficients for the inner and outer control loops are different.

The high-level control loop should be configurable so that it can be activated/deactivated while the program is running. If it is deactivated, the upstream pump will run at a flowrate set by the user in the same manner as the downstream pump. If activated, the pressure-control loop will follow a pressure setpoint entered by the user.

Low level speed control



High level pressure control