High Data Rate Vehicle Dynamic Testing with MOOS

Jordan Britt Jamie Colbert

GPS and Vehicle Dynamics Laboratory
Auburn University

MOOS_DAWG'11

Outline

Introduction

- GPS and Vehicle Dynamics Laboratory
- Often taking sizable amount of high speed data

Figure: Research Vehicles

Motivation
Main Thrusts

- Assess robustness of MOOS data logging beyond just size
  - Multiple high rate devices
  - Large number of Channels
- Tasked research: Data Acquisition.
- Ease of Use

Tasked Research
Project Outline

- Project Goal: Validate Simulation of triple-trailer behavior to assess vehicle safety
- Auburn's Responsibility: Instrument a triple-trailer with all necessary (and unnecessary) sensors and DAQ system.
- Requirements: Cheap, Fast (3-months), High data rates (~100Hz).
- Implemented by new RA in Mechanical Engineering (Unfamiliar with C++ / MOOS)

Tasked Research

Sensors Used

Sensor	Quan	Data	Rate(Hz)
u-blox	2	GPS	1

Tasked Research

Sensors Used

Sensor	Quan	Data	Rate(Hz)
u-blox	2	GPS	1
Novatel	4	GPS	5

Tasked Research

Sensors Used

Sensor	Quan	Data	Rate(Hz)
u-blox	2	GPS	1
Novatel	4	GPS	5
RT-2500	1	KF	100

Tasked Research

Sensors Used

Sensor	Quan	Data	Rate(Hz)
u-blox	2	GPS	1
Novatel	4	GPS	5
RT-2500	1	KF	100
RT-3100	1	KF	100

Tasked Research

Sensors Used

Sensor	Quan	Data	Rate(Hz)
u-blox	2	GPS	1
Novatel	4	GPS	5
RT-2500	1	KF	100
RT-3100	1	KF	100
crossbow	1	IMU	20

Tasked Research

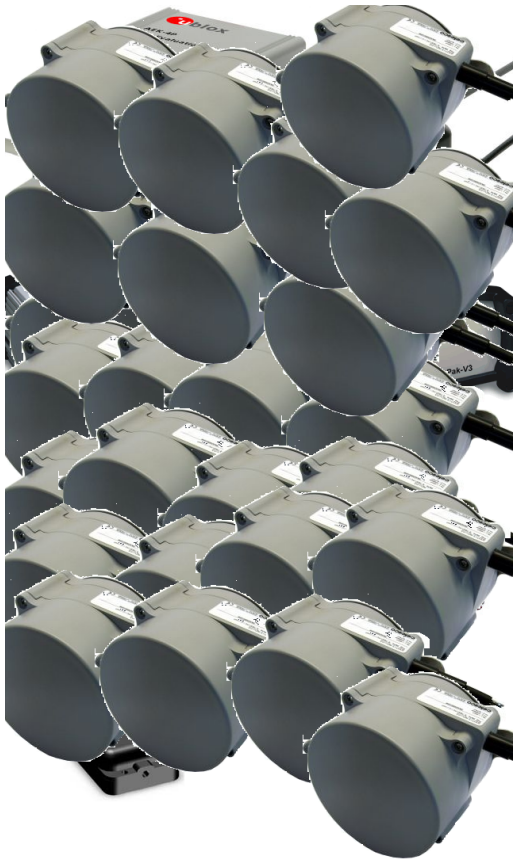
Sensors Used

Sensor	Quan	Data	Rate(Hz)
u-blox	2	GPS	1
Novatel	4	GPS	5
RT-2500	1	KF	100
RT-3100	1	KF	100
crossbow	1	IMU	20
Memsense	1	IMU	150

Tasked Research

Sensors Used

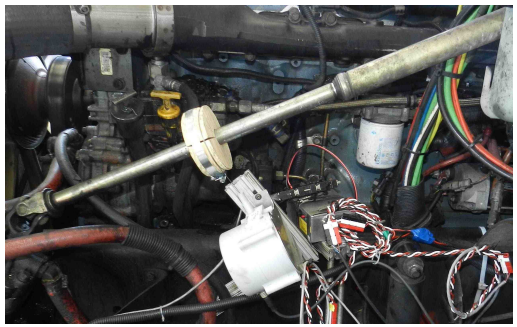
Sensor	Quan	Data	Rate(Hz)
u-blox	2	GPS	1
Novatel	4	GPS	5
RT-2500	1	KF	100
RT-3100	1	KF	100
crossbow	1	IMU	20
Memsense	1	IMU	150
String Pots	1	Steer	100
	20	Susp	100
	6	Angle	100



Tasked Research

Sensors Used

Sensor	Quan	Data	Rate(Hz)
u-blox	2	GPS	1
Novatel	4	GPS	5
RT-2500	1	KF	100
RT-3100	1	KF	100
crossbow	1	IMU	20
Memsense	1	IMU	150
String Pots	1	**Steer**	100
	20	Susp	100
	6	Angle	100



Tasked Research

Sensors Used

Sensor	Quan	Data	Rate(Hz)
u-blox	2	GPS	1
Novatel	4	GPS	5
RT-2500	1	KF	100
RT-3100	1	KF	100
crossbow	1	IMU	20
Memsense	1	IMU	150
String Pots	1	Steer	100
	20	**Susp**	100
	6	Angle	100

Tasked Research

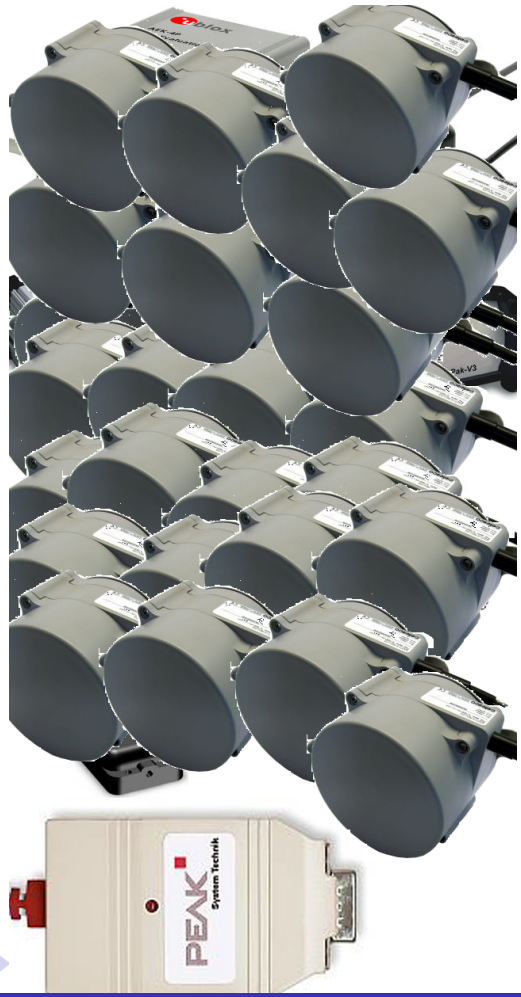
Sensors Used

Sensor	Quan	Data	Rate(Hz)
u-blox	2	GPS	1
Novatel	4	GPS	5
RT-2500	1	KF	100
RT-3100	1	KF	100
crossbow	1	IMU	20
Memsense	1	IMU	150
String Pots	1	Steer	100
	20	Susp	100
	6	**Angle**	100

Tasked Research

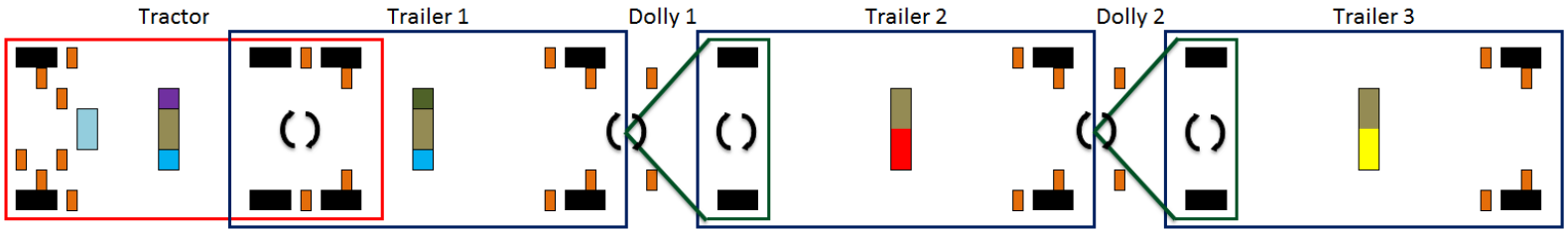
Sensors Used

Sensor	Quan	Data	Rate(Hz)
u-blox	2	GPS	1
Novatel	4	GPS	5
RT-2500	1	KF	100
RT-3100	1	KF	100
crossbow	1	IMU	20
Memsense	1	IMU	150
String Pots	1	Steer	100
	20	Susp	100
	6	Angle	100
CAN	1	Engine Params	100



Tasked Research
Location on Trucks

- Location of Sensors on Trucks



Color	Item Description	Connection Type	Data Rate (Hz)
	Sting Pot	CAN	100
	Engine CAN	CAN	100
	Novatel	RS-232	5
	U-Blox	USB	1

Color	Item Description	Connection Type	Data Rate (Hz)
	MemSense	USB	150
	Crossbow	RS-232	20
	RT-2500	TCP-IP	100
	RT-3100	TCP-IP	100

Tasked Research
Double Lane Change

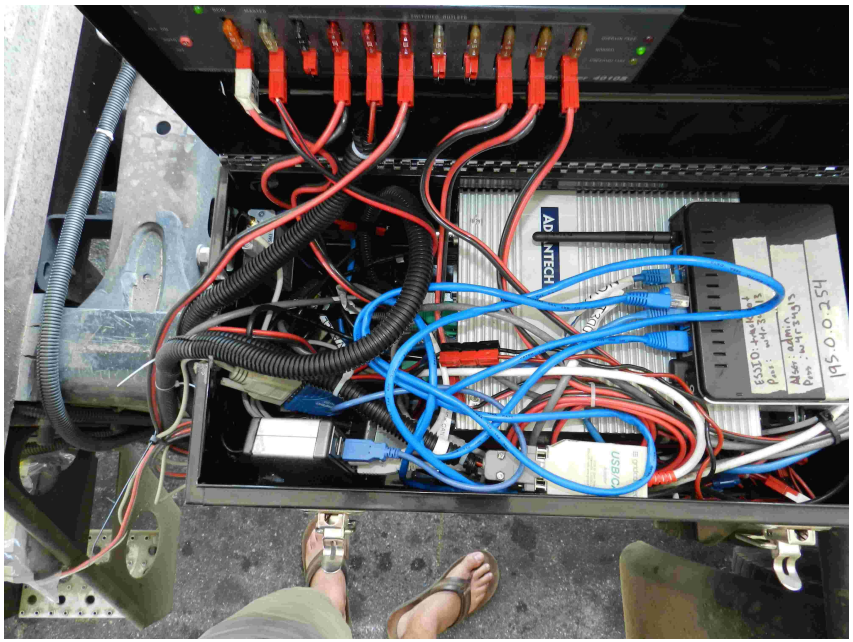
Movie

Tasked Research
On This PC

- Log all these sensors on this PC:
  - Advantech ARK-5280
  - 1.6GHz Pentium M
  - 2Gig DDR RAM

Data Types

- Data was input over:
  - Serial
  - USB
  - TCP/IP



Data Statistics

- This Comprised 201 Messages
- Logging roughly 600 Kb/s

Failure

- Initially attempted to log just the two RT units (100Hz) and the xbow (20Hz)
- PC was completely bogged down
  - Only 38 messages
  - CPU usage 80-90%

Flushing Causes Clogging

Logging issues

- High CPU usage with logging, required code altering

```
490  bool CMOOSLogger::Iterate()
491  {
492      double dfTimeNow = MOOSTime();
493
494  »       // see if we are currently logging, if not exit sub
495  »       if (!m_bLogging)
496  »       »       return true;
497
498      //look to do a synchronous log....
499      if(m_bSynchronousLog)
500      {
516
517
518      //check monitored variables
519      if(dfTimeNow-m_dfLastMonitorTime>DEFAULT_MONITOR_TIME)
520      {
546
547      //are we requested to do wild card logging?
548      if(m_bWildCardLogging)
549          HandleWildCardLogging();
550
551
552      //finally flush all files to be safe
553      m_SyncLogFile.flush();
554      m_AsyncLogFile.flush();
555      m_SystemLogFile.flush();
556
557
558
559      return true;
560  }
```

```
490  bool CMOOSLogger::Iterate()
491  {
492      double dfTimeNow = MOOSTime();
493
494  »       // see if we are currently logging, if not exit sub
495  »       if (!m_bLogging)
496  »       »       return true;
497
498      //look to do a synchronous log....
499      if(m_bSynchronousLog)
500      {
516
517
518      //check monitored variables
519      if(dfTimeNow-m_dfLastMonitorTime>DEFAULT_MONITOR_TIME)
520      {
546
547      //are we requested to do wild card logging?
548      if(m_bWildCardLogging)
549          HandleWildCardLogging();
550
551
552      //finally flush all files to be safe
553      m_SyncLogFile.flush();
554      m_AsyncLogFile.flush();
555      m_SystemLogFile.flush();
556
557
558
559      return true;
560  }
```

Flushing Causes Clogging

- With the flush statements removed we were able to log all sensors at the their desired rates.
  - CPU usage down to 30%

A Better Plunger

Logging issues

- However It should probably be changed to be look like this:

```
490 bool CMOOSLogger::Iterate()
491 {
492     double dfTimeNow = MOOSTime();
493
494     // see if we are currently logging, if not exit sub
495     if (!m_bLogging)
496         return true;
497
498     //look to do a synchronous log....
499     if(m_bSynchronousLog)
500     {
516
517
518     //check monitored variables
519     if(dfTimeNow-m_dfLastMonitorTime>DEFAULT_MONITOR_TIME)
520     {
546
547     //are we requested to do wild card logging?
548     if(m_bWildCardLogging)
549         HandleWildCardLogging();
550
551
552     //finally flush all files to be safe
553     m_SyncLogFile.flush();
554     m_AsyncLogFile.flush();
555     m_SystemLogFile.flush();
556
557
558
559     return true;
560 }
```

```
131 bool CMOOSLogger::ShutDown()
132 {
133     //finally flush all files to be safe
134     m_SyncLogFile.flush();
135     m_AsyncLogFile.flush();
136     m_SystemLogFile.flush();
137
138     return CloseFiles();
139 }
140
141
142 bool CMOOSLogger::CloseFiles()
143 {
144     if(m_AsyncLogFile.is_open())
145     {
146         m_AsyncLogFile.close();
147     }
148
149     if(m_SyncLogFile.is_open())
150     {
151         m_SyncLogFile.close();
152     }
153
154     if(m_SystemLogFile.is_open())
155     {
156         m_SystemLogFile.close();
157     }
158
159     //crucially make sure teh zipping thread has sto
160
161 #ifdef ZLIB_FOUND
```

Issues with Iterate at 100Hz

Timing Issues

- Needed to send a pulse at 100Hz
  - Failed to operate at that rate
  - Would operate at 90Hz, but fluctuated
  - Is there an alternative to using the sleep command for iterate?
- Curious what rates people often reliably control to.

Read My Mind Methods

- Didn't log all intended messages ...oops
  - Alternatives to wild-card logging?
  - Maybe something that ...
- Didn't log some messages with intended precision
  - We've gotten around this by:

```
//! converts the given value to a string with a given percision
template <class T>
inline string precise_to_string (const T& t, unsigned int precision) {
»       stringstream ss;
»       ss << setprecision(precision) << t;
»       return ss.str();
}
```

Plotting Real-time Data

- uMS is great but, is it possible plot any given signal in real-time?

Conclusions

- MOOS is quite robust
- Very capable of handling multiple processes and channels operating a high data rates
- Very (first time) user friendly

Acknowledgements

- National Transportation Research Center Inc (NTRCI)
  - National Transportation Research Center, Inc., works to promote the development and deployment of advanced transportation technologies through research, testing and commercialization for the benefit of the transportation industry and our transportation systems.
- Federal Highway Administration
  - The Federal Highway Administration is funding part of this project and others across the range of issues that are critical to the transportation industry through the Exploratory Advanced Research (EAR) Program. For more information, see the EAR Web site at http://www.fhwa.dot.gov/advancedresearch/about.cfm
- All Gavlab members that assisted on this project.

Suggestions / Questions / Comments

Real Talk

Suggestions / Questions / Comments

Wrap Up

1. Changing Location of flush statements
2. What rates can you reliably control to
3. Alternatives to wild-card logging
4. uMS plotting