# MACHINE LEARNING AND ADAPTIVE CONTROL FOR IMPROVING SERVO PERFORMANCE

*Arvid Linder* *Carl Sjöstedt* *Claes Thunberg*
*Emil Strömblad* *Karl Karlsson* *Gustav Åberg*
*William Wärn*

## Abstract

Tightening emission regulations on road traffic has led to a big push for more efficient and less polluting engines, drivelines and cars in general. As of right now, a big push for electric vehicles is being made, however, the internal combustion engine is still being developed. Every possible aspect of the engine that can be made more efficient is scrutinized. In this project the control of a throttle is studied. The throttle has a non-linear friction behaviour which makes it difficult to control. The throttle will "stick" causing it to not move and have jittery movement. Another problem with the throttle control is the "limp-home" region. The "limp-home" region is at the angle that the throttle defaults to when no current is placed on the electric motor. This angle is slightly open to make sure that the car is always driveable, even if the control of the throttle is lost.

A new controller that is able to adapt parameters when the car is running and classify if it is in a region where it might have gotten stuck, minimizing the effect of external disturbances (such as a change in temperature or pressure) is to be developed. Precise control in all environments allow the engine to always run at optimal conditions and hence lower fuel consumption and increase efficiency could be obtained.

## Setup

A setup where the throttle was tested as hardware in the loop was used, which means that the throttle could be actuated without being installed in a real engine. This allows quick and easy access and testing of the throttle. A PC was used to run the engine model and generate the reference throttle angle and a Raspberry Pi 3B+ and some other electrical components were used to translate the signals between the throttle and the PC.

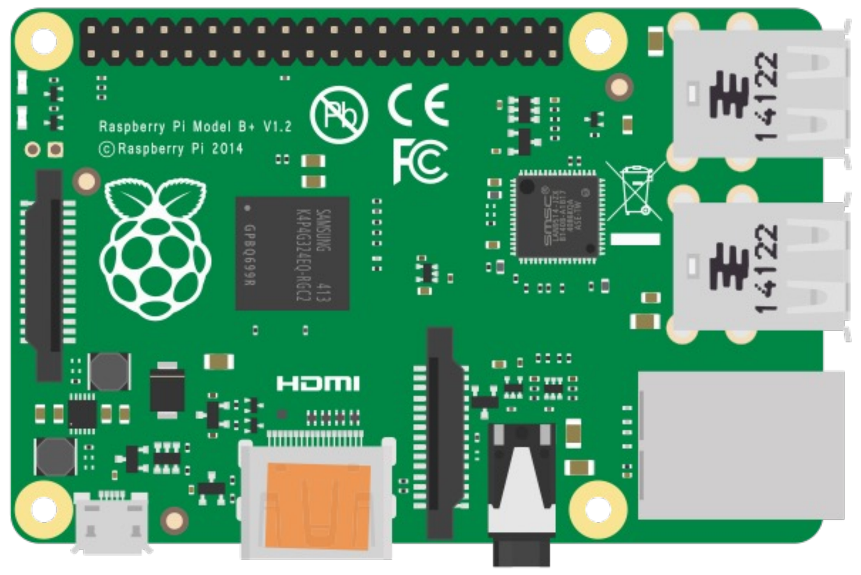

Figure 1: Raspberry Pi 3 model B+.

## Model

The throttle model from TSFS09, Modelling and Control of Engines and Drivelines is used. The non-linear friction of the throttle was modelled using coulomb friction.

$$T_s(\theta) = \begin{cases} m_{lh}^{+} + k^{+}(\theta - \theta_{lh}^{+}) & \text{if} \quad \theta > \theta_{lh}^{+} \\ m_{lh}^{+}(\theta - \theta_{lh})/(\theta_{lh}^{+} - \theta_{lh}) & \text{if} \quad \theta < \theta_{lh} < \theta \leq \theta_{lh}^{+} \\ m_{lh}^{-}(\theta_{lh} - \theta)/(\theta_{lh} - \theta_{lh}^{-}) & \text{if} \quad \theta < \theta_{lh}^{-} < \theta \leq \theta_{lh} \\ m_{lh}^{-} - k^{-}(\theta_{lh}^{-} - \theta) & \text{if} \quad \theta > \theta_{lh}^{+} \end{cases}$$

It is apparent that the friction is non-linear around the limp-home region. The coulomb friction model is piecewise linear and this makes it relatively easy to implement the non-linear friction. The full model was utilized but in a "dummy-state" only to give reasonably sized desired angles for the throttle.
A PID controller with limp-home and friction compensation was implemented that was able to handle these non-linearities well.

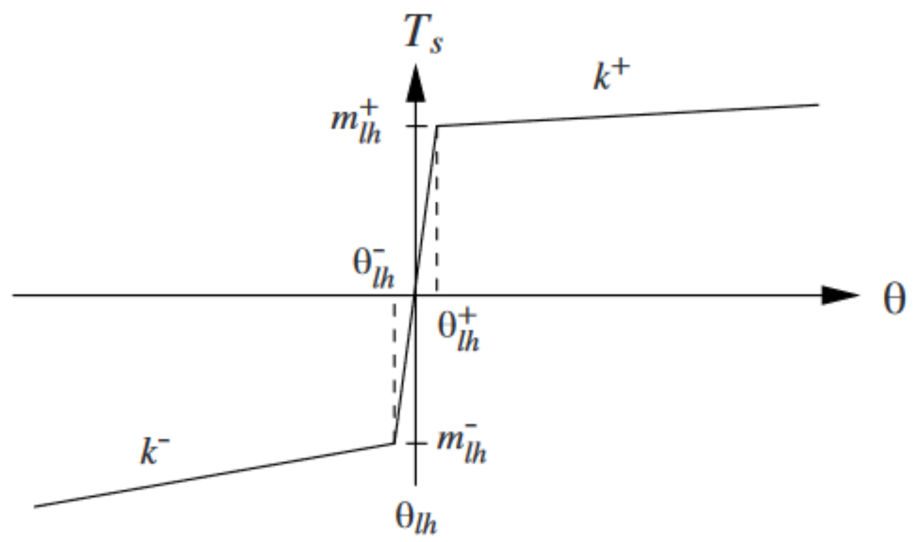


*Figure 2: Spring torque, $T_s$, as function of throttle angle, $\theta$.*

## Machine Learning

Predicting the "stick" behaviour is not simple and could seem random to the untrained eye. Machine learning methods can be used to help identifying the friction parameters when they change, giving improved handling of the characteristics for the system.
An adaptive algorithm which saves the data from the throttles latest 10 minutes was created. Using the current information the "friction-boxes" could be created and adapted for the latest and most accurate characteristics of the throttle.

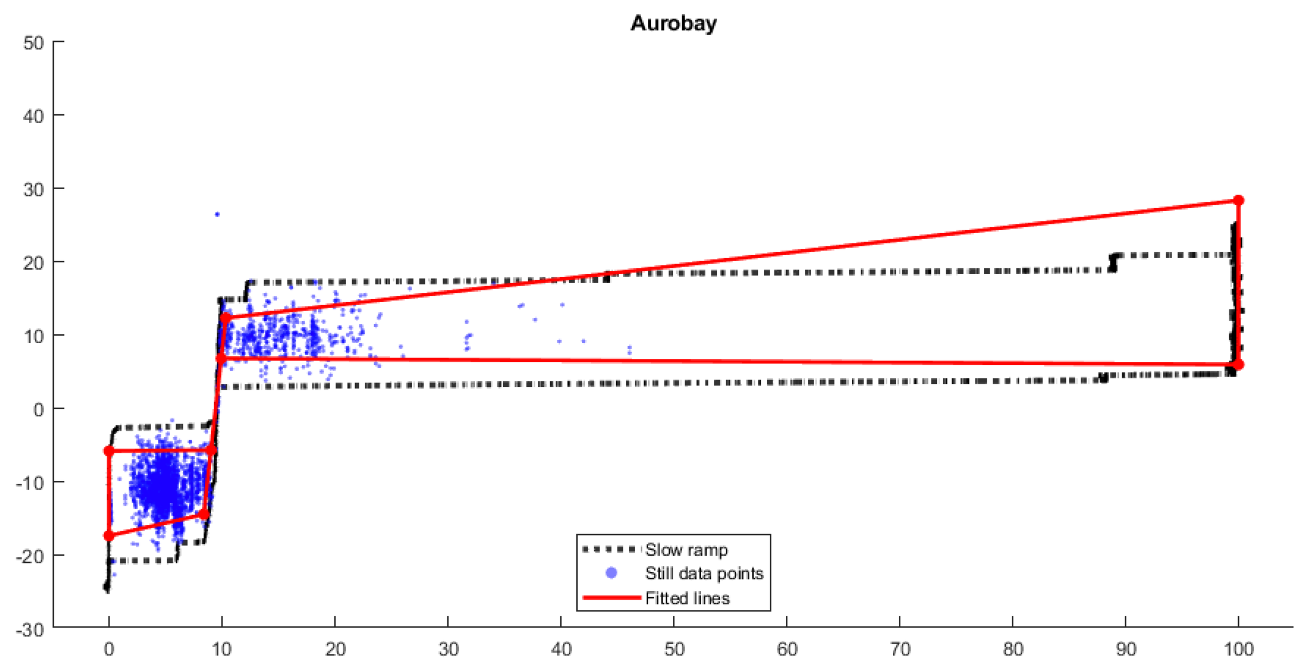


*Figure 3: Calculated hysteresis boxes from the Aurobay drive cycle. Control signal, u, as a function of angle, $\theta$.*

## Results

Slow ramping were used to examine the friction changes depending on temperature. There were no major differences between the different cases for either the new or old throttle. The small changes that can be seen in the figure below occurs due to stick-slip friction and therefore no visible changes in friction due to temperature can be observed.

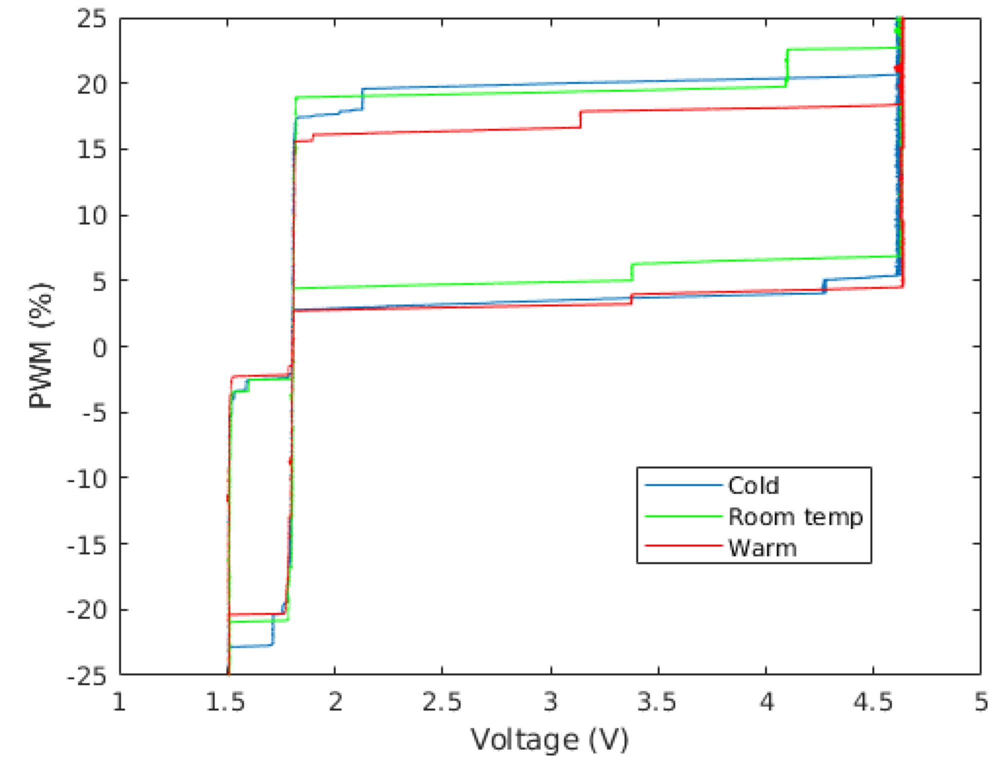


*Figure 4: PWM signal (%) as function of voltage, V.*

When having a manually tuned PID-controller the results show a reasonable reference following of the controller. Not only for a step response but also when the controller was applied to a highly variable drive cycle such as the drive cycle supplied from Aurobay. The errors during certain timesteps in this drive cycle could be relatively large. However, the controller manages to eliminate the error during these time frames very quickly.

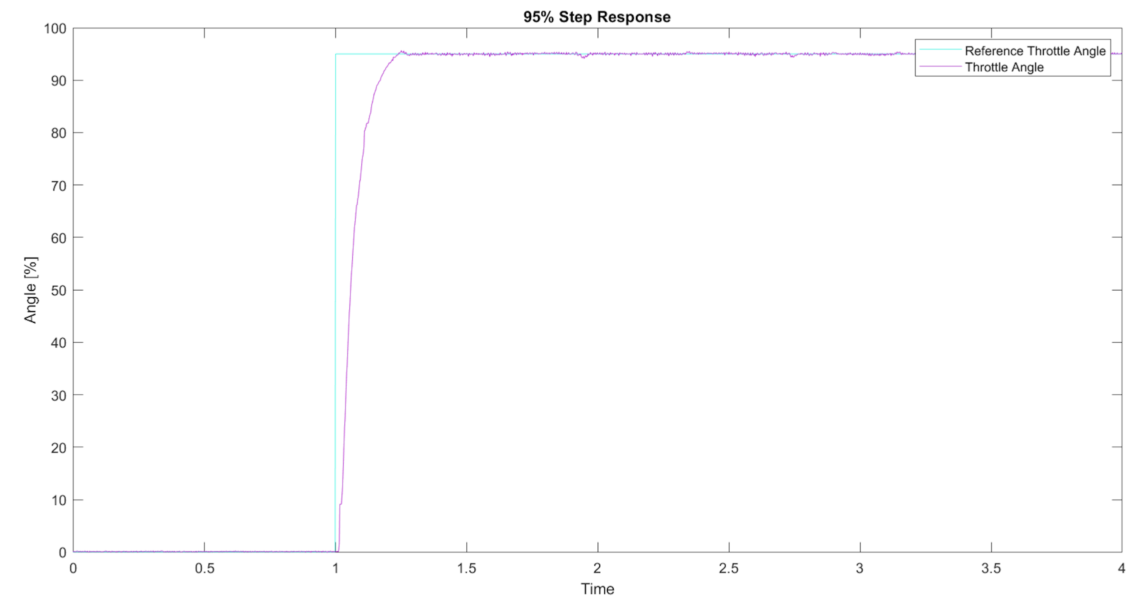


*Figure 5: The performance of the adaptive throttle control with a 90% step response.*

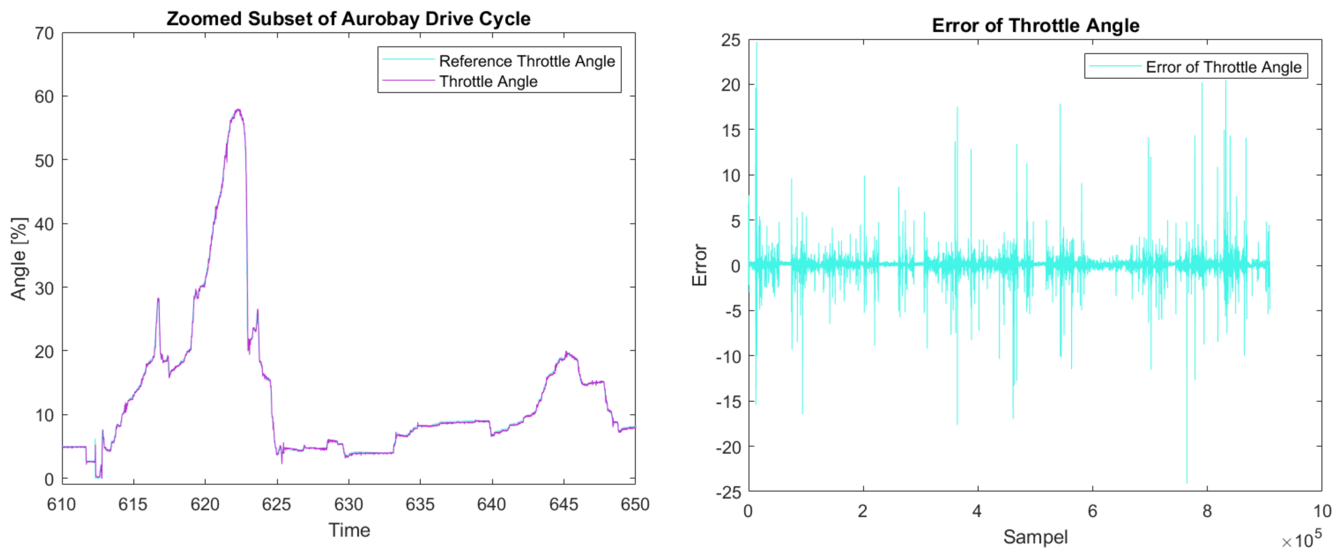


*Figure 6: The performance of the adaptive throttle control with the Aurobay drive cycle.*

A sensitivity analysis was conducted using only the friction and limp home compensation blocks without the PID-controller since this would have corrected some of the faults caused by bad parameter estimation and therefore make the analysis less accurate.

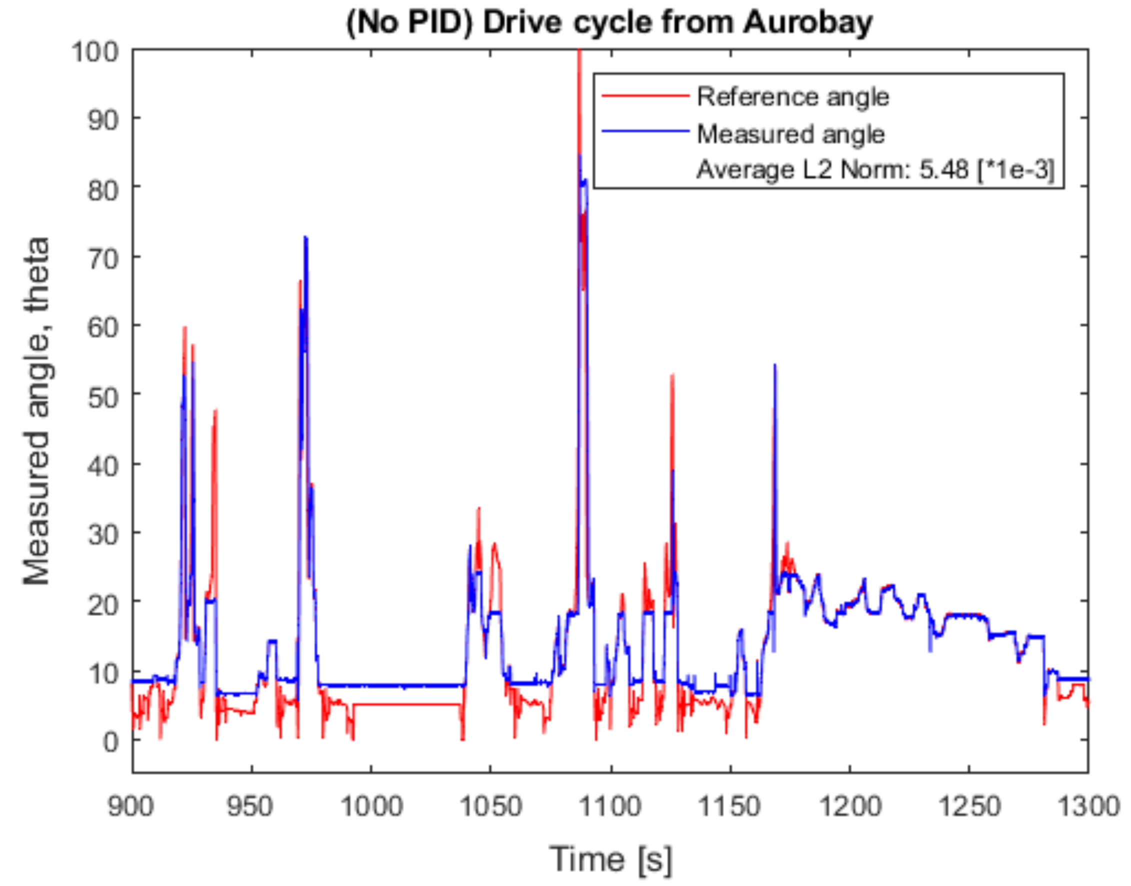


*Figure 7: Sensitivity analysis of a section from the Aurobay drive cycle.*

## Conclusions

Firstly, no major impact could be discovered by varying the operating temperature for the throttle. It could yield a better result with more gathered data. Also, no significant difference was discovered in performance between the old and the new throttle.

Using the machine learning methods, large data from drive cycles that excites the system is required to obtain a reliable result for those methods.

## Future work

- Further improvements and interesting ways forward for the project are to implement a model predictive controller (MPC).
- A problem during the project was a rather extensive delay from a command sent to the Raspberry Pi to the returning response. This was seen clearly when doing steps but is also assumed to have negative impact on the general control of the system.
- Neural network models could be a way forward for the project.
- Some small improvements could be done to the tuning parameters of the controller and/or make these adaptive as well.