

Predictive Dynamic Engine Speed Reduction in Mobile Hydraulic Equipment

Travis Wiens

Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, Canada
E-mail: t.wiens@usask.ca

Abstract

This paper presents an analysis of the potential for engine speed reduction in hydraulic equipment, taking into account not only the minimum engine speed required to meet the current flow demand, but also the minimum speed capable of accelerating the engine to meet increased flow demand in the near future. This is a predictive task, as it requires an estimate of the operator's intention to increase flow demand. We present an analysis of the potential for engine speed reduction using a work cycle from a 40 ton excavator loading a truck, which results in a 33% reduction in the mean engine speed with no reduction in useful work rate. We also present an engine speed control algorithm to perform this predictive task. This controller is easy to tune and requires only a small amount of information about the plant and work cycle. A simulation study is performed that demonstrates the controller's performance and studies the effect of tuning parameters.

Keywords: Fuel efficiency, Diesel engine, Speed Reduction

1 Introduction

The hydraulic system of an excavator is typically powered by a diesel engine turning one or more variable-displacement pumps. While there are other loads on the engine (air conditioning, generator, cooling, etc.), the vast majority of engine power is consumed by the pump and other loads are generally tolerant of varying engine speed. In most cases, the engine speed of conventional excavators is fixed at a near-constant speed and the hydraulic system flow is controlled by varying the pump displacement.

Typical excavator systems exhibit lower specific fuel consumption at lower engine speeds (and higher torque), due to a combination of lower auxiliary function loading, and lower internal frictional losses [1]. Some commercially models of excavator feature an "efficiency mode" where the constant engine speed is lowered [2].

Lowering the engine speed also limits the useful work rate, as the pump flow is limited to the pump displacement multiplied by the engine speed. Thus, the pump flow may be controlled by either adjusting the displacement or engine speed. If the pump is not running at full displacement, the engine speed can be lowered and displacement increased without affecting flow. However, while the pump displacement has a response time in tens of milliseconds, a diesel engine's response time can be on the order of hundreds of milliseconds, too slow to meet dynamic requirements of operator flow demands. Due to the slow response of the engine, the chosen engine speed must not only be sufficient to meet the current flow demand, but also the flow demand expected in the near future, during the time required to accelerate the engine. Some researchers have developed systems to dynamically adjust engine speed to meet the current load and recent history (typically for hybrid architectures [3-6 and others]), but it is believed that

this paper is the first to consider this dynamic effect in a predictive manner.

2 Speed Reduction Potential

In order to study the potential for engine speed reduction in a practical work cycle, we examined the pump flow of a previously recorded excavator data set [7]. This was recorded for a 40 ton excavator loading a truck on the same level with a 90° swing, which is believed to be representative of a typical excavator work cycle. One pump (of two on the machine) was considered, but the same analysis could be applied to the combined effect. This data cycle was scaled to match the power available in the target engine.

At each time step in the dataset, we used a dynamic simulation to find the minimum speed that is capable of fulfilling all future flow demands if the engine is accelerated at full available torque. This was calculated using a simplified system, modelled as an ideal variable-displacement pump with instantaneous response, driven by a turbo-charged diesel engine. This model is intended to be a generic representation of a typical system, with simple parameters that are easy to understand, rather than an attempt to accurately model a specific system.

The engine load is modelled as an ideal pump with inertia:

$$\dot{\omega} = \frac{1}{J}(T_e - T_p) \quad (1)$$

where ω is the engine shaft speed, J is combined inertia of engine and pump, T_e is the engine torque and T_p is the pump torque, given by

$$T_p = P_s \frac{D}{2\pi} \quad (2)$$

where P_s is the pump pressure (from the recorded data set), and D is the pump displacement (in terms of volume per revolution).

The pump displacement controller is assumed to have a fast reaction time (relative to the slow engine dynamics) and sets the displacement to

$$D = \hat{Q} \frac{2\pi}{\omega} \quad (3)$$

where \hat{Q} is the pump flow demand (from the dataset). This displacement is limited to the range of 0 to D_{max} , the maximum pump displacement.

As this analysis is interested in the engine acceleration time, the engine torque is the maximum available, limited by both the engine's torque curve, as well as a rate-limited torque rise to model the turbocharger response. The engine's torque curve was modelled as an eight-point lookup table with respect to shaft speed. This was based on published data for a four cylinder 4.4L turbocharged diesel engine with air to air charge cooling, shown in Figure 1 [8]. The data was extrapolated to zero torque at zero speed.

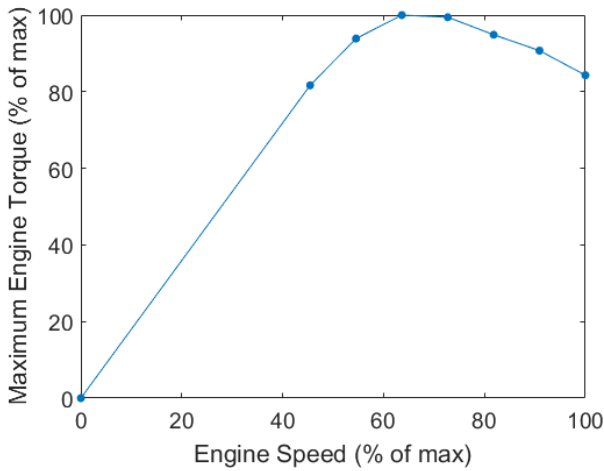


Figure 1: Diesel engine torque curve [8].

The turbocharger and smoke map response was modeled as a constant rate-limit on the engine torque, with the time taken for the torque to rise from zero to the engine maximum denoted as Δt_{turbo} . This is a relatively simple model, but is believed to capture the gross engine dynamics without losing generality by targeting a specific engine.

Parameters used for the simulation study may be found in Table 1.

Starting at each time step in the data set, the system of nonlinear ordinary differential equations above is solved for a given initial engine speed, with the torque rise limiter initially set at the torque required to meet the current torque demand. The initial engine speed is then varied to solve for the minimum engine speed, ω_{min} such that the system can accelerate to meet all future flow demands without the required engine displacement exceeding D_{max} . This was achieved using the nonlinear root finding algorithm fzero in Matlab, based on Forsyth's algorithm [9].

2.1 Results

The resultant minimum engine speed for one cycle of the data set is shown in Figure 3 along with the flow demand, based on the pressure demand shown in Figure 2. This cycle is believed to be representative of the dataset. Notice that the minimum engine speed tracks the small changes in flow demand, but needs to lead the demand for the rapid increases around time 227 and 234 s.

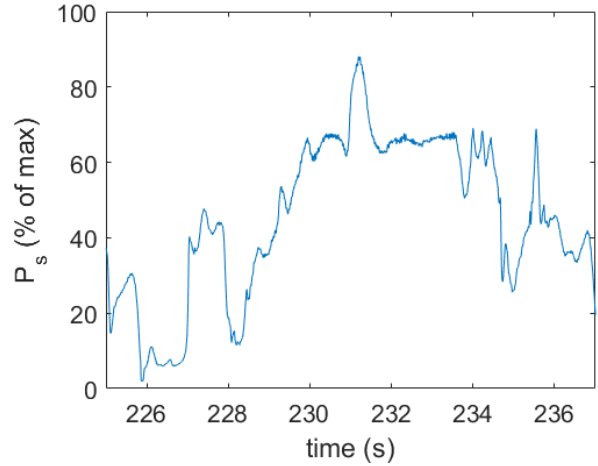


Figure 2: Pressure demand for a single cycle of the excavation dataset. Confidentiality concerns prevent presentation of the entire dataset or non-normalized data.

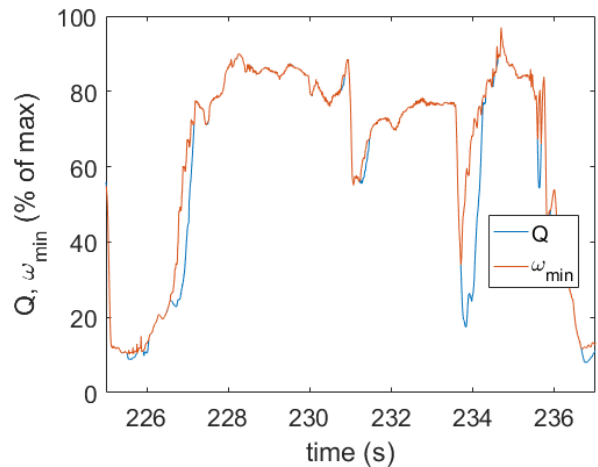


Figure 3: Flow demand and engine speed required to meet future flow demands. This shows one cycle of the dataset.

Figure 4 shows a map of minimum required engine speed capable of accelerating to meet future flow demand vs current flow demand. An application with a slowly varying flow demand would result in all points lying on the diagonal, while an application with a more high frequency content in the flow demand would have more points toward the upper part of the plot. In this case, 95% of all points fall below the red line. The black line shows a histogram of all points. The average engine speed for this dataset can be reduced by 34.3% below a constant speed capable of doing the same work, with corresponding fuel savings dependent on the engine's BSFC curve.

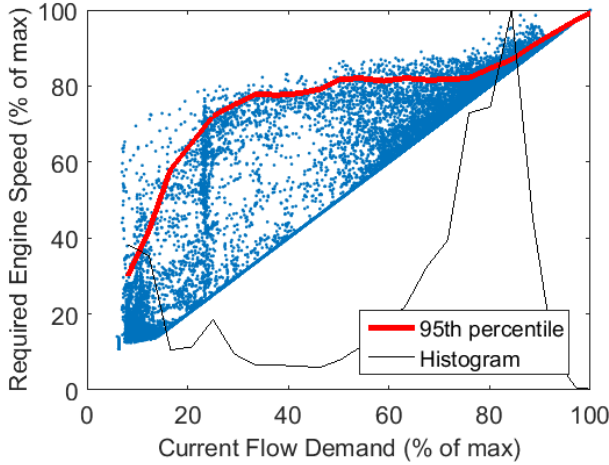


Figure 4: Engine speed required to meet future flow demands for a simulated 40-ton excavator, using an experimentally recorded work cycle. Each dot represents one data point in time. The black line shows a histogram of the time spent at a given flow demand.

2.2 Effect of Turbo Lag

In order to quantify the effect of turbocharger dynamics, the above analysis was repeated, varying the torque rise rate limit. Figure 5 shows the 95th percentile of required engine speed for torque rise values of 500 ms and 1000 ms. The mean engine speed reduction for the 500 ms case is 35.1% (compared to 34.3% for the 1000 ms case). Although the difference in curves is significant, the overall engine speed reduction difference is more modest for this work cycle (as shown in Fig 6), as much of the time is spent in the right-hand part of the curve where the difference is smaller. This effect would be more noticeable for a work cycle with more time spent at lower flow demand.

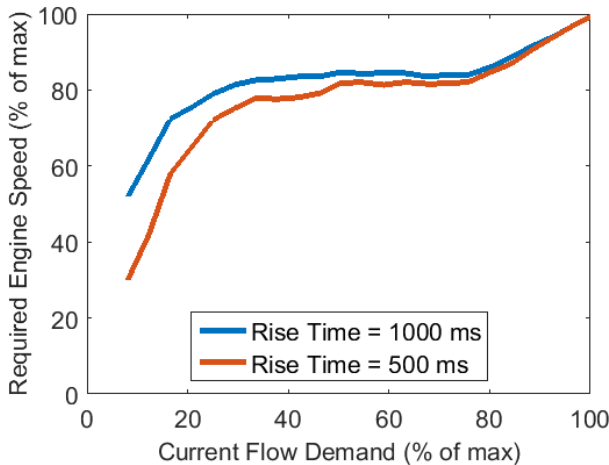


Figure 5: Engine speed required to meet future flow demands 95% of the time for varying torque rise times.

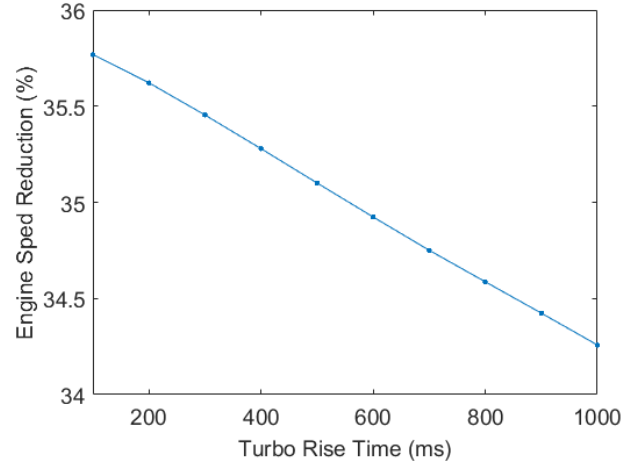


Figure 6: Potential for average engine speed reduction with respect to turbocharger rise time.

3 Control Scheme

The above analysis presents a best-case scenario for reducing the engine speed with no reduction in output pump flow. This section presents a preliminary controller which attempts to achieve this by predictively setting the engine speed based on the current flow demand. The predictive model is continually updated, in retroactive response to over- or under-prediction of the flow demand, so that it can track changes in operator style or task at hand.

The proposed controller is based on a speed adjustment factor, $K(\hat{Q})$, defined so that

$$\hat{\omega} = \frac{\hat{Q} + (Q_{max} - \hat{Q})K}{\frac{D_{max}}{2\pi}} \quad (4)$$

where $\hat{\omega}$ is the controller's requested engine speed, Q_{max} is the flow at maximum engine speed and displacement. This means that if $K = 0$ then the desired engine speed is set just sufficient to meet the current demand flow with no margin for future increases, and if $K = 1$ then the engine speed is set to the maximum.

This adjustment factor is implemented as an N-element lookup table with linear interpolation between values. The elements are updated based on the excess displacement, defined as

$$D_e(k) = D(k - d) - D_{max} \quad (5)$$

where k is the time step and d is the estimated engine response time (in time steps). The controller attempts to minimize this excess displacement while avoiding it being negative (not meeting flow demand). At each time-step the adjustment factor is modified by

$$\Delta K(\hat{Q}(k)) = \begin{cases} -\Delta K^+ & \text{if } D_e > 0 \\ 0 & \text{if } D_e = 0 \\ \Delta K^- & \text{if } D_e < 0 \end{cases} \quad (6)$$

This could be practically implemented in a real load-sensing system by monitoring the load-sense margin pressure: if the pump is able to maintain margin $D_e \geq 0$, otherwise $D_e < 0$.

This adjustment is used to modify the two lookup table points bracketing the current flow. If a subscript 1 denotes the lookup table point just below the current flow and a 2 denotes the point equal to or just above it, then the table is updated according to

$$K_1(k+1) = K_1(k) + \Delta K(\hat{Q})\Delta t\phi_1 \quad (7)$$

$$K_2(k+1) = K_2(k) + \Delta K(\hat{Q})\Delta t\phi_2 \quad (8)$$

$$\phi_1 = \frac{Q_2 - \hat{Q}}{Q_2 - Q_1} \quad (9)$$

$$\phi_2 = \frac{\hat{Q} - Q_1}{Q_2 - Q_1} \quad (10)$$

where Δt is the sample time. The lookup table is saturated to ensure no points fall outside the range of $0 \leq K \leq 1$.

3.1 Parameter Tuning

The proposed controller has two tuning parameters, ΔK^+ and ΔK^- which control how the adjustment factor adapts to past history. A large ΔK^+ will tend to reduce the engine speed more aggressively, while a large ΔK^- will aggressively react to not meeting flow demand, by increasing the engine speed. Large values of both will tend to cause the system to adapt quickly, while small values will tend to smooth out variations.

For a stationary operator model, the expectation of the probability of meeting the flow demand ($D_e > 0$) is

$$\hat{p} = \frac{\Delta K^-}{\Delta K^- + \Delta K^+} \quad (11)$$

For the case where $\hat{p} \gg 0.5$ then the minimum time to adapt K from 1 to 0 is

$$\Delta t_{adapt} = \frac{1}{\Delta K^+} \quad (12)$$

Therefore given a desired success probability and speed of adaptation, the above two equations can be used to set the required parameters.

The third tuneable parameter is the estimated engine response time, d . This parameter controls how far ahead in time the controller tries to predict the engine's response. It should be selected to be close to the actual engine's response time (which varies with operating point), but need not be exact.

3.2 Simulation Results

The proposed controller was simulated using the same excavator truck-loading work cycle described above. The $K(Q)$ lookup table was initialized with zeros, the most aggressive engine speed reduction.

The goal of this controller is to meet the desired probability of meeting the flow demand (in this case $\hat{p} = 0.95$). This is shown in Figure 7, with the simulated p calculated from the Boolean success in meeting flow demand, with a first order filter with time constant 50 s. As desired, the simulated probability quickly increases, before settling around the target value of 95%. Note that this simulation was carried out with an aggressive initial condition; if the $K(Q)$ lookup table was initialized with ones, the probability would start at 100% and fall to the desired value, resulting in a higher initial performance, but lower fuel savings.

Figure 8 shows the flow demand and the simulated actual flow, with the flow demand generally being met, except for some flow peaks and rapid increases (which is expected with the $\hat{p} = 0.95$ target).

The evolution of the engine speed adjustment factor is shown in Fig 9. By the end of the run (red), the curve has settled to a position where the engine speed is aggressively reduced for low and high flow demands, with smaller engine speed reductions in the middle. This curve is strongly dependent on the operator's aggressiveness and the work cycle performed.

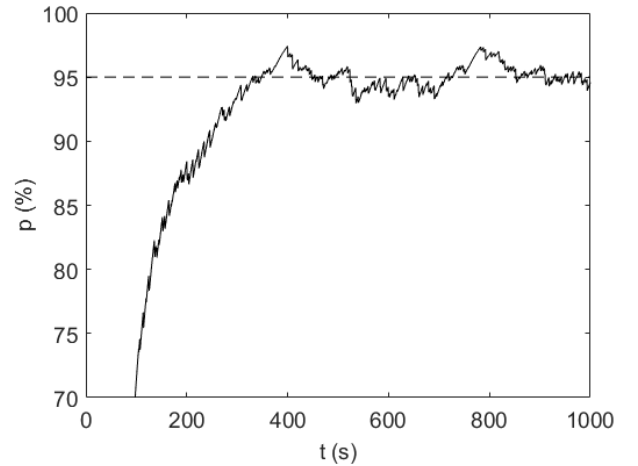


Fig 7: Probability of meeting flow demand (filtered with a 50 s time constant). The target value is 95%.

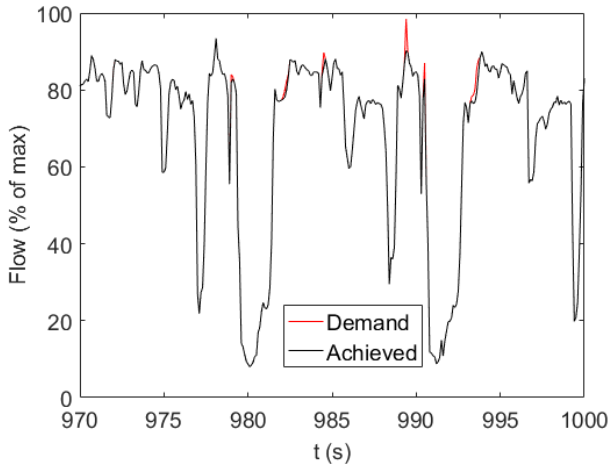


Fig 8: Flow demand and the achieved performance.

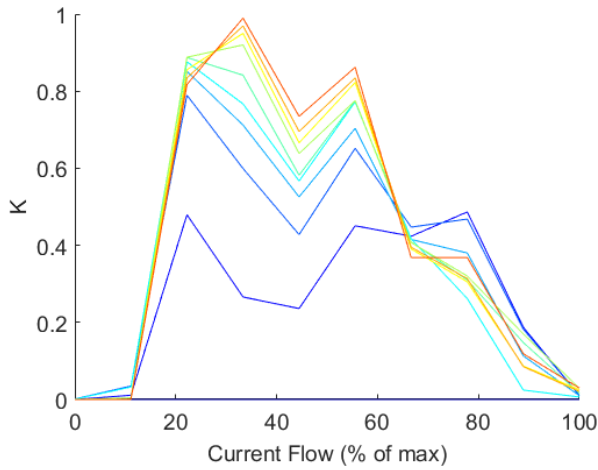


Fig 9: Evolution of $K(Q)$ lookup table, plotted at ten evenly spaced points in time between $t=0$ (blue) and $t=1000$ s (red).

The effect of the \hat{p} tuning parameter is shown in Fig 10 and 11. Fig 10 shows the final adapted engine speed vs current flow demand lookup table after running for 1000 s on the same excavator loading cycle. The blue line ($\hat{p} = 80\%$) shows a more aggressive reduction in engine speed (at the cost of not meeting flow demand approximately 20% of the time), while the red line ($\hat{p} = 99\%$) shows the effect of more priority applied to meeting the flow demand. Figure 11 shows the resultant mean engine speed over the cycle, demonstrating the tradeoff of flow performance with engine speed reduction.

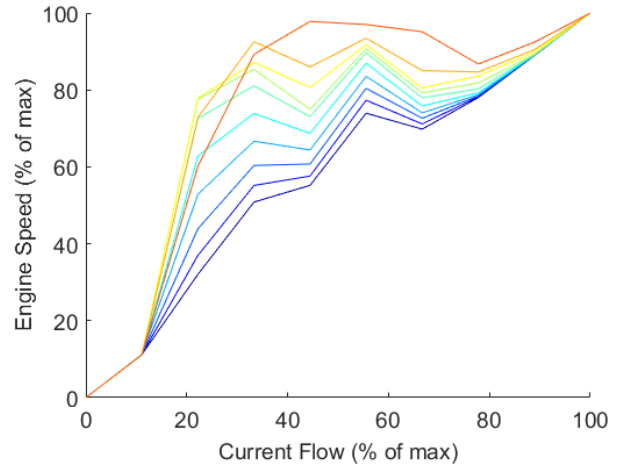


Fig 10: Engine speed based on the adapted $K(Q)$ lookup table after 1000 s of running the excavator work cycle, showing ten evenly spaced values of \hat{p} between 80% (blue) and 99% (red).

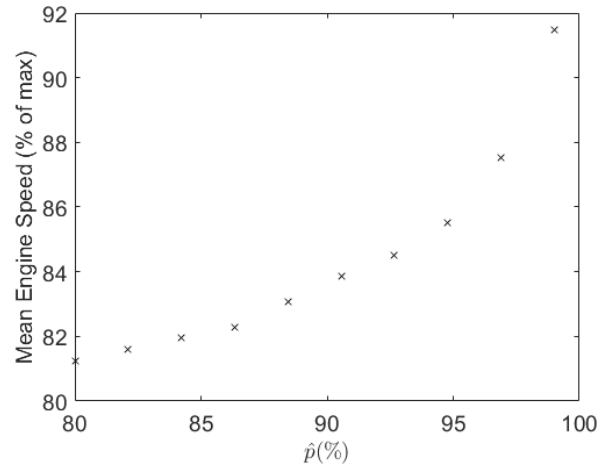


Fig 11: Effect of the \hat{p} tuning parameter on the mean engine speed.

The adaptation dynamics also have an effect on the controller. Figures 12 and 13 show the effect of the controller adaptation time scale, Δt_{adapt} , which scales the adaptation gains. This parameter controls how quickly the controller adapts to change in operating style and also how long it “remembers” a prior operating style. These plots show Δt_{adapt} varying from 10 s (blue) to 500 s (red), after the controller is initialized at $t = 0$. Clearly the fastest settings are tracking the short-term operator inputs rather than learning and predicting the behavior. It is also possible that the controller exhibits some nonlinear instability for very fast settings, although this has not been studied (it appears that the lowest Δt_{adapt} would be set by predictive ability rather than stability in any case). At higher settings, the response is not particularly sensitive to the value of Δt_{adapt} , and a general value can likely be used, with little need for fine tuning.

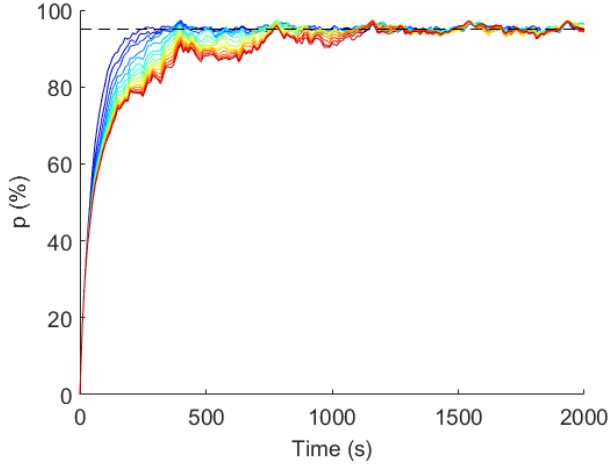


Fig 12: Likelihood of meeting flow demand, for Δt_{adapt} varying from 10 s (blue) to 500 s (red). The desired value of 95% is shown by the dashed line.

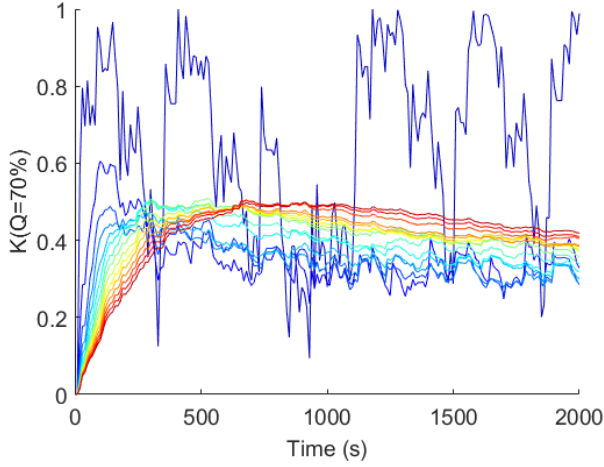


Fig 13: Adaptation dynamics for the same varying adaptation times as Fig 12, in this case showing the K value for 70% flow.

4 Conclusions

The presented results show the potential for reducing engine speed while not significantly reducing the useful work output of an excavator. In this case, the engine speed may be reduced by a maximum of 34% relative to a constant engine speed capable of doing the same work, which would be expected to provide significant fuel savings.

We also present a controller that predicts the future flow demand based on the current flow, and can automatically adapt to changes in operator style. This controller attempts to meet a preset desired probability of meeting flow demand by adjusting a lookup table relating the required engine speed to the current flow demand. Simulation studies suggest that this controller behaves as expected on a simple engine model. Future work will apply this controller to a real engine including pump dynamics, and will consider its specific BSFC curve.

Nomenclature and Base Parameters

Table 1: Nomenclature and Selected Simulation Parameters

Symbol	Denotation	Unit	Base Value
F_i	Force	N	variable
d	Estimated engine lag	sample s	500
D	Pump displacement	m ³ /rev	variable
D_e	Excess displacement	m ³ /rev	variable
D_{max}	Maximum pump displacement	m ³ /rev	100 × 10 ⁻⁶
J	Inertia of pump and engine	kg m ²	1.723
K	Speed adjustment factor		variable
N	Number of points in $K(Q)$ table		10
\hat{p}	Desired probability of meeting flow demand		0.95
P_s	Pump pressure	Pa	variable
T_e	Engine torque	Nm	variable
T_e	Engine torque	Nm	variable
T_{max}	Max engine torque	Nm	Fig 1
T_p	Pump torque	Nm	variable
Q	Pump flow	m ³ /s	variable
\hat{Q}	Pump flow demand	m ³ /s	variable
ΔK^+	Control adaptation gain	1/s	0.010
ΔK^-	Control adaptation gain	1/s	0.190
Δt_{adapt}	Control adaptation time	s	100
Δt_{turbo}	Turbocharger rise time	s	1.0
Δt	Sample time	s	1 × 10 ⁻³
ω	Shaft speed	rad/s	variable
ω_{max}	Maximum shaft speed	rad/s	230.4
ω_{min}	Minimum shaft speed to meet flow demand	rad/s	variable

References

- [1] Goering, C., Stone, M., Smith, D., and Turnquist, P., 2003. *Off-road Vehicle Engineering Principles*. American Society of Agricultural Engineers, St. Joseph, MI.
- [2] Bennick, C. 2012. "Excavators: Get to Know the Working Modes" *Equipment Today*, Sept 2012.
- [3] Kim, H., Choi, J. and Yi, K., 2012. Development of supervisory control strategy for optimized fuel consumption of the compound hybrid excavator. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 226(12), pp.1652-1666.
- [4] Lin, X., Pan, S.X. and Wang, D.Y., 2008. Dynamic simulation and optimal control strategy for a parallel hybrid hydraulic excavator. *Journal of Zhejiang University-SCIENCE A*, 9(5), pp.624-632.
- [5] Hippalgaonkar, R. and Ivantysynova, M. 2015. Optimal Power Management for DC Hydraulic Hybrid Multi-Actuator Machines - Part 1: Theoretical Studies, Modeling and Simulation. *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 138, Issue 5
- [6] Hippalgaonkar, R. and Ivantysynova, M. 2015. Optimal Power Management for DC Hydraulic Hybrid Multi-Actuator Machines - Part 2: Machine Implementation and Measurement. *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 138, Issue 5
- [7] Wiens, T. and Bitner, D., 2016. "An Efficient, High Performance and Low-Cost Energy Recovering Hydrostatic Linear Actuator Concept. *Proceedings of the 2016 Bath/ASME Symposium on Fluid Power and Motion Control*, Bath, UK.
- [8] Perkins Engines Company Limited, 2005, "Technical Data 1100 Series", Publication No TDP 1502E, Peterborough, UK.
- [9] Forsythe, G. E., M. A. Malcolm, and C. B. Moler, 1976, *Computer Methods for Mathematical Computations*, Prentice-Hall.