L4: identification of the engine requirements for a wheel loader

P. Santana González

***Abstract*—This work aims to find a model for the engine speed of a wheel loader using a small dataset of time-series data of various variables (features), not all of which may be relevant towards predicting the required engine speed. In the end, it turns out that a model with very few input features can still produce a significantly representative prediction model. Learned how to approach a black-box identification problem and use MATLAB-implemented tools -both using a graphical user interface (GUI) toolbox and the command-line interface (CLI)- to solve it.**

I. Results

A. Model selection

First, the data is loaded and normalized per feature. Then, it is split in a train and validation test (10,000 + 4,000 split). In a real case scenario, only the train data should have been used to create a normalization model, which would then be applied to the validation (we would normalize the validation set using the mean and variance of the train set). However, because the data consists on two consecutive cuts of time-series measurements, the differences between validation and training data are expected to be almost negligible and how they are normalized is not considered of relevance.

Once the splits are made, various ARX and an ARMAX model are computed for different system orders, while using the same order for each feature (i.e. a 4th order system would go as far as z^{-4} in all of the variables) and including all features. Figure 1 shows various mean square errors (MSEs) for these two models and different system order values. From this quick experiment it is observed that the ARX model performs as good as the ARMAX taking way less time to compute, so since it is not worth to consider more complex models ARX will be kept as model of choice. It is also noted that the optimal model order when using all features is around 2.

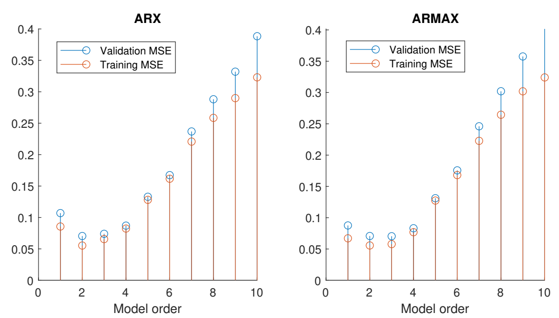

Fig. 1: MSEs for ARX ARMAX models using all features and different system orders (orders were kept the same for all features).

	Cov(each,engine_rps)	Idx
throttle	0.76647	8
digging	0.34708	12
force_lift	0.3098	5
boom_trate	0.24191	2
bucket_trate	0.1926	4
tilt_joystick	0.16809	9
drive_revs	0.13519	10
bucket_angle	-0.08251	3
boom_angle	0.073224	1
gear_index	0.053809	11
lift_joystick	-0.018198	7
force_tilt	0.013993	6

Fig. 2: Correlation between each feature the output Engine RPS, sorted from highest to lowest. The column Idx (index) only serves to identify each feature. The higher a feature on the table, the more representative it is when predicting the output.

B. Feature selection

Since it is desired to identify the most impactful parameters on the engine rps, dimensionality reduction techniques such as principal component analysis (PCA) are not desirable, since they will just transform the features into new ones but require, after all, all of them at the beginning of the algorithm, so this approach wouldn't help finding the most relevant ones as clearly.

For making an hypotheses on which features to ignore and test model regression, two tools are used. First, a covariance matrix among all features is computed: features with high common covariance are highly correlated and thus it could be that one of them holds most of the information, so it wouldn't hurt the model to remove the other. Then, a vector holding the values of covariance between each feature and the engine rps is computed, as shown in figure 2: the higher this covariance the more correlated input and output are and thus the more relevant that feature would be towards predicting the rps.

After running some experiments it is seen that these assumptions hold. In the end, a series of ARX-221 models are computed removing one feature at a time according to the order of relevance told by said covariance vector. As shown in 4, after the removal of each feature, the MSE of the model slightly increases. The difference is not too notorious, so a compromise should be assumed when deciding how many variables to keep. While a 12-feature model like this is not heavy to compute (the available machine can make a prediction of the whole available data in less than a second,

	ARX221 MSE
Baseline	0.070453
Remove highest cov(feat, y)	0.73739
Remove lowest cov(feat, y)	0.073614
Keep only 3 highest cov(feat, y)	0.081768
Remove 3 lowest cov(feat, y)	0.077759

Fig. 3: Experimental MSE results for some experiments removing some features, all of them for an ARX-221 model space. Baseline is the best validation MSE obtained in the experiments depicted in figure 1. The references of "highest" and "lowest" point to the correlation hierarchy depicted in figure 2.

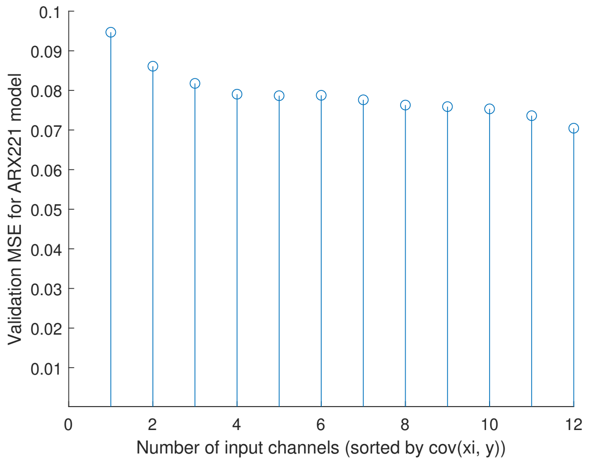

Fig. 4: MSE of the model ARX-221 when using only some of the available features. The number in the X axis tells how many features were used; they were picked in descending order from 2. For example, for X=5 the model considered the features (8,12,5,2,4).

while this would happen over several minutes in real life), it is usually desired to work with small models.

Taking it to the extreme, an ARX-221 model consisting only of one single feature (throttle), can already make quite accurate predictions on the RPS. However, not to oversimplify, in the end the 3 top correlation features were included. The final proposed model of this paper is an ARX-221 including features (8,12,5), showing a MSE of 0.082 on the validation dataset. An example of its performance is displayed in figure 5.

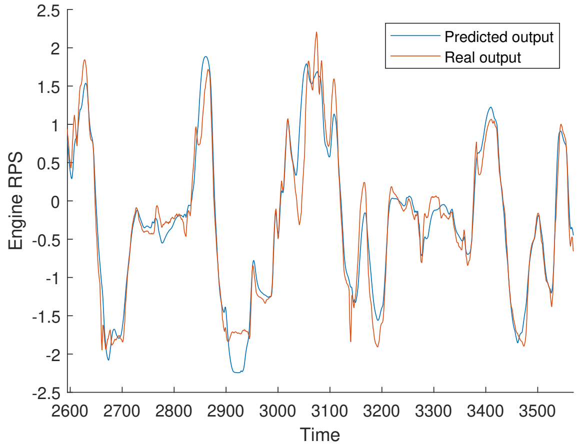

Fig. 5: Predicted vs. real engine RPS, using validation dataset. Computed with an ARX-221 model using only features 8,12,5.

II. Conclusions

Feature selection was easily successfully achieved by using correlation between each feature the output. While a covariance matrix was also computed in the end it proved easier to follow just the former. In the end, it was found that a simple ARX-221 with a single feature was enough to obtain a reasonable prediction on the output. It was also seen that the biggest pitfalls of the model happen anyway even including more features, which only slightly improve the small-size discrepancies between prediction and validation data. In the end, the 3 most relevant features were picked for a model proposal: they were found to be the Throttle, the Digging signal the Lift Force. Adding more features seems to reduce the bias of the model, which is already low enough not to be an issue of interest.

Acknowledgments: Thanks to Robert and Martin.