ADAPTIVE MODELLING AND PREDICTIVE CONTROL OF AN IC ENGINE

S. W. Wang*, D. L. Yu*, J. B. Gomm*, M. Beham, G. F. Page* and S. S. Douglas***

** Control Systems Research Group, School of Engineering,*
Liverpool John Moores University, Byrom Street, Liverpool, L3 3AF, UK
Email:D.Yu@livjm.ac.uk
*** BMW AG Group, Munich, Germany*

Abstract: This paper presents modelling of internal combustion (IC) engine with adaptive neural networks. A radial basis function network model with both centres and weights adapted and a model with only weights adapted are compared with a fixed parameter model. The developed models are used in model based predictive control (MPC) to form an adaptive nonlinear MPC scheme and applied to engine speed tracking control. The modelling and control are based on a generic mean value engine model and consists of three submodels that describe the fuel mass flow dynamics, the intake manifold filling dynamics and the crankshaft speed. Adaptive MPC is shown superior over the fixed parameter model based control. *Copyright © 2005 IFAC*

Keywords: engine modelling, engine control, adaptive neural networks, model predictive control

1. INTRODUCTION

Automotive IC (Internal Combustion) engine control is one of the most complex control problems for control system engineers and researchers. Due to the increasing requirements of governments and customers, car manufacturers always try to reduce substantially emissions and fuel consumption while maintaining the best engine performance. To satisfy these requirements, a variety of variables need to be controlled, such as engine speed, engine torque, spark ignition timing, fuel injection timing, air intake, air-fuel ratio and so on. These variables are complicatedly related to each other. Moreover, car engines have several different operating modes including start up, idle, running and braking. Engine dynamics are severely non-linear and multivariable because of these factors.

From an enormous amount of research about the modelling and control of automotive engines, it has been shown that neural networks not only provide a simple model structure, but also capture the inherent nonlinearities and the dynamics of automotive engines with satisfactory accuracy (De Nicolao, *et al.*, 1996; Isermann, *et al.*, 2001; Tan, *et al.*, 2000; Vinsonneau, et al., 2003;). For highly nonlinear and uncertain systems, neural network models with fixed parameters are not sufficiently accurate. Instead, adaptive neural networks are more appropriate for this kind of modelling problems (Chen, *et al.*, 1992). In this paper, the dynamic modelling abilities of fixed and adaptive neural networks for IC engines are tested and analysed.

From the control point of view, MPC, as a recent developed practical approach, is a potential control strategy for many complex nonlinear systems. In order to obtain the best control performance for this model-based control strategy, it is essential to choose a model as accurate as possible. This paper presents a MPC application based on fixed and adaptive neural networks. The purposes are to investigate the control ability of MPC for IC engines and analyse the suitability of these networks for this control strategy.

The paper consists of the following. Section 2 introduces an engine simulation package and the corresponding modification. Section 3 discusses three neural network modelling methods of car engines, using the knowledge derived from Section 2. The modelling performance of the different methods in uncertain conditions are compared and analysed. Section 4 provides a neural network model based predictive control strategy for the engine crankshaft speed and the control results are discussed. Conclusions and remarks are reported in Section 5.

2. THE ENGINE SIMULATION

An engine simulation has been provided by Hendricks (2000). It is called a generic mean value engine model and consists of three submodels that describe the fuel mass flow dynamics, the intake manifold filling dynamics and the crankshaft speed. The latter two submodels were used in this paper. All the variables in this section are defined in the notation.

2.1 The intake manifold filling dynamics

The intake manifold filling dynamics is analysed from the viewpoint of the air mass conservation inside the intake manifold. It includes two nonlinear differential equations about the manifold pressure and the manifold temperature.

The manifold pressure is mainly a function of the air mass flow past throttle plate, the air mass flow into the intake port, the EGR (Exhaust Gas Recirculation) mass flow, the EGR temperature and the manifold temperature. It is described as

$$\dot{p}_i = \frac{\kappa R}{V_i}\left(-\dot{m}_{ap}T_i + \dot{m}_{at}T_a + \dot{m}_{EGR}T_{EGR}\right) \qquad (1)$$

The manifold temperature can be computed using the following differential equation.

$$\dot{T}_i = \frac{RT_i}{p_i V_i}\begin{bmatrix} -\dot{m}_{ap}(\kappa-1)T_i + \dot{m}_{at}(\kappa T_a - T_i) \\ +\dot{m}_{EGR}(\kappa T_{EGR} - T_i) \end{bmatrix} \qquad (2)$$

Here, the EGR mass flow was not considered and simply set to be zero. The air mass flow past throttle plate is related with the throttle position and the manifold pressure. The air mass flow into the intake port is a function of the crankshaft speed and the manifold pressure. Then the manifold pressure can be described by an input-output mapping function.

$$\dot{p}_i = f(u, p_i, n, T_i) \qquad (3)$$

2.2 The crankshaft speed

The crankshaft speed is derived based on the conservation of the rotational energy on the crankshaft.

$$\begin{aligned}\dot{n} = &-\frac{1}{In}\left(P_f(p_i,n) + P_p(p_i,n) + P_b(n)\right) \\ &+\frac{1}{In}H_u\eta_i(p_i,n,\lambda)\dot{m}_f(t-\Delta\tau_d)\end{aligned} \qquad (4)$$

Both the friction power k and the pumping power k are related with the manifold pressure p_i and the crankshaft speed n. The load power P_b is a function of the crankshaft speed n only. The indicated efficiency η_i is a function of the manifold pressure p_i, the crankshaft speed n and the air/fuel ratio λ. The engine port fuel mass flow $\dot{m}_f$ is described by the following equation.

$$\dot{m}_f = \frac{\dot{m}_{ap}}{\lambda L_{th}} \qquad (5)$$

Similarly, the crankshaft speed can be derived by another input-output mapping function.

$$\dot{n} = f(p_i, n) \qquad (6)$$

2.3 Introducing uncertainty for the engine simulation

In order to analyse the performance of different adaptive engine models in uncertainty conditions, modification is made for the engine simulation to introduce the uncertainty of ambient temperature, which is uniformly distributed between –35 to +35 degrees centigrade.

3. NEURAL NETWORK MODELLING

3.1 Data collection

A set of Random Amplitude Signals (RAS) shown in Fig. 1 was designed for the throttle position to obtain a representative set of input-output data. The range of this excitation signal was bounded between 20 and 40 degrees. The length of each amplitude was set to be 5 seconds which is 10 times the sampling time of the simulation system. Three sets of data samples, including the crankshaft speed, the intake manifold pressure and temperature, were collected for the neural network training and testing. Each set contains 2000 data samples.

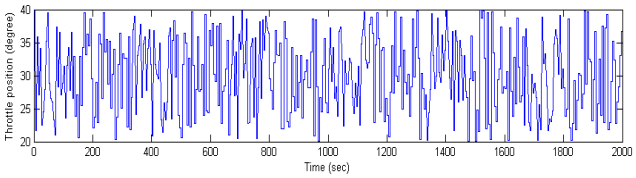

Fig. 1. RAS of the throttle position

3.2 Neural network structure

As shown in Fig. 2, the RBF neural network was chosen to construct a second-order engine model with four inputs and two outputs.

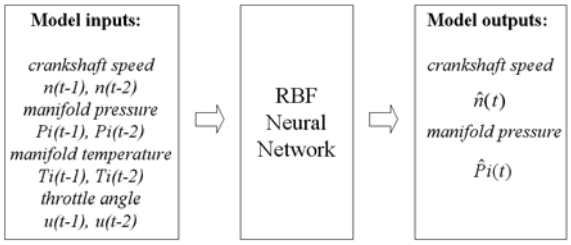

Fig. 2. The structure of RBF neural network

The ith output of the neural network model at time t is

$$y_i(t) = \sum_{j=1}^{nh} \phi_j(t) w_{ji} \tag{7}$$

where $\phi_j(t)$ is the jth activation function and w_{ji} is the weight. The Gaussian RBF, which has the form of the multi-dimensional bell-shaped curve, was used as the activation functions.

$$\phi_j(t) = \exp\left(-\frac{\left\| x(t) - c_j(t) \right\|^2}{\sigma_j^2} \right) \tag{8}$$

where $x(t)$ is RBF neural network inputs, $c_j(t)$ is the jth centre and σ_j is the corresponding width.

3.3 The three training algorithms

In order to obtain the best modelling performance and analyse the modelling ability, three different training algorithms were used.

Algorithm 1:
BLS (Batch Least Squares) + K-means clustering

This algorithm is widely used for off-line training. The centres are set by the K-means clustering method. The widths are computed by the p-nearest neighbours method. The batch least squares method is responsible for training the weights *W* by using equation (9).

$$W = \left(\Phi^T \Phi \right)^{-1} \Phi^T Y \tag{9}$$

where W is the matrix of weights, Φ is the matrix of activation function outputs and Y is the matrix of training targets.

Algorithm 2:
RLS(Recursive Least Squares) + K-means clustering

The recursive least squares method is used for on-line training. It helps to construct on-line system models which should be more accurate and flexible than the off-line models. The equations of this algorithm are summarized as follows (Ljung, 1999).

$$L(t) = \frac{P(t-1)\varphi(t)}{\lambda(t) + \varphi^T(t) P(t-1) \varphi(t)} \tag{10}$$

$$\hat{w}(t) = \hat{w}(t-1) + L(t)\left[y(t) - \varphi^T(t) \hat{w}(t-1) \right] \tag{11}$$

$$P(t) = \frac{1}{\lambda(t)} \left[P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda(t) + \varphi^T(t) P(t-1) \varphi(t)} \right] \tag{12}$$

$\lambda(t)$ is set to be 0.985 and the initial conditions are

$$\hat{w}(0) = 2.22 \times 10^{-16} \times U_{nh \times 2}$$
$$P(0) = 1 \times 10^4 \times I_{nh \times nh}$$

where I is unit matrix and U represents a matrix whose components are one.

At each sampling time, the parameters $L(t)$, $\hat{w}(t)$ and $P(t)$ are updated orderly with the activation function $\varphi(t)$. Hence, in this algorithm, only the weights are adapted recursively, the centres are fixed after running the K-means clustering program for a certain set of training data.

Algorithm 3:
RLS (Recursive Least Squares) + recursive K-means clustering

The difference between algorithm 3 and algorithm 2 lies in the adaptation of the centres. In algorithm 3, instead of using the fixed centres, the centres are adapted in each sampling period by the recursive K-means clustering method (Chen, *et al.*, 1992). It helps to realise real on-line modelling especially when the off-line training data for the K-means clustering is not available. To adapt the centres recursively, two procedures should be followed.

(i) Compute distances $a_i(t)$ between the sampled input data $x(t)$ and each centre $c_i(t-1)$. Find a minimum distance and use *k* to represent its index.

$$a_i(t) = \left\| x(t) - c_i(t-1) \right\| \tag{13}$$
$$k = \arg\left[\min\left\{ a_i(t) \right\} \right] \tag{14}$$

(ii) Update the corresponding centre

$$c_i(t) = c_i(t-1) \quad 1 \le i \le nh \tag{15}$$
$$c_k(t) = c_k(t-1) + \alpha_c(t)\left(v(t) - c_k(t-1) \right) \quad i \ne k \tag{16}$$

The learning rate is

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1+\text{int}[t/nh]^{1/2}} \qquad (17)$$

where int[ ] means the integer part of the argument.

3.4 Comparison of the modelling results

The three training algorithms mentioned in section 3.2 were used for a RBF neural network to model the intake manifold pressure and the crankshaft speed of the simulation engine. 11 hidden nodes were chosen for the second-order RBF model. First 1000 data samples (1~1000) were used for training and another 1000 data samples (1001~2000) were used for testing. The three sets of modelling results are plotted in Fig. 4, 5 and 6. The corresponding MAE (Mean Absolute errors) of the modelling outputs are shown in Table 1.

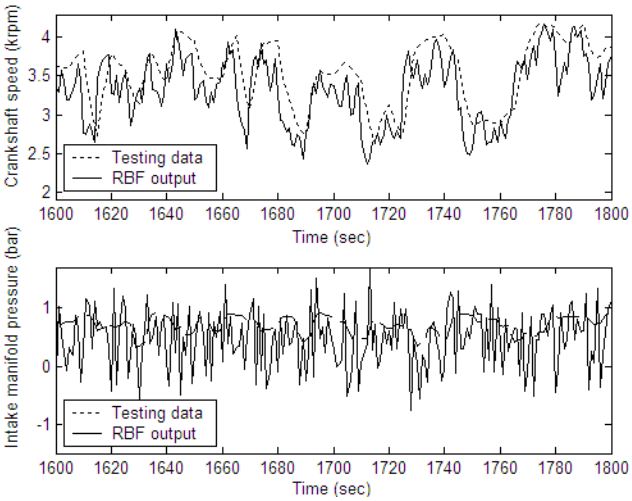

Fig. 4. Modelling results of algorithm 1 (Displayed data: 1600~1800)

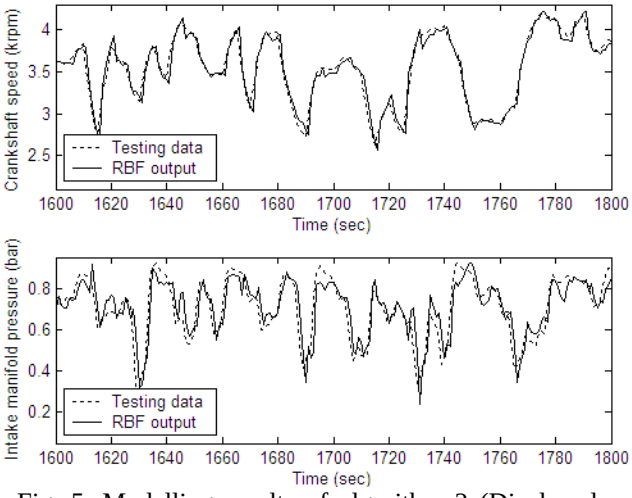

Fig. 5. Modelling results of algorithm 2 (Displayed data: 1600~1800)

For the three training methods, algorithm 1 has the biggest modelling error, which is because of its limited modelling ability for uncertainty conditions. Compared with algorithm 1, the modelling errors of algorithms 2 and 3 are significantly reduced, which shows that the adaptive neural networks are more suitable for modelling uncertain IC engines.

From Fig. 5 and 6, it can be seen that the manifold pressure are more difficult for modelling compared with the crankshaft speed. This is because it has more complex dynamics especially in uncertain operating conditions. The different dynamics of the two variables determine that the performance of

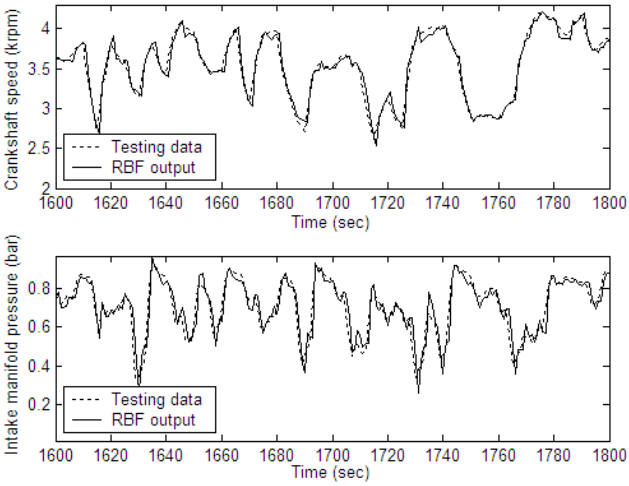

Fig. 6. Modelling results of algorithm 3 (Displayed data: 1600~1800)

Table 1. MAE of modelling outputs

Training algorithm	MAE (n)	MAE (Pi)
Algorithm 1	0.2571	0.4517
Algorithm 2	0.0605	0.0508
Algorithm 3	0.0594	0.0408

algorithm 3 is better than algorithm 2 for the manifold pressure, but for the crankshaft speed, there is not a significant improvement. It illustrates that algorithm 2 can work well for the variables with small variations and algorithm 3 is more appropriate than algorithm 2 for modelling the variables with complex dynamics in uncertain conditions.

4. MPC FOR THE CRANKSHAFT SPEED

Based on the obtained neural network models in Section 3, a predictive control strategy was realised for the crankshaft speed control.

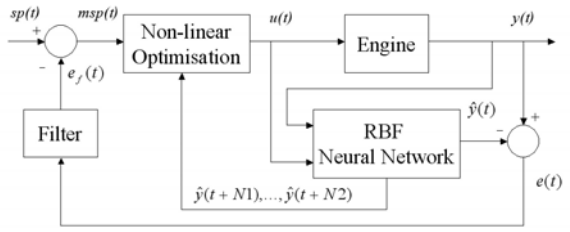

Fig. 7. The neural network model-based predictive control strategy

As shown in Figure 7, the neural network is used to predict the engine output for N2 steps ahead. The nonlinear optimiser minimises the errors between the set point and the engine output by using the cost function,

$$J = \sum_{i=t+N1}^{t+N2} \left[r(i) - y_{nn}(i)\right]^2 + \lambda \sum_{i=t}^{t+Nu} \left[u(i) - u(i-1)\right]^2 \quad (19)$$

N1 and *N2* define the prediction horizon. λ is a control weighting factor which penalises excessive movement of the control input, the throttle position *u(t)*. *Nu* is the control horizon.

Three engine models obtained in section 3 were used in this MPC strategy. The parameters of the nonlinear optimisation was chosen as *N1=1, N2=4, λ=0.5, Nu=0.* Three sets of control results and tracking errors for different set points are shown in Figure 8, 9 and 10.

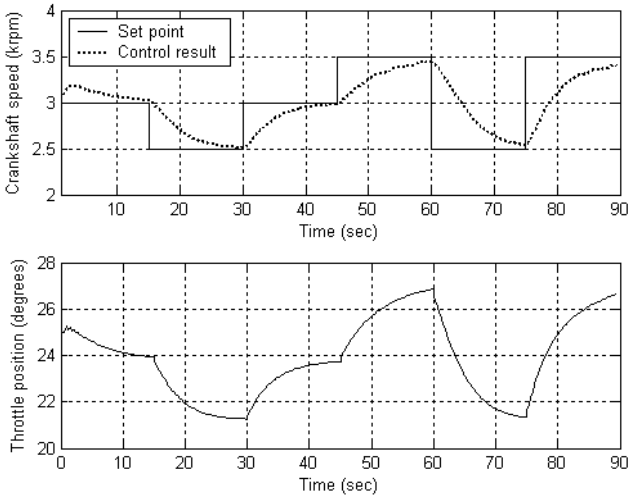

Fig. 8: Crankshaft speed control result based on algorithm 1 (Tracking error: 0.2525)

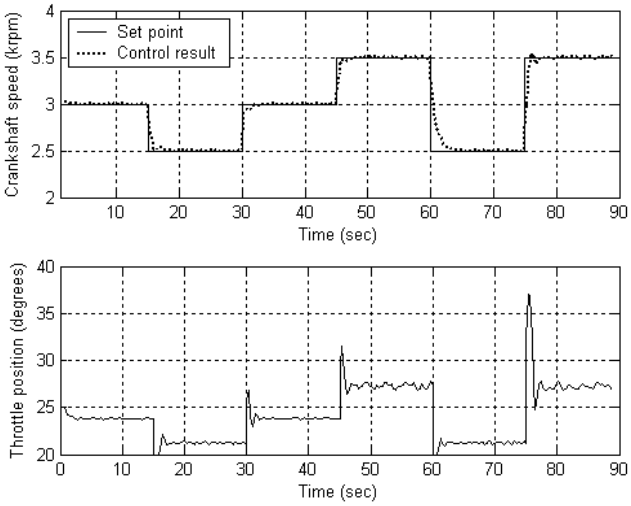

Fig. 9: Crankshaft speed control result based on algorithm 2 (Tracking error: 0.0447)

From Fig. 8, it can be seen that even the MPC controller based on the inaccurate model trained by algorithm 1 cannot control the crankshaft speed to be desirable values. In Fig. 9 and 10, the control results have been improved significantly, which benefits from their more accurate engine models. However, the improvement of the control result in Fig. 10 is not obvious enough compared with the result in Fig. 9, although its tracking error is smaller. This is due to

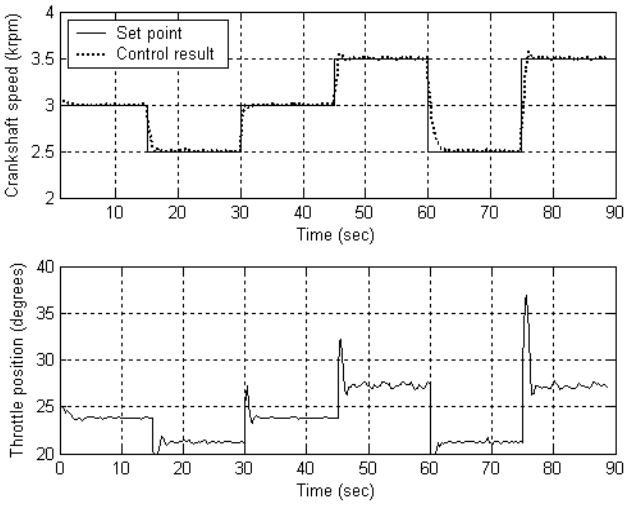

Fig. 10: Crankshaft speed control result based on algorithm 3 (Tracking error: 0.0439)

the strong control ability of the MPC controller covers the inaccuracy of the model trained by algorithm 2. Nevertheless, further comparison should be made by using real engine data which contains more complexities and uncertainties.

5. CONCLUSIONS

This paper has compared the IC engine modelling ability of RBF neural networks trained by three algorithms in uncertain conditions. The control performance of MPC controllers based on the three

engine models is also analysed. Some conclusions can be obtained as follows.

(1) The adaptive RBF neural networks trained by recursive least square method with fixed or recursive centres are appropriate for modelling IC engines in uncertain operating conditions.

(2) The MPC strategy based on adaptive RBF neural network models can achieve good crankshaft speed control results. It is a potential control method for controlling other engine variables such as the ignition angle and the fuel injection angle to improve current engine control strategies.

(3) It is needed to do further investigation about the modelling ability and the suitability for MPC of training algorithm 2 and 3 in more uncertain and complex operating conditions.

NOTATION OF THE ENGINE SIMULATION

p_i	manifold pressure (bar)
P_b	load power (kW)
H_u	fuel lower heating value (kJ/kg)
k	pumping power (kW)
$\dot{m}_{at}$	air mass flow past throttle plate (kg/sec)
k	friction power (kW)
T_a	ambient temperature (degrees Kelvin)
$\dot{m}_{EGR}$	EGR mass flow (kg/sec)
$\dot{m}_{ap}$	air mass flow into intake port (kg/sec)
u	throttle position (degrees)
T_{EGR}	EGR temperature (degrees Kelvin)
n	crankshaft speed (krpm)
I	crank shaft load inertia (kg m^2)
η_i	indicated effiency
$\dot{m}_f$	engine port fuel mass flow (kg/sec)
λ	air/fuel ratio
T_i	intake manifold temperature (degrees Kelvin)
$\Delta\tau_d$	injection torque delay time (sec)
L_{th}	stoichiometeric air/fuel ratio (14.67)
κ	ratio of the specific heats=1.4 for air
R	gas constant (here 287×10^{-5})
V_i	manifold + port passage volume (m^3)

REFERENCES

Chen S., S.A. Billings and P. M. Grant. (1992). Recursive hybrid algorithm for non-linear system identification using radial basis function networks. *Int. J. Control* **55**(5), 1051-1070.

De Nicolao G., Scattolini R. and Siviero C. (1996). Modelling the volumetric efficiency of IC engines: parametric, non-parametric and neural techniques. *Control Eng. Practice* **4**(10), 1405-1415.

Hendricks, E., (2000). A Generic Mean Value Engine Model for Spark Ignition Engines. *41st Simulation Conference, SIMS 2000, DTU, Lyngby, Denmark.* also available at www.iau.dtu.dk/~eh/

Isermann R. and Muller N. (2001). Modelling and Adaptive Control of Combustion Engines with Fast Neural Networks. *European Symposium on Intelligent Technologies, Hybrid Systems and their Implementation on Smart Adaptive Systems, Tenerife, Spain.*

Ljung Lennart, (1999). *System Identification—Theory for the User*. 361-369. Prentice-Hall.

Tan Yonghong and Mehrdad Saif, (2000). Neural-networks-based nonlinear dynamic modelling for automotive engines. *Neurocomputing*, **30**, 129-142.

Vinsonneau J.A.F., Shields D,N., King P.J. and Burnham K.J. (2003). Polynomial and neural network spark ignition engine intake manifold modelling. *Proc. 16th Int. Conf. on Systems Engineering, ICSE'*, **2**, 718-723.