

## **Design of ON-Line Tuned Idle Speed Controller for an Automotive Engine By Using NCD**

**Ali Majeed Mahmood\***

**e-mail:** [alimmj81@yahoo.com](mailto:alimmj81@yahoo.com)

Received on: 11/7/2011

Accepted on: 22/11/2011

**Abstract:** This paper attempts to tune any controller without the knowledge of mathematical model for the system to be controlled. For that purpose, the optimization algorithm of MATLAB / Nonlinear Control Design Blockset (NCD) is adapted for On-line tuning for controller parameters. To present the methodology, a PID controller is verified with the physical plant using the engine speed control System where the problem of maintaining the engine idle speed at a reference value with the applied load is studied. A Proportional Integral Derivative (PID) Controller is used to solve this problem, but to get the best controller parameters the (NCD) Blockset is used for tuning the PID parameters. Simulation shows promising results in the idle speed response by comparing NCD tuning results with the trial and error results. The analytical results are carried out MATLAB / SIMULINK.

**Keywords:** PID controller, NCD Blockset, Automotive Engine, Idle speed.

---

\* Control and Systems Engineering Department / University of Technology

## 1. Introduction

Idle speed is the rotational speed the engine runs on when throttle pedal isn't pressed measured in revolutions per minute, or rpm, of the crankshaft. At idle speed, the engine generates enough power to run reasonably smoothly and operate the alternator, water pump and power steering, but If the engine is operating additional number of accessories, particularly air conditioning, the idle speed must be increased to ensure that the engine generates enough power to run smoothly and operate the accessories. Most air conditioning-equipped engines have an automatic adjustment feature in the carburetor or fuel injection system that raises the idle when the air conditioning is running [1]. Idle Speed Control is one of the common problems in automotive engines,. Several controllers applied for idle speed, in 1999 DaeEun Kim and Jaehong Park [2] used Neural network control for reducing engine speed fluctuation at idle It is suggested in this paper that the idle speed control system should use a nonlinear function mapping control with the feature extraction about engine speed fluctuation, in 2005 Elena De Santis et al [3] used digital idle speed control of Automotive Engine using hybrid models. In 2008, TERUJI SEKOZAWA [4] proposed an engine model-based feed-forward idle control system. The model-based control system is demonstrated to achieve better control results. The control unit continually monitors the engine state using several sensors and selects better control actions to achieve what is demanded from an engine under different defined operating modes. P.K. Wong et al. [5] in 2008 applies automotive engine idle speed control optimization using least squares support

vector machine and genetic algorithm. In 2009 Pavkovic, D. et al [6] used Adaptive Kalman Filter-Based Load Torque Compensator for Improved SI Engine Idle Speed Control The adaptive estimator is characterized by a fast response and good noise suppression ability. In 2010 Xue Jinlin [7] used Reinforcement Learning for Engine Idle Speed Control A control method of neural network controller with reinforcement learning is proposed to implement idle speed control of an automobile engine to reduce fluctuation of the idle speed.

## 2. Electronic Control Unit (ECU)

In automotive electronics, ECU is a generic term for any embedded system that controls one or more of the electrical systems or subsystems in a motor vehicle. It does this by reading values from a multitude of sensors within the engine, interpreting the data using multidimensional performance maps (called Look-up tables), and adjusting the engine actuators accordingly. Sometime referred to as the car's computer as shown in Figure 1. Before ECUs, air/fuel mixture, ignition timing, and idle speed were mechanically set and dynamically controlled by mechanical and pneumatic means [1], [2].



Figure 1: Electronic Control Unit (ECU)

## 2.1 Functions of ECU

- 1) Control of idle speed
- 2) Electronic valve control
- 3) Control of Air/Fuel ratio
- 4) Control of variable valve timing
- 5) Control of ignition timing

## 2.2 Control of idle speed

Most engine systems have idle speed control built into the ECU. The engine RPM is monitored by the crankshaft position sensor which plays a primary role in the engine timing functions for fuel injection, spark events, and valve timing. Idle speed is controlled by a programmable throttle stop or an idle air bypass control stepper motor. Early carburetor-based systems used a programmable throttle stop using a bidirectional DC motor. Early TBI systems used an idle air control stepper motor. Effective idle speed control must anticipate the engine load at idle. Changes in this idle load may come from power steering systems, power brake systems, and electrical charging and supply systems. Engine temperature and transmission status, and lift and duration of camshaft also may change the engine load and/or the idle speed value desired. A full authority throttle control system may be used to control idle speed, provide cruise control functions and top speed limitation [3].

## 3. Nonlinear Control Design (NCD) Blockset [8]

The (NCD) is a MATLAB tool that helps to tune design parameters in a nonlinear Simulink model by optimizing time-based signals to meet user-defined constraints by graphically placing constraints within a time-domain window. The NCD Blockset automatically converts time domain

constraints into a constrained optimization problem and then solves the problem using the optimization routines taken from the Optimization Toolbox. The constrained optimization problem formulated by the NCD Blockset iteratively calls for simulations of the Simulink system, compares the results of the simulations with the constraint objectives, and uses gradient methods to adjust tunable parameters to better meet the objectives. The NCD Blockset allows introducing uncertainty into plant dynamics, specify lower and upper limits on tunable parameters, and alter termination criterion. The progress of an optimization while the optimization is running can be followed from command window, and the final results are available in the MATLAB workspace when an optimization is complete. Intermediate results are plotted after each simulation. It allows the user to terminate the optimization before it has completed, to retrieve the intermediate result or change the design. NCD uses optimization algorithms to find parameter values that allow a feasible solution to the given constraints. NCD automatically converts the constraint bound data and tunable variable information into a constrained optimization problem. Basically, the NCD Blockset attempts to minimize error and generates constraint errors at equally spaced time points (with spacing given by the Discretization interval defined in the Optimization Parameters dialog box) beginning at the simulation start time and ending at the simulation stop time. For upper bound constraints, it is defined the constraint error as the difference between the simulated output and the constraint boundary. For lower bound constraints, it is defined the constraint error as the difference between the constraint boundary and the simulated output. Figure (2) shows

an example usage of NCD Output block in a Simulink model of the sample plant including a PID controller.

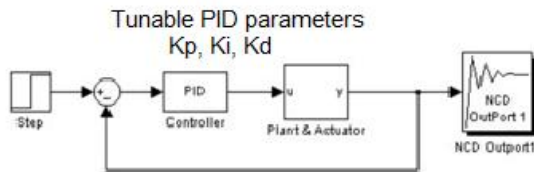


Figure (2): A Simulink model with NCD Output Block.

#### 4. Design of Idle Speed Controller

An idle speed ON-Line tuned PID controller is designed to maintain the engine speed to the desired level. The model used in this paper is described in [4]. A closed-loop idle-speed control system is shown in Figure 3. The reference input  $N_r$  sets the desired idle-speed. The engine idle speed  $N$  should agree with the reference value  $N_r$  and any difference such as the load-torque  $T$  is sensed by the speed-transducer and the error detector. The controller will operate on the difference and provide a signal to adjust the throttle angle to correct the error.

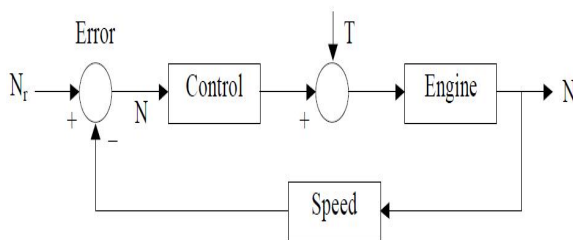


Figure (3): Block diagram of the Idle speed control model

The engine model uses a programmed relationship between torque and speed, modulated by the throttle signal. The idle

speed control is studied as a control problem, to keep the engine from stopping with additional load. The controller uses only the throttle angle to compensate the drop in the speed of engine, by extending the area for the air flow at the throttle valve. As shown in Figure 4 below:



Figure (4): The Throttle Valve

#### 5. Controller Design

##### 5.1 Proportional–Integral–Derivative controller (PID)

PID is a generic control loop feedback mechanism (controller) widely used in industrial control systems – a PID is the most commonly used feedback controller. A PID controller calculates an "error" value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process control inputs. The PID controller calculation (algorithm) involves three separate constant parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D. Heuristically, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. The weighted sum of these three actions is used to adjust the process via a

control element such as the position of a control valve, or the power supplied to a heating element. As shown in Figure 5.

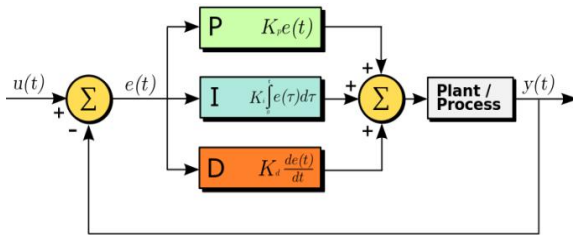


Figure (5): Block diagram PID controller

within the Simulink Library Browser. Figures (6 and 7) show the block diagram of Engine Model with PID controller and NCD blockset.

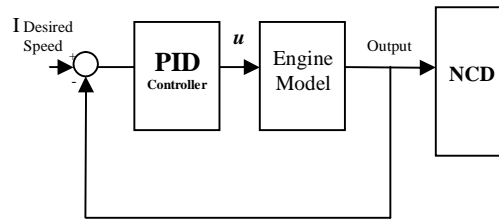


Figure (6): Block diagram of Engine Model with PID controller and NCD blockset

### 5.2 PID controller with NCD Blockset

To use the NCD Blockset, it only requires to include a special block, the NCD output block, in Simulink diagram and to connect that block to any signal in the model to signify that user wants to place some kind of constraint on the signal. NCD output block can be found under NCD

The closed loop system with PID is simulated with applying a variable value of a uniform load torque to the closed loop system, as shown in Figure (8).

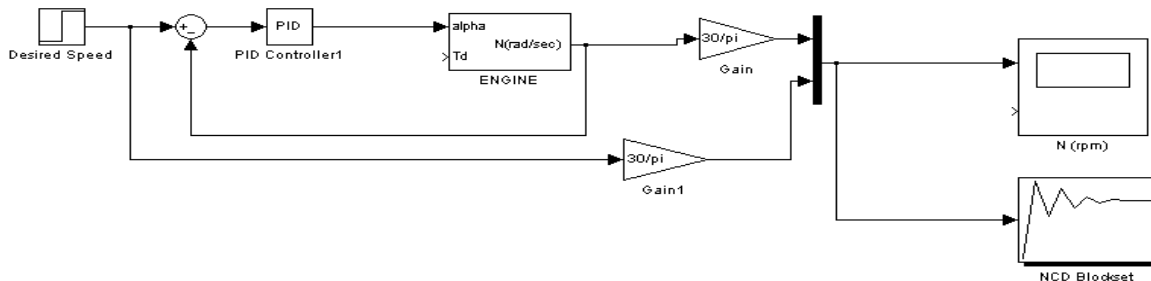


Figure (7): Block diagram of Engine Model with PID controller and NCD blockset in MATLAB/SIMULINK

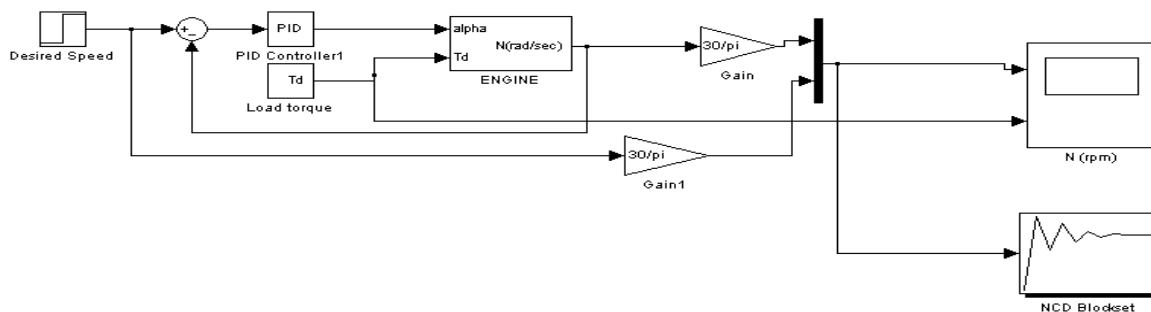
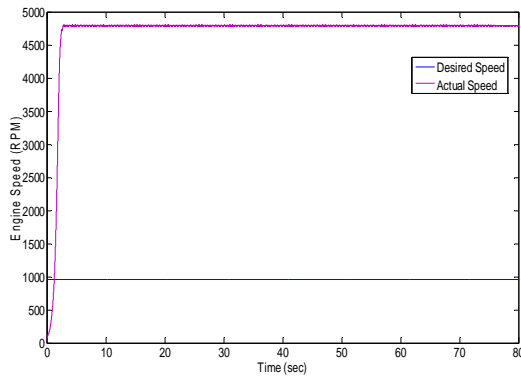


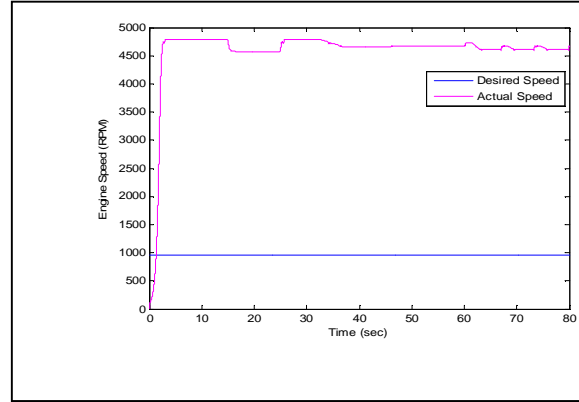
Figure (8): Block diagram of Engine Model with PID controller and NCD blockset in MATLAB/SIMULINK with a uniform load torque

**6. Simulation Results**

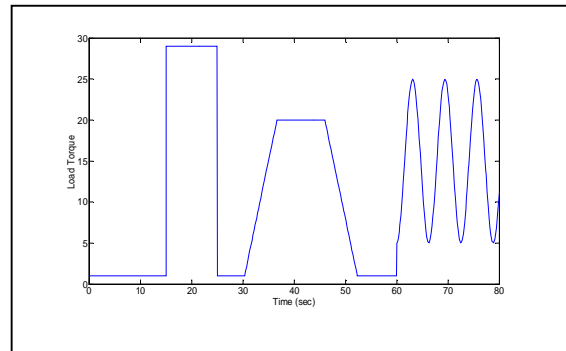
Consider the desired engine speed is (955) rpm and a variable value of a uniform load torque is applied, firstly the simulation is done without controller as shown in Figures (9 and 10), where it is shown that the system without controller is unable to track the desired Engine speed where the system goes to high speed exceeding the desired level about (4750) rpm. In order to eliminate the error steady state and to get better tracking performance a PID controller is applied. As shown in Figure(11), the system response with PID controller has the ability to get the desired performance. The PID controller parameters are selected by trial and error method. Thus the PID coefficients are  $K_p=0.42 \times 10^{-2}$ ,  $K_i=0.12 \times 10^{-2}$  and  $K_d=0.04 \times 10^{-3}$  respectively. To enhance the system response and to get best parameters for PID controller, the NCD is used to on-line tuning for controller parameters with following parameters  $K_p=0.9 \times 10^{-2}$ ,  $K_i=0.18 \times 10^{-2}$  and  $K_d=0.05 \times 10^{-3}$ . As shown in Figure (12), it is noticed that the settling time and error steady state are reduced and the Peak Overshoot and Peak Undershoot are decreased too especially with high applied load Torque.



Figure(9): System Response without Controller and without Load Torque

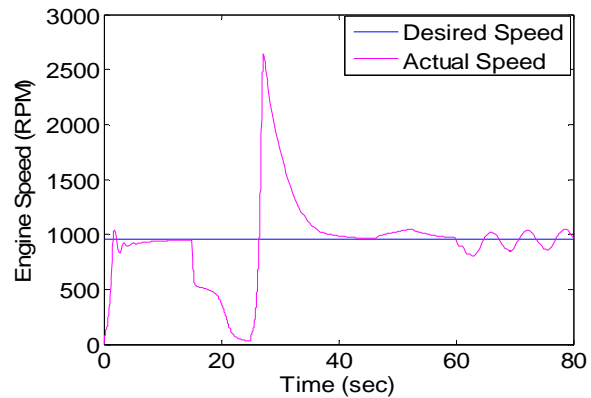


(a) Engine speed with load torque



(b): Applied load torque

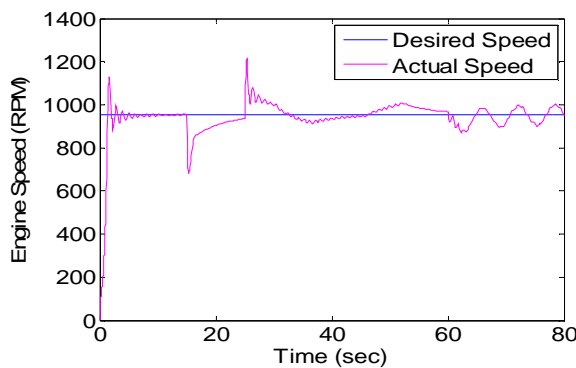
Figure(10): System Response without Controller and with Load Torque



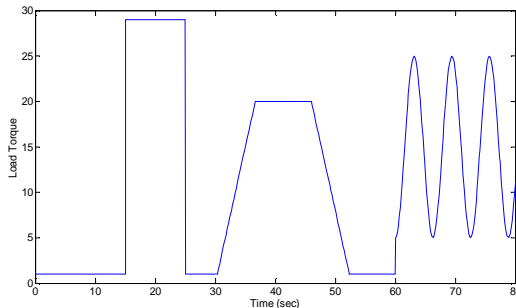
Figure(11): System Response with PID controller and the same Load Torque



(a) NCD Blockset



(b): Engine speed with load torque



(c): Applied load torque

Figure(12): System Response with PID controller and Uniform Load Torque Based NCD Blockset for tuning PID parameters

## 7. Conclusions

1- When compared NCD with trial and error method, which is used widely in industry, this method offers a more scientific and logical approach to a difficult problem of tuning control systems. Also, when the control strategy is not well-known, unlike a PID controller, tuning by trial and error method will be a time-consuming process or almost insoluble.

2- NCD is adapted for real-time executions and it can tune any controller without the mathematical model knowledge of the system it is controlling.

3- Using PID controller with NCD achieve an improvement on the speed response by improving the steady state error.

## References:

- [1] Kassakian, J.G; Wolf, H.-C.; Miller, J.M.; Hurton, C.J., 1996. "Automotive electrical systems circa 2005 - IEEE Spectrum". IEEE Spectrum (IEEE) 33 (8): 22. doi:10.1109/6.511737.
- [2] DaeEun Kim & Jaehong Park," Neural network control for reducing engine speed fluctuation at idle", IEEE 1999.
- [3] Elena De Santis, Maria Domenica Di Benedetto, Giovanni Girasole" DIGITAL IDLE SPEED CONTROL OF AUTOMOTIVE ENGINES USING HYBRID MODELS" , Center of Excellence DEWS,2005.

[4] TERUJI SEKOZAWA , “Model-based Control and Learning Control Method for Automobile Idle Speed Control using Electric Throttle”, Department of Industrial Engineering & Management Kanagawa University, Yokohama 221-8686, JAPAN, Issue 2, Volume 3, February 2008.

[5] P.K. Wong, L.M. Tam, K. Li and H.C. Wong, “Automotive engine idle speed control optimization using least squares support vector machine and genetic algorithm”, journal ,2008.

[6] Pavkovic, D., Deur, J. and Kolmanovsky, I. “Adaptive Kalman Filter-Based Load Torque Compensator for Improved SI Engine Idle Speed Control”,Control Systems Technology, IEEE , 2009.

[7] Xue Jinlin , Gao Qiang and Ju Weiping ,“Reinforcement Learning for Engine Idle Speed Control” Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference.

[8] MathWorks Inc., (2002), MATLAB® Nonlinear Control Design Blockset User’s Guide, Version 1.1.6, July 2002.