Prof. Bernd Finkbeiner, Ph.D.
Dipl.-Inf. Rüdiger Ehlers
Markus Rabe, M.Sc.
Sebastian Hahn, B.Sc.

Summer term 2012
Problem Set 3
Due: Wednesday, 9th May 2012

Embedded Systems

This assignment deals with constructing a simple electronic ignition controller for a four-cylinder four-stroke Otto motor. This case study illustrates the sources of hard real-time constraints and how (and in how far) real-time can be dealt with in a high-level design model. In particular, the problem provides hands-on experience on how to structure a typical embedded software design and how to validate the design steps using high-level modeling and simulation.

Introduction

An Otto motor is a combustion engine that uses a spark from a spark plug for controlled ignition of the air-fuel mixture in the combustion chamber. The task of an ignition controller attached to an Otto motor is to ensure adequate firing times of the spark plugs (typically one spark plug per cylinder). Early or late firing is detrimental to thermodynamic efficiency and can furthermore lead to spontaneous adiabatic ignition of the air-fuel mixture instead of controlled ignition, leading to mechanical stress and "engine knock". Ignition control is thus central to efficiency, environmental impact, lifetime, and noise comfort of Otto-type combustion engines. Current ignition controllers use knock sensors and feedback control for optimizing the ignition times, starting from a base timing being provided by a lookup table. For the sake of this assignment, we will not dig into knock sensing and feedback control, but go for purely lookup-table based ignition timing, which was the prevalent technology in the late eighties. In fact, for the sake of easier testability, we will also omit the lookup table and provide a manual slider for setting the relative ignition timing.

Ignition timing

Traditionally, ignition timing is not communicated as a time entity, but as a geometric entity: ignition "times" are reported as angles before/after the top dead center (TDC) of piston motion, where the angle is measured between ignition-time crankshaft position and crankshaft position in TDC (cf. Figure 1). We will use the convention that ignition before TDC will be denoted by negative ignition angles while ignition after TDC is assigned positive angles.

In a four-stroke engine, each cylinder fires once upon every two turns of the crankshaft. Hence, in an n-cylinder engine, the cylinders fire at angles

$$\psi = \frac{720^\circ}{n}(i-1) + \phi \ ,$$

if we set our coordinate system such that angle 0 coincides with the ignition TDC of cylinder 1. Here, i is the firing sequence element number and ϕ is the ignition angle relative to TDC. Given

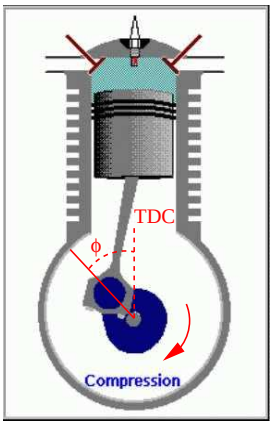

Figure 1: Ignition angle ϕ

a four-cylinder engine with the firing sequence $1, 2, 4, 3$, we thus have to fire the sparks of for the cylinders $i = 1 \ldots 4$ at "times" (i.e., angles)

$$\begin{aligned} \psi_1 &= \phi, \\ \psi_2 &= \frac{720°}{n} + \phi, \\ \psi_3 &= \frac{720°}{n} \cdot 3 + \phi, \text{ and} \\ \psi_4 &= \frac{720°}{n} \cdot 2 + \phi. \end{aligned}$$

In order to allow such controlled firing, cam-shafts and crank-shafts of Otto motors are equipped with rotary sensors. A standard setup is to have one sensor (called "*major mark*" in the remainder) issuing a signal whenever cylinder 1 reaches its ignition TDC and another sensor (called "*minor mark*" in the remainder) issuing 60 signals per crank-shaft rotation, thus yielding a resolution of $6°$. If we want to achieve higher resolution wrt. ignition timing — a resolution below $1°$ is well desirable — then we need to interpolate by using offset timing based on the rotational speed of the engine. This obviously requires to accurately measure the rotational speed based on the sensory output and to calculate and implement the necessary time offsets.

Problem 1 (Stateflow: Measuring Real-Time Constraints)

In the following, engine speed is always given in *rotations per minute*. Calculate the following values:

1. The minimal and maximal temporal separation between
   a) two successive ignitions of the same cylinder,
   b) the ignitions of two successive cylinders in the firing sequence,

   if engine speed is in the range $750/\text{min}$ to $6500/\text{min}$ and the ignition angle ϕ against TDC is kept constant between the ignitions.
2. The temporal resolution of ignition timing needed when ignition angles are to be accurate to $\pm 0.5°$ and engine speed is in the range $750/\text{min}$ to $6500/\text{min}$.

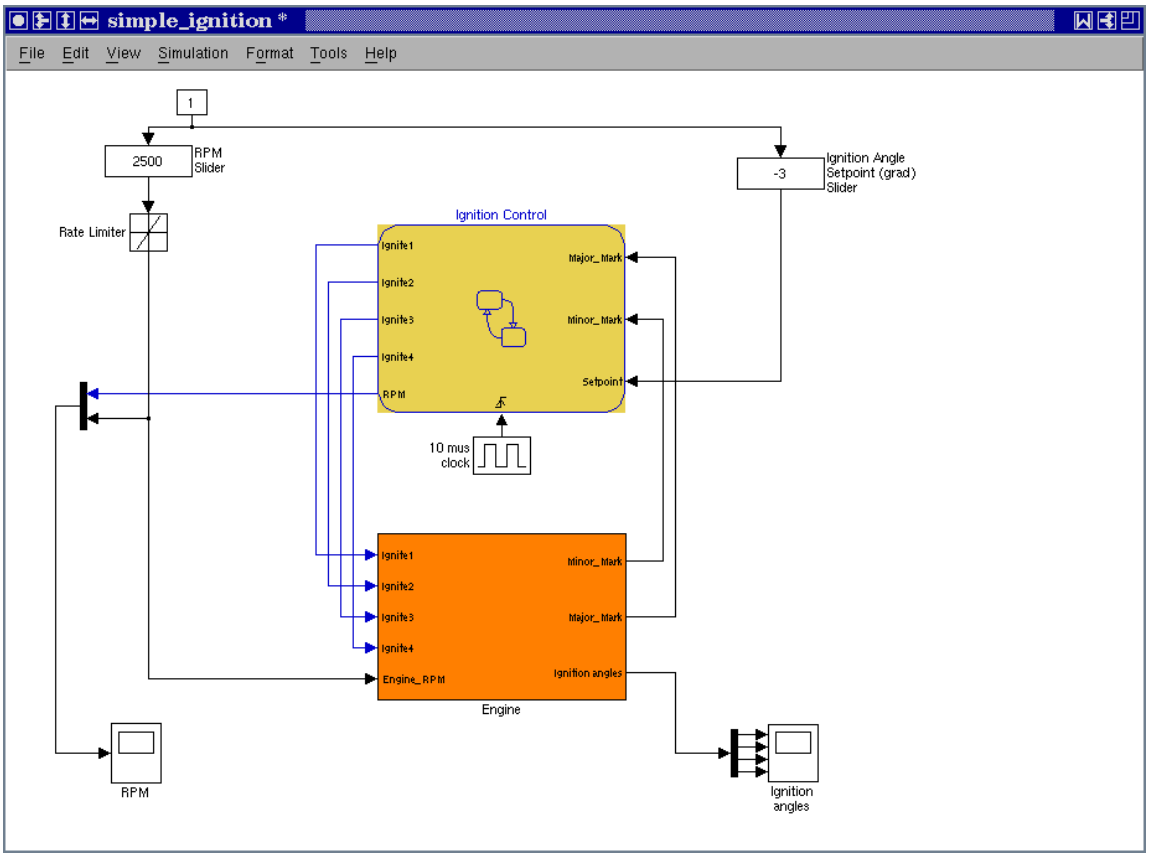

Figure 2: The overall system model

3. The accuracy of ignition angles obtained when maximum deviation in ignition timing is $\pm 5\mu$s, if engine speed is in the range $750/\text{min}$ to $6500/\text{min}$.

4. The minimum temporal separation between two *minor mark* signals if there are 60 signals per crank-shaft rotation and engine speed is in the range $750/\text{min}$ to $6500/\text{min}$.

Please state all temporal quantities in seconds or powers-of-ten thereof (e.g., μs).

Problem 2 (Stateflow: Constructing the Controller)

Download the following model:

`http://react.cs.uni-saarland.de/teaching/embedded-systems-12/downloads/ignition.mdl`

This Matlab model file contains a simulink model mimicking an Otto motor, as seen from the interface to a (simplified) electronic ignition control. It contains two major building blocks: the Simulink-based environment model mimicking the motor (called "Engine" in the model) and an empty Stateflow block (called "Ignition Control") for the ignition control. The interface between these two blocks (cf. Figure 2) consists of the Boolean signals

- "Major_Mark" (featuring a rising edge whenever cylinder 1 reaches the ignition TDC) and
- "Minor_Mark" (yielding 60 equidistant rising edges upon every crank-shaft rotation)[1]

from the engine to the controller, and the four Boolean signals

- "Ignite1" to "Ignite4" (which generate a spark on the corresponding spark plug upon a rising edge of the signal)

from the controller to the engine. Furthermore, the controller has a real-valued input

- "Setpoint" (for the relative ignition angle ϕ)

which is connected to a manual slider "Ignition angle setpoint" that you can open by double-left-click on the box; the setpoint may then be modified by moving the slider with the mouse. The controller also has a real-valued output

- "RPM" (to be connected to the engine speed measurement logic developed in part , sub-problem 2)

which is linked to an oscilloscope as a debugging aid (the oscilloscope shows the actual motor RPM with a magenta line and the measured one as a yellow line). The controller is driven by a clock generator of 10μs period, yielding a transition rate of 100kHz.

In analogy to the controller setpoint input, the engine model has a real-valued input

- "Engine_RPM" (controlling the engine speed)

which is controlled by a slider (open by double-left-click) and additionally linked to the same oscilloscope as the measured "RPM" output of the stateflow block in order to ease debugging. If you move the slider during simulation then the actual change rate of the engine RPM is bounded in order to prevent pysically meaningless discontinuous changes. Consequently, the engine follows the slider with some damping.

Finally, the engine model has outputs reporting the actual ignition angles of the four cylinders, again being piped to an oscilloscope. Should you miss firing a cylinder then it will engage into spontaneous adiabatic ignition at a random angle between $0°$ and $15°$ after TDC. For simplicity, this adiabatic ignition is shown on the same oscilloscope, i.e. the oscilloscope will display ignition angles after TDC under these circumstances.

Based on this model, do the following:

1. Devise a reasonable system architecture for measuring the engine speed, for calculating the angular and temporal offsets needed for controlling the ignition timing, and for generation and distribution of the ignition signals.
2. Construct a sub-chart for measuring the engine speed and pipe its output (if necessary after converting it from "rotations per (milli-)second" to "rotations per minute") to the "RPM" output of the Ignition Control. Test it by simulation and find a good compromise between

[1] Please ignore the negative edges, which are also equidistant in the model and thus could be used to double the angular precision. In actual technical realizations, they are often far more less accurate than the rising edges (or vice versa, in which case you would only use the falling edges).

a) the stability and accuracy of the measurements when the motor is running with constant speed and

b) the inertia it shows when adapting to changes in the engine speed.

You should start from a simple architecture, yet may later on invest more complexity into a better reconciliation of requirements (2a) and (2b).

3. Construct components for calculating, based on the current (measured) engine speed and the ignition angle setpoint, the angular and timing offsets best matching the ignition angle setpoint.

4. Construct components for accurately generating the ignition signals based on the angular and temporal offsets calculated by the component(s) of subproblem 3. Connect them to the "Ignition1" to "Ignition4" ports of the Ignition Control and test the whole system by simulation.

5. Perform extensive tests of the system over the whole range of engine speeds and ignition angles, both with stable and with varying engine speeds and ignition-angle setpoints. Do in particular address the following issues:

   a) What worst-case accuracy do you get when the engine is running with constant speed?

   b) In which operational mode (wrt. engine speed and ignition angle) do you get worst-case behavior?

   c) Is the error homogeneous over the range, or does it show peculiar patterns, e.g. increased error when asked to fire ignition slightly before/exactly at/slightly after a minor mark?

   d) Do you encounter transient errors that are hard to reproduce, e.g.

      - occasional misfiring (i.e. adiabatic ignition, which shows by ignition after TDC), in particular upon change of the ignition setpoint, or
      - occasional, transient jumps of the actual ignition angle by e.g. approximately $6°$ (i.e., ca. one minor mark) when changing the ignition setpoint gently, or
      - other occasional misbehavior?

Submit your *final* MDL-file that you obtain from subproblems 1-4 to `es12@react.cs.uni-saarland.de`. Write your solutions to subproblem 5 in your paper submission.