In computer systems functionality correctness of the result, is usually in focus. Dynamics is on the other hand very important for networking, but in other environments. Scientists have realized that mechanism that they can use is Control Theory, mechanism that has proven to give excellent results in many other science fields, starting from mechanical engines.

Segway…gyroscope, control theory, for balancing;

Terminology: Plant is engine or anything that is part of our system that we want to control. It has output that we can observe and if it deviates from the state or behavior that is desired. If it does, we can react to return it to desired state by changing the input.

There are two main approaches to control theory:

-Open loop:

There is just one input for the device. On that input we can set some desired value. The other part of the system then tries to get the state that is set in the input. Typical example of such system is thermostat that controls heating system after desired temperature of the room is set.

-Example:

Car with the speed control is fixed on 65 miles. It can be tuned on for ﬂat road, but there will be problems in the mountains because more power is needed if car is going uphill and less if car is going downhill. So in this example we can see that open loop fails if we don’t know everything that influences the system, if we make errors in estimation or modeling and if things change to rapid.

-Closed loop or feedback control:

Here you observe the signal, and compare it to the reference value. As soon as you have some deviation, you have reason for the controlling device to act with purpose of getting the system in equilibrium state. Classical approach is to look at the difference between desired value and the output value, and based on it behavior of effector is controlled.

Server and remote procedure call example.

Reference value is constant in time. Error is the difference between reference and observed value. Depending of the error and the predefined controller behavior for specific model, behavior of the effector is influenced. Observed value is very important. We need to pick parameter that influences the system for controller to react in the way it is desired. In every such system there is some settling time. Depending of the predefined controller behavior it can be shorter or longer. It can overshoot or converge smoothly. Even after the settling time there is still some small deviation present, that is called steady state error. Depending on the settling time response can be undamped(when K is too big, oscillates), underdamped(overshoots), critically damped(smooth conversion) and overdamped(slow and smooth). After measuring the system it is possible to predict systems behavior depending on the past performances and control parameters.

In reality for communication between pole and some computer we need to convert signal from analog to digital form. For that we can use Laplace transform for continuous signal and Z transform for discrete signal.

Slide 26.

Dividing by Z is shifting things in the future. Multiplying is delaying things. Value at point K is in Z domain equal Z to do power of minus K. Coefficient is just put in front of the Z. Examples for IMPULS, UNIT STEP, are on the slides 28 AND 29. Scaling of signal is done by adding scale value in front. Sum of signals is made by adding the coefficients of the signals at times where they intersect. GEOMETRIC SIGNAL is on the page 33. You can see that Y(z) here depends only on value “a”, and it can in some way be seen as a coefficient that controls the behavior of the system. Understanding how a changes the function can help us to predict the behavior of the whole system.

Auto regressive model with input – you take the values at K-1 and compute signals at time K.

Transfer function in the Z transform can be calculated dividing y(z) by u(z). Then we have examples of transfer functions: CONSTANT(40), TIME DELAY(41), It is easy to combine different functions in Z domain. Rest of the examples are skipped because we didn’t go into details. On the slide 51 there is sum of all things that are needed to understand poles in Z domain. Steady state gain can be computed by dividing two functions for infinity as their elements.

There are two basic controller implementations in Z domain:

In proportional control we have only one coefficient on which the transfer function depends.

In integral control you look also in the time step before the current step, so you integrate over time.

Proportional controller is much faster, but integral is more accurate and less overshoot because of looking in the past steps. There is also third option that combines the two controllers for best performance.

Real example:

There is some database system that needs to be backed up periodically, but when backup process is started, system starts to perform poorly. If backup priority is set to low, we don’t fix the problem. The solution is achieved with adding specific code that dynamically decides for how long is backup going to sleep. Parameter that is used is response time.

This is really one nice step towards autonomy, meaning that it can detect and recover from some specific problems.