

2012

A novel Control Design approach for sever subsystems: the concept of active disturbance rejection and a case study

John Ping
Intel Corporation

Zhiqiang Gao
Cleveland State University, Z.GAO@csuohio.edu

Rahul Khanna
Intel Corporation

Follow this and additional works at: https://engagedscholarship.csuohio.edu/enece_facpub
How does access to this work benefit you? Let us know!

Repository Citation

Ping, John; Gao, Zhiqiang; and Khanna, Rahul, "A novel Control Design approach for sever subsystems: the concept of active disturbance rejection and a case study" (2012). *Electrical Engineering & Computer Science Faculty Publications*. 259.
https://engagedscholarship.csuohio.edu/enece_facpub/259

This Article is brought to you for free and open access by the Electrical Engineering & Computer Science Department at EngagedScholarship@CSU. It has been accepted for inclusion in Electrical Engineering & Computer Science Faculty Publications by an authorized administrator of EngagedScholarship@CSU. For more information, please contact library.es@csuohio.edu.

A NOVEL CONTROL DESIGN APPROACH FOR SERVER SUBSYSTEMS: THE CONCEPT OF ACTIVE DISTURBANCE REJECTION AND A CASE STUDY

Contributors

John Ping

Data Center and Connected Systems
Group, Intel Corporation

Zhiqiang Gao

Center for Advanced Control
Technologies, Cleveland State
University

Rahul Khanna

Software and Services Group, Intel
Corporation

The fast data center growth and cloud computing implementations drive the demands for a higher server system power efficiency to reduce data center energy cost. In this article, a novel control strategy is explored for power optimization to key components in a server system, using the voltage regulator (VR) control as an illustrative example. The new approach is based on the unique active disturbance rejection control (ADRC) principle, which actively estimates, and compensates for, disturbances to the system caused by dynamic load changes rather than passively reacting to them as most existing methods do. Hence the controller is inherently efficient in rejecting the disturbances in real time. Without any hardware changes, this methodology leads to substantial power saving in a highly dynamic load environment in a simulation study.

Introduction

The US data center industry is in the midst of a major growth period stimulated by increasing demand for data processing and storage^{[1][2]}. Financial services, Internet communication and entertainment, media, and global commerce all drive fast growth of the data center, along with a significant increase in energy consumption and its associated cost from the server system and data center infrastructure. The server system power efficiency becomes a frontline issue in server architecture, design, and research^{[3][4][5]}.

An Intel server system is shown in Figure 1. Under the hood of a modern server, we see many subsystems or circuits that are separately controlled. The server subsystem system controls can be characterized in several categories: voltage regulator control, power energy control and optimization, and thermal management and control. At the OS level, the control issues could be workload control, performance optimization, and so on. Each of these subsystems is quite different in its dynamics, but they all seek better control means to improve efficiency, robustness, smartness, and yet, at the same time, retain ease of use and intuitiveness. The improvement of control methodology or strategy in each subsystem in the server could result in a major improvement of the overall server system in terms of power efficiency, performance, and adaptation.

Undoubtedly automatic control systems play a crucial role in server systems and yet their design and tuning have not been the focus of our work until recently. Our default solution for many years has been the conventional proportional-integral-derivative (PID) controller that dates back to early 1900s^{[6][7][16]}. It is still widely used in server subsystems today due primarily to

“The improvement of control methodology or strategy in each subsystem in the server could result in a major improvement of the overall server system in terms of power efficiency, performance, and adaptation.”



Figure 1: Intel server system
(Source: Intel Corporation, 2012)

its simplicity and our familiarity with it. But perhaps we can no longer ignore its intrinsic shortcomings, including but not limited to the following:

- It is mostly tuned by trial-and-error, leaving much room for systematic improvement.
- It has limited ability to reject disturbances, such as load changes and process dynamics variations, which is the primary function in any control systems and this imposes unnecessary constraints on server systems.
- It regulates the system by reacting to the deviations in the process variables, such as voltage and temperature, from their desired values, also known as setpoints, wasting energy in the process, especially during high dynamic load change in server operations.

It is our belief that to overcome such shortcomings we must make a fundamental change in how we approach the problem for server subsystem control: instead of passively reacting to disturbances, we propose an active disturbance rejection (ADR) paradigm where the disturbance information is gathered and used preemptively in limiting the disturbance impact on the system. That is, we propose a method that will help eliminate the deviation before it appears, therefore saving the energy that would be otherwise needed in correcting the deviation.

Such a design principle has been discussed in depth before^{[8][9][10]}. The key in general is to find a way of getting ahead of the curve in mitigating set-point deviation, as opposed to always playing catch-up like PID does most of time. The focus of this article is to creatively adapt the ADR principle to server problems, utilizing all our relevant knowledge of server dynamics. The key to the solution is how we obtain the disturbance information and fully taking advantage of it in helping the controller to get ahead in mitigating disturbance effects.

“Active disturbance rejection (ADR): the disturbance information is gathered and used preemptively in limiting the disturbance impact on the system.”

“Eliminate the deviation before it appears.”

This new way of thinking about server control problems is rooted in our understanding of a control system's primary task as that of disturbance rejection, upon which system performance is evaluated. Acting on the source of the deviation, that is, disturbance, as opposed to deviation itself, gives us the advantage of getting ahead, of treating the cause, not the symptom. As will be demonstrated, this has a profound impact on future energy saving in the server market.

In this article, we use a typical CPU VR control subsystem as an example to apply the ADR principles. The server in a data center normally runs in a high dynamic workload environment with the various tasks running above the operating system making the CPU current changes drastically in real time; this makes it a tough disturbance to deal with for the VR controller. From the perspective of efficiency, however, any improvement in the VR controller in handling each single load change will add up to potentially significant energy savings in a highly dynamic environment with big swings in load current. It is in this environment that we'll design and validate the ADR methodology to actively reject the disturbance in the CPU VR system and compare the performance and energy consumption with respect to step load changes, as it is compared to the standard PI controller currently used; with a given average dynamic load fluctuation, we derive the energy saving over a period of time.

The article is organized as follows: the following section, "Background: What Is the Control Anyway?" describes the related work and background of the control algorithms. The next section, "Active Disturbance Rejection," introduces the ADRC algorithm. Next, the section "Active Disturbance Rejection in a Server VR Subsystem" describes the ADRC control method on a server VR subsystem. "Comparing ADRC to Existing Solutions" provides an analysis of the result and makes the comparison between the PID and ADRC in terms of control performance and power efficiency. This is followed by a summary of the article.

"Today, automation has been built into the very fabric of modern society, from massive production lines of consumer goods to individual homes and personal electronic devices."

"Over a period of two centuries, control technology has emerged as a crucial centerpiece in all engineered systems, ..."

Background: What Is Control Anyway?

Since not all server design engineers are well versed in the concepts and terminology of controls, we start with this basic question. Automatic control is a technology that has played a crucial role in industry ever since the era of the steam engine and the industrial revolution in the 18th century. Today, automation has been built into the very fabric of modern society, from massive production lines of consumer goods to individual homes and personal electronic devices. From the vantage point of control engineers, everything is a part of a process, or system, within which all variables are in some way interdependent to each other. The objective of control system design is to make such dependency, in a particular case, meet a predetermined goal or set of criteria. Over a period of two centuries, control technology has emerged as a crucial centerpiece in all engineered systems, simply because all such systems have a goal to reach, a need to satisfy, and the resources to reach the goal. To satisfy the need is what we call the act of control.

The act of control can be divided into two phases: collecting information and acting on it. Using the CPU VR control as an example, the goal is to provide a constant voltage supply for CPU to function. The information that can be collected are values of process variables such as voltage and current at various points in the circuit. Such information is used by the controller to adjust the amount of power supplied to the CPU—not too much, not too little, just right! That is, in a perfect CPU VR system, the power supplied to the CPU is exactly what it needs, resulting in a voltage supply that is kept at a constant 3.3 volts, despite huge, unpredictable swings in load current.

Perfect control, of course, doesn't exist in the real world. For instance, when we turn on a washing machine at night, the light may dim momentarily, indicating a voltage dip when the load current suddenly increases. The same thing happens in the VR control system: when the load currently unexpectedly increases, the voltage dips, the extent of which shows the ability of the “disturbance rejection” of the controller, a primary criterion and a central task in control design.

Curiously, little has changed since the beginning of the modern era in how we perceive and solve the disturbance rejection problem in control: we wait, we see, and we react to the deviation in the process variable from its desired value, or setpoint, the deviation caused by disturbances. Much progress has been made in all aspects of control engineering, techniques, hardware, and software, and so on, but this reactive paradigm has endured over two hundred years, crystallized in the dominant industrial control technology known as PID^[6], a technology defined by how it react in three ways to the setpoint deviation, that is, tracking error, proportional, integral, and derivative, as shown in equation 1.

$$u = K_p e + K_i \int e dt + K_d \dot{e} \quad (1)$$

where u is the control signal, e is the error between the process output and its desired value, and $\{K_p, K_i, K_d\}$ are controller gains. Over 95 percent of industrial controllers are of this type^[6], an alternative to which is discussed below.

Active Disturbance Rejection

Emerging after World War II as a distinct engineering discipline, automatic control has been synonymous with feedback largely thanks to Norbert Wiener's brainchild of Cybernetics^[17]. Wiener calls it “control by informative feedback,” which means that “when we desire a motion to follow a given pattern the difference between this pattern and the actually performed motion is used as the new input to cause the part regulated to move in such a way as to bring its motion closer to that given by the pattern.” In other words, the control mechanism first sees the deviation and then acts on it in order to reduce it. Such conception by Wiener influenced generations of control scientists and engineers and dominated the field ever since the publication of his book in 1948. Many, if not most, control textbooks have the word *feedback* in the

“Much progress has been made in all aspects of control engineering, ... but this reactive paradigm has endured over two hundred years.”

title. The renowned historian of control engineering, Otto Mayr, goes as far as saying that “this field is essentially based upon a single idea, that of the feedback loop” and there was never a serious debate or reflection on it, or was there?

The success, as well as the occasional problems of oscillation, of steam engines some 200 years ago attracted the attention of many scholars, and engineers. Among them, Jean-Victor Poncelet, a prominent French scholar and engineer, in the early 1800s conceived of a very different idea of control: measure the load disturbance to the engine and cancel it out with the adjustment of steam flow *before* the engine speed is affected^[9]. Some 100 years later, Russian scholars revived Poncelet’s idea and developed an entirely different theory and practice of automatic control that is called “combined system” or dual channel, where disturbances are measured and the information is used to make a much more effective control system^{[18][19]}. A few scholars and engineers in England and the United States also discovered the benefits of adding the so called “feed forward” element to the control system, as shown in nature and manmade systems’ control systems alike^{[6][20]}.

The Origin of ADRC

Conceived by Jingqing Han in the mid-1990s, Active Disturbance Rejection Control (ADRC) is in the same vein of the invariance principle of the Soviet scholars a few decades earlier, exposed to him when he was a budding graduate student in Moscow. By late 1980s, Han, well established as one of the top control theorists in China, openly challenged the modern control paradigm in the vein of Kalman Filter and mathematical control science, predicated on accurate model of the reality^[21]. Han believed that such conception of the problem and presumption in its solution could be called a theory of mathematical models, but not of controls. Han believed that the Soviet scholar got it right: control systems are about disturbances; in particular, they are about how one strives to make the controlled variables, or process outputs, invariant under the assault of disturbances ubiquitously internal and external.

The background of Han’s 1989 paper^[21] is that PID had dominated industrial controls for decades and no serious researchers could ignore the reality any longer and avoid the question “why?” If there was competition in engineering practice between PID and its users against the vast edifice of modern control theory and its creators and builders in academia, PID would have won hands down and everyone knows that. What is not so clear was the reason behind such a big divide between how control is practiced and how it is researched and taught. It took a scholar of the highest caliber to pinpoint the cause: our reliance on mathematical models and a misconception of what control engineering really is.

What a mathematical model represents is the known dynamics of the process being controlled; but the real task of control, the reason to have a control system in the first place, is to deal with the unknowns and unpredictable, also known as disturbances. Renowned control theorist Roger Brockett once said “If there is no uncertainty in the system, the control, or the environment, feedback control

“PID had dominated industrial controls for decades and no serious researchers could ignore the reality any longer and avoid the question “why?””

“If there is no uncertainty in the system, the control, or the environment, feedback control is largely unnecessary” – Roger Brockett.

is largely unnecessary.”^[22] But modern control theory largely proceeded along the lines of the following: given the mathematical model, design a control law to achieve some measure of optimality, which is a valid question in itself but not necessarily the *only* control problem out there. Hence the theory/practice dichotomy and the eighty-year dominance of PID. The question was “What can we do about it?” The answer, according to Han, was ADRC.

From 1989 to the time of his passing in early 2008, Han dedicated the last two decades of his life to an alternative to PID and he came up with much more than just a replacement algorithm. ADRC, according to Han, “inherits from PID the quality that makes it such a success: the error-driven, rather than model-based, control law; it takes from modern control theory its best offering: the state observer; it embraces the power of nonlinear feedback and puts it to full use; it is a useful digital control technology developed out of an experimental platform rooted in computer simulations.”

In other words, Han concluded that a viable control law cannot be model driven. The success of PID demonstrates the effectiveness and practicality of the error-driven control paradigm. At the same, being a theorist he recognized that the vast research in modern control brought us its crown jewel, the state observer, which can be creatively used to extract the disturbance information from the already available input-output data. ADRC “actively” uses this information to cancel the disturbance out whenever possible, *before* it does any damage, in direct contrast to PID, which only passively reacts to the changes produced by the disturbances after it runs its course through the process.

Another barrier broken through via ADRC is the linear-nonlinear divide in control theory. Instead of the linearizing the nonlinear dynamics so that they can fit into the well-developed linear system theory, Han demonstrated in the ADRC framework that one could purposely add nonlinearity into the PID structure to make it more effective. This and other discoveries are only made possible because the computer simulation provided us with a platform where control research could be done experimentally, like other physical sciences, instead of as a branch of mathematics. Han emphasized that it is through experimental research ADRC was *discovered*, as opposed to *derived*.

In summary, ADRC can be viewed as a distinctly different conception of what control is; as a way of conducting an experimental science; and finally as a new control system platform, absorbing the error-driven mentality of PID but adding to it a proactive disturbance rejection facility that makes control truly “active.”

Illustration of Active Disturbance Rejection for a Second Order Plant

The conception and the methodology of ADRC obviously is quite general and fundamental, applicable to most control systems across disciplinary boundaries, so much so that any concrete application of it would come with its limitations pertaining only to that application, which is sometimes mistaken

“A viable control law cannot be model driven.”

“ADRC was discovered, as opposed to derived.”

for limitations in general. With this in mind, we introduce a second order, nonlinear, uncertain, and time-varying process and demonstrate how the problem can be reformulated with the guidance of ADRC principles.

Although the ADRC method is applicable, in general, to n th order, nonlinear, time-varying, multi-input and multi-output systems (MIMO), for the sake of simplicity, its basic concept is illustrated here using the second-order motion control problem in equation 2.

$$\ddot{y} = p(y, \dot{y}, w, u, t) \quad (2)$$

of which

$$\ddot{y} = u \quad (3)$$

is an idealization corresponding to Newton's law of motion $f = ma$. Between the totally unknown system of equation 2 and the idealized motion of equation 3, the actual motion system can be described as

$$\ddot{y} = f(y, \dot{y}, w, t) + bu \quad (4)$$

That is, $p(y, \dot{y}, w, u, t)$ can be meaningfully separated as

$$p(y, \dot{y}, w, u, t) \approx f(y, \dot{y}, w, t) + bu \quad (5)$$

Adopting a disturbance rejection framework, the motion process in equation 2 can be seen as a nominal, double integral, plant in equation 3 scaled by b and perturbed by $f(y, \dot{y}, w, t)$. That is, $p(y, \dot{y}, w, t)$ is the generalized disturbance and the focus of the control design.

Contrary to all existing conventions, Han proposed that $f(y, \dot{y}, w, t)$ as an analytical expression perhaps is not required or even necessary for the purpose of control design. Instead, what is needed is its value estimated in real time. Specifically, let \hat{f} be the estimate of $f(y, \dot{y}, w, t)$ at time t , then

$$u = (-\hat{f} + u_0)/b \quad (6)$$

reduces equation 1 to a simple double-integral plant

$$\ddot{y} \approx u_0 \quad (7)$$

which can be easily controlled.

“The control of a complex nonlinear, time-varying, and uncertain process in equation 2 is reduced to the simple problem in equation 7.”

This demonstrates the central idea of active disturbance rejection: the control of a complex nonlinear, time-varying, and uncertain process in equation 2 is reduced to the simple problem in equation 7 by a direct and active estimation and rejection (cancellation) of the generalized disturbance, $f(y, \dot{y}, w, t)$. The key difference between this and all of the previous approaches is that no explicit analytical expression of $f(y, \dot{y}, w, t)$ is assumed here. The only thing required, as stated above, is the knowledge of the order of the system and the approximate value of b in equation 4. The bu term in equation 4 can even be viewed as a linear approximation, since the nonlinearity of the actuator can be seen as an external disturbance included in f .

Obviously, the success of ADRC is tied closely to the timely and accurate estimate of the disturbance. A simple estimation such as $\hat{f} = \hat{j} - u$ may very well be sufficient for all practical purposes, where \hat{j} denotes an estimation of j .

The Extended State Observer and the Control Law

There are also many observers proposed in the literature, including the unknown input observer, the disturbance observer, the perturbation observer, and the extended state observer (ESO). See, for example, a survey in Tian and Gao^[9]. Most require a nominal mathematical model. A brief description of the ESO of equation 1 is described below. The readers are referred to Tian and Gao^[14], Goa^{[10][11]}, and Sun and Gao^[12] Zheng and Gao^[13] for details, particularly for the digital implementation and generalization of the ESO in Ping and Gao^[15].

The ESO was originally proposed by J. Han^[23]. It is made practical by the tuning method proposed by Goa^[11], which simplified its implementation and made the design transparent to engineers. The main idea is to use an augmented state space model of equation 1 that includes f , short for $f(y, j, w, t)$, as an additional state. In particular, let

$$x_1 = y, x_2 = j, \text{ and } x_3 = f \quad (8)$$

The augmented state space form of equation 1 is

$$\begin{aligned} \dot{x} &= Ax + Bu + Eb \\ y &= Cx \end{aligned} \quad (9)$$

$$\text{with } A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix}, C = [1 \ 0 \ 0], E = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Note that $x_3 = f$ is the augmented state and $b = \dot{f}$ is a part of the jerk; that is, the differentiation of the acceleration, of motion and is physically bounded.

The state observer

$$\begin{aligned} \dot{z} &= Az + Bu + L(y - \hat{y}) \\ \hat{y} &= Cz \end{aligned} \quad (10)$$

with the observer gain $L = [l_1 \ l_2 \ l_3]^T$ selected appropriately, provides an estimate of the state of equation 9, $z_i \cong x_i$, $i = 1, 2, 3$. Most importantly, the third state of the observer, z_3 , approximates f . The ESO in its original form employs nonlinear observer gains. Here, with the use of linear gains, this observer is denoted as the linear extended state observer (LESO). Moreover, to simplify the tuning process, the observer gains are parameterized as

$$L = [3\omega_o, \ 3\omega_o^2, \ \omega_o^3]^T \quad (11)$$

where the observer bandwidth, ω_o , is the only tuning parameter.

With a well-tuned observer, the observer state z_3 will closely track $x_3 = f(y, j, w, t)$. The control law

$$u = (-z_3 + u_0)/b \quad (12)$$

“The success of ADRC is tied closely to the timely and accurate estimation of the disturbance.”

“The ESO (extended state observer) is simplified its implementation and made the design transparent to engineers.”

then reduces equation 4 to equation 7, that is,

$$\ddot{y} = (f - z_3) + u_0 \approx u_0 \tag{13}$$

An example of such u_0 is the common linear proportional and derivative control law

$$u_0 = k_p(r - z_1) - k_d z_2 \tag{14}$$

where r is the set point. The controller tuning is further simplified with

$$k_d = 2\omega_c \text{ and } k_p = \omega_c^2 \tag{15}$$

where ω_c is the closed-loop bandwidth^[11]. Together, equations 10 through 15 are collectively denoted as the parameterized linear ADRC, or LADRC.

“Apply ADRC to a Romley Server CPU PVTT power rail voltage regulation subsystem, and compare the simulation results.”

Active Disturbance Rejection in a Server VR Subsystem

In this section, we apply ADRC to a Romley Server CPU PVTT power rail voltage regulation subsystem, and compare the simulation result with traditional PID control in the next section.

Sandy Bridge CPU VTT Voltage Regulator

The Romley PVTT VR is designed to provide power to the VCCPPA, VCCPCA, VCCPDTTA pins of the Sandy Bridge processor. The VR switching regulator is a single phase synchronous buck converter as shown in Figure 2. It consists of two MOSFETs, one inductor, and one capacitor.

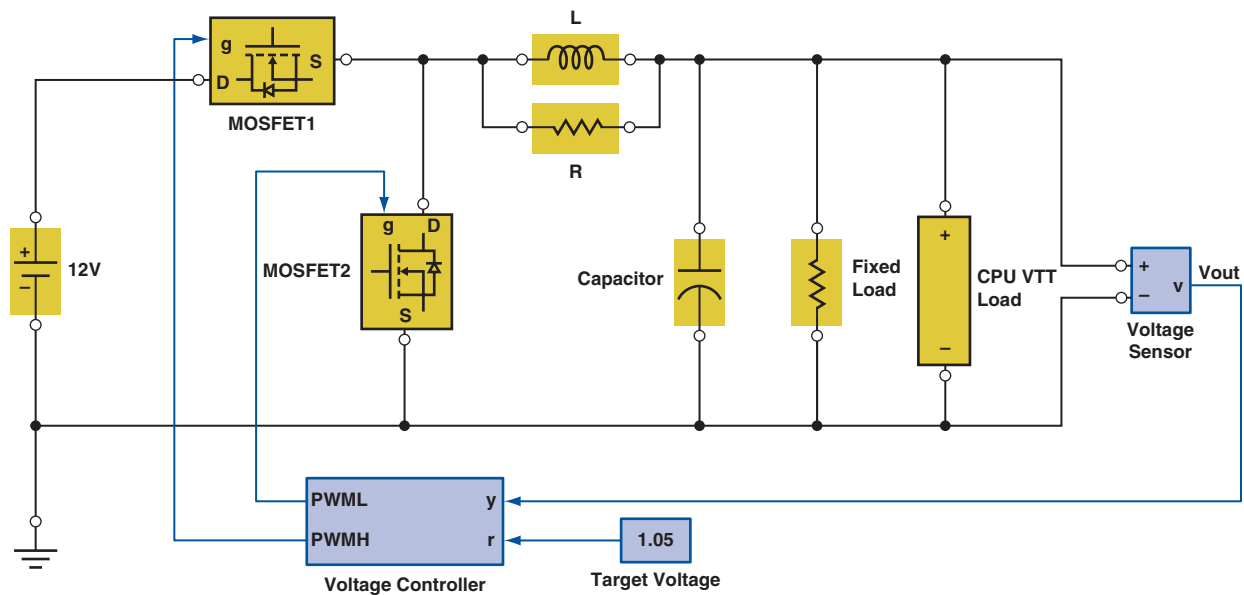


Figure 2: VTT VR circuit
(Source: Intel Corporation, 2012)

It converts the 12 V to 1.05 V V_{out} or 1.0 V V_{out} . It is capable of providing a maximum load of 20 A, the maximum step load size is 14 App, and the maximum step load slew rate is 20 A/ μ s. The frequency of the pulse-width modulation (PWM) is 500 kHz.

The MOSFETs are turned on and off to alternate between connecting the inductor to source voltage to store energy in the inductor and discharging the inductor into the load, and the capacitor smooths the ripple of voltage output from the inductor. The PWM control the MOSFETs open and close the time ratio to determine the output voltage level.

The control object of the controller is to deal with voltage deviation caused by the CPU VTT dynamic load changes and maintain the desired voltage level by adjusting the PWM duty ratio.

MATLAB Modeling of the Voltage Regulator

To be able to test ADRC in simulation, a MATLAB model is built to describe the CPU PVTT buck converter circuit. Based on the original circuit implemented in the Romley Rosecity Server Reference board, we created the model to describe the CPU PVTT VR circuit as shown in Figure 3.

“A MATLAB model is built to describe the CPU PVTT buck converter circuit.”

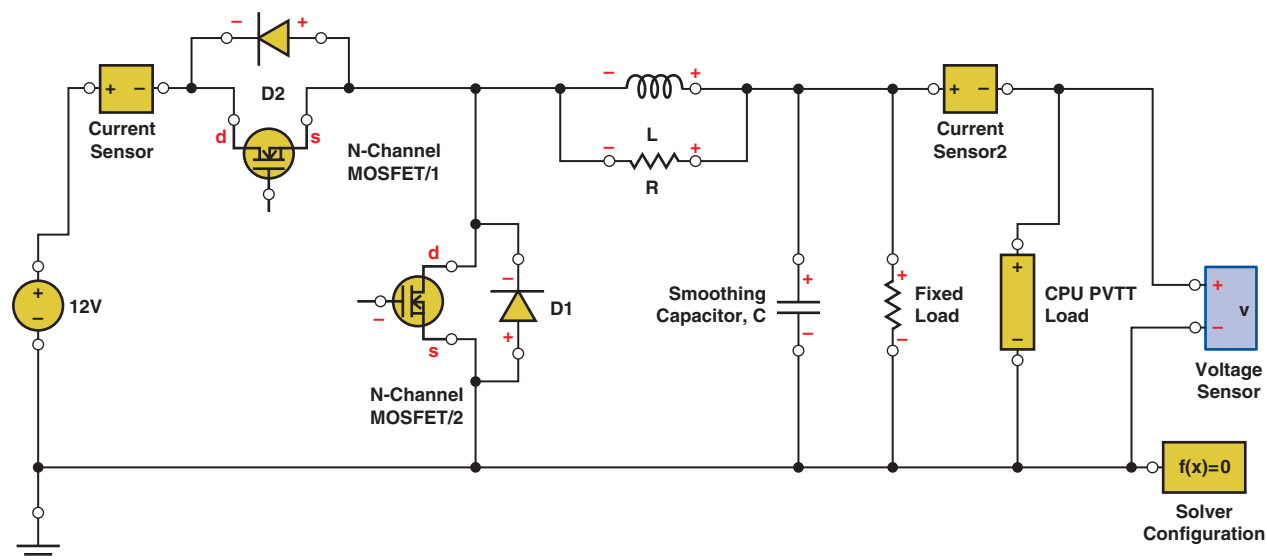


Figure 3: PVTT VR circuit modeling in MATLAB

(Source: Intel Corporation, 2012)

The CPU PVTT load connects to the output of the VR circuit to simulation CPU PVTT load changes. Current sensors are added to the input and output of the VR circuit to get the current reading in real time, and a voltage sensor is applied to the output side; thus the power data can be derived with product of the voltage and current.

Active Disturbance Control Design

As described earlier, the ADRC control law is given as follows:

$$\begin{aligned} \dot{z}(t) &= Az(t) + Bu(t) + L(y(t) - \hat{y}(t)) \\ \hat{y}(t) &= Cz(t) \end{aligned} \tag{16}$$

where

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, B = \begin{bmatrix} 0 \\ b_0 \\ 0 \end{bmatrix}, L = \begin{bmatrix} 3\omega_0 \\ 3\omega_0^2 \\ \omega_0^3 \end{bmatrix}, C = [1 \ 0 \ 0],$$

Here, ω_0 is the bandwidth of the observer. The control law is

$$u = \frac{\omega_c^2(r - z_1) - 2\omega_c z_2 - z_3}{b_0} \tag{17}$$

where r is the set point and ω_c is the control bandwidth. ADRC has three design parameters, b_0 , ω_0 , and ω_c , which can be easily *tuned*^{[8][9][10][11]}.

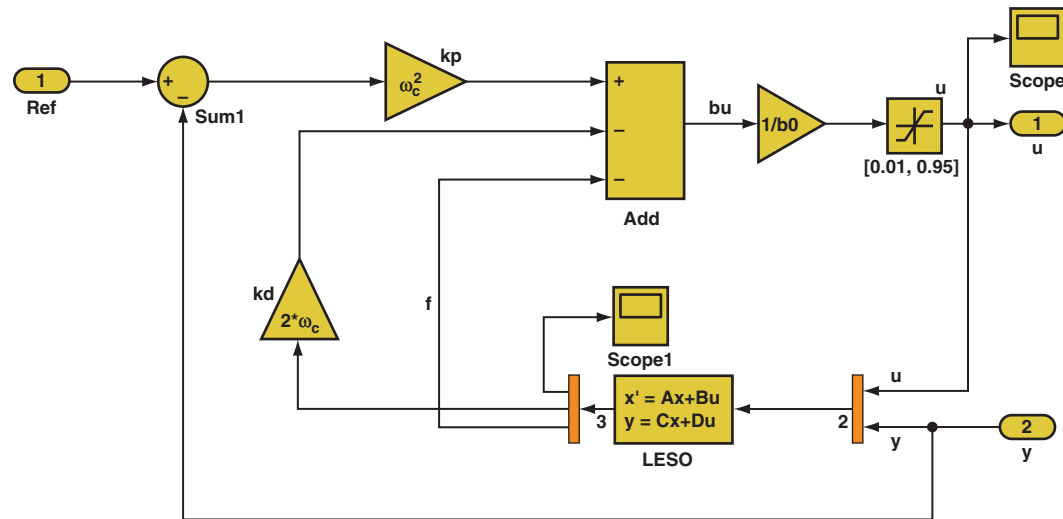


Figure 4: ADRC simulation block diagram in MATLAB
(Source: Cleveland State University, 2003, 2012)

“The model of the ADRC is built in MATLAB and connected with the CPU VTT VR model build from last section.”

The model of the ADRC is built in MATLAB as shown in Figure 4, and when connected with the CPU VTT VR model built from last section, we get a fully controlled CPU VTT voltage regulator simulation model, which is shown in Figure 5. A cyclic step load resource to simulate the CPU VTT dynamic load changes is added to the input of the VTT VR model. The setup point to the controller is set to 1.05 V to the ADRC controller to regulator the voltage to 1.05 V.

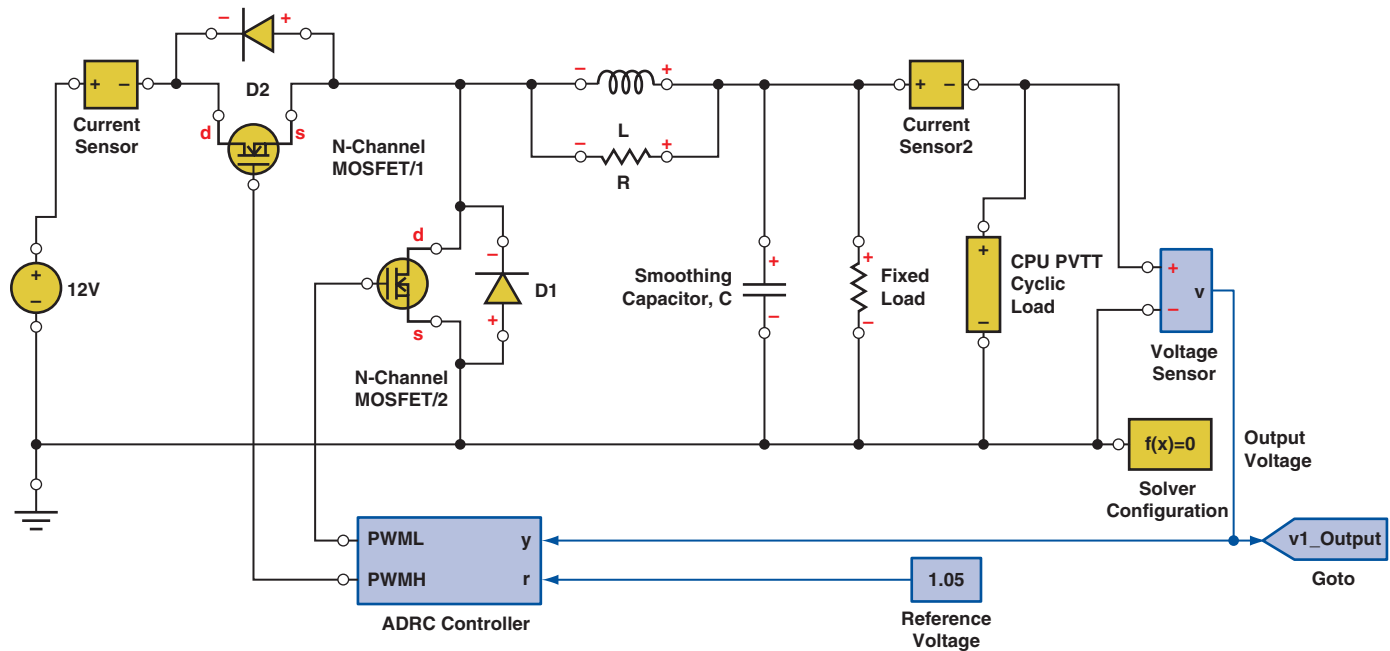


Figure 5: PVTT with ADRC controller modeling in MATLAB

(Source: Intel Corporation, 2012)

ADRC Simulation Result

The simulation result of the ADRC is shown in Figure 6. The top chart is the voltage output, and the lower chart is the simulated CPU VTT cyclic step load change between 0-3 A in the frequency of 200 Hz (for further testing it is an idea to use the maximum load step change as 0-15 A or 0-50 A). The rising curve at the beginning of the output voltage is the control system

“ADRC simulation result is shown with cyclic step load.”

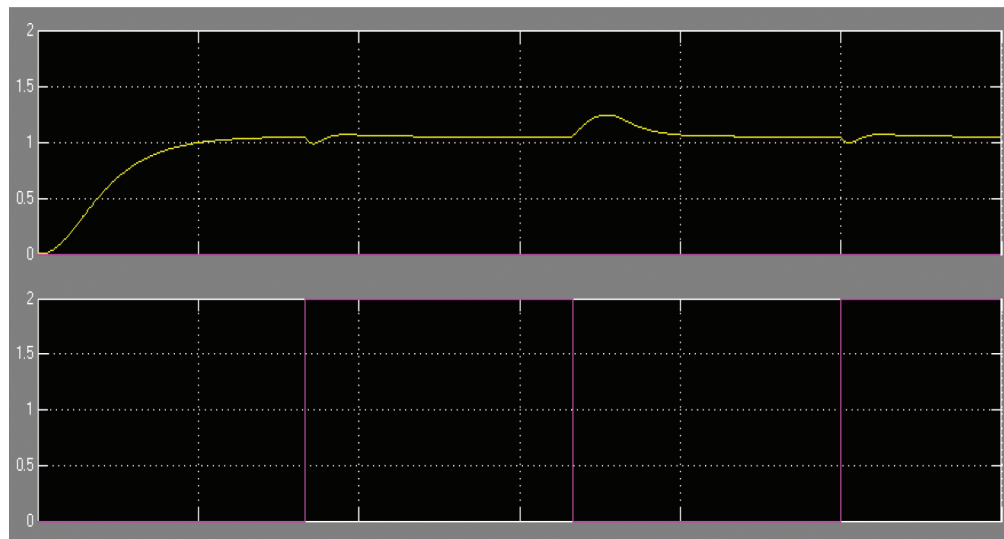


Figure 6: Simulation result of ADRC control

(Source: Intel Corporation, 2012)

“As the intrinsic characteristic of the ADRC, it generates more smooth control to the VR circuit and results in power savings.”

“PID is well known as an empirical design with users assuming little knowledge of the plant dynamics. On the other hand, most design methods based on control theory, classical or modern, require detailed and accurate knowledge of plant dynamics in the form of a mathematical model.”

transient response when the system starts. After the voltage reaches the desired voltage level at 1.05 V and is in steady state, the cyclic 0-3 A step loads are applied to the output of the regulator. From the simulation result, we can see that the ADRC can quickly correct the overshoot and undershoot caused by the dynamic step load change and quickly recover to the desired voltage without any oscillation. The control action is effective and efficient, thus resulting in a power saving by avoiding unnecessary control effort. In the next section, we compare the ADRC control with tradition PID control and show how much power it can save by ADRC with the same cyclic load over the certain period of time.

Comparing ADRC to Existing Solutions

In this section we compare the ADRC and PID to control the same VR circuit while adding in the same load changes. As the intrinsic characteristic of the ADRC, it generates more smooth control to the VR circuit and results in power savings. We will quantify the power savings based on the simulation comparison result.

Simulation Setup

A Simulink model is set up in MATLAB to compare the ADRC and PID as shown in Figure 7. Two identical VR circuit models we made in the last section are put into the comparison model, and the exact same CPU cyclic loads are applied to each VR circuit. The upper VR circuit model is connected with a PID controller; the lower VR controller is connected with an ADRC controller. To make a real-time comparison, the output voltage, output current, and the control signal from the controller output are fed into the simulation scope so that we can visualize the difference between these two control methodologies. Specifically, the VR input voltage and current are multiplied and have the integration over time to make the energy consumption comparison between these two control methods for the same VR circuit. In addition, the Integral of Absolute Errors (IAE) of the VR voltage output is calculated for each control method for comparison, the purpose of the extraction of IAE data is to make a common reference parameter to make a fair comparison. We make the above comparisons under the condition that the FAE with these two control methods are about the same.

Controller Tuning

In addition to performance, especially disturbance rejection ability, the comparison between controllers must include the ease of use, which consists of two aspects: 1) what does the user need to know to perform the controller design? And more importantly 2) how easy it is to adjust the controller parameters in order to meet different design specifications?

PID is well known as an empirical design with users assuming little knowledge of the plant dynamics. On the other hand, most design methods based on control theory, classical or modern, require detailed and accurate knowledge of plant dynamics in the form of a mathematical model. In practice, the

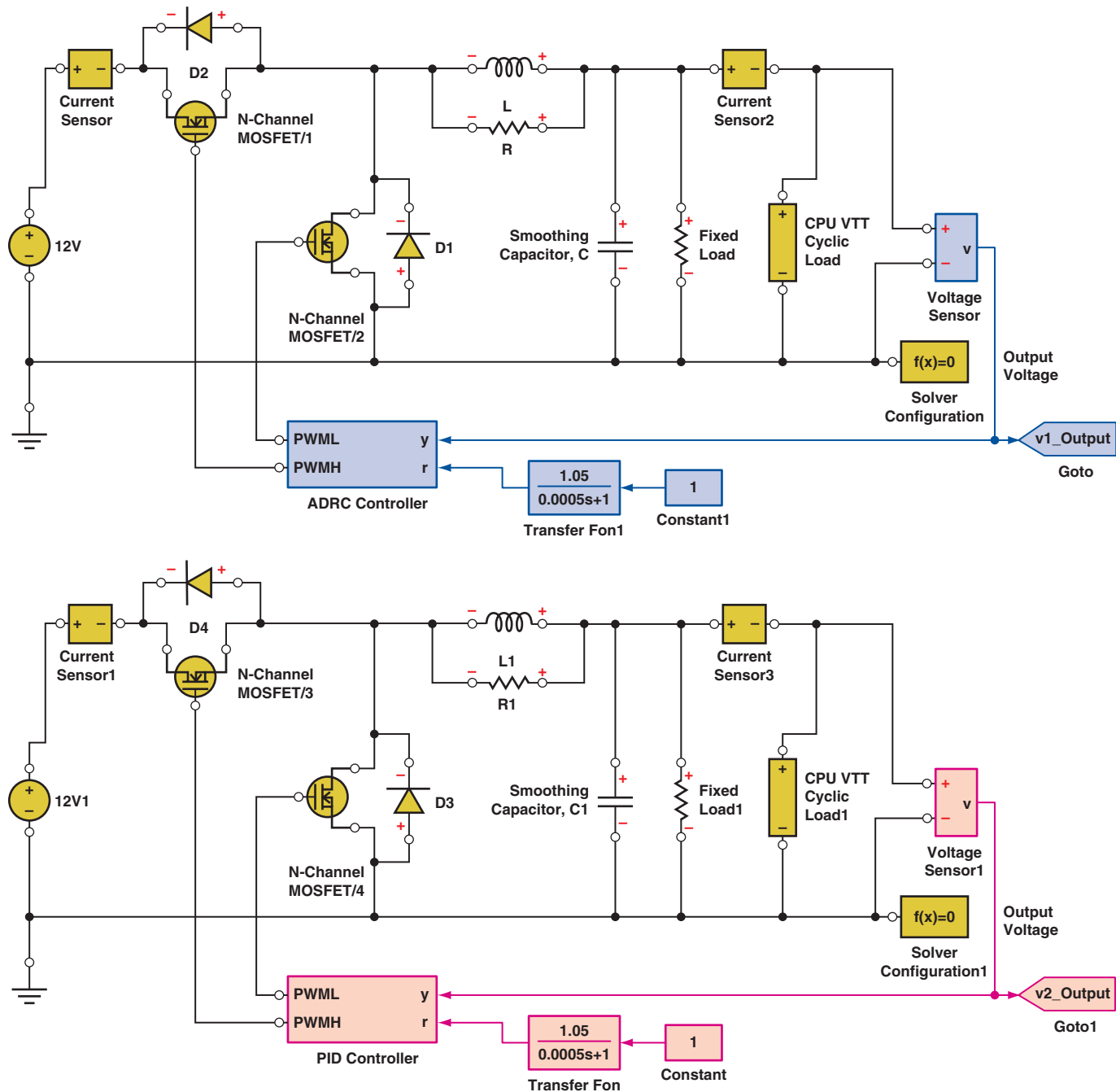


Figure 7: MATLAB Modeling to compare ADRC and PID control
(Source: Intel Corporation, 2012)

PID controller is mostly tuned based on the user’s experience and model-based controllers are tuned based on the identification or estimation of the parameters of the plant model.

ADRC design and tuning require a different mindset: it presumes that the users are familiar with the physics of the physical process but not necessarily its detailed dynamic relationship between the input and output. Based on such

“There are two key parameters in ADRC: the observer bandwidth and the controller bandwidth.”

knowledge the user chooses the order of the plant, n , to be used in design, which is not necessarily the actual order of the plant but, instead, is the order in which the controller will force the plant to behave. Once n is chosen, the users need to know, or acquire the information of, how the change in input u , *approximately* translates to the change in the n th derivative of the output y , as described in the parameter b in equation 9. Such information can be easily obtained as the initial rate of temperature change in a step response test for a thermal system.

Once the order of the plant is selected and the parameter b is obtained, the tuning of ADRC is quite straightforward. Shown in equations 10 through 15, there are two key parameters in ADRC: the observer bandwidth and the controller bandwidth. All observer gains are functions of the former and all controller gains the latter. The observer bandwidth is in general several times higher than that of the controller, to ensure that the state estimation converges fast enough for the controller, although there are exceptions. Once the ratio of the two bandwidths is fixed, the only tuning parameter is the controller bandwidth, which is the measure of the aggressiveness of the control system.

With such single parameter tuning, practical optimality or tradeoff is easily obtained. It is obvious to the users that, increasing bandwidth from low to high, the tracking and disturbance rejection are improved, but at the costs of increased sensitivity to measurement noises, the larger amount of energy exerted, and the reduced stability margin. Seeing both sides, it will not be hard for the user to choose a compromise.

PID Tuning

In PID tuning, we strive for fairness in comparison. Since PID is usually tuned by experience in practice, in a time-consuming process, duplicating that in our simulation is challenging. Instead, we take advantage of the MATLAB embedded PID autotuning tool to get the optimal coefficient value of K_p , K_i , and K_d . The MATLAB PID autotuner is a tool capable of computing the parameters of a regulator connected to the VR circuit automatically, without major user interaction apart from initiating the operation. The autotuner avoids tuning a PID regulator manually, which is not consistent and may not be optimal. The basic steps of a tuning process of the autotuner may be summarized as follows:

1. Observing the process behavior, eventually stimulating it somehow and turning this knowledge into a description of the process behavior
2. Establishing the desired closed loop behavior on the basis of the obtained process description
3. Computing the PID controller parameters in order to achieve the desired closed loop behavior.

Comparison Results

The comparison simulation result is shown in Figure 8. The top chart is for output voltage of the VR circuit, the second chart from top is the cyclic load, which simulates the CPU load frequent changes applied to the VR. The third

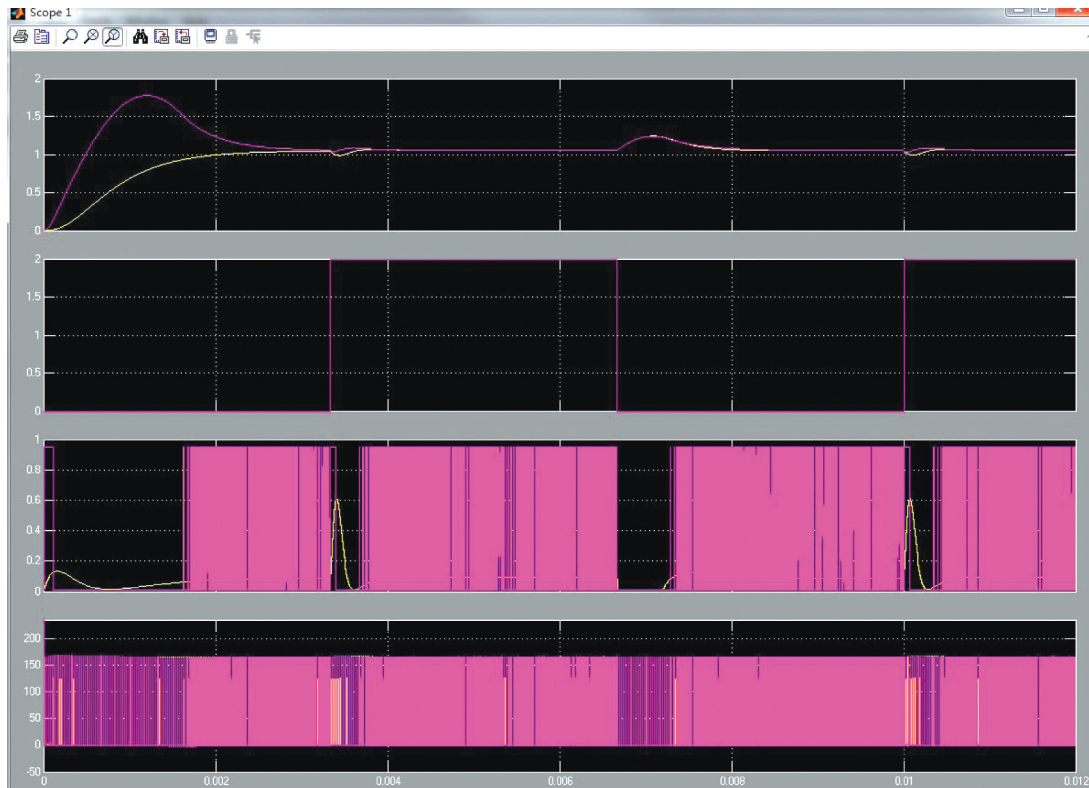


Figure 8: Comparison simulation result
(Source: Intel Corporation, 2012)

chart is the control signal output (PWM duty ration) from each controller. The bottom chart is the output current applied to the CPU. The purple line is PID, and the yellow line is ADRC.

Based on the comparison, we made the following observations:

1. For both ADRC and PID control, the output voltage all reach to the desired value 1.05 V after start transient and reach to steady state.
2. With the load step change, both ADRC and PID can correct the voltage back to 1.05 V with small overshoot or undershoot.
3. The major differences between ADRC and PID are the control signal output. The ADRC control is smooth and only acts when it is needed. PID does a busy control and it is very hard to maintain the output voltage at the same 1.05 V. Theoretically, the more efficient control will result in power savings, and we will look at how power saving ADRC can be provided quantitatively in the next step.

Energy Consumption Comparison between ADRC and PID

Figure 9 plots the integration of the input power to the VR circuit with both ADRC and PID control method; the integration of the power over time is the energy consumption. The energy consumption (yellow line)

“Compare the Integration of the power over time between ADRC and PID control method.”

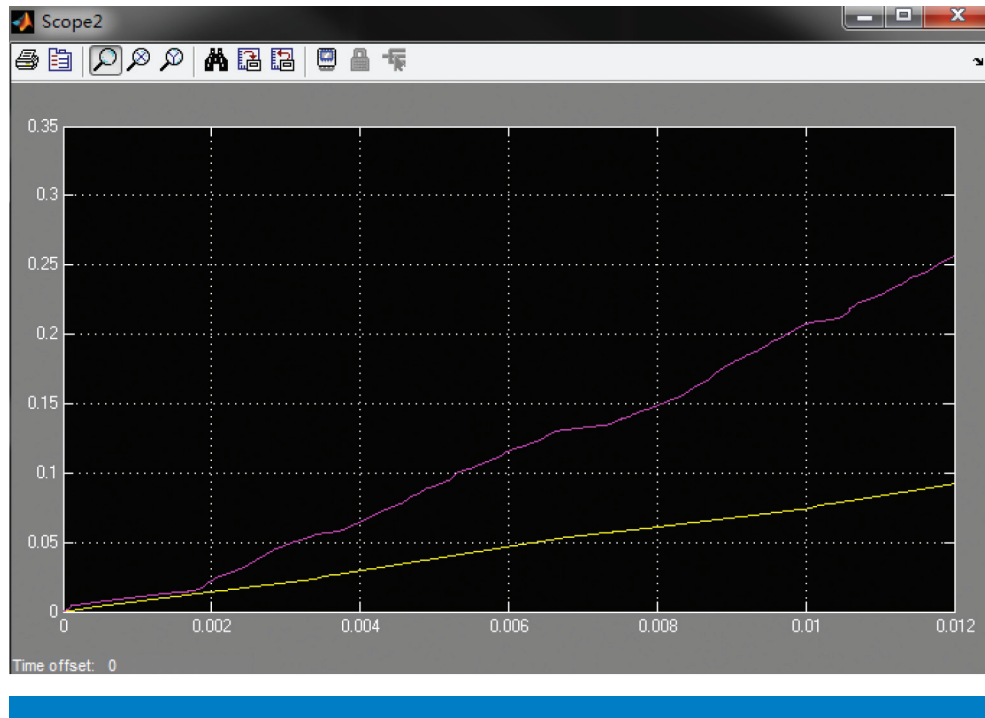


Figure 9: Energy consumption comparison between ADRC and PID
(Source: Intel Corporation, 2012)

with ADRC control method (purple line) is obviously less than the energy consumption with the PID control method. With the time last, the gap of energy consumption between ADRC and PID is significant. The energy consumption is calculated between time 0.0025 seconds and 0.012 seconds. The reason to choose 0.0025 seconds as the start time is because at 0.0025 seconds it has reached steady state after transient for both ADRC and PID to make a fair comparison.

Table 1 shows the quantitative energy consumption different between the ADRC and PID while the output voltage IAE between the ADRC and PID are about the same.

	Energy Consumption (Watt X second) (input voltage 12 V)	IAE
ADRC	0.0919	3.3927e-04
PID	0.2358	3.3784e-04

Table 1: Energy Consumption Comparison between ADRC and PID
(Source: Intel Corporation, 2012)

“ADRC control method save major power versus PID control method.”

Based on the data shown in Table 1, ADRC saves about 68 percent energy versus the PID control method for this CPU VTT VR circuit.

Power Saving Estimation at System Level

The simulation timespan for the above data is 0.0095 second, so assuming the same cyclic load is applied to the VR, we can derive what the power consumption is in hours, days, and a year. Table 2 gives a comparison about the energy consumption for various time spans. In a year, only the ADRC in the single VTT VR controller will save about 131.4 kWh of energy for the server. If the same control methodology applied to each VR in the server, and in a data center, the energy and cost saving would be tremendous.

	1 hour	1 day	1 year	1 year energy saving per VR
ADRC	0.0097 kWh	0.23 kWh	83.95 kWh	131.4 kWh
PID	0.0248 kWh	0.59 kWh	215.35 kWh	

Table 2: Energy Saving For Various Timespans
(Source: Intel Corporation, 2012)

Summary

Design principles pertaining to control systems in server subsystems are examined in this article to distinguish two different paradigms: the reactive PID and active disturbance rejection. It is shown how the ADRC principle can be systematically applied to facilitate advanced control development for server subsystems. One class of such subsystems, the CPU VR control, is used to illustrate how the concept fits and how the corresponding control algorithm is developed and validated in simulation, with encouraging results. Much work is ahead to further test the concept in hardware implementation and in the expansion of the investigation into other Server subsystems.

Acknowledgments

The authors would like to thank Ms. Qinling Zheng for her assistance in simulation.

References

- [1] U.S. EPA, "Report to congress on server and data center energy efficiency," Tech. Rep., Aug. 2007.
- [2] J.G. Koomey, "Estimating Total Power Consumption by Servers in the U.S. and the World."
- [3] P. Bohrer, E. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, and R. Rajamony, "The case for power management in web servers," Power Aware Computing, Jan 2002.
- [4] Intel, "First the Tick, Now the Tock: Intel Microarchitecture (Nehalem)," 2009.
- [5] L. Barroso and U. Hölzle, "The case for energy-proportional computing," IEEE Computer, Jan 2007.

"Scale the energy saving to a year."

"Design principles pertaining to control systems in server subsystems are examined in this article to distinguish two different paradigms: the reactive PID and active disturbance rejection."

- [6] K. J. Astrom and T. Hagglund, "PID Control," The Control Handbook, W.S. Levine, Ed. p. 198, CRC Press 1996.
- [7] N. Minorsky, "Directional Stability and Automatically Steered Bodies," J. Am. Soc. Nav. Eng., vol. 34, 1922, p. 280.
- [8] Z. Gao "On Disturbance Rejection Paradigm in Control Engineering," Proceedings of the 2010 Chinese Control Conference, July 29–31, 2010, Beijing, China.
- [9] G. Tian, Z. Gao, "From Poncelet's Invariance Principle to Active Disturbance Rejection," Proceedings of the 2009 American Control Conference, June 10–12, 2009, pp. 2451–2457.
- [10] Z. Gao, "Active Disturbance Rejection Control: A Paradigm Shift in Feedback Control System Design," Proc. of the 2006 American Control Conference, Minneapolis, June 14–16, 2006, pp. 2399–2405.
- [11] Z. Gao, "Scaling and Parameterization Based Controller Tuning," Proc. of the 2003 American Control Conference, June 2003, pp. 4989–4996.
- [12] B. Sun and Z. Gao, "A DSP-Based Active Disturbance Rejection Control Design for a 1 kW H-Bridge DC-DC Power Converter," IEEE Transactions on Industrial Electronics. Volume 52, Issue 5, pp.1271–1277, Oct. 2005.
- [13] Q. Zheng and Z. Gao, "On Practical Applications of Active Disturbance Rejection Control," Proceedings of the 2010 Chinese Control Conference, July 29–31, 2010 Beijing, China.
- [14] G. Tian, Z. Gao, "Frequency Response Analysis of Active Disturbance Rejection Based Control System," Proceedings of the 2007 IEEE Multi-conference on Systems and Control, Singapore, October 1–3, 2007.
- [15] Z. Ping and Z. Gao, "An FPGA-Based Digital Control and Communication Module for Space Power Management and Distribution Systems," Proceedings of the 2005 American Control Conference, Portland, June 8–10, 2005.
- [16] Stuart Bennet, "The Past Of PID Controllers," Annual Reviews in Control 25 (2001) pp. 43–53
- [17] Nobert Wiener, *Cybernetics*, Wiley, New York, 1948.
- [18] G. Schipanov, "Theory and methods of designing automatic regulators," *Automatika in Telemekhanika*, vol. 4, no. 1, pp. 49–66, 1939.
- [19] B. N. Petrov, "The invariance principle and the conditions for its application during calculation of linear and nonlinear systems," Proc. Of 1st IFAC Congress, Moscow, USSR, Butterworth. Ltd., London, England, vol. 1, pp. 117–126; 1961.

- [20] William Ross Ashby, *An Introduction to Cybernetics*, Wiley, New York, 1956.
- [21] J. Han, “Control theory: Model approach or control approach,” *Syst. Sci. Math.*, vol. 9, no. 4, pp. 328–335, 1989, (in Chinese).
- [22] Roger Brockett, “New Issues in the Mathematics of Control,” in *Mathematics Unlimited – 2001 and Beyond*, B. Engquist and W. Schmid Ed., pp. 189–220, Springer, 2001.
- [23] J. Han, “A class of extended state observers for uncertain systems,” *Control and Decision*, vol. 10, no. 1, pp. 85–88, 1995, (in Chinese).

Author Biographies

John Ping is a system architect in the Intel Data Center and Connected Systems Group. He has a Bachelor of Technology degree in electrical engineering from East China University of Science and Technology (Shanghai), and a master of science in electrical engineering and a doctor of engineering degree from Cleveland State University. His expertise and research interests span server system architecture, optimization of performance, power and energy in server nodes, racks and IP data centers. Another expertise and research area is advanced control algorithms. He can be reached at john.ping@intel.com

Zhiqiang Gao is associate professor of electrical engineering and director of the Center for Advanced Control Technologies at Cleveland State University. He received his PhD in electrical engineering from the University of Notre Dame in 1990. Employing an experimental science philosophy to research and a humanistic orientation to teaching, Dr. Gao and his team of researchers bring creative solutions to real world control problems and vitality of thinking to the young minds. He spent seventeen years developing ADRC from an obscure concept into a proven industrial control solution.

Rahul Khanna is a platform architect at Intel Corporation involved in development of energy efficient algorithms. Over the past 17 years he has worked on server system software technologies including platform automation, power/thermal optimization techniques, reliability, optimization, and predictive methodologies. He has authored several technical papers and book chapters in the areas related to energy optimization, platform wireless interconnects, sensor networks, interconnect reliability, predictive modeling, motion estimation, and security and holds 27 patents. He is also the co-inventor of the Intel IBIST methodology for high-speed interconnect testing. His research interests include machine learning based power/thermal optimization algorithms, narrow-channel high-speed wireless interconnects and information retrieval in dense sensor networks. Rahul is member of IEEE and the recipient of three Intel Achievement Awards for his contributions in areas related to advancements of platform technologies. He is the author of book *A Vision for Platform Autonomy: Robust Frameworks for Systems*. Rahul Khanna can be reached at rahul.khanna@intel.com