

# THE DESIGN OF A NEW STATE SPACE DIGITAL POWER SUPPLY PROTOTYPE

K. Shu<sup>†</sup>, J.X. Zhang, Institute of High Energy Physics, Beijing, China

## Abstract

Based on system identification and DLQR methods, a self-adaptive magnet power supply can be implemented by the new controller structure without any change in the power supply main circuit. Now only some raw tests have been taken which show its stableness, but the State Space Digital Controlling may turn out to be with potential. It makes power supplies with different capability easy to interchange without any more effort on tuning their parameters according to their loads, and can be extended to accommodate MIMO systems [1-3].

## BACKGROUND

The modern magnet digital power supplies are mostly composited as the following structure (see Fig 1).

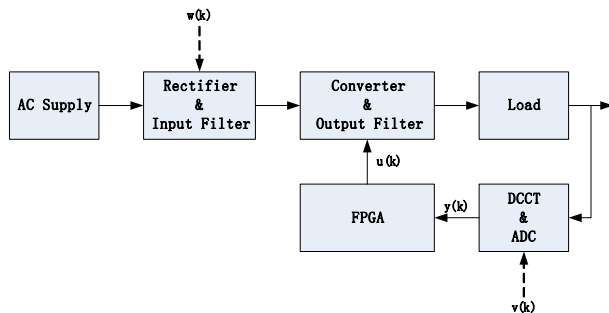


Figure 1: Magnet power supply framework.

The controller is running inside FPGA or DSP, which samples the feedback signal such as load voltage or magnet current, then computes the control variable to be translated into PWM signals (with switching power supply for example), and finally makes the feedback meeting the control index.

In the practice of the digital magnet power supplies in BEPCII and ADS, the conventional PID controlling, which is still powerful, has shown some inconvenience. As the controller is built and optimized on the system under control, the parameters such as the PID gain, the integral and derivative constant are decided on field tests and often with a lot of work. Still this work must be done all over again with a different magnet, or even with the same one in case that the environment conditions may shift over time.

## CONTROLLER STRUCTURE

This prototype takes the same hardware framework as conventional digital power supplies. However, the controller design, which is called the State Space Digital Controller, is totally different (e.g. Fig. 2).

<sup>†</sup> shuk@ihep.ac.cn

The masked grey box indicates that the actual power and magnet hardware underlying is “black” to users, because we will not use a full set of sensors and sampling. Instead, a state observer (or several observers) is used to estimate the state variables of the objects under observation, which are combined by the switching or other executor, the output filter, the magnet load, and the DCCT-AD sampling channel. This observer has the same input and output characteristic as the control objects, and iterates in the standard state space equation form:

$$\begin{aligned} \hat{X}(k+1) &= A\hat{X}(k) + Bu(k) \\ \hat{Y}(k) &= C\hat{X}(k) + Du(k) \end{aligned} \quad (1)$$

This iteration will be calculated in FPGA with a certain time constant that will soon be determined. Then, instead of direct feedback, we use approximate state vector feedback law as:

$$u(k) = -K\hat{X}(k) + r(k). \quad (2)$$

The L matrix is added to make the output estimation converge to the actual controlled variable fast enough, for the initial state is unknown. Since the estimated state variables are calculated by FPGA thus a lot faster than field samplings, this feedback structure may have a better dynamic behaviour than traditional design. The main disadvantage of using state feedback instead of output feedback is that the static error cannot be eliminated, so a type I servo subsystem (a PI loop) plays its role.

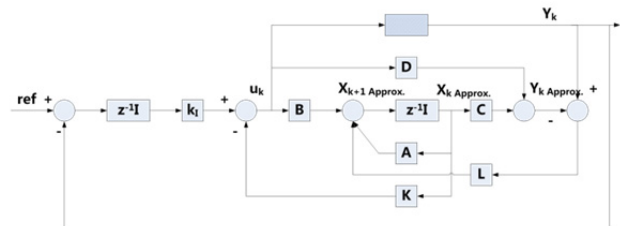


Figure 2: The controller structure.

## OBSERVER IDENTIFICATION

To make the controller online in situ, the observer must be identified online instead of being derived manually from circuit equations. There are two approaches to identify the system. One is to identify with noise signal, which is valid in theory but impractical in our situation because in the high precision controlling, the noises are probably covered by the ADC imprecision and the process and measurement noise  $w(k)$  and  $v(k)$  are often deeply colored thus cannot stimulate the proper subject patterns. So instead we use an active identification source called by

the Pseudo Random Number Generator formed by the LFSR array and been implemented in the FPGA. When the identification takes place, this generator gives a sequence of random reference to make the patterns in attentive frequency range arouse. Then this number sequence and the object output sample with the corresponding time mark are delivered into a FIFO memory waiting to be fetched by the computing unit as identification IO data pairs (Fig. 3).

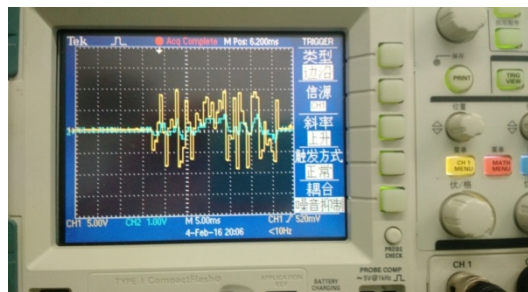


Figure 3: The RNG outputs.

It is important to note that the ID time interval must be carefully chosen to fit the hardware to be identified and the RNG and sampling FIFO must be synchronized to it. If the time interval is too large, we may not get an operating state space model because the dynamic process has been missed; contrariwise, the time interval too small will require too many IO pairs to cover this process which may exceed our computing capability as the algorithm must be carried out by the implemented core on FPGA (Fig. 4). A practical choice is one tenth of the object time constant. However, it can certainly be smaller to get a more precise model, given a sufficient hardware capability (we use Altera Cyclone II FPGA series).

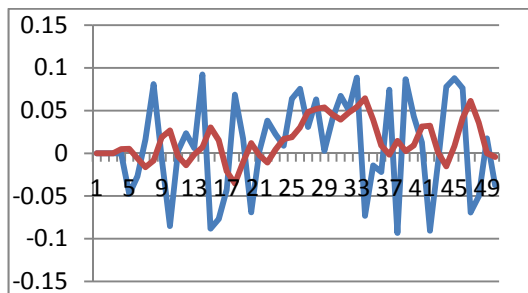


Figure 4: The sampled IO pairs.

The identification algorithm, considering our power supply situation whose reference is digital without any shifting while feedback is sampled by DCCT-ADC channel and is contaminated by measurement noise, is chosen to be PO-MOESP (the Past Output-MIMO Output Error State sPace algorithm), which is a well-developed sub-space model identification method running in time domain and requires only the IO data pairs and apriori order  $n$  ( $n$  can be large enough and will approach automatically to a more closer value).

## CONTROLLER DESIGN

After the identification, the object open loop equation is now given by formula 1. This design must automatically calculate the state feedback  $K$  and observer error gain  $L$ . The state space controller design can be carried out by close loop pole replacement. However, the  $z$ -field poles are not intuitive and hard to determine, and which is worse, the identified model is often with I-O direct path ( $D$  matrix, often quite small, but not neglectable). This does not necessarily mean that the actual hardware underlying has direct IO connection, but just a computing optimization result. However, it makes the virtual object with zeros, which makes the situation more complicating and even breaks down its close loop stability. So the optimizing control strategy is used to decide the state feedback matrix  $K$ .

A well-developed optimal quadratic controller, named by Digital Linear Quadratic Regulator, is used to adapt the high order and multi variable situation we face here. By a properly chosen  $Q$  and  $R$  matrix, a discrete algebraic Riccati equation is solved and the state feedback  $K$  matrix is then given. The details are well described in many researches which we will not discuss here. After that a stability check is taken by calculating the eigenvalues of the close loop iteration matrix  $A-BK$ , which should all be inside the unit circle, and then the observer error gain  $L$  matrix is calculated by pole replacement which places the eigenvalues of  $A-BK-LC$  to those specified by the vector  $1/4$  of the  $A-BK$  pole vector that is, in MATLAB language, described by  $L=place((A-BK)',C',P)'$ , where  $P=(eig(A-BK))'/4$ . Remember that the whole algorithm is run by implemented soft core in FPGA, so all the required calculating, including the  $Q$  and  $R$  matrix specification, the DARE solving and the Ackermann method for  $L$  calculation, is coded in C++, with the help of Eigen C++ template library for linear algebra.

Then the close loop combined by perimeter PI loop and inner state space loop, the observer iteration and the control law is given by the following formula:

$$\begin{aligned} \hat{X}(k) &= (A - BK - LC + LDK)\hat{X}(k-1) + (B - LD)I_e(k-1) \\ I_e(k) &= k_e e(k) + I_e(k-1) \\ u(k) &= -K\hat{X}(k) + I_e(k) \end{aligned}$$

The error is taken into calculation with  $e(k)$ , while  $I_e$  is the PI result.

## THE IMPLEMENTATION IN FPGA

All the controller parameters are downloaded from RAM to registers through the onboard databus after the identification and controller design algorithm is properly done. In addition to other peripherals required, the total VHDL design is formed.

A rngen module generates a customizable random reference sequence to drive the identification process, while a synchronized sample module storing the IO data pairs, which are then read by embedded core and the controller

parameters are to be calculated and written to the hardware, by which the control signal is generated according to  $u(k)$ .

We make a lot of efforts implementing the identification and DLQR algorithm on FPGA and SOPC so that the whole access can be automatically done without any work in PC or MATLAB. Limited by the onboard RAM capacity, the ID data pairs  $N$  cannot be too large, as the ID algorithm computing complexity increases by  $N^2$  (but the controller generation complexity remains constancy). So the ID generating and sampling rate must be set according to the time constant of the target, which arouses the problem that the target may have several parts which are of close influence on the control index but varies widely in frequency band. For example, the output filter is a lot faster in dynamic response than the magnet in the current controlling. So the ID process can be performed multiple times to fit in different time constant to generate multi-observers, whose outputs will then be iterated by different triggers and be combined to form a total estimation.

The operation procedure of this new prototype is shown in Fig. 5.

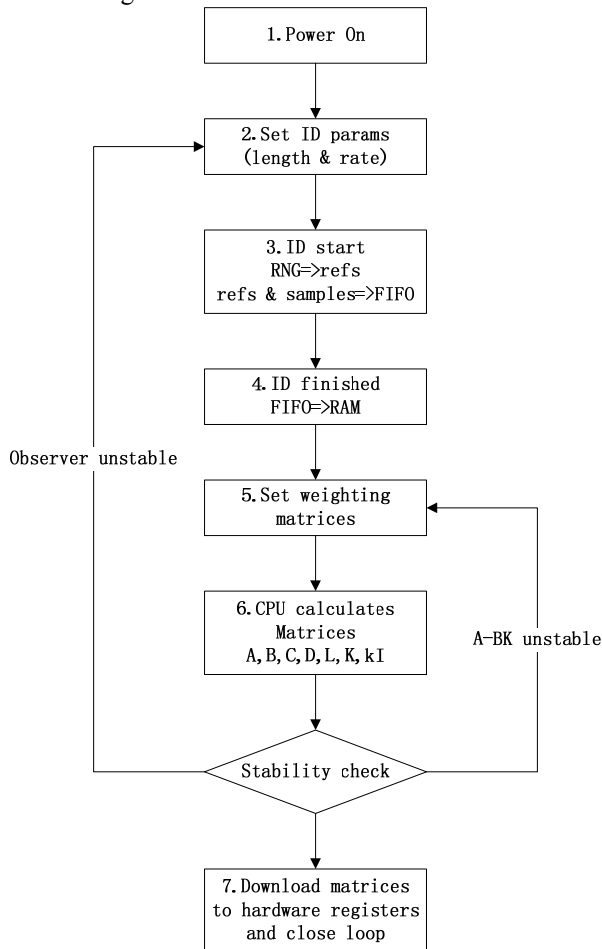


Figure 5: The operating procedure.

There are no apriori knowledge needed from the users expect the target time constant estimation. As the ID process can be run by order and the controlling parameters are generated automatically afterwards, the State Space Digital Controller we provided here is self-adaptive to any change in loads or environment.

The system is still in testing and now we can provide some early results by SSDC as a close loop step response (e.g. Fig. 6).

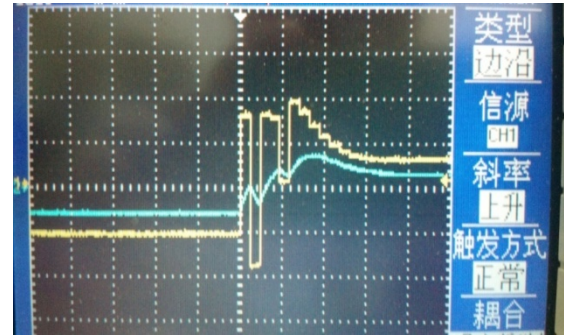


Figure 6: A SSDC step response

The line with step transitions indicates the control signal; the other line shows the feedback. In order to make it clear, the control variable is turned into true forms and then be transferred by a DA chip to voltage signal. This is not a good enough result, due to some nonlinear parts we use in the testing loads.

### CONCLUSION

This magnet power supply prototype focuses on the fast dynamic response and self-adaptive controller design. The identification of the control targets may show its significance in a lot of situations in future, and the self-adaptive design can save some on-site adjustment. Moreover this design takes a similar hardware configuration as modern digital magnet power supplies, which makes it suitable in many cases.

### REFERENCES

- [1] Eigen, <http://eigen.tuxfamily.org>
- [2] Haverkamp, L.R.J., (2001). *State space identification – Theory and practice*. [online] [https://www.researchgate.net/publication/27345172\\_State\\_space\\_identification\\_-\\_Theory\\_and\\_practice](https://www.researchgate.net/publication/27345172_State_space_identification_-_Theory_and_practice) (04/05/2016)
- [3] Peter Van OVERSCHEE, & Bart DE MOOR, (2002). *Subspace Identification for Linear Systems Theory - Implementation – Applications*. Boston: KLUWER ACADEMIC PUBLISHERS.