To appear in *Proc. IEEE Multi-Conference on Systems and Control (MSC)*, Sydney, 21-23 Sep. 2015.
(Final version, single-column format)

Application of Fading-Memory Polynomial Filters to the Control of an Electric Motor

Hugh L. Kennedy

***Abstract*—Recursive fading-memory polynomial (FMP) filters are used to support the frequency-domain design of digital lead/lag compensators, with two degrees-of-freedom (2-DOF) in a dc motor-control application. Digital proportional-integral-derivative (PID) and linear-state-space observer techniques are also applied for comparison. In this tutorial-style treatment, practical matters regarding system identification and discrete-time controller implementation, on a personal computer, are discussed.**

I. Introduction

Frequency-domain controller-design provides a direct link between system measurement and system analysis processes; it also allows feedback controller synthesis to be viewed as a filter design problem [1]-[9]. While the transfer function of the process to be controlled (i.e. the plant) cannot be changed, the transfer function of the compensator can be contrived to shape the loop frequency-response (i.e. around the loop from point A to point B in Fig. 1) to yield an integrated closed-loop system with the desired transient/steady-state response and satisfactory stability (i.e. gain and phase) margins. This perspective has the potential to be particularly useful when designing digital compensators for computer implementation in discrete-time control systems because it provides a direct link to digital signal processing (DSP) techniques [10].

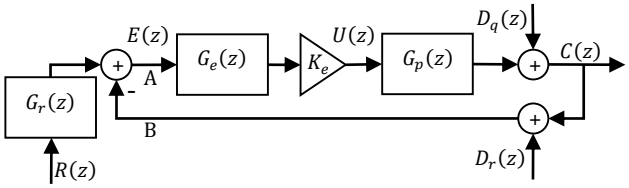

Fig. 1. The assumed 2-DOF controller and system structure. $R(z)$, $C(z)$, $E(z)$, $U(z)$, $D_q(z)$ and $D_r(z)$ are the $\mathcal{Z}$ transforms of the reference input $r(n)$, system output $c(n)$, error signal $e(n)$, control signal $u(n)$, plant-disturbance input $d_q(n)$ and sensor-noise input $d_r(n)$, respectively.

Frequency-domain compensator design is guided by the following basic principles [11]:

- High loop-gain is desirable wherever the closed-loop controller is required to exert its influence; for instance:
  - At low-frequencies, to reduce steady-state errors when tracking simple reference-inputs (e.g. steps and ramps) in servomechanisms;
  - At low-to-medium frequencies, for disturbance attenuation/rejection in regulators and an improved transient response in servomechanisms (i.e. faster rise time, faster settling time and reduced overshoot).
- Low loop gain at high frequencies is required to reduce sensor measurement noise and interference.
- The loop transfer function must attenuate frequencies wherever the phase lag is at (or some integer multiple of) 180 degrees, to guarantee stability in a closed loop configuration. This ensures that stable negative-feedback does not become unstable positive-feedback. The margins of stability, thus the ability of the closed-loop system to remain stable in the presence of plant modelling errors (e.g. unexpected system gains and delays), increase as the attenuation increases.

Lag and lead compensators are commonly used as loop-shaping instruments – Lag compensators are low-pass filters, with a backward phase shift over the transition band; lead compensators are high-pass filters, with a forward phase-shift over the transition band. Lag compensators are used to attenuate high-frequencies, with a phase lag as an undesirable side-effect; whereas lead filters are used to provide a forward phase-shift, with high-frequency amplification as an undesirable side-effect.

Lag compensators behave somewhat like an integrator; whereas lead compensators have differentiator-like properties. They may therefore be combined to emulate the properties of proportional/integral/derivative (PID)

H. L. Kennedy is with the Defence and Systems Institute within the School of Engineering, University of South Australia, Adelaide, SA 5095, Australia (e-mail: hugh.kennedy@unisa..edu.au).

controllers. PID controllers are indeed a specialized compensator form, with a marginally stable pole at $z = 1$ due to the integrator and up to two zeros. The absence of a pure integrator in a lag/lead compensator made it a popular type of analogue controller. As charge cannot be accumulated forever without dissipation, active circuitry is required to implement an analogue integrator (for infinite gain at dc). While this constraint does not directly apply to digital systems, care must still be taken to avoid numerical overflow due to persistent plant-output/reference-input offsets and integrator wind-up due to actuator saturation [12]. The lag and lead filters considered in this paper may also be used as digital filtering elements within a greater PID control architecture [16],[20].

Lag and lead components are clearly complementary. It would be ideal to have a single filter cut high-frequency gain (for improved noise immunity) *and* provide a phase lead (for improved stability) to permit the application of a large controller gain (for improved transient and stead-state behaviour); however in practice, a compromise must be reached with lead and/or lag filters tuned to yield an appropriate balance.

In Section II, the proposed controller structure is outlined. Details of the compensator design process, using fading-memory polynomial (FMP) and fading-memory sinusoidal (FMS) filters are given in [13]. However, only the FMP filters are employed here, as the particular problem considered requires *low-frequency* magnitude and phase manipulation. Sections III & IV are used to highlight some practical considerations regarding the use of FMP-based compensators in a real motor-control application. Performance of the proposed design approach, using FMP components, in the motor control problem is compared with some candidate design alternatives of similar complexity; namely, an empirically-tuned PID controller and a linear-state-space (LSS) servomechanism with an observer [14]. The paper closes with a summary, some recommendations and concluding remarks in Section IV. This paper is the second part of a two-part series on fading-memory filters in control. Design issues are considered in part one, using simulation [13]; whereas practical/implementation issues are considered here using a real control problem.

II. Overview of the Proposed Approach

The proposed polynomial filters are useful as frequency compensation elements, applied to the error signal, inside a controller with one degree-of-freedom (1-DOF). In this configuration, where the plant is discretized using a zero-order hold to yield the transfer function $G_p(z)$, the error transfer function $G_e(z)$ and controller gain K_e in the forward path are tuned to yield satisfactory reference tracking and disturbance rejection behaviour with sufficient stability margins, using an FMP filter.

In a 2-DOF configuration (see Fig. 1) $G_e(z)$ and K_e are primarily tuned for disturbance rejection and stability; a different filter $G_r(z)$ is then used to "shape" the reference input [15]-[17], with an optional gain factor K_r to help remove offset errors in the absence of a closed-loop integrator. For instance, a low-pass $G_r(z)$ filter could be used to excise high-frequency content from a step input that would otherwise excite complex poles near the unit circle in the closed-loop system. High-order reference-input filters may be useful in "fly-by-wire"-type control systems, where the plant is required to follow irregular and rapidly changing inputs from an operator, that do not have simple low-order forms.

III. Method and Results

Polynomial lead and lag compensators (i.e. the FMP filters) were designed and implemented in software running on a personal computer which was used to control a hobby-grade (brushed, dc) electric motor. In this application, the plant input is an electric potential (V) and the output is rotational speed, in units of revolutions per second (rps). An optical encoder disk with five black and five white sectors was attached to the drive shaft of the motor and a photoelectric sensor was placed in front of the disk's face to estimate the rotation rate. The sensor outputs a 4 V signal when a white sector passes by. The analogue pulse-train which is output by the sensor was digitized using an analogue-to-digital (A2D) converter at a rate of 40 kHz and processed using a low-level measurement thread coded using the C programming language. The measurement thread determined the instantaneous inter-pulse period using a simple threshold crossing algorithm then wrote the value to a shared buffer on the arrival of very new pulse.

The buffer was read asynchronously by a high-level control and user-interface thread, coded using the C# programming language. Periodic read events were triggered by a software timer at a rate of 20 Hz. The contents of the buffer were flushed and averaged on each read event. A lock mechanism was used to coordinate buffer access. The control thread then closed the loop by determining an appropriate input voltage for the motor. The control action was sent immediately to the motor via a digital-to-analogue (D2A) converter and held constant for one sample period. As control-action computation time was negligible, relative to the control loop period (T) of 0.05 s, there was little to be gained by waiting for the next control cycle to apply the action. Thus a fast response, possibly with some random variation (i.e. "jitter"), was favoured over a slower (delayed) deterministic response [18].

The C# software layer provides a graphical interface and allows the user to select, configure and visualize the operation of a variety of different control algorithms. Digital PID and a digital LSS algorithm were selected for

comparison in this work because they are of similar complexity and capability, thus they are all likely to be used in the same sorts of applications. The polynomial lag and lead compensators, with the FMP filters, were used in variants of what will be referred to as the "FMP controller".

The PID algorithm was implemented using the conventional "positional form", with all terms in the forward path. The D term was implemented using Euler's backward difference method, $G_d(z) = K_d \frac{(z-1)}{Tz}$; forward differences were used for the I term, $G_i(z) = K_i \frac{Tz}{(z-1)}$ [14].

The LSS algorithm allows closed-loop poles to be arbitrarily placed using internal state feedback while steady-state errors are eliminated using output feedback and an integrator. A full-order current Luenberger observer was used to estimate the internal states [14].

To allow a fair comparison between the algorithms, the lag and lead compensators were combined with a parallel integrator to yield FMP controllers with PI- and PID-type characteristics, respectively.

Three different scenarios were considered –

- *Baseline* scenario;
- *Noise* scenario, where the logic to average the pulse intervals was deactivated, so that the motor speed was inferred using only the most recent pulse interval; and
- *Delay* scenario where the plant input and plant output were passed through a two-sample delay-line to simulate controller-plant communication delays.

An attempt was made to tune all controllers for reasonable performance in all scenarios. However, to better illustrate the operation of the proposed lag and lead controllers, two polynomial implementations were used: one using a FMP lag filter, for the noise scenario; the other using a FMP lead filter, for the delay scenario. For a fair comparison, two PID tunings were also used – with and without the D component, for the delay and noise scenarios respectively.

The motor input was restricted to the 0 V to 5 V range; however, the motor reaches a near-maximum speed of approximately 200 rps at around 3 V and the speed only increases slightly as the voltage is further increased. There is also some 'stickiness' and hysteresis at low voltages, as the motor only begins to turn when the initial voltage is above 2 V. It then stops turning when the voltage falls below 1 V.

Reference input step responses for the various controllers are shown in Fig. 2 to Fig. 6. In these figures, a sliding time window of 10 seconds is shown (x axis). The solid black line is the reference input $r(nT)$, and the dashed gray line is the measured plant output $c(nT)$, in revolutions per second, ranging from 0 to 250 rps (scale shown on right-hand y axis). The solid gray line is the plant input voltage, or control signal $u(nT)$, ranging from 0 to 5 V (scale shown on the left-hand y axis). In automatic (closed-loop) mode, the user is able to adjust the reference input r, using a slider control in the graphical user interface to give sudden or gradual changes. Input steps of ± 50 rps are shown in the figures. In manual (open-loop) mode, the user adjusts the plant input u directly.

Using the manual mode of operation, the motor input was abruptly increased from 0 V to 3 V to create a step function input for system identification purposes [19]. The least-squares time-domain method described in [14] was then used. A second-order model with two-poles and no zeros fitted the data well. With T = 0.05 s, the pole locations were estimated to be at 0.9658 and 0.2717, yielding the following discrete-time model of the plant:

$$G_p(z) = \frac{1.7263}{z^2 - 1.2375z + 0.2624}. \qquad (1)$$

The LSS controller was designed to yield (three) repeated closed-loop poles at z = 0.75. Moving the poles closer to the origin for a faster response resulted in excessive noise amplification thus a rapidly fluctuating control signal and an excessively large control signal for step inputs, leading to actuator 'saturation' at 5 V then overshoot due to integrator 'windup'. Logic to handle integrator windup was included in the C# layer, although it was not activated in this study. The observer was designed with repeated poles at z = 0.25. After converting (1) into an equivalent state-space representation in controllable canonical form [14], these poles yielded a LSS controller with observer gains

$K_{\text{obs}} = [0.4413 \quad 0.4272]^{\text{T}}$

state feedback gains

$K_{\text{fbk}} = [0.1595 \quad -0.0125]$

and an integrator gain of

$K_{\text{int}} = 0.0091.$

See Fig. 2 for a plot of the closed-loop system response of this controller.

The PID controller was initially tuned using $K_p = 0.05$, $K_i = 0.05$ and $K_d = 0.00$, to yield a PI controller, for reasonable performance in the baseline and noise scenarios. Relatively low P & I gains and a zero D gain were required to reduce noise amplification. See Fig. 3 for a plot of the closed-loop system response of this controller.

The PID controller was subsequently tuned using $K_p = 0.05$, $K_i = 0.05$ and $K_d = 0.005$ for reasonable performance in the delay scenario. A small amount of derivative control dramatically decreased the settling time in the delay scenario but degraded performance in the other scenarios. See Fig. 4 for a plot of the closed-loop system response of this controller.

The FMP lag compensator, was tuned to reduce the effects of pulse measurement error using $B = 1$, $q = 2$ and $\sigma = -0.5$ to create G_e (see [13] for a definition and description of these filter parameters). When combined with an integrator using $K_i = 0.05$ and an error gain of $K_e = 0.05$, this gave a nominal gain margin of 5.6573 at 0.0764 cycles per sample and a nominal delay margin of 7.6275 samples at 0.0210 cycles per sample. See Fig. 5 for a plot of the closed-loop system response.

The FMP lead compensator was tuned to improve the stability margins in the delay scenario using $B = 2$, $q = -1$ and $\sigma = -1.0$ to create G_e (see [13] for a definition and description of these filter parameters). The G_e filter provides a positive phase shift at all frequencies with a maximum phase lead of approximately 30 degrees near 0.1 cycles per sample and a maximum gain of 7.2 dB near 0.25 cycles per sample. Using the plant model G_p as a guide, and with G_e, K_e and G_i combined, this compensator results in a theoretical gain margin of 4.9867 at 0.1751 cycles per sample and a delay margin of 11.0543 samples at 0.0195 cycles per sample for $K_e = 0.05$ and $K_i = 0.05$. As desired, the delay margin of the lead compensator is significantly greater than the delay margin of the lag compensator, for a similar gain margin, which results in greater damping. See Fig. 6 for a plot of the closed-loop system response.

The FMP lag and lead compensators both used reference filters with $B = 0$, $q = 0$ and $\sigma = -2.0$ (see [13] for a definition and description of these filter parameters). These settings result in a moderate G_r, with only a slight low-pass effect, primarily due to the short filter memory.

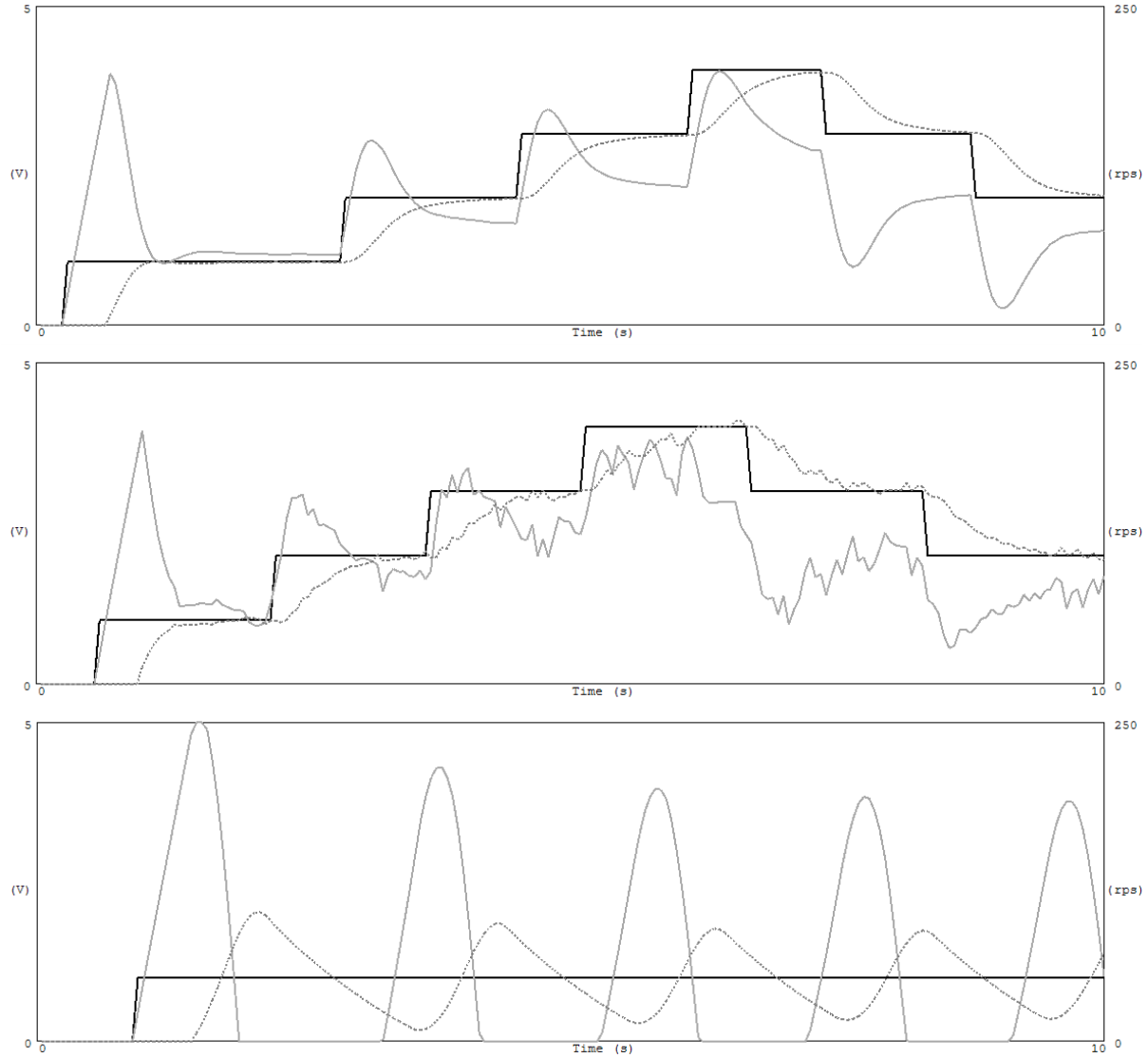

Fig. 2. Response of the LSS motor controller. In the baseline (top), noise (middle), and delay (bottom), scenarios. See text for subplot description.

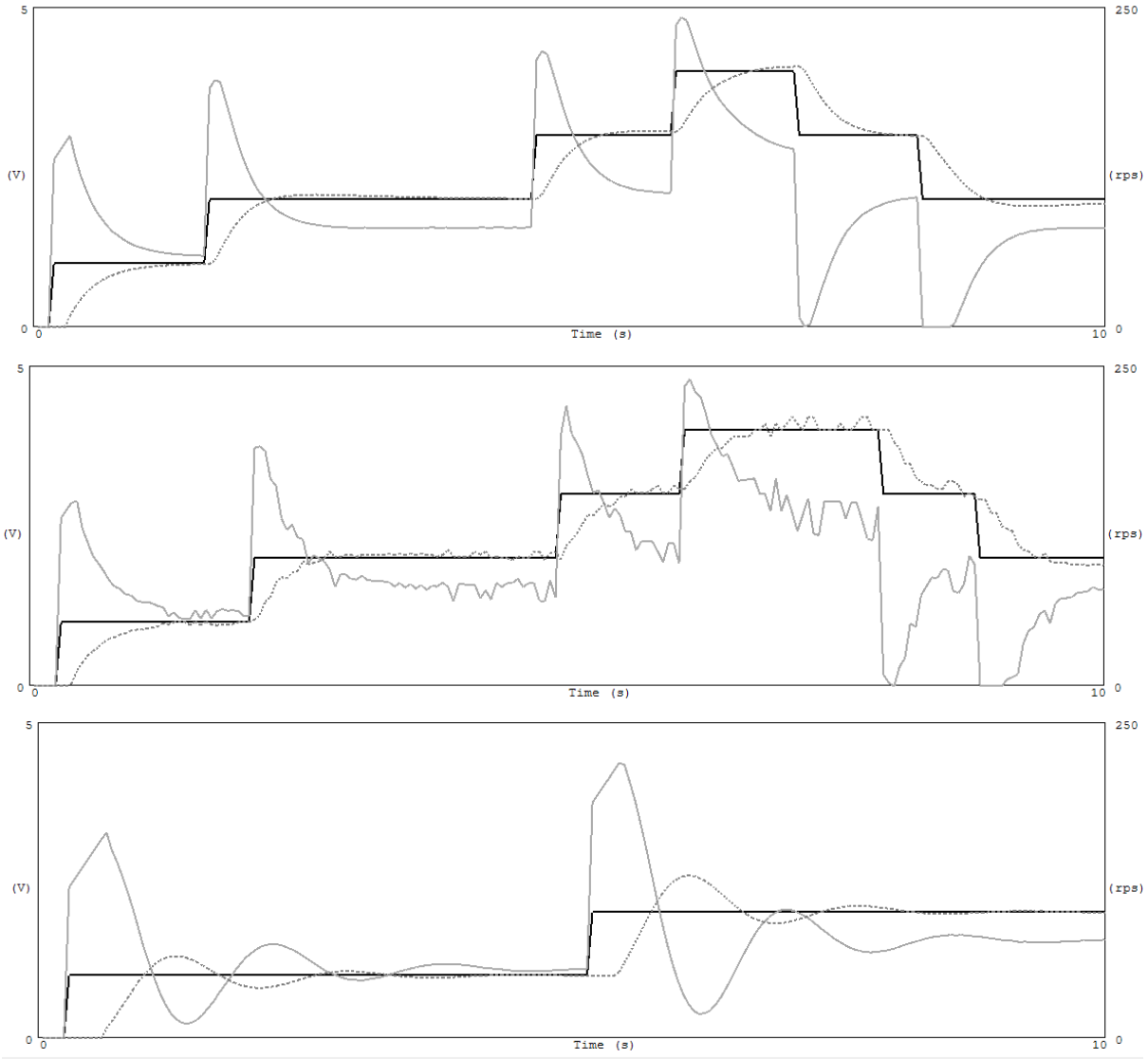

Fig. 3. Response of the PI motor controller. In the baseline (top), noise (middle), and delay (bottom), scenarios. See text for subplot description.

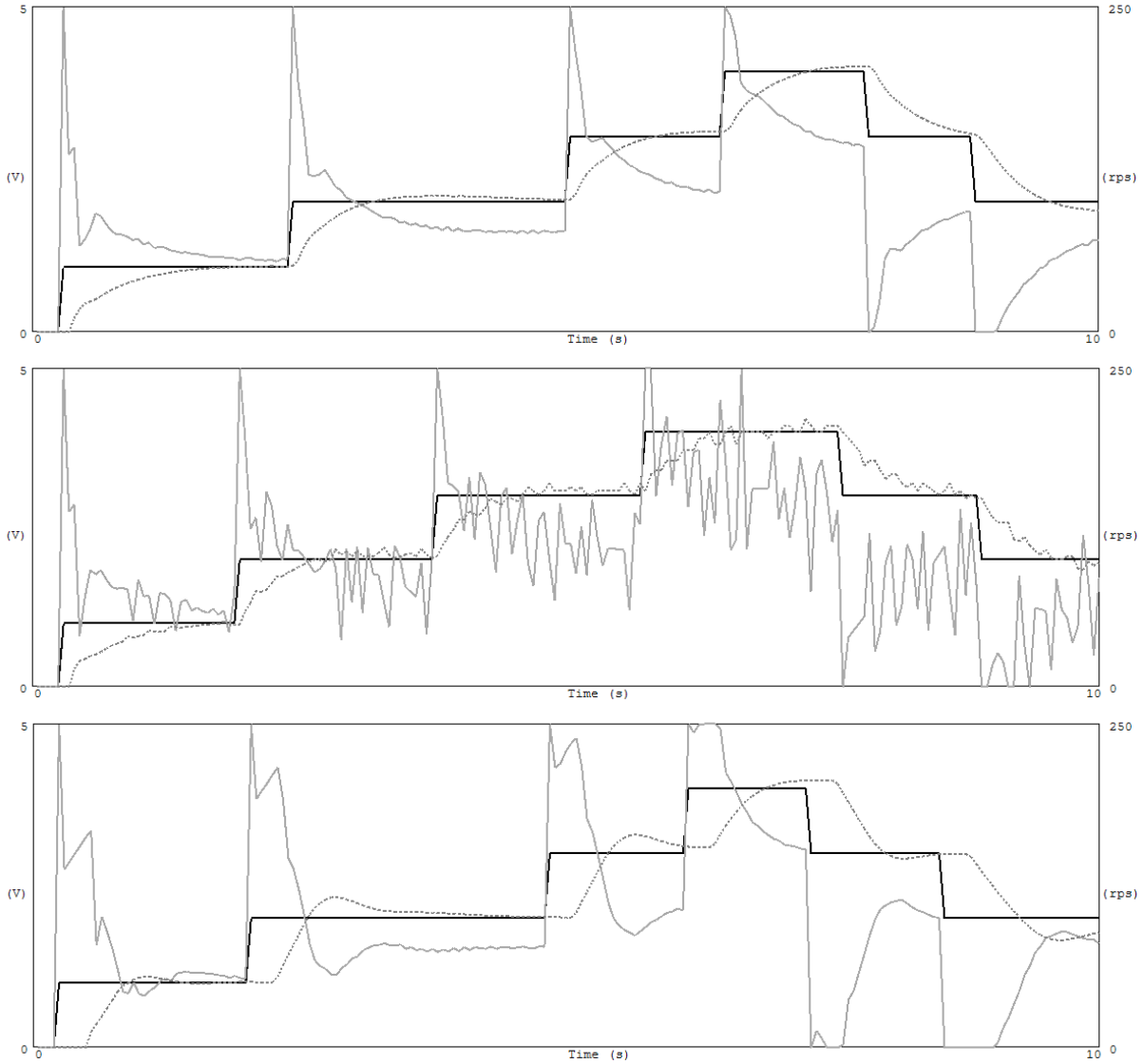

Fig. 4. Response of the PID motor controller. In the baseline (top), noise (middle), and delay (bottom), scenarios. See text for subplot description.

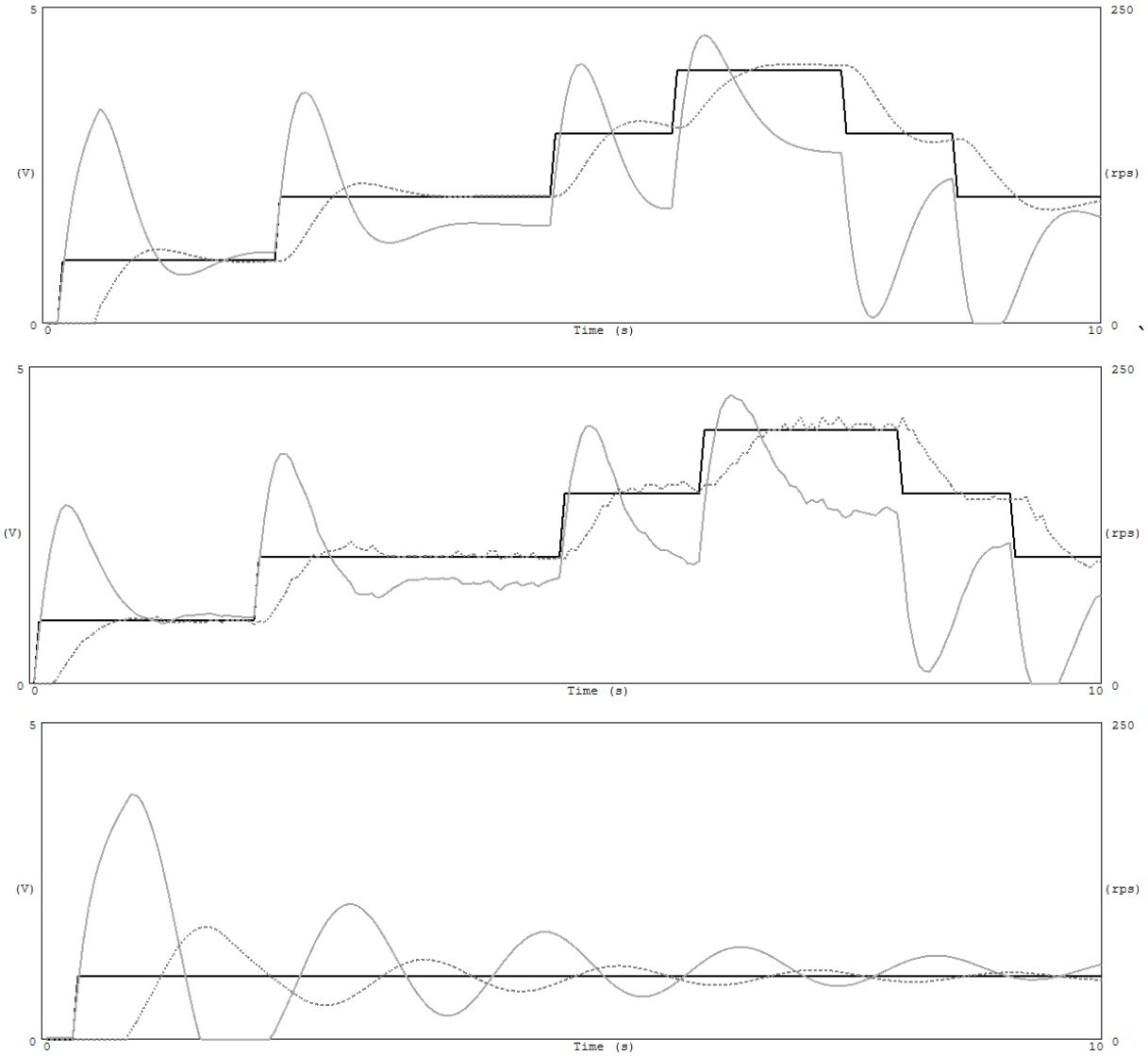

Fig. 5. Response of the FMP motor controller with a polynomial lag filter. In the baseline (top), noise (middle), and delay (bottom), scenarios. See text for subplot description.

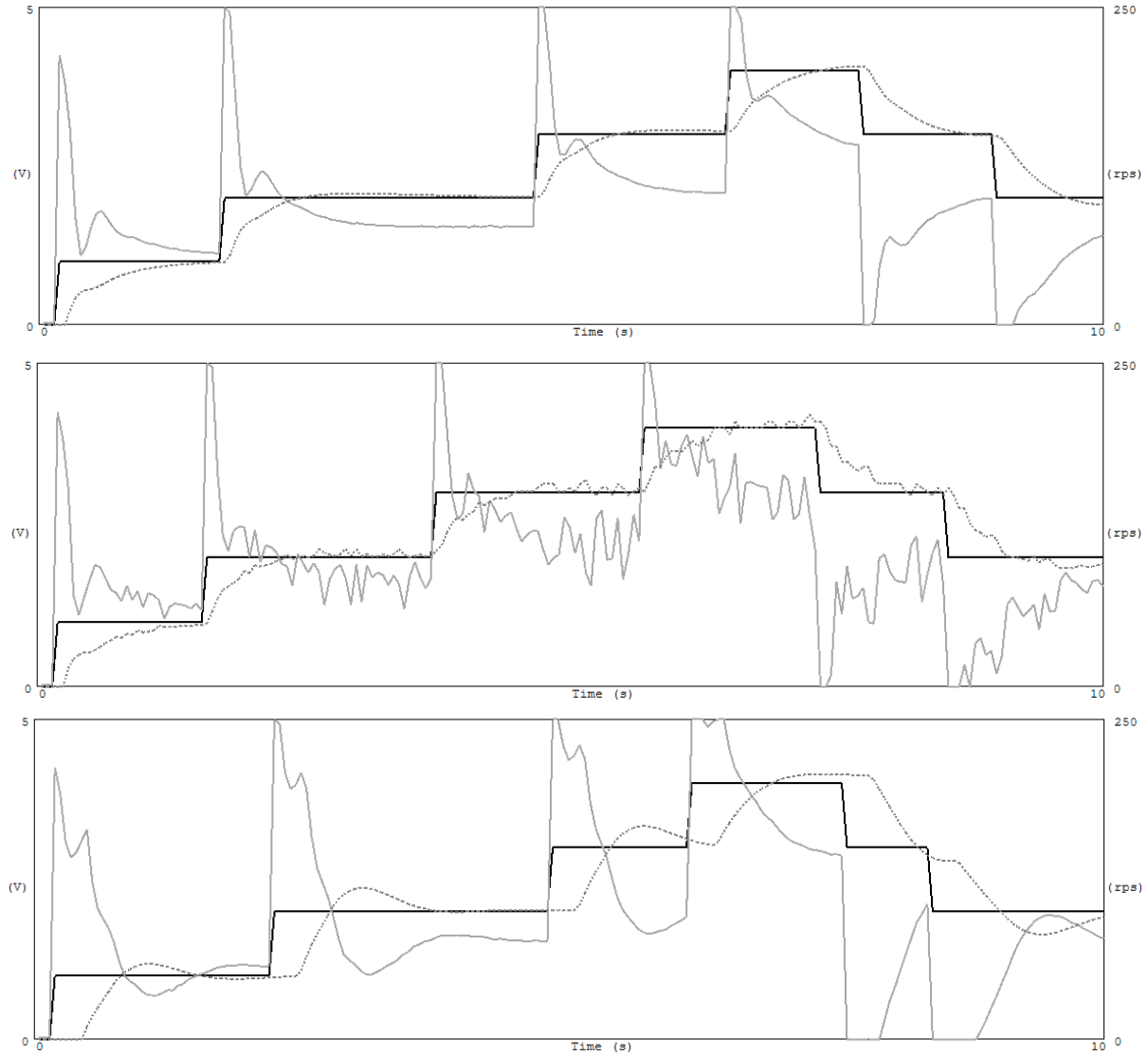

Fig. 6. Response of the FMP motor controller with a polynomial lead filter. In the baseline (top), noise (middle), and delay (bottom), scenarios. See text for subplot description.

IV. Discussion

Given the flexibility of the three algorithms (PID, LSS and FMP) considered in the motor-control experiment and the lack of any real system requirements, it is difficult to draw meaningful performance conclusions. All algorithms could be tuned to give reasonably similar behavior. However as is already well known and widely appreciated, one obvious difference between the controller types is the ease and speed with which the PID algorithm can be designed and tuned for simple plants. The LSS and FMP controller coefficients were generated using hand-coded Matlab ® scripts then imported into the C# tool. Once the scripts were written, controller design was a fairly straightforward process. PID does not need this supporting design 'infrastructure', although it may be utilized if available.

Even though the LSS and FMP controllers use a model of the plant, the plant model was not perfect, mainly due to the non-linearities discussed earlier; therefore some empirical tuning was still required. The plant model did however somewhat reduce the time spent tuning. The use of a plant model can be both an advantage and a disadvantage – depending on the quality of the model. The PID controller is unaffected by the model because an empirical tuning approach was used; the LSS controller is strongly affected by the model; whereas the FMP controller allows the model to be used to help the developer understand the relationships between the various conflicting design parameters during the tuning process [9].

If controller development time is the sum of

a) plant modeling;
b) controller coding and implementation; and
c) controller integration, test and tuning activities;

then the production of a good plant model, and the use of a controller design technique that utilizes the derived model, transfers development time from activity c) to activity a). Unlike industrial control problems, for the simple motor control application considered here, there were no safety or productivity penalties associated with online tuning of the closed-loop system. Thus development effort was equally allocated to modeling and tuning activities. This also meant that the appeal of theoretical approaches (such as LSS), or semi-empirical approaches (such as FMP), relative to empirical approaches (such as PID), was somewhat diminished.

Regardless, the simple model and system identification approach appeared to be adequate, as the LSS output more-or-less matched expectations. The predicted closed-loop response of the FMP controller for the model plant also closely matched the actual response for the real plant. Theoretical or predicted step responses, produced via modelling, are plotted in Fig. 7; actual step responses are plotted in the top subplot of Fig. 5 and Fig. 6; isolated and exported step responses, from a 'rolling start', are also plotted on the same axes in Fig. 8. According to the system model (see Fig. 7), the lag filter is slightly under-damped; the lead filter has a faster initial rise rate, reduced overshoot but a slightly extended settling time. All of these characteristics are apparent in the response of the real system (see Fig. 8), which indicates that the plant model was reasonable. The theoretical step response and the estimated gain/phase margins could therefore be used to guide and constrain the compensator tuning process.

In addition to the step input, various pulsed and free-form inputs were also tried during system identification. The location of the (dominant) slow pole was reasonably consistent and reproducible, for all input waveforms and on all occasions; the location of the fast pole was somewhat more variable. The step input waveform was mainly used for simplicity and reproducibility. It was also found to be ideal for precise location of the slow pole. More elaborate waveforms may be better for high-order systems [8]; however for various practical reasons, simple steps are still widely used [19].

The time-domain least-squares approach to system identification is perhaps more compatible with LSS controller because the poles and zeros of the plant, thus the form of the recursive observer, are determined directly. The frequency response is then derived by evaluating the discrete-time transfer-function around the unit circle. A frequency-domain system-identification approach, driven by sinusoidal inputs to stimulate critical frequencies, probably would have suited the FMP controller better. Model perfection is not however critical for the successful application of the FMP controller. As a consequence, the designer is able to choose between time spent identifying the system and time spent tuning the compensator.

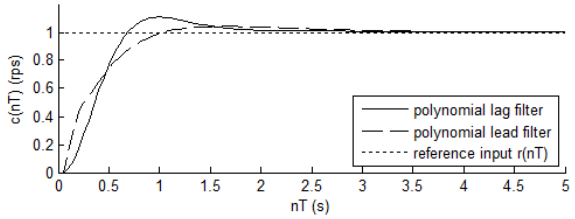

Fig. 7. Predicted response of the FMP controller for two different types of compensator and a unit-step reference input.

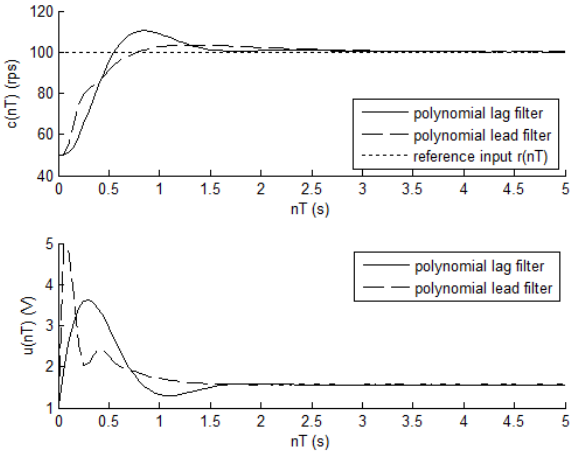

Fig. 8. Actual response of the FMP controller for two different types of compensator and a reference input step of 100 rps.

The FMP controller with the polynomial lead filter gave very similar performance to the PID controller (see Fig. 6 and Fig. 4). Both controllers improved the response in the delay scenario at the expense of the response in the noise scenario. The FMP controller with a polynomial lag filter resulted in slightly better noise suppression than the PI controller; however, the PI step response settled slightly faster in the delay scenario (see Fig. 5 and Fig. 3).

One of advantages in using the lag and lead filters in the FMP controller is the extra loop-shaping flexibility. A single polynomial filter with adjustable lag and lead properties may be used to replace the P and D branches of a PID controller [20]. Furthermore, use of the polynomial filter eliminates the need for customized low-pass derivative filters to reduce noise in the D path. On the one hand, when designing a derivative filter, one of the design issues is the balance between low-frequency phase linearity and high-frequency noise attenuation [21]-[24]; however, in a loop shaping context it is not clear which of these requirements is the more important. On the other hand, when designing a lead filter, the designer only needs to consider gain and phase, using stability margins as a guide.

The LSS controller implementation is slightly more involved than the other controllers. Only first- and second-order variants were coded, using in-line math operations. As a result, no attempt was made to increase the model order to accommodate the system delay, which would presumably have improved the performance in the delay scenario. In the absence of any remedial measures, the LSS controller was only marginally stable in the delay scenario. Had it not been for the actuator saturation which limited the control action, the system probably would have been unstable. The LSS controller performed reasonably well in the noise scenario and the step response appeared to be critically damped (or at least over-damped) in the baseline scenario, as per the intent of the design (see Fig. 2).

The PID controller performed much better than the LSS controller in the delay scenario, but due to the use of the D term, it performed much worse than the LSS controller in the noise scenario. The non-zero D term was also responsible for the spike (or "kick") in the control signal when the step reference input is first applied (see Fig. 4). Of all the controllers, PID was the easiest controller to tune for a given scenario. However, the FMP controller has many more degrees of tuning freedom, due to the filter design process, which can be both a good and bad feature, depending on the circumstances. The lack of PID flexibility was most evident in the noise scenario, where little could be done in the PI filter to reduce noise amplification, other than setting K_d to zero scenario (see the middle subplot of Fig. 3). Reducing K_p further did help somewhat in this respect but it also slowed the transient response in the baseline scenario.

The logic to average the pulse periods in the C# layer, common to all controllers, had a significant impact on the dynamics of the closed-loop system. This process is part of the controller; however, it appears as part of the plant, because it is required to generate the output for observation. The measurement process is therefore integrated with the plant and included in the system identification process, giving rise to the fast pole of the plant. Smoothing of some kind is essential to reduce noise and to fully utilize all pulse measurements that are collected by the C layer over each

period of the C# timer – around 70 pulses per timer tick when the motor is operating near maximum speed. At these speeds, even when the pulse train from the optical encoder is digitized at rate of 40 kHz, an error of just one sample in the identification of the pulse edges in the C layer potentially results in a speed error of around 10% in the C# layer. Smoothing the pulse-period measurements hides these errors. Various other smoothing algorithms were considered – for instance, a first-order low-pass IIR filter in the C layer. While this approach allows the behavior of the smoother to be configured, averaging all pulses over a finite time interval was found to give a smoother output and better closed-loop performance in general. As the degree of smoothing decreased, the measurement pole moved to the left along the real z axis, towards zero. Decreased smoothing increased the stability margins and allowed faster responses to achieved; however, it also increased measurement noise and degraded steady-state performance; as a consequence, derivative and phase-lead filters could not be used effectively in any of the scenarios.

V. Conclusion

When compared with other compensator design techniques, the proposed method, using FMP filters [13]:

- Does not suffer from distortion associated with s-to-z mappings because design is performed entirely in the z domain.
- Is more flexible than simpler methods involving closed-form expressions for first- and second-order components, which are not readily generalized to forms of higher order.
- Eliminates the need for guesswork when cascading multiple low-order units or when manually assigning single poles or zeros in the z plane.

The PID controller, the LSS controller, and the FMP controller involving the polynomial lag and lead filters, could all be tuned for similar performance in the simple scenarios considered here. Therefore, the three methods should probably be regarded as being different controller design *processes*, rather than different controller *realizations*. However, the FMP controller has the greatest number of tuning parameters, due to the flexible filter-design process. This feature was found to be most useful in the noise scenario where the lag compensator was able to slightly outperform the PI filter, albeit at the expense of performance in the delay scenario.

In broader control engineering problems, the best solution is determined by many constraints that were not contemplated in this study, it is therefore difficult and unwise to draw definite performance conclusions here. The experiments performed did however reveal the following:

- It was gratifying to use the PID algorithm because a very good controller could be designed and implemented with a minimum of time and effort.
- It was satisfying to use the LSS algorithm because less guess work was required to tune the controller due to the utilization of the plant model; however, a few design iterations were still required.
- It was reassuring to use the FMP algorithm because the plant model and frequency analysis could be used to
  - Justify an initial controller design,
  - Guide the fine-tuning process and
  - Set approximate upper bounds on the controller parameters via the stability margins.

Like PID and LSS, the proposed digital FMP filters may be used to design simple, effective and flexible controllers.

References

[1] Y. Chen, "Replacing a PID controller by a lag-lead compensator for a robot-a frequency-response approach," *IEEE Trans. Robotics Automation*, vol. 5, no. 2, pp. 174-182, Apr. 1989.

[2] W.C. Messner, M.D. Bedillion, Lu Xia and D.C. Karns, "Lead and lag compensators with complex poles and zeros design formulas for modeling and loop shaping," *IEEE Control Syst.*, vol. 27, no. 1, pp. 44-54, Feb. 2007.

[3] W. Messner, "Formulas for asymmetric lead and lag compensators," in *Proc. American Control Conference (ACC '09)*, pp. 3769-3774, 10-12 Jun. 2009.

[4] Qi Zhang and W.C. Messner, "Root locus design with Complex Proportional-Integral-Lead compensation," in *Proc. American Control Conference (ACC)*, pp. 693-698, Jun. 29 - Jul. 1, 2011.

[5] W. Messner, "Some advances in loop shaping controller design with applications to disk drives," *IEEE Trans. Magnetics*, vol. 37, no. 2, pp. 651-656, Mar. 2001.

[6] R. Zanasi and Stefania Cuoghi, "Direct method for digital lead-lag design: analytical and graphical solutions," in *Proc. IEEE Conference on Automation Science and Engineering (CASE)*, pp. 804-809, 24-27 Aug. 2011.

[7] Jie Feng and M. C. Smith, "When is a controller optimal in the sense of $\mathcal{H}_\infty$ loop-shaping?," *IEEE Trans. Automatic Control*, vol. 40, no. 12, pp. 2026-2039, Dec. 1995.

[8] Yuichi Chida, Yoshiyuki Ishihara, Takuya Okina, Tomoaki Nishimura, Koichi Ohtomi and Ryo Furukawa, "Identification and frequency shaping control of a vibration isolation system," *Control Engineering Practice*, vol. 16, no. 6, pp. 711-723, Jun. 2008.

[9] J. C. Doyle, B. A. Francis and A. R. Tannenbaum, *Feedback Control Theory*, Dover Publications, 2009.

[10] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing* (4th Edition), Prentice Hall, 2006.

[11] G. F. Franklin, J. D. Powell and M. L. Workman, *Digital Control of Dynamic Systems* (2nd Edition), Addison-Wesley, 1990.

[12] Hwi-Beom Shin and Jong-Gyu Park, "Anti-Windup PID Controller With Integral State Predictor for Variable-Speed Motor Drives," *IEEE Trans. Industrial Electronics*, vol. 59, no. 3, pp. 1509-1516, Mar. 2012.
[13] H. L. Kennedy, "Numerical Derivation of Fading-Memory Polynomial and Sinusoidal Filters for Discrete-Time Control Systems," *in Proc. IEEE Multi-Conference on Systems and Control (MSC)*, 21-23 Sep., 2015 (to appear).
[14] K. Ogata, *Discrete-Time Control Systems*, New Jersey, Prentice Hall, 1987.
[15] U. Boettcher, D. Fetzer, H. Li, R. A. de Callafon and F. E. Talke, "Reference Signal Shaping for Closed-Loop Systems With Application to Seeking in Hard Disk Drives," *IEEE Trans. Control Syst. Technology*, vol. 20, no. 2, pp. 335-345, Mar. 2012.
[16] T. Hägglund, "A unified discussion on signal filtering in PID control," *Control Engineering Practice*, vol. 21, no. 8, pp. 994-1006, Aug. 2013.
[17] V. Vijayan and R. C. Panda, "Design of PID controllers in double feedback loops for SISO systems with set-point filters," *ISA Trans.*, vol. 51, no. 4, pp. 514-521, Jul. 2012.
[18] S. J. Ovaska and O. Vainio, "Predictive compensation of time-varying computing delay on real-time control systems," *IEEE Trans. Control Syst. Technology*, vol. 5, no. 5, pp. 523-526, Sep 1997.
[19] S. Ahmed, B. Huang and S. L. Shah, "Novel identification method from step response," *Control Engineering Practice*, vol. 15, no. 5, pp. 545-556, May 2007.
[20] H. Kennedy, "Recursive Digital Filters With Tunable Lag and Lead Characteristics for Proportional-Differential Control," *IEEE Trans. Control Syst. Technology*, to appear (2015).
[21] M. A. Al-Alaoui, "Class of digital integrators and differentiators," *IET Signal Process.*, vol. 5, no. 2, pp. 251-260, Apr. 2011.
[22] B.T. Krishna and S.S. Rao, "On design and applications of digital differentiators," in *Proc. Fourth International Conference on Advanced Computing (ICoAC)*, pp. 1-7, 13-15 Dec. 2012.
[23] R. H. Brown, S. C. Schneider and M. G. Mulligan, "Analysis of algorithms for velocity estimation from discrete position versus time data," *IEEE Trans. Industrial Electronics*, vol. 39, no. 1, pp. 11-19, Feb. 1992.
[24] F. Dietrich, J. Maass and A. Raatz, "Computationally Efficient Adaption of the Window Size of Discrete Position Differentiators," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 4, pp. 1377-1384, Aug. 2013.