Transient Enhancement in Add-on Feedforward Algorithms for High-performance Mechatronic Systems

Xu Chen* and Masayoshi Tomizuka*

* Department of Mechanical Engineering, University of California, Berkeley, CA, 94720, USA (e-mails: {maxchen, tomizuka}@me.berkeley.edu).

Abstract: In this paper, we analyze the problem of control injection in feedforward compensation, and discuss the transient enhancement in add-on feedforward algorithms. In contrast to the common feedforward injection at the plant input, the updated reference feedforward design is discussed with the advantage of accelerated transient. Such a configuration also brings convenience from the viewpoints of filter implementation and uniformed design in industrial mass production. Two implementation examples are provided from advanced manufacturing and precision motion control.

Keywords: feedforward control, vibration rejection, iterative learning control, mechatronics

1. INTRODUCTION

Combining feedforward and feedback controls is a standard practice in control engineering. This paper discusses the problem of control-effort injection in such a combined control scheme. More specifically, consider a general closed-loop control system in Fig. 1. The dotted lines denote the distribution of the feedforward efforts, which are “added on” to the baseline feedback system. From the viewpoint of injecting the add-on feedforward command, we can classify the two designs as:

- updated control (UC), where feedforward \( u_{ff} \) is directly injected at the input of the plant \( P \);
- updated reference (UR), where \( r_{ff} \) is added as an update of the reference \( r \).

For feedforward control, updated reference is equivalent to control injection at the location of the feedback error \( e \) in Fig. 1. In both cases, the dynamics between \( r_{ff} \) and \( y \) equals the complementary sensitivity function \( T = PC/(1 + PC) \).

Both UC and UR feedforward control are vastly implemented in practice. Popular examples can be found, for instance, in general two-degree-of-freedom motion control (Sugie and Yoshikawa, 1986; Li and Tomizuka, 1999; Liu et al., 2005), iterative learning control (Amann et al., 1996; Bristow et al., 2006; Moore et al., 2006; Ahn et al., 2007; Mishra, 2009), and adaptive or sensor-based feedforward compensation (Bodson et al., 1994; Zhang et al., 2000; Widrow and Walach, 2007). These schemes are all central in mechatronics and industrial motion control. For designing the feedforward filters, representative techniques include: scalar gain approximation such as the P-type iterative learning control (Saab, 1994; Bristow et al., 2006; Ahn et al., 2007), discrete-time Taylor expansion (Widrow and Walach, 2007), zero-phase-error-tracking (ZPET) design (Tomizuka, 1987), and \( H_2/H_\infty \) design (Moore et al., 2008; Nie and Horowitz, 2011).

Despite the rich literature about feedforward control and design, not much investigations are available about the difference between UC and UR. From transfer-function analysis, it seems reasonable to always be able to equivalently transform from one scheme to the other, as the structural difference between the feedforward commands in UR and UC is that \( r_{ff} \) in UR is filtered by one additional filter \( C \). In this paper, further investigations are made to examine the functional differences between UC and UR. We remark first, that transfer-function analysis reveals the steady-state performance but not the transient properties of the system. For the class of add-on or plug-in feedforward control, the transient at different injection points are different. Analysis and solutions of the problem is important for plug-in compensation, which is central in many practical problems such as iterative learning control (where add-on feedforward command is injected to the closed loop at different iterations) and feedforward
Similar to Li and Tomizuka (1999); Sugie and Yoshikawa (1986); Liu et al. (2005), we refer to the basic combined

2. SIMULATION AND EXPERIMENTAL SET UP

The experimental results in this paper are obtained on a testbed of a wafer scanner/stepper system shown in
Fig. 2. Such systems are essential for photolithography in fabrication of integrated circuits in the semiconductor
industry. There are two moving stages in the testbed, mounted on air bearings and actuated by epoxy-core linear
permanent magnet motors. The stage positions are measured by laser interferometers. A LabVIEW real-time
system with field-programmable gate array (FPGA) is used to execute the control commands. The frequency
response of the upper reticle stage is shown in Fig. 3. The input and the output are respectively the motor voltage
command and the position of the moving stage.

Additional evaluations are performed on a benchmark hard disk drive (HDD) simulation tool developed by IEEJ,
Technical Committee for Novel Nanoscale Servo Control (2007), a technical committee consisting of both industrial
and academic experts in the field. The benchmark has been frequently used in the disk drive community for algorithm
verification and performance comparison.

Both the wafer stage and the HDD are typical examples of integrated control systems in mechatronics, where we
commonly have integrator or mass-damper type of plant dynamics (recall Fig. 3), due to the working principles of
motors and actuators.

2. SIMULATION AND EXPERIMENTAL SET UP

The experimental results in this paper are obtained on a testbed of a wafer scanner/stepper system shown in
Fig. 2. Such systems are essential for photolithography in fabrication of integrated circuits in the semiconductor
industry. There are two moving stages in the testbed, mounted on air bearings and actuated by epoxy-core linear
permanent magnet motors. The stage positions are measured by laser interferometers. A LabVIEW real-time
system with field-programmable gate array (FPGA) is used to execute the control commands. The frequency
response of the upper reticle stage is shown in Fig. 3. The input and the output are respectively the motor voltage
command and the position of the moving stage.

Additional evaluations are performed on a benchmark hard disk drive (HDD) simulation tool developed by IEEJ,
Technical Committee for Novel Nanoscale Servo Control (2007), a technical committee consisting of both industrial
and academic experts in the field. The benchmark has been frequently used in the disk drive community for algorithm
verification and performance comparison.

Both the wafer stage and the HDD are typical examples of integrated control systems in mechatronics, where we
commonly have integrator or mass-damper type of plant dynamics (recall Fig. 3), due to the working principles of
motors and actuators.

2. SIMULATION AND EXPERIMENTAL SET UP

The experimental results in this paper are obtained on a testbed of a wafer scanner/stepper system shown in
Fig. 2. Such systems are essential for photolithography in fabrication of integrated circuits in the semiconductor
industry. There are two moving stages in the testbed, mounted on air bearings and actuated by epoxy-core linear
permanent magnet motors. The stage positions are measured by laser interferometers. A LabVIEW real-time
system with field-programmable gate array (FPGA) is used to execute the control commands. The frequency
response of the upper reticle stage is shown in Fig. 3. The input and the output are respectively the motor voltage
command and the position of the moving stage.

Additional evaluations are performed on a benchmark hard disk drive (HDD) simulation tool developed by IEEJ,
Technical Committee for Novel Nanoscale Servo Control (2007), a technical committee consisting of both industrial
and academic experts in the field. The benchmark has been frequently used in the disk drive community for algorithm
verification and performance comparison.

Both the wafer stage and the HDD are typical examples of integrated control systems in mechatronics, where we
commonly have integrator or mass-damper type of plant dynamics (recall Fig. 3), due to the working principles of
motors and actuators.
Fig. 6. Command decomposition in UR

![Command decomposition in UR](image)

Fig. 7. 2DOF design with updated control feedforward

![2DOF design with updated control feedforward](image)

Fig. 8. 2DOF design with updated reference feedforward

We focus on repeated tracking problems. ILC can also handle repeated disturbances. The analysis is analogous and omitted here.

4.2 Iterative Learning Control

Iterative learning control (ILC) is a standard technique in controlling repeated process (e.g., transportation of the workpieces in a product line in manufacturing). Using the errors made in the previous cycle of the process, ILC updates the control strategies as demonstrated in Fig. 10. In this UC ILC scheme, manipulation is performed to the final control command:

$$u_{j+1} = u + u_{ff,j+1} = u + [Q \{u_{ff,j} + [L] e_j\}]$$

where $j$ is the iteration number, $L$ is the learning filter that extracts the desired information from the previous error $e_j$ $\hat{=} r - y_j$, and $Q$ is the robustness filter to control the learning process.

The added $u_{ff,j+1}$ affects the output by

$$y_{j+1} = \left[PC \over 1 + PC\right]r + \left[P \over 1 + PC\right]u_{ff,j+1}$$

The new errors can be shown to satisfy

---

1 In this paper, we use $[G]\,r$ to represent the output of the transfer function $G$ w.r.t. the input $r$.

2 We focus on repeated tracking problems. ILC can also handle repeated disturbances. The analysis is analogous and omitted here.
Fig. 11. Sensor-based feedforward control in HDDs

\[ e_{j+1} = \left[ Q \left( 1 - L \frac{P}{1 + PC} \right) \right] e_j + \left[ \frac{1 - Q}{1 + PC} \right] r \]  \hspace{1cm} (7)

On the other hand, for UR ILC, the feedforward term is injected via

\[ r_{j+1} = r + r_{ff,j+1} = r + \left[ Q \right] \left\{ r_{ff,j} + \left[ L \right] e_j \right\} \]

Analagous to UC ILC,

\[ y_{j+1} = \left[ \frac{PC}{1 + PC} \right] r + \left[ \frac{PC}{1 + PC} \right] r_{ff,j+1} \]  \hspace{1cm} (8)

and

\[ e_{j+1} = \left[ Q \left( 1 - L \frac{PC}{1 + PC} \right) \right] e_j + \left[ \frac{1 - Q}{1 + PC} \right] r \]  \hspace{1cm} (9)

From (7) and (9), perfect learning is achieved by \( Q = 1 \); \( L = [P/(1+PC)]^{-1} \) in UC ILC and \( L = [PC/(1+PC)]^{-1} \) in UR ILC. In both cases, the feedforward command is of add-on type, with iteration-dependent signal properties.

Comparing (2) with (6), and (4) with (8), we see that the 2DOF feedforward and ILC are structurally quite similar, and \{\( PC/(1+PC) \), \( P/(1+PC) \}\} are the two central transfer functions determining the transient and steady-state performances.

4.3 Sensor-based Feedforward Disturbance Rejection

A significant issue in practical precision systems is to deal with large external disturbances. Such disturbances may come from expectable error sources in the working environment (e.g., the vibration of a HDD in a personal laptop playing loud music or movies), or unexpectedly as a shock disturbance to the system. The latter is a particularly important issue in advanced manufacturing, where the required working condition of the fabrication device is usually so demanding that a sudden shock may pause or even stop the manufacturing process.

In sensor-based feedforward compensation, motion sensing elements such as accelerometers are applied to measure the (expectable and unexpectable) disturbances. Take the example of hard disk drive control. The sensor measurement detects the vibration reflected on the PCB board, which is correlated to the actual disturbance to the HDD actuator, as shown in Fig. 11. Feedforward design aims at generating the compensation signal to cancel the effect of the disturbance \( d \) on the position error signal (PES).

It is clear that the construction of feedforward should depend not only on \( P_1 \) and \( P_2 \), but also on the location of feedforward control injection. Fig. 11 can be first simplified to the general block diagram in Fig. 12. Here the effect of the external vibration is lumped as an input disturbance to the closed loop, with the related dynamics explained by

\[ G_d = P/(1 + PC) \]  \hspace{1cm} (10)

Ideally, to minimize PES, we need

\[ G_d P_1 - GW P_2 = 0 \Leftrightarrow W = \frac{G_d P_1}{GP_2} \]

With \( G_d = P/(1 + PC) \) and the above table of \( G \), we have

\[ W = \frac{P_1}{P_2} \frac{G_d}{GP_2} \]

The above are the ideal-case feedforward controller structures, if the filters are stable and realizable in practice. Again, the steady-state performance is the same if \( W \) is correctly configured according to the aforementioned discussions, yet transient exists when feedforward is injected to the closed-loop in Fig. 12. The transient dynamics depends on the dynamics of \( G \), namely \( P/(1 + PC) \) in UC and \( PC/(1 + PC) \) in UR.

5. TRANSIENT PERFORMANCE

Following the comparison and steady-state equivalence condition of UC and UR in Section 4, this section analyzes the transient performance of the two feedforward control allocation schemes. Let \( P = N_P/D_P \), \( C = N_C/D_C \), where \( N_{(j)} \) and \( D_{(j)} \) denote respectively the numerator and denominator polynomials of a transfer function. Then

\[ P \]

It can be observed that the above transfer functions differ only in the location of the zeros. For the response to the signal discontinuities in Figs. 4 and 6, the pursued goal in this section is to show that (10) provides faster transient than (11) under a properly designed feedback loop.

We first note that in general motion control, the roots of \( N_C \) (i.e., zeros of \( C \)) are always designed to be stable. Simple evaluation verifies this point in PID and lead-lag based controls. For instance, consider the PID controller

\[ C(s) = k_p + k_i \frac{1}{s} + k_ds = \frac{k_ps + k_i + k_ds^2}{s} \]  \hspace{1cm} (12)

where all the PID gains are positive. The Routh test reveals that all zeros of \( C(s) \) are stable for all \( k_p > 0 \), \( k_i > 0 \), and \( k_d > 0 \).
0. The same conclusion can be made for the causal version (for practical implementation) where \( C(s) = k_p + k_i \frac{1}{s} + k_d \frac{\tau}{s + \frac{1}{\tau}} \) \((\tau \text{ is a small positive scalar)}\). The above are for designs in the continuous-time domain. After discretization via bilinear transform \( s = 2(z - 1)/[T_s(z + 1)]\) \((T_s \text{ is the sampling time)}\), the left half of the \( s \) plane is mapped to the inside of the unit circle on the \( z \) plane, and the digital version of the controller preserves the stability of the zeros. Hence in both continuous- and discrete-time designs, the controller has stable zeros. Indeed, non-minimum-phase zeros of \( C \) (if there are any) will become non-minimum-phase zeros of the loop transfer function \( L = PC \). From various fundamental limitations of feedback control (Doyle et al. (1992); Stein (2003)), open-loop unstable zeros will amplify the waterbed effect and place direct constraints on the achievable servo performance. Hence, unless really necessary, \( C \) should be designed to have stable zeros.

The situation is different for the roots of \( DC \) (poles of \( C \)). For the case of PID control in (12), \( C \) has a marginally stable pole. For lead-lag controllers, a stable pole close to the imaginary axis will occur due to the module of \((s + b)/(s + a)\). Actually, from basic frequency-response concepts, marginal poles at low frequencies are always needed for high-gain feedback to achieve good low-frequency servo.

It is well-known that stable zeros can accelerate the transient and unstable zeros reduce the speed of transient and can even introduce undesired undershoot in the step response. Hence from the design principles of the feedback loop, (10) is better than (11) from the viewpoint of transient performance. Indeed, (10) is the closed-loop complementary sensitivity function and the transfer function from the reference \( r \) to the output \( y \). In any (linear and even nonlinear) feedback design, great efforts are made to make sure good tracking of \( r \) by \( y \), both in the steady-state and the transient stages. While usually little or no direct attention is paid to the transient and steady-state performances of \( P/(1 + PC) \). Hence from either the perspective of stable and unstable zeros, or fundamental goals of feedback design, \( PC/(1 + PC) \), and hence UR feed-forward allocation, would have better and more consistent transient performance.

Next, validation and some additional design properties are provided via a practical example. Consider the plant \( P(s) = 1/(0.2556s^2 + 0.279s) \), which is the nominal model of one stage of the wafer-scanner system in Section 2.

Closing the (negative) feedback loop with \( C(s) = 3000(1 + 2\frac{1}{4} + 0.012s) \) yields \(^3\)

\[
\frac{P(s)C(s)}{1 + P(s)C(s)} = \frac{36s^2 + 3000s + 6000}{0.2556s^3 + 36.279s^2 + 3000s + 6000} = \frac{3.9124s}{(s + 2.05)(s^2 + 139.9s + 11450)}
\]

(13)

Fig. 13. Transient comparison: step response

Comparing the central transfer functions for updated reference (14) and updated control (13), we see that the slowly convergent mode \( e^{-2.05009s} \) (corresponding to the pole at \(-2.05009\)) is firstly removed in (14), and hence accelerating the transient. Meanwhile, the marginally stable zero in (13) is replaced with a strictly stable one in (14). It is a standard result that such a zero improves the transient speed: for instance, from initial-value theorem, the step response of (14) has an initial slope of 140.8451 (assuming zero-initial conditions for \( P(s) \) and \( C(s) \) while the step response of (13) has a zero initial slope—much slower than the response of (14). Rigorous analysis using the theory of distributions can provide the mathematical expressions of different transient responses in (14) and (13). Limited by space, the results are omitted in this paper.

Figs. 13 and 14 compare the step and the impulse responses for \( P/(1 + PC) \) and \( PC/(1 + PC) \). The results support the mathematical analysis: for both types of transient evaluation, \( PC/(1 + PC) \) has much shorter transients compared to \( P/(1 + PC) \).

6. SIMULATION AND EXPERIMENTS

The analysis in preceding sections suggests that the same steady-state performance can be obtained in UC and
UR feedforward designs, and that UR provides better transient responses than UC. Certainly, $T = PC/(1 + PC)$ also has a standard shape in feedback design (it is designed to approximate identity), and the shape of $P/(1 + PC)$ is largely plant dependent. Such properties also make UR feedforward beneficial from the viewpoints of filter implementation and industrial mass production. This section validates the results with two application examples.

6.1 Simulated regulation control on a HDD

This example uses the HDD benchmark to verify the influence of feedforward allocations in regulation control. Here the reference is zero. A set of baseline disturbances consisting of white sensor noises, periodic disturbances from the disk rotations, and non-repeatable colored noises, is applied to the plant. For testing the transient performance with respect to shock resistance, we additionally applied (i) a step disturbance, and (ii) a half sinusoidal disturbance to the system. Both tests reveal the transient properties of the closed loop. In particular, (ii) is a standard testing procedure in HDD industry.

The overall control scheme is as shown in Fig. 12. The feedback controller in this case comes from the benchmark, which gives a baseline bandwidth of 1.19 kHz, a gain margin of 5.45 dB, and a phase margin of 38.2 deg. The dynamics $P_1$ and $P_2$ between the disturbance source and the HDD control loop are selected such that their ratio $P_1/P_2$ matches that of experimental measurements provided in White (1997).

Figs. 15 and 16 show the transient difference in this sensor-based feedforward control. In Fig. 16, a step shock disturbance is injected at 1 sec. Although the sensor measurement is always turned on, in practical situations, the feedforward command is turned on after a disturbance detector reports large errors. In this simulation, a delay of 0.2 sec is used to mimic the effect of the disturbance detector. From the results in Fig. 15, the PES immediately reduces after turning on the feedforward compensation, and both algorithms performed well at steady state. The difference of the transient performance is however also self-explanatory. In UC feedforward (FF), the longer transient of $P/(1 + PC)$ makes the settling time much longer than the case of UR FF. Similar observations can be made in the case of half-sinusoidal shock response in Fig. 16.

6.2 Experiments of tracking control on a wafer scanner

The second example compares the feedforward performance for combined 2DOF and ILC tracking control in the wafer scanner testbed. The reference scanning trajectory is as shown in the top plot of Fig. 17. Also shown in the figure is the baseline tracking result without feedforward control. After turning on the 2DOF feedforward control, the tracking error improved as shown in Fig. 18. The results indicate first, that both feedforward schemes significantly improves the tracking error—the maximum magnitude of the error in Fig. 17 is close to $2 \times 10^{-5}$ m while in Fig. 18 it is about one magnitude smaller. Second, injection at the reference provides improved transient response—at the transition regions of the trajectory (at around 0.2 sec, 0.75 sec, 1 sec, and 1.5 sec), the UR feedforward with $T^{-1}$ provides much smaller errors compared the UC feedforward with $P^{-1}$.

$T^{-1}$ and $P^{-1}$ are both not strictly stable in this example, and approximated by ZPET method (Tomizuka, 1987), according to the remark at the end of Section 4.1. The bias in the error difference is due to the slight difference in the initial condition at time 0, and the fact that there is model uncertainty in the plant. Such bias is immediately removed in Fig. 19, where online ILC is applied and information from the physical plant (instead of just the models) comes into the learning scheme.
In this paper, we have compared the performance differences between updated control and updated reference in feedforward design, and revealed the often neglected transient comparisons in feedforward control allocation. We have shown, via analysis of three common feedforward schemes, the conditions of steady-state equivalence, and the intrinsic faster transient performance in updated reference control. The results are general for motion control and mechatronics, and are important for various types of add-on feedforward designs.

7. CONCLUSION

In this paper, we have compared the performance differences between updated control and updated reference in feedforward design, and revealed the often neglected transient comparisons in feedforward control allocation. We have shown, via analysis of three common feedforward schemes, the conditions of steady-state equivalence, and the intrinsic faster transient performance in updated reference control. The results are general for motion control and mechatronics, and are important for various types of add-on feedforward designs.

REFERENCES


