Learning Control Strategies for High-Rate Materials Testing Machines

Proc. IMechE, Part I: J. of Syst. and Control Eng., 225(8):1125–1135.

Michael Schlotter*, Andrew Plummer†

DRAFT

Hydraulic high strain rate materials testing machines are required to track a user-defined velocity profile during tensile or compression tests in the face of sudden large impact forces. Due to delays and limited bandwidth of the actuation system, causal feedback/feedforward controllers fail to compensate for these disturbances. This paper presents more suitable non-causal learning control strategies, which anticipate the impact and take corrective action in advance. Two control strategies are discussed: The first comprises an iterative algorithm, which calculates a command signal correction by passing the velocity error observed in the previous test through an inverse model linearised around the target velocity. In the second approach, a detailed nonlinear inverse model is used to obtain a command signal from demand motion and force data. It is concluded that the first method is superior if two or more iterations can be performed.

Keywords: iterative control, learning control, motion control, electrohydraulic servosystems, materials testing

*Centre for Power Transmission and Motion Control, Department of Mechanical Engineering, University of Bath, UK. Email: m.schlotter@bath.ac.uk

†Centre for Power Transmission and Motion Control, Department of Mechanical Engineering, University of Bath, UK. Email: a.r.plummer@bath.ac.uk

Contents

Nomenclature

c_l	cross-piston leakage
k	actuator stiffness coefficient
A_1, A_2	actuator areas
B	bulk modulus
C_p, C_x	linearisation coefficients
F	specimen force
K_v	valve constant
M	mass of moving parts
P_r, P_s	return/supply pressure
P_1, P_2	actuator chamber pressures
Q_1, Q_2	actuator chamber flows
T_d	delay samples
T_s	sample time
U	piston velocity
V_1, V_2	actuator chamber volumes
W	drive signal
X	valve spool displacement
X_v	normalised valve spool displacement
Y	piston displacement
δ	delay
ζ_1, ζ_2	valve/actuator damping
ω_1, ω_2	valve/actuator eigenfrequency

1 Introduction

Iterative motion control of hydraulic actuators originated in the automotive area [1] and is still widely used in structural and materials testing applications [2, 3]. The objective is to reproduce a certain target motion profile specified by the user or measured in field tests. This strategy falls within the framework of iterative learning control (ILC), which has received significant attention from the control community in the last two decades [4, 5].

ILC aims to improve the transient response of systems operating repetitively over a fixed time interval. In contrast to non-learning controllers, which cause the same tracking errors in each pass, ILC uses existing error information from previous trials to modify the control signal for the next repetition. As control signals are computed off-line, non-causal signal processing such as forward shifting and zero-phase filtering can be employed. Hence, learning controllers can anticipate repeating disturbances and respond preemptively. This feature proves to be particularly valuable for plants subjected to rapid transient disturbances, which can not be rejected well by traditional closed-loop controllers due to delays and limited bandwidth of the actuation system.

Numerous ILC algorithms have been proposed, and design methods range from manual tuning of PD-type learning controllers to more systematic approaches based on quadratically optimal design and H_∞ synthesis [6]. Commonly, ILC performance is assessed by the asymptotic value of the tracking error after an infinite number of iterations, and most ILC concepts can be shown to yield satisfactory results. In practice, however, the transient learning behaviour, i.e. the reduction rate of the error magnitude during the first few iterations, is a more important performance criterion. Although monotonic convergence can usually be achieved, the convergence rate can be relatively slow especially with simple PD-type ILC algorithms. This is a major issue for destructive testing applications, considering that a specimen is destroyed during each test.

By using a plant inversion algorithm, the tracking error can, in theory, be reduced to zero in only one iteration. Of course, high-order ill-conditioned plants, modelling errors, non-minimum phase zeros, and inherent physical limitations of the machine make this impossible. Nevertheless, ILC strategies based on inverted dynamics can be successful, see [7, 8] for example. Experimentally identified frequency domain models are commonly used in the testing industry for that purpose, so that a priori knowledge about the plant and specimen dynamics is not necessary. As the characteristics of hydraulic actuation systems and the specimen are nonlinear, adaptive algorithms are occasionally employed to update the linear plant model during the iteration process depending on the current operating point.

Inverse plant models must be derived analytically for certain systems, which do not

lend themselves to experimental frequency response measurements. Examples in the literature include a uni-directionally operating catapult for crash test simulations [9]. In this application with purely inertial loads, the target acceleration command is perturbed with the acceleration error from the previous test. This modified target acceleration signal is then passed through a nonlinear inverse model, which calculates the new drive signal. A different control algorithm for high speed materials testing machines is presented in [10]. In this case, a drive signal correction is derived with a linear inverse model from the measured velocity error of the previous test. This correction is then added to the original drive signal for the next run.

This paper briefly describes the challenges encountered with high-rate testing in Section 2. The physical modelling and linearisation of a typical high-rate testing machine is outlined in Section 3. Section 4 shows how the linear iteration method described in [10] can be improved by considering not only the velocity error but also the force measurements from the previous test for drive signal profiling. In Section 5, a nonlinear approach for calculating the command signal from with demand motion and force information is presented. Section 6 compares the performance of the algorithms and discusses their usefulness in practice.

2 High-Rate Materials Testing

High rate materials testing is used for automotive crash worthiness assessment, sports equipment safety checks, and material characterisation. The aim is to extend or compress a specimen until destruction with a pre-defined velocity, which may be constant or time-varying. Typical test velocities range from 0.1 m/s to 20 m/s at forces of up to 500 kN.

A high strain rate materials testing machine is of similar appearance to a conventional hydraulic machine for quasi-static tensile, compression or fatigue testing. However, there are differences in the construction to enable much higher actuator velocities: large supply and return line accumulators, high flow rate 3-stage valves, a low friction cylinder, and special grips which only grip the specimen once the actuator has been accelerated up to speed. A picture and a simplified schematic of the hydraulic circuit is shown in Figure 1.

From the control perspective, the medium speed range ($\approx 1-8$ m/s) is the most critical. At low speeds, conventional feedback controllers can successfully reduce velocity errors, and at high speeds, when the inertial energy of the moving parts is large compared to the strain energy required to break the specimen, the drop in velocity during the loading period is small even without any compensation. At medium speeds, however, large tracking errors can be observed as the impact energy is large compared to the inertial energy and closed-loop control strategies are too slow to take corrective action. In these

operating conditions, a well designed iterative learning controller will yield significant tracking performance improvements.

An example of an experimental high strain rate tensile test with iterated drive signal is shown in Figure 2. Velocity, command signal, and force obtained with a steel specimen at a constant target velocity of 4 m/s are plotted. Velocity data is derived from LVDT position measurements, filtered by a non-causal, zero-phase 4th order Butterworth filter with 500 Hz cut-off frequency, which eliminates phase distortion by processing the input data in forward and reverse directions [11]. The force data is unfiltered, and some high-frequency oscillations due to load cell ringing can be observed. The actual test (i.e. the period when the specimen is loaded) and the drive signal profiling window are marked with dark and light grey shading, respectively.

For the first run, a constant drive signal is derived based on the known steady-state drive signal/velocity characteristics of the machine. The settling time is around 35 ms, and the specimen is hit after 43.2 ms. The force quickly rises to 15 kN causing the velocity to decrease down to a minimum of 2.75 m/s at 46.2 ms. The specimen breaks shortly afterwards at 46.8 ms.

The total loading period lasts 3.6 ms only, which is similar to the valve delay. It is therefore obvious that closed-loop disturbance rejection is not feasible. Instead, the drive signal for the next test is profiled with a linear iterative method. The profiling window spans from 10 ms before to 5 ms after the test; outside of this window the signal is left unchanged.

The initial part of the second run with modified command is similar to before, but the specimen is hit slightly earlier at 41 ms. This can be explained by mounting tolerances and the slight acceleration of the actuator shortly before impact. The velocity drops to 3.5 m/s at 42.6 ms before rising again to 4.3 m/s when the specimen breaks at 44.4 ms. After the break, the piston is accelerated further before hitting the bump stops – however, the velocity after the test is irrelevant. Therefore, after one iteration, the maximum velocity error has been reduced from 1.25 m/s to 0.5 m/s. With further iterations, a reduction to around 15–20% of the initial velocity error can be expected.

All iterative strategies rely on the machine and specimen characteristics remaining unchanged from one test to the next. Purely inertial loads satisfy this requirement by default, but tensile testing forces will vary due to slight differences between specimen and changes in test velocity after drive signal adjustments. Particular attention should be paid to the run-up being equal between tests, which is the period from initially accelerating the actuator until gripping the specimen. It must be assured that the initial position of the actuator is correct, the specimen aligned properly, and that the drive signal is timed accurately.

3 Machine Dynamics

In the following sections, the nonlinear system equations for the circuit in Figure 1 are presented, a linearised model is derived, and the influence of actuator forces is discussed.

3.1 Nonlinear Model

Fluid compressibility, piston leakage, and nonlinear valve orifice characteristics are taken into account. Valve spool displacement is approximated by a second-order transfer function with delay. For brevity, only positive spool displacements (open flow path between supply and actuator chamber 1) are considered in the derivation below; the relationships for negative displacements are analogous.

Piston force balance with specimen load F:

$$P_1 A_1 - P_2 A_2 = M\dot{U} + F \tag{1}$$

Cylinder flow equations with bulk modulus B and leakage coefficient c_l:

$$Q_1 = A_1 U + \frac{V_1}{B}\dot{P}_1 + (P_1 - P_2)c_l \tag{2a}$$

$$Q_2 = A_2 U - \frac{V_2}{B}\dot{P}_2 + (P_1 - P_2)c_l \tag{2b}$$

Valve orifice equations with valve constant K_v for positive spool displacements ($X > 0$):

$$Q_1 = K_v X \sqrt{P_s - P_1} \tag{3a}$$

$$Q_2 = K_v X \sqrt{P_2 - P_r} \tag{3b}$$

Spool dynamics relating demand signal W to spool displacement X:

$$X = \frac{\omega_1^2 \mathrm{e}^{-s\delta}}{s^2 + 2\zeta_1\omega_1 s + \omega_1^2}\, W \tag{4}$$

3.2 Linearisation

It is assumed that the actuator is symmetric, and the piston in centre position. The model is linearised for steady-state operation at a velocity $\mathcal{U}$ with static actuator force $\mathcal{F}$. In this case, the piston acceleration is zero and flows Q_1 and Q_2 are equal. From (1) and (3):

$$P_1 - P_2 = \frac{\mathcal{F}}{A} \quad \text{and} \quad P_1 + P_2 = P_s + P_r \tag{5}$$

therefore

$$P_1 = \frac{1}{2}\left(P_s + P_r + \frac{\mathcal{F}}{A}\right) \quad \text{and} \quad P_2 = \frac{1}{2}\left(P_s + P_r - \frac{\mathcal{F}}{A}\right) \tag{6}$$

Equating (2a) and (3a), and substituting (6) for P_1, P_2 leads to

$$A\mathcal{U} = K_v X \sqrt{\frac{P_s - P_r - \mathcal{F}/A}{2}} - \frac{\mathcal{F}}{A} c_l \tag{7}$$

Using the small perturbation technique for the valve orifice equations (3) gives

$$q_1 = \frac{\partial Q_1}{\partial X} x + \frac{\partial Q_1}{\partial P_1} p_1 = C_{x1} x - C_{p1} p_1 \tag{8a}$$

$$q_2 = \frac{\partial Q_2}{\partial X} x + \frac{\partial Q_2}{\partial P_2} p_2 = C_{x2} x + C_{p2} p_2 \tag{8b}$$

where the lower case symbols p, q, x represent small changes of pressure, flow, and spool displacement around the operating point chosen for linearisation. Considering (6), the linearisation coefficients can be expressed as

$$C_x := C_{x1} = C_{x2} = \sqrt{\frac{P_s - P_r - \mathcal{F}/A}{2}} \tag{9a}$$

$$C_p := C_{p1} = C_{p2} = \frac{K_v X}{\sqrt{2(P_s - P_r - \mathcal{F}/A)}} \tag{9b}$$

Equating (8) with (2) for small perturbations in the cylinder flow equations, solving for p_1, p_2, and substituting the results into (1) leads to the linear second order actuator model

$$u = \frac{\frac{2kAC_x}{M} x - \frac{s + k(C_p + 2c_l)}{M} f}{s^2 + k(C_p + 2c_l)s + \frac{2kA^2}{M}} \tag{10}$$

with stiffness coefficient $k = \frac{B}{V_1} = \frac{B}{V_2}$. The inputs x and f are small perturbations in spool displacement and specimen force, respectively, and the output u represents small changes around the linearisation velocity.

At zero force,

$$C_{x,\mathcal{F}=0} = K_v \sqrt{\frac{P_s - P_r}{2}} \tag{11}$$

and

$$C_{p,\mathcal{F}=0} = \frac{K_v x}{\sqrt{2(P_s - P_r)}} \overset{(7)}{=} \frac{A\mathcal{U}}{P_s - P_r} \tag{12}$$

Normalising x so that it equates to steady-state actuator velocity gives

$$x_v = \frac{C_x}{A} x \tag{13}$$

and substituting (12, 13) into (10) leads to the second order linear time-invariant actuator model

$$u = \frac{\frac{2kA^2}{M} x_v - \frac{s + k\left(\frac{A\mathcal{U}}{P_s - P_r} + 2c_l\right)}{M} f}{s^2 + k\left(\frac{A\mathcal{U}}{P_s - P_r} + 2c_l\right) s + \frac{2kA^2}{M}} \tag{14}$$

With

$$\omega_2^2 = \frac{2kA^2}{M} \quad \text{and} \quad \zeta_2 = \zeta_{21}\mathcal{U} + \zeta_{20} = \sqrt{\frac{kM}{8(P_s - P_r)^2}}\,\mathcal{U} + \sqrt{\frac{kMc_l^2}{2A^2}} \tag{15}$$

equation (14) can be written as

$$u = \frac{1}{s^2 + 2\zeta_2\omega_2 + \omega_2^2} \cdot \left(\omega_2^2 x_v - \frac{s + 2\zeta_2\omega_2}{M} f\right) \tag{16}$$

3.3 Influence of Actuator Force

With $\mathcal{F} \neq 0$, the linearisation coefficients depend on the actuator load and the valve spool displacement. A non-zero static actuator load changes the valve spool displacement – steady-state velocity relationship (13) to

$$x_v = \frac{C_x(\mathcal{F})}{A} x = \frac{K_v}{A} \sqrt{\frac{P_s - P_r - \mathcal{F}/A}{2}}\; x \tag{17}$$

This can be split into a part for $\mathcal{F} = 0$ and a force dependent factor $r(\mathcal{F})$, which may be used to modify the gain (or lookup table) relating x_v to x:

$$x_v = r(\mathcal{F}) \cdot \frac{C_{x,\mathcal{F}=0}}{A} x = \sqrt{1 - \frac{\mathcal{F}/A}{P_s - P_r}} \;\cdot\; \frac{K_v}{A} \sqrt{\frac{P_s - P_r}{2}}\; x \tag{18}$$

The block diagram in Figure 3 shows the structure of the plant model including force dependent actuator model (16) and spool dynamics (4). The block with the nonlinear curve represents a calibration table relating steady-state drive signal to actuator velocity. In the simplest case, it is a constant gain as shown in equation (13). In practice, however, it is implemented as lookup table based on measured values for various operating points. This can account for valve nonlinearities, e.g. deadband due to spool overlap.

The factor $r(\mathcal{F})$ defined in equation (18) effectively implements the standard steady-state square-root velocity/force characteristic of an hydraulic actuator. During a test, the load is not static of course, but increases rapidly when the specimen is gripped. Part of the specimen breaking energy comes from the decreasing inertial energy of the moving mass, hence only a fraction of the measured force is actually "seen" by the actuator. This is accounted for by the factor c_1, which converts the actual measured force to an

equivalent steady-state force

$$\mathcal{F} = c_1 F \tag{19}$$

A value $c_1 = 0$ implies that the specimen force has no effect on the pressures in the actuator, whereas $c_1 = 1$ implies that inertial energy does not contribute at all to the breaking energy. In reality, c_1 is larger at low velocities, and small at the higher end of the speed range.

In case of C_p, a non-zero steady state actuator force changes equation (12) to

$$C_p = \frac{A\mathcal{U} + \frac{\mathcal{F}}{A} c_l}{P_s - P_r - \frac{\mathcal{F}}{A}} \tag{20}$$

The damping coefficient ζ_2 therefore increases with load, which is implied by equation (10). This effect is not considered in the current implementation, because its influence on the tracking error and ILC performance is significantly lower than that of the force dependent changes in steady state velocity. Furthermore, estimating ζ_2 as a function of steady state velocity *and* static force would cause difficulties during experimental parameter identification. It may be feasible in a lab environment, but the process is too complicated for commercial use. As a workaround, machine operators can slightly increase the value of ζ_{20} if strong specimen are tested.

4 Linear Iterative Control

An iterative controller calculates a new drive signal from measured motion and force data, so the linear model derived in the previous section must be discretised and inverted. A regular "forward" model is also derived for model identification and parameter tuning.

4.1 Model Discretisation

The first order force gain, which is only used in the forward model (c.f. Figure 3), is discretised with the Euler approximation $s \approx (1 - z^{-1})/T_s$:

$$\frac{s + 2\zeta_2\omega_2}{M} \approx \frac{(1 - z^{-1})/T_s + 2\zeta_2\omega_2}{M} = \frac{2\zeta_2\omega_2}{M}\left(1 + \frac{1 - z^{-1}}{2\zeta_2\omega_2 T_s}\right) \tag{21}$$

Discrete valve and actuator transfer functions are calculated to have the same pole positions and steady-state gain as the continuous time model. Using the pole mapping technique guarantees that there are no non-minimum phase zeros, which would cause instability in the inverse model.

Actuator dynamics:

$$A(z) = \frac{u(z)}{x_v(z)} = \frac{b_a}{(1 - z^{-1}\mathrm{e}^{s_{a1}T_s})(1 - z^{-1}\mathrm{e}^{s_{a2}T_s})} \tag{22}$$

Valve dynamics:

$$V(z) = \frac{x(z)}{w(z)} = \frac{b_v}{(1 - z^{-1}\mathrm{e}^{s_{v1}T_s})(1 - z^{-1}\mathrm{e}^{s_{v2}T_s})} \tag{23}$$

s_{a1}, s_{a2}, s_{v1}, s_{v2} are the complex poles of the continuous-time valve and actuator models (4, 16), and T_s is the sample interval. The gains b_a and b_v are set to give unity steady-state gain. The pure dead time in the valve model is also replicated to the nearest sample interval by

$$D(z) = z^{-T_d} \tag{24}$$

where T_d is an integer chosen to yield

$$T_d \approx \frac{\delta}{T_s} \tag{25}$$

4.2 Inverse Model Structure

The block diagram in Figure 4 shows the structure of the inverse model. Inputs are demand velocity, measured velocity, and measured force. Velocity can be derived from zero-phase filtered LVDT position data. Fourth order Butterworth filters with cut-off frequencies between 300–500 Hz are suitable, depending on the amount of noise. If acceleration is also measured, then more accurate velocity information can be obtained using complementary filters [12, 13].

Demand velocity and measured velocity are passed independently through the inverse actuator model

$$A^{-1}(z) = \frac{(1 - z^{-1}\mathrm{e}^{s_{a1}T_s})(1 - z^{-1}\mathrm{e}^{s_{a2}T_s})}{b_a} \tag{26}$$

in order to ensure the correct operating point for the nonlinear gain relating steady-state velocity to drive signal voltage. The measurement is then subtracted from the demand to get the correction signal. This is multiplied by $1/r(\mathcal{F})$ to account for the reduced actuator speed under load, passed through the inverse valve model

$$V^{-1}(z) = \frac{(1 - z^{-1}\mathrm{e}^{s_{v1}T_s})(1 - z^{-1}\mathrm{e}^{s_{v2}T_s})}{b_v} \tag{27}$$

and shifted by T_d samples. Finally, all samples outside the profiling window are set to

zero, and the resulting signal is passed through a phase compensated low-pass filter

$$L = \frac{1-p}{1-pz^{-1}}, \qquad p = 0.7 \tag{28}$$

to make it less likely that the valve is driven into saturation.

4.3 System Identification

Valve spool dynamics (4) can be estimated from data sheets, and approximate actuator model parameters can be calculated with equation (15). These initial values must then be tuned by comparing the measured machine response with model output. Figure 5 shows a graphical user interface developed for identification tasks. Appropriate steps are:

1. Estimate model parameters from known physical values of the machine.
2. Perform velocity calibration, i.e. tests without specimen and constant valve drive signals to determine nonlinear gain relating x_v to x at zero load.
3. Accelerate the piston and shut the valve when the cylinder is in mid-position. The resulting oscillations can be used to adjust ω_2 and ζ_{20}.
4. Use square-wave drive signals around different mean velocities to adjust ζ_{21} and the valve model parameters. Valve dynamics dominate in these tests as the valve has lower bandwidth than the actuator.
5. Perform some tests with different strength specimen for fine-tuning and adjusting load influence factor c_1.

It is advisable to check the match between model and plant before the first iteration, whenever the target velocity is changed, or a new type of specimen is tested (e.g. to adjust c_1 depending on velocity and ζ_{20} depending on specimen strength, as discussed in Section 3.3).

5 Nonlinear Learning Control

Learning control using a nonlinear inverse model differs from the linear case in that the superposition principle does not apply. Hence, it is not possible to derive drive signal corrections, which are then added to the original drive signal. Instead, any perturbations have to be applied to the inputs of the inverse model.

5.1 Nonlinear Model Implementation

The nonlinear forward and inverse models implement the equations

$$\frac{V_1}{B}\dot{P}_1 + c_l(P_1 - P_2) - K_{v1}X\sqrt{P_s - P_1} + A_1U = 0 \tag{29a}$$

$$\frac{V_2}{B}\dot{P}_2 - c_l(P_1 - P_2) + K_{v2}X\sqrt{P_2 - P_r} - A_2U = 0 \tag{29b}$$

$$P_1A_1 - P_2A_2 - M\dot{U} - F = 0 \tag{29c}$$

which follow from (1–3) by eliminating Q_1 and Q_2. Valve spool dynamics are again represented by the linear second order model (4). The chamber volumes are dependent on cylinder displacement:

$$V_1 = V_{10} + A_1Y \quad \text{and} \quad V_2 = V_{20} - A_2Y \tag{30}$$

and so is the supply pressure:

$$P_s = P_{s0} - d_{acc}Y \tag{31}$$

V_{10}, V_{20} are the chamber volumes at piston displacement $Y = 0$, P_{s0} is the initial accumulator pressure, and d_{acc} an accumulator discharge coefficient. The linear equation (31) ignores the adiabatic nature of the accumulator discharging process, but this is an acceptable simplification in practice.

For the forward model with inputs $X(t)$ and $F(t)$, (29) is a system of ordinary differential equations (ODE) with solutions $P_1(t)$, $P_2(t)$, $Y(t)$. The inverse model has inputs $F(t)$, $Y(t)$, $U(t)$, $\dot{U}(t)$ and solutions $P_1(t)$, $P_2(t)$, $X(t)$, which makes (29) a Hessenberg index-2 system of differential algebraic equations (DAE) in the form

$$\dot{m} = f(t, m, n) \tag{32a}$$

$$0 = g(t, m) \tag{32b}$$

with $m = [P_1, P_2]^T$, $n = X$. Differentiating (29c) with respect to time, substituting $\dot{P}_1$, $\dot{P}_2$ from (29a), (29b), and rearranging gives

$$\begin{bmatrix} \dot{P}_1 \\ \dot{P}_2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ A_1 & -A_2 & 1 \end{bmatrix} \begin{bmatrix} \frac{B}{V_1}\left(-c_l(P_1 - P_2) + K_{v1}X\sqrt{P_s - P_1} - A_1U\right) \\ \frac{B}{V_2}\left(c_l(P_1 - P_2) - K_{v2}X\sqrt{P_2 - P_r} + A_2U\right) \\ -M\ddot{U} - \dot{F} \end{bmatrix} \tag{33}$$

This reduces the problem to solving a semi-explicit DAE system of index 1 with nonsin-

gular Jacobian g_m:

$$\dot{m} = f(t, m, n) \tag{34a}$$

$$0 = g(t, m, n) \tag{34b}$$

Systems of the form (34) can be solved numerically with the standard Matlab solver ode-15s as well as with common directional block diagram languages like Simulink [14]. The price for this convenience is having to differentiate the force and motion inputs, which may necessitate rather aggressive filtering of measured data, depending on noise content.

5.2 Drive Signal Calculation

Initial parameters for the model are estimated from actual machine data. These can then be tuned with system identification procedures similar to those listed in Section 4.3.

The first test run is carried out with a drive signal profile based on the steady-state voltage/velocity calibration data for the particular machine. In contrast to the linear approach, only the force data in conjunction with *demand* motion is used as input for the inverse model after the first trial. In theory, the output is a drive signal that maintains the demand motion under the influence of the measured force. Because the force profile will differ in subsequent tests due to the changed actuator velocity with the new drive signal, more than one iteration may be necessary.

The residual error should be small, but even after multiple iterations with identical specimen, the result will not be accurate due to inherent model inaccuracies. There are two options to reduce this error: linearise the model and use the algorithm from Section 4 for further iterations, or perturb the demand velocity input of the nonlinear inverse model with a fraction of the velocity error. Ultimately, the achievable tracking error for a given test is determined by physical limitations of the machine such as limited valve flow and slew rate. This is highlighted by the last equation in (33). The occurence of the force derivative $\dot{F}$ implies that very large actuator flows are required if the force changes rapidly. In general, fast valves, a stiff frame and actuation system, large accumulators, as well as large inertia of the moving parts are desirable characteristics for this type of testing machine.

6 Algorithm Performance

It is now examined whether considering the force influence improves the existing linear inverse controller, and how the nonlinear strategy compares with the linear method.

The main performance criteria are the velocity errors after up to three iterations, and the sensitivity to changes in operating conditions such as target velocity and specimen strength. Tests are simulated with a very detailed model of a high-rate machine, which has a rated maximum force of 100 kN and a rated maximum velocity of 20 m/s [10]. Using simulation results rather than experimental data gives an idea of the upper performance limits, and has the advantage that conditions are repeatable – there are no differences in specimen strength, zero mounting tolerances, and no problems with varying machine parameters such as oil temperature, friction, and sensor noise. The simulation model is similar to the nonlinear forward model described in Section 5, but additionally accounts for limited valve slew rate, spool hysteresis, adiabatic accumulator discharging, as well as load cell and frame dynamics. The specimen has a simple velocity independent stress-strain relationship.

Simulations at target speeds of 2.5 m/s and 5 m/s were performed with specimen breaking forces of 50 kN and 100 kN. Figure 6 shows the results of the initial runs with constant drive signals, which are used for iterations with the different algorithms. Crucial performance parameters are the maximum and arithmetic mean velocity errors defined by

$$e_{max} = \max_i |U_{i,actual} - U_{i,target}| \qquad \forall\, i : F_i > F_{threshold} \tag{35a}$$

$$e_{mean} = \frac{1}{n} \sum_i |U_{i,actual} - U_{i,target}| \qquad \forall\, i : F_i > F_{threshold} \tag{35b}$$

where n is the number of samples for which the measured force is higher than a certain threshold, here set to 1 kN.

Table 1 lists the errors for the initial run and three subsequent iteration steps with the following methods: linear algorithm without force compensation ($c_1 = 0$), linear algorithm with force compensation ($c_1 > 0$), and the nonlinear strategy. In the last case, the first two iterations were performed with force data alone. After two iterations with the nonlinear algorithm, two different methods were tried for the third one: using the nonlinear model with force data and perturbed velocity signal, and using the linear model with $c_1 = 0.5$.

First Iteration For the linear method, setting $c_1 = 1$ gives the best results overall. In particular, the maximum errors are reduced significantly. The mean errors only vary with the stronger specimen. As expected, the nonlinear method is particularly beneficial at low velocities and high specimen strengths. In the other cases it gives similar results to linear control with force correction.

Second Iteration The linear algorithm converge quickly in all cases. The results with the nonlinear method are not improved by a second iteration step, which may be partly due to the fact that the specimen characteristics in these simulations are independent of velocity, hence the data fed into the inverse nonlinear model is virtually the same after the initial and the second run.

Third Iteration Again, all tests with the linear method show good convergence. With $c_1 = 1$, the maximum velocity errors are now under 20% of the original values with constant drive signal, which is a good result. In the nonlinear cases, perturbing the velocity demand in addition to considering the measured force is not effective. However, performing further iterations with the linear algorithm improves matters.

Summary The presented linear and nonlinear learning control strategies reduce the maximum and mean velocity tracking errors of high strain rate tests. The linear method converges quickly, especially if the force influence is considered. This is an important fact in practice, where more than one or two iterations are often unacceptable due to the limited number of specimen.

The nonlinear algorithm performs well with one iteration only. Subsequent steps do not lead to further error reductions if the specimen characteristics are velocity independent. Using it for the first step and then iterating with the linear model may be beneficial, especially at low velocities and high forces.

7 Conclusions and Further Work

A high-rate materials test requires accurate tracking of a specified velocity profile under the influence of high impact forces. The typical test duration measured from grabbing the specimen until its destruction lies in the region of 1–10 ms. Inevitably, causal closed-loop velocity control methods fail due to signal latency, plant delays, and limited actuator bandwidth.

Learning controllers, however, utilise error information from past tests to anticipate repeating disturbances and allow the preemptive modification of the drive signal. Linear and nonlinear strategies based on inverse plant models have been presented, and it has been shown that they reduce the tracking error significantly from the first repetition onwards. The linear iteration method, which feeds the velocity errors through an inverse model, is particularly useful if multiple iterations can be performed. The nonlinear method, which calculates a new drive signal based on force measurements and the target

velocity profile, may be beneficial for the first iteration, especially if large velocity tracking errors are observed in the initial test.

The algorithms are currently being implemented in commercial machines, and measured results as well as user feedback will help to get an idea of what really works in practice, and what does not. Further work will focus on using ILC together with closed-loop control for improved velocity tracking at low speeds. This will permit control of drive signal corrections by displacement rather than elapsed time from the start of the test, which eliminates errors caused by different run-up times.

References

[1] **Cryer, B., Nawrocki, P., Lund, R.** A road simulation system for heavy duty vehicles. *SAE Paper*, 1976, 760361:13.

[2] **Dodds, C., Plummer, A.R.** Laboratory road simulation for full vehicle testing-a review. *7th Symp. on Int. Automotive Technology, SIAT*, 2001.

[3] **Plummer, A.R.** Control techniques for structural testing: a review. *Proc. IMechE, Part I: J. Systems and Control Engineering*, 2007, 221(2):139–169.

[4] **Bristow, D.A., Tharayil, M., Alleyne, A.G.** A survey of iterative learning control. *IEEE Control Systems Magazine*, June 2006.

[5] **Ahn, H.-S., Chen, Y.Q., Moore, K.L.** Iterative learning control: brief survey and categorization. *IEEE Trans. Systems, Man, and Cybernetics*, 2007, 37(6).

[6] **Norrlof, M., Gunnarsson, S.** Experimental comparison of some classical iterative learning control algorithms. *IEEE Trans. Robotics and Automation*, 2002, 18(6).

[7] **Kinosita, K., Sogo, T., Adachi, N.** Iterative learning control using adjoint systems and stable inversion *Asian J. Control*, 2002, 4(1).

[8] **Daley, S., Hatonen, J., Owens, D.** Hydraulic servo system command shaping using iterative learning control. *Control 2004, Bath, UK.*

[9] **Plummer, A.R.** Iterative acceleration control of a hydraulic actuator for vehicle crash simulation. *Symp. Power Transm. and Motion Control*, Bath, 2002, 191–202.

[10] **Plummer, A.R.** Iterative velocity control for a high speed hydraulic actuator. *Symp. Power Transm. and Motion Control*, Bath, 2001, 339–354.

[11] **Oppenheim, A.V., Schafer, R.W.** *Digital Signal Processing.* Prentice-Hall, 1989, pp. 311–312.

[12] **Stoten, D.P.** Fusion of kinetic data using composite filters. *Proc. IMechE, Part I: J. Systems and Control Engineering*, 2001, 215(5):483–497.

[13] **Plummer, A.R.** Optimal complementary filters and their application in motion measurement. *Proc. IMechE, Part I: J. Systems and Control Engineering*, 2006, 220(6):489–507.

[14] **Shampine, L.F., Reichelt, M.W., Kierzenka, J.A.** Solving Index-1 DAEs in MATLAB and Simulink. *SIAM Review* 1999, 41:538–552.

List of Figures

List of Tables

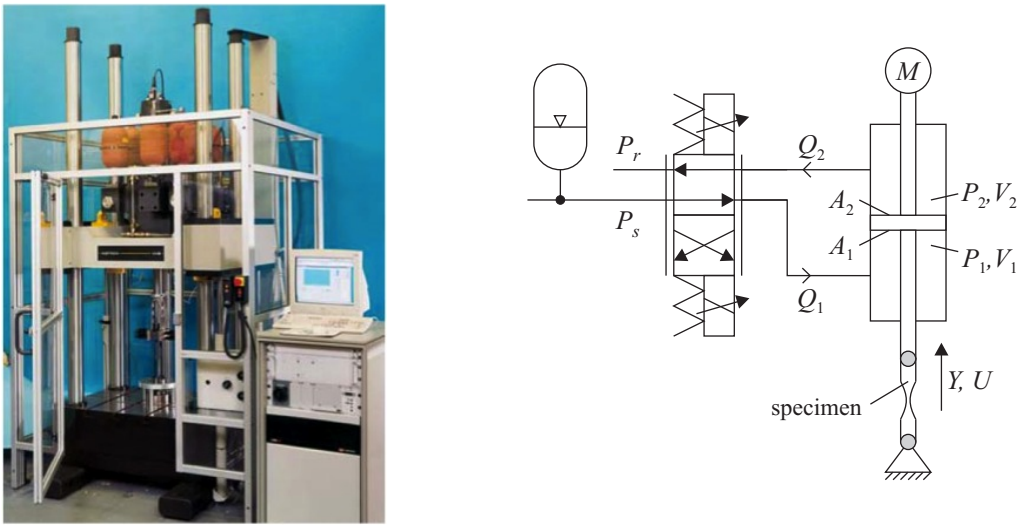
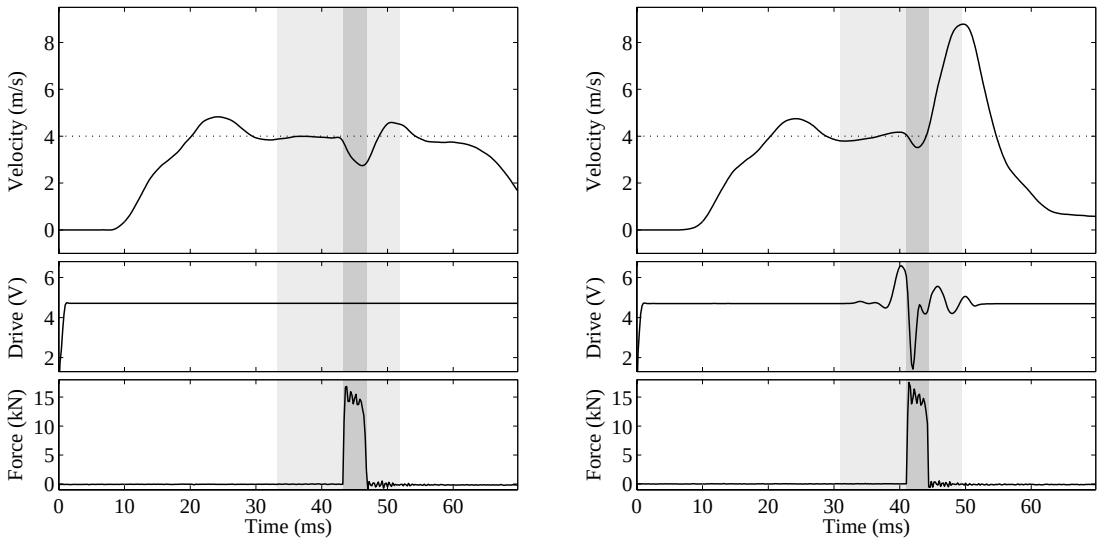


Figure 1: Photograph and simplified hydraulic circuit of a materials testing machine.



(a) Initial test with constant drive signal.

(b) Results after first iteration.

Figure 2: High strain rate tensile test of a steel specimen and a target velocity of 4 m/s.

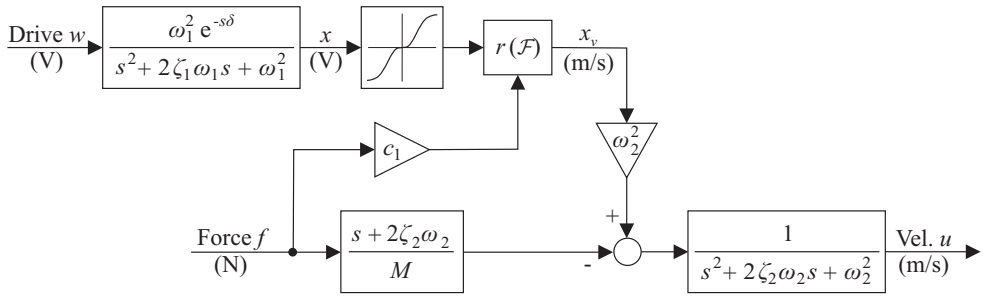


Figure 3: Block diagram of the linearised plant model.

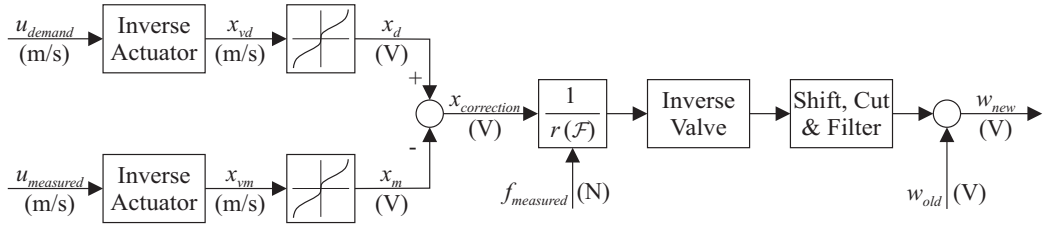


Figure 4: Block diagram of the linear inverse model.

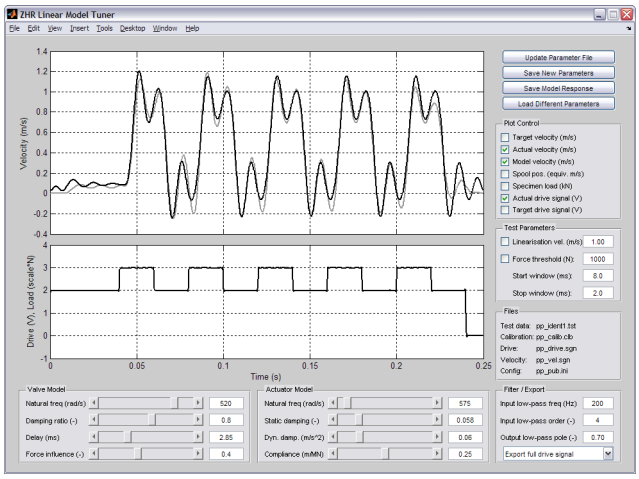
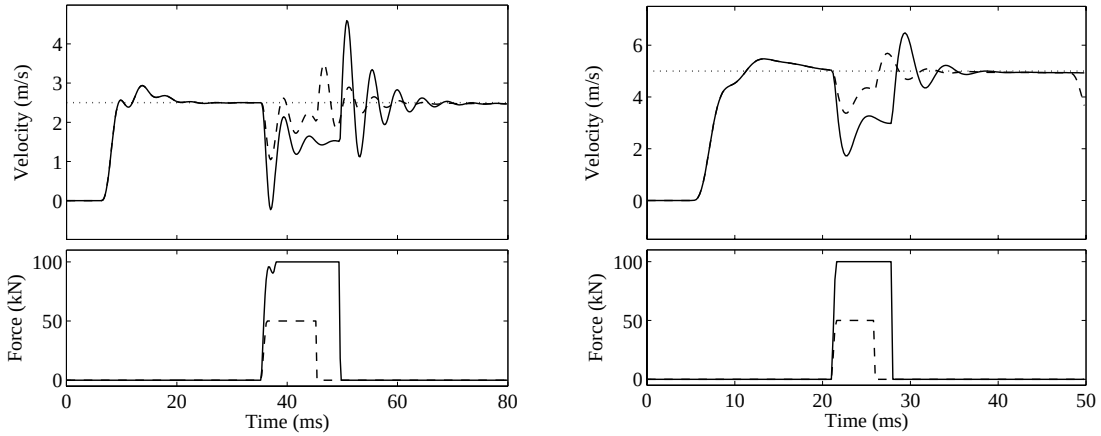


Figure 5: Graphical user interface for parameter tuning.



(a) Target speed 2.5 m/s.

(b) Target speed 5 m/s.

Figure 6: Simulated tests with specimen breaking forces of 50 kN and 100 kN at two different target speeds.

(a) Maximum errors

Target velocity		2.5 m/s		5 m/s	
Specimen Force		50 kN	100 kN	50 kN	100 kN
Initial run		1.45 (100%)	2.73 (100%)	1.63 (100%)	3.28 (100%)
Iteration 1	linear, $c_1 = 0$	0.70 (48.3%)	1.60 (58.3%)	0.68 (41.9%)	1.60 (48.7%)
	linear, $c_1 = 0.5$	0.67 (46.1%)	1.46 (53.6%)	0.64 (39.1%)	1.39 (42.5%)
	linear, $c_1 = 1$	0.63 (43.3%)	1.28 (46.9%)	0.58 (35.5%)	1.05 (32.2%)
	nonlinear	0.70 (47.9%)	0.94 (34.6%)	0.59 (36.2%)	1.10 (33.6%)
Iteration 2	linear, $c_1 = 0$	0.43 (29.3%)	1.02 (37.2%)	0.39 (23.9%)	0.97 (29.7%)
	linear, $c_1 = 0.5$	0.40 (27.7%)	0.91 (33.1%)	0.36 (22.1%)	0.82 (25.0%)
	linear, $c_1 = 1$	0.37 (25.7%)	0.75 (27.5%)	0.33 (20.5%)	0.61 (18.7%)
	nonlinear	0.87 (59.7%)	0.98 (35.8%)	0.59 (36.4%)	1.01 (30.9%)
Iteration 3	linear, $c_1 = 0$	0.30 (20.3%)	0.70 (25.6%)	0.27 (16.6%)	0.67 (20.3%)
	linear, $c_1 = 0.5$	0.29 (19.9%)	0.62 (22.8%)	0.27 (16.5%)	0.57 (17.3%)
	linear, $c_1 = 1$	0.29 (19.6%)	0.53 (19.5%)	0.27 (16.5%)	0.45 (13.7%)
	nonlin. w/ vel.	0.82 (56.4%)	0.96 (35.1%)	0.65 (40.0%)	1.05 (32.2%)
	nonlin. w/ lin.	0.84 (57.9%)	0.62 (22.5%)	0.46 (28.5%)	0.72 (22.0%)

(b) Arithmetic mean errors

Target velocity		2.5 m/s		5 m/s	
Specimen Force		50 kN	100 kN	50 kN	100 kN
Initial run		0.53 (100%)	1.10 (100%)	0.98(100%)	2.10 (100%)
Iteration 1	linear, $c_1 = 0$	0.24 (44.7%)	0.49 (45.0%)	0.27 (27.7%)	0.89 (42.2%)
	linear, $c_1 = 0.5$	0.23 (44.4%)	0.42 (38.1%)	0.26 (26.8%)	0.71 (33.7%)
	linear, $c_1 = 1$	0.23 (44.3%)	0.36 (32.5%)	0.26 (27.1%)	0.44 (21.0%)
	nonlinear	0.37 (69.5%)	0.45 (40.9%)	0.29 (29.4%)	0.49 (23.2%)
Iteration 2	linear, $c_1 = 0$	0.17 (32.5%)	0.31 (28.5%)	0.19 (19.8%)	0.44 (21.1%)
	linear, $c_1 = 0.5$	0.17 (32.3%)	0.30 (27.1%)	0.19 (19.9%)	0.32 (15.4%)
	linear, $c_1 = 1$	0.17 (32.3%)	0.32 (29.1%)	0.20 (20.1%)	0.32 (15.2%)
	nonlinear	0.29 (55.3%)	0.38 (34.5%)	0.22 (22.1%)	0.34 (16.2%)
Iteration 3	linear, $c_1 = 0$	0.13 (24.7%)	0.25 (22.5%)	0.16 (16.3%)	0.25 (12.0%)
	linear, $c_1 = 0.5$	0.13 (24.5%)	0.25 (22.5%)	0.16 (16.2%)	0.24 (11.2%)
	linear, $c_1 = 1$	0.13 (24.4%)	0.26 (23.5%)	0.16 (15.9%)	0.25 (11.9%)
	nonlin. w/ vel.	0.26 (49.0%)	0.34 (31.1%)	0.26 (26.5%)	0.46 (22.0%)
	nonlin. w/ lin.	0.22 (42.6%)	0.29 (26.8%)	0.19 (19.8%)	0.31 (14.6%)

Table 1: Absolute velocity errors in (m/s) and relative errors compared to test with constant drive signal.