Iterative Learning Control (ILC)

Bo Bernhardsson and Karl Johan Åström

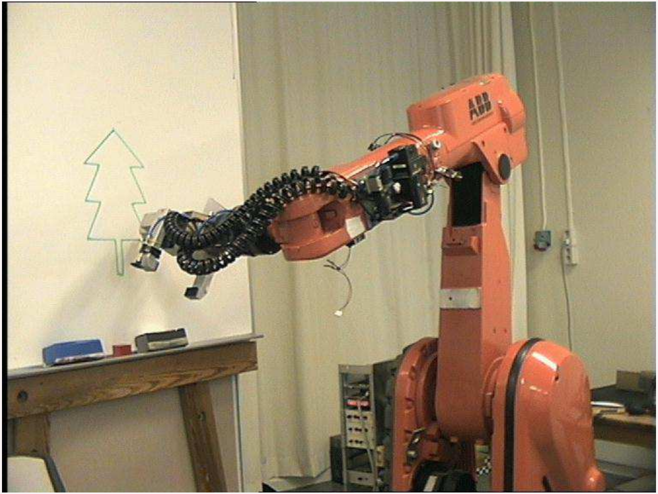
Department of Automatic Control LTH,
Lund University

ILC

- ILC - the main idea
- Time Domain ILC approaches
- Stability Analysis
- Example: The Milk Race
- Frequency Domain ILC
- Example: Marine Vibrator

Material:

ILC and Repetitive Control



Picture from the master thesis project by Domenico Scalamogna, 2000-2001

ILC and Repetitive Control

Suitable when: Repetitive task + Repetetive disturbances

Use knowledge from previous iteration to improve next iteration

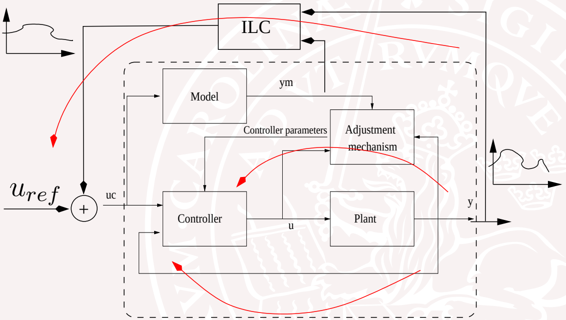
Examples

- CD-disc radial control + eccentricity
- Industrial robot laser cutting + backlash and friction
- Signal transmitter + amplifier nonlinearities

Different versions (time-interval $[0, T]$, n= iteration index):

- Repetitive control $x_{n+1}(0) = x_n(T)$
- ILC $x_n(0) = x_0$, where x_0 is the same for all n

About time scales



- ILC (batch)
- Adaptation (slow)
- 'Ordinary' feedback (fast)

NOTE: ILC works on whole signal sequences and modifies u_{ref} after 'off-line' calculations.

ILC

Consider the system

$$y(t) = Tu(t)$$
$$e(t) = r(t) - y(t)$$
$$u_{new}(t) = u(t) + L(q)e(t)$$

If T is linear and there are no disturbances then $L = T^{-1}$ gives $e_{new}(t) \equiv 0$

Is it a good strategy? Why/why not?

ILC update law

Process at iteration k:

$$y_k(t) = T_r(q)r(t) + T_u(q)u_k(t)$$
$$e_k(t) = r(t) - y_k(t)$$

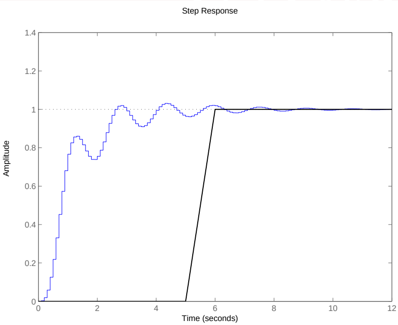
A common update law for ILC is

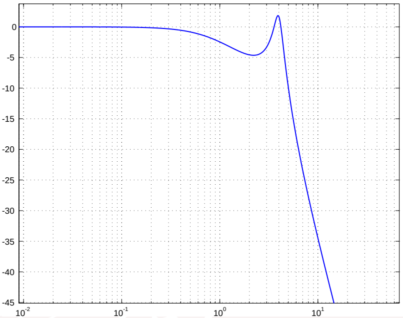
$$u_k(t) = Q(q)[u_{k-1}(t) + L(q)e_{k-1}(t)]$$

where Q and L are linear filters (need not to be causal!).

Example

$$G(s) = \frac{\omega_0^2}{(s+1)(s^2+2\zeta_0\omega_0 s+\omega_0^2)}$$, sample rate $h = 0.1$.



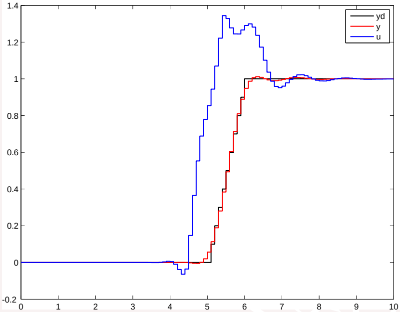


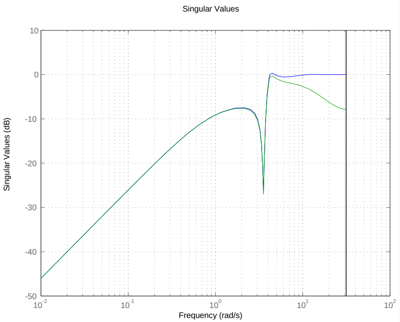
Demonstration

Example

$L(q) = kq^n$ with $k = 1$ and $n = 6$.

Q = 2nd order zero-phase low pass filter (`filtfilt`) with bandwidth $w_f = 20$





Left: After 10 iterations, Right: Gain of $(I - LT_u)$ and $Q(I - LT_u)$

```
g = 1/(s+1)*w0^2/(s^2+2*w0*z0*s+w0^2);
gd = c2d(g,h);
u1 = (t>4.5);
y1=lsim(gd,u1,t);
yr = max(0,min(1,-5:h:5))';
...
delta = 6; k=1;
wf=20; lp = wf/(s+wf); lpd = c2d(lp,h);
[b,a]=tfdata(lpd,'v');
for iter = 1:itermax-1
  u(:,iter+1)=filtfilt(b,a,u(:,iter)+...
     k*[e(delta+1:end,iter);zeros(delta,1)]);
  y(:,iter+1) = lsim(gd,u(:,iter+1),t);
  e(:,iter+1) = yr - y(:,iter+1);
end
sigma(1-k*z^delta*gd);  hold on
sigma(lpd*lpd*(1-k*z^delta*gd))
```

Error Analysis

With $G := Q(I - LT_u)$, $H := QL$ and $\tilde{r} = (I - T_r)r$ we have

$$u_k = Gu_{k-1} + H\tilde{r}$$

If G is contraction then the iteration converges and

$$e_\infty = (I - T_u(I - G)^{-1}H)\tilde{r}$$

If T_u is right invertible one can prove that

$$e_\infty = T_u(I - G)^{-1}(I - Q)T_u^{-1}\tilde{r}$$

$Q = I$ gives $e_\infty = 0$. But smaller Q gives better robustness.

Main Convergence Result

Theorem (Norrlöf, 1999)

Given a SISO LTI system and ILC algorithm

$$y_k(t) = T_r(q)r(t) + T_u(q)u_k(t)$$
$$u_k(t) = Q(q)[u_{k-1}(t) + L(q)e_{k-1}(t)]$$

convergence will be achieved if

$$\mid 1 - L(e^{i\omega t_s}) \cdot T_u(e^{i\omega t_s}) \mid < \mid Q^{-1}(e^{i\omega t_s}) \mid$$

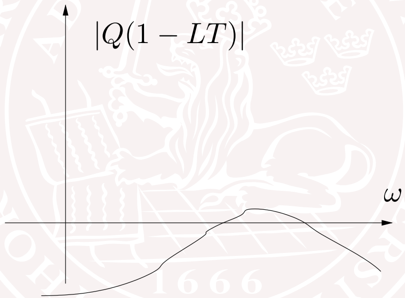
for all $\omega \in [-\pi, \pi]$ *where* t_s *is the sampling time.*

Divergence

What if

$$|1 - L(j\omega)T(j\omega)| > |Q^{-1}(j\omega)|$$

for some frequencies?



Can do some iterations, enough to reduce the worst parts. (Low frequency errors will decay rapidly, some others will grow ...)

Generalized Convergence Result

Theorem (Ardakani, Khong, Bernhardsson, 2015)

Given a SISO LTI system and ILC algorithm

$$y_k(t) = T_r(q)r(t) + T_u(q)u_k(t)$$
$$u_k(t) = Q(q)[u_{k-1}(t) + L(q)e_{k-1}(t)]$$

convergence on finite time intervals $[0, T]$ *will for any* T *be achieved if there is* $r > 0$ *so*

$$\mid 1 - L(re^{i\omega t_s}) \cdot T_u(re^{i\omega t_s}) \mid < \mid Q^{-1}(re^{i\omega t_s}) \mid$$

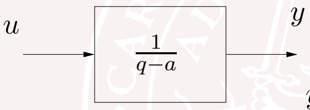
for all $\omega \in [-\pi, \pi]$ *where* t_s *is the sampling time.*

Causal and anti-causal filters

L and Q are allowed to be non-causal !

Causal representation (stable if $|a| < 1$):

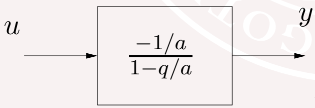
$$y(t+1) = ay(t) + u(t)$$



$$y(t) = u(t-1) - au(t-2)\ldots$$

Anti-causal representation (stable if $|a| > 1$):

$$y(t) = \frac{1}{a}y(t+1) - \frac{1}{a}u(t)$$



$$y(t) = -\frac{1}{a}u(t) - \frac{1}{a^2}u(t+1)\ldots$$

Causal and anti-causal filters

A general transfer function can be split into two parts

$$H(q) = \underbrace{H_+(q)}_{causal} + \underbrace{H_-(q)}_{anti-causal}$$

To implement $Q(q)$ one often uses a noncausal zero-phase low pass filter.

The command `filtfilt` gives a filter of the form $\frac{B(q)}{A(q)}\frac{B(q^{-1})}{A(q^{-1})}$

ILC Design

ILC design approaches

- Heuristic design
- Model Based design
- Optimization Based design

Heuristic design

Heuristic design procedure [MN]:

1. Choose the Q filter as a low pass with cut-off frequency such that the band-width of the learning algorithm is sufficient.
2. Let $L(q) = \kappa \cdot q^{\delta}$.Choose κ and δ such that the stability criterion above is fulfilled. Often it suffices to choose δ as the time delay and $\kappa : 0 < \kappa \leq 1$ to get a stable ILC system.

Why it works: With a well tuned controller $T_u \approx 1$ up to the cut-off frequency

Model based ILC design

Model based design procedure:

1. Build a model of the relations between the ILC input and the resulting correction on the output (i.e. find a model $\widehat{T}_c$ of T_c).
2. Choose a filter $H_b(q)$ such that it represents the desired convergence rate for each frequency. Normally this means a high-pass filter
3. Calculate L by $L(q) = \widehat{T}_c^{-1}(q)(1 - H_b(q))$.
4. Choose the Q filter as a low pass with cut-off frequency such that the band-width of the resulting ILC is high enough and desired robustness is achieved.

Optimization based design procedure

Impulse response matrix model

$$\mathbf{y_k} = \mathbf{T_r}\mathbf{r} + \mathbf{T_u}\mathbf{u_k}$$

where $\mathbf{T_r}$ and $\mathbf{T_u}$ are quadratic matrices of size equal to the number of time points.

Given the cost:

$$\mathbf{J}_{k+1} = \mathbf{e}_{k+1}^T \mathbf{W}_e \mathbf{e}_{k+1} + \mathbf{u}_{k+1}^T \mathbf{W}_u \mathbf{u}_{k+1} + \lambda(\mathbf{u}_{k+1} - \mathbf{u}_k)^T (\mathbf{u}_{k+1} - \mathbf{u}_k)$$

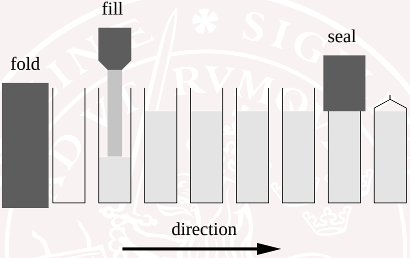
by minimizing it w.r.t. $\mathbf{u}_{k+1}$ we derive the following algorithm.

1. Build a model of the relations between the ILC input and the resulting correction on the output (i.e. find a model $\widehat{\mathbf{T}}_c$ of $\mathbf{T}_c$. The matrix $\widehat{\mathbf{T}}_c$ is simply the lower triangular Toeplitz matrix created from the impulse response of $\widehat{T}_c(q)$)
2. Choose the weight matrices as $\mathbf{W}_e = I$ and $\mathbf{W}_u = \rho I$ with $\rho > 0$, choose also $\lambda > 0$
3. $\mathbf{Q}$ and $\mathbf{L}$ are calculated according to

$$
\begin{aligned}
\mathbf{Q} &= ((\rho + \lambda) \cdot I + \widehat{\mathbf{T}}_c^T \widehat{\mathbf{T}}_c)^{-1}(\lambda \cdot I + \widehat{\mathbf{T}}_c^T \widehat{\mathbf{T}}_c) \\
\mathbf{L} &= (\lambda \cdot I + \widehat{\mathbf{T}}_c^T \widehat{\mathbf{T}}_c)^{-1} \widehat{\mathbf{T}}_c^T
\end{aligned}
$$

4. Use the ILC updating equation $\mathbf{u}_{k+1} = \mathbf{Q}(\mathbf{u}_k + \mathbf{L}\mathbf{e}_k)$ with $\mathbf{u}_0 = 0$.

Example - The Milk Race [Grundelius]



The motion is performed stepwise, step time T. Want:

- To shorten the motion time T
- To control the slosh inside ($s(t) \leq s_{max}$)
- To reduce the residual slosh after the motion

Determine best acceleration profile $u(t)$ that minimizes T.

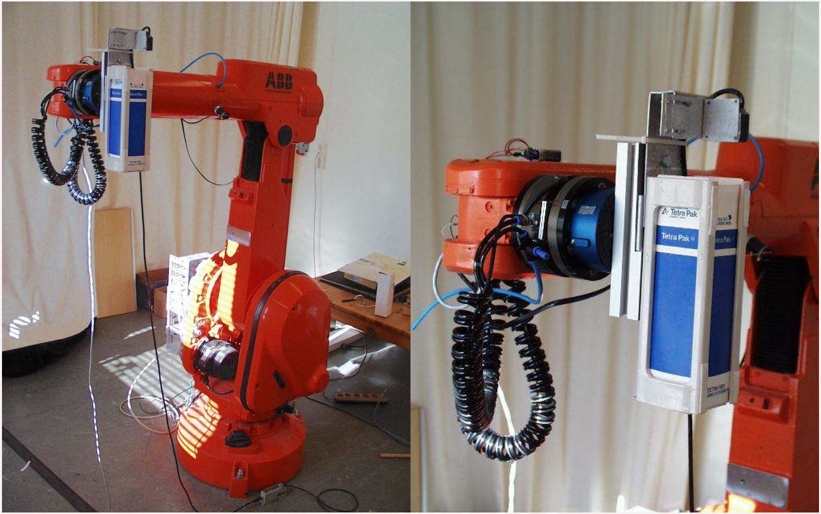
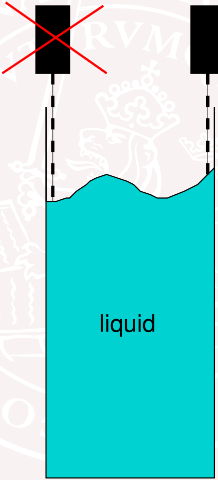


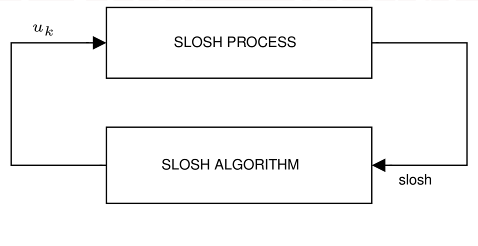
ABB
Tetra Pak
Tetra Pak

During the motion the slosh is measured by a laser sensor.

No direct feedback is applied from these measurements, but they are used in the ILC after each motion.

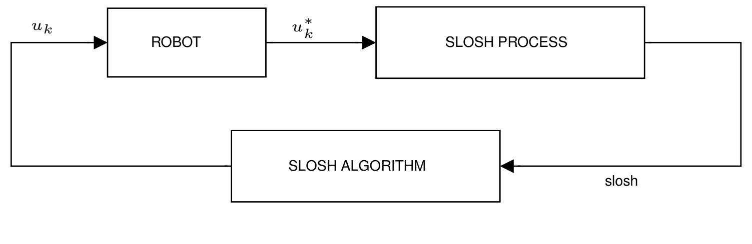


The acceleration profile was calculated iteratively by an ILC algorithm (SLOSH ILC). Optimization based ILC respecting the slosh constraints.



Assumes reliable, accurate acceleration u_k can be applied to the package all iterations (i.e. a perfect robot).

In practice we need two ILC schemes:

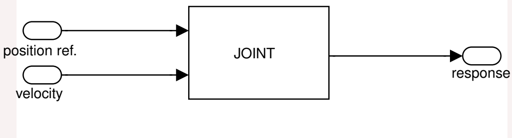


At each iteration the robot tries to track the acceleration u_k but it applies to the container a different acceleration $\hat{u}_k^*$. Solution: ILC applied to the Robot joints (ROBOT ILC) in order to improve the tracking of u_k.

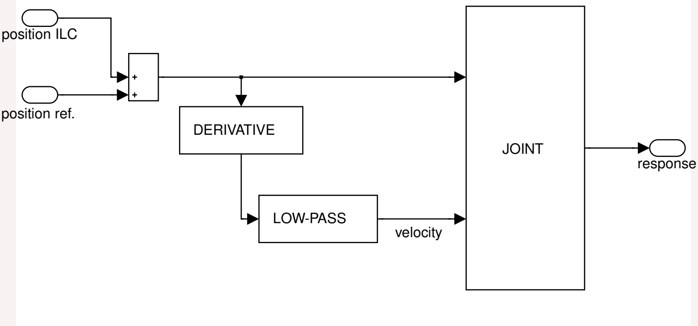
Outer and Inner Loops

1. Calculate the initial acceleration u_0 by solving the Minimum Energy Problem and calculate the slosh reference $r(t)$
2. Consider the acceleration reference $u_k(t)$
    - 2.1. Execute an iteration of ROBOT ILC and measure the acceleration performed $u_k^*(t)$
    - 2.2. If $u_k^*(t)$ doesn't approximate well $u_k(t)$ go to step 2.1.
3. Execute an iteration of SLOSH ILC by reproducing on the robot the acceleration $u_k^*(t)$ performed in the last iteration of ROBOT ILC.
4. Calculate the new acceleration $u_k(t)$ using the SLOSH ILC algorithm. Let $k \leftarrow k + 1$
5. If the slosh behavior needs to be improved go to step 2.

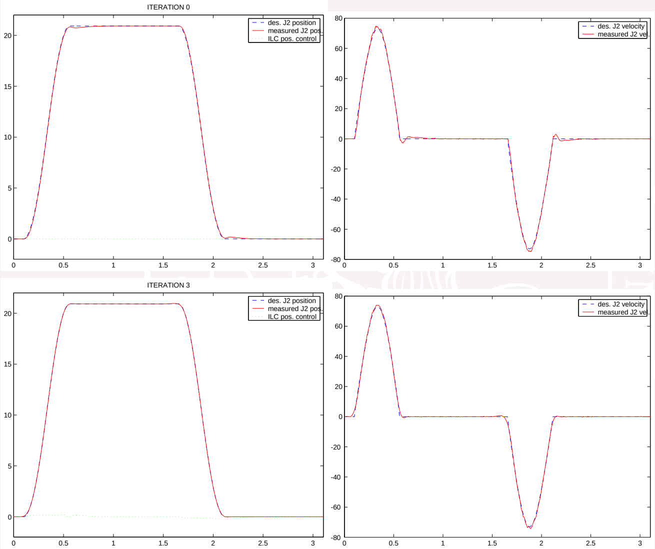
Joint Scheme:



The Heuristic ILC algorithm is applied in ROBOT ILC. It was applied to joint 2 and joint 3 of the robot. ILC Scheme applied:



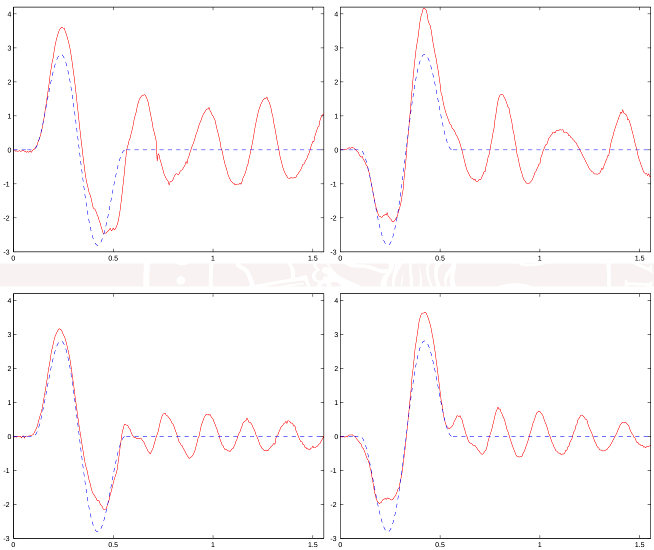
Outer loop iteration 0 on joint 2 ($T = 0.46$ s.)



$k_{ROB} = 0$

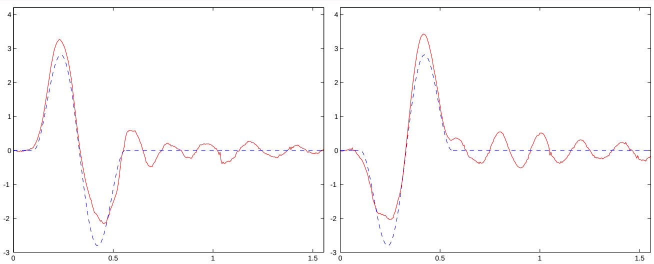
$k_{ROB} = 3$

Slosh behavior ($T = 0.46$ sec.)

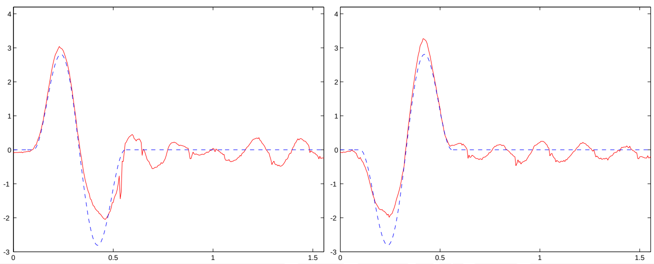


$k_{SLOSH} = 0$

$k_{SLOSH} = 1$

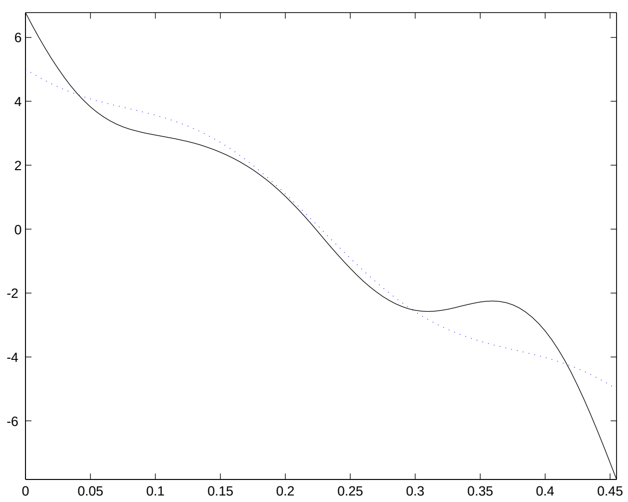


$k_{SLOSH} = 2$



$k_{SLOSH} = 3$

Acceleration profiles ($T = 0.46$ s.)



Initial acceleration (dotted) and acceleration after three iterations of SLOSH ILC (full)

Don’t push it too much...

Example: Robot tracking

Choose a wanted reference motion which should be possible to achieve without high energy input signals. The robot can’t change direction instantly. Respect the limitations.

If you are measuring angles on the motor side, you may very well improve the tracking on the motor side, but it may degrade the motion on the arm side !!

Typically enough with a limited number of iterations, say 10 or so. Errors might increase with too many iterations if models are bad.

Frequency based ILC

Similar as above, but in frequency domain

$$u_{k+1}(f) = Q_2(f)u_k(f) + Q(f)G^{-1}(f)(R(f) - Y(f))$$

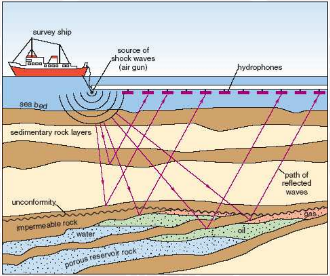
Filters chosen as

$$Q(f) = \begin{cases} 0.1 - 0.5 & \text{for frequencies we want the ILC to be active} \\ 0 & \text{otherwise} \end{cases}$$

$$Q_2(f) \equiv 1$$

Note: Can use (non-causal) filter $L = G^{-1}$ without much effort.

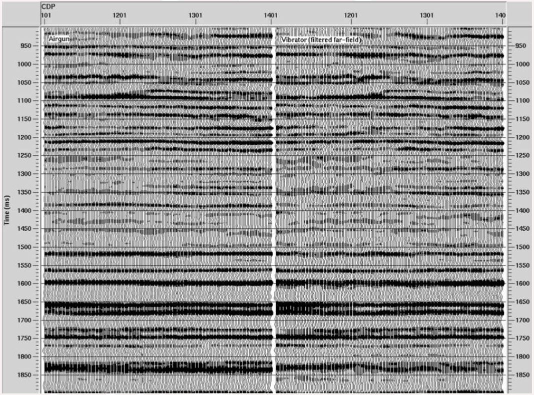
Example - Marine Vibrators



How to do seismic surveying:

- Generate a HUGE acoustic signal
- Pick up echoes using a HUGE (kilometers) sensor array
- Do some signal processing (correlation analysis)

Output from seismic survey



Higher frequencies -> Great resolution near surface structure

Lower frequency -> Better characterization of structure at depth

Acoustic Sources

Air guns have traditionally dominated the market

Higher peak pressures than most other man-made sources, except explosives

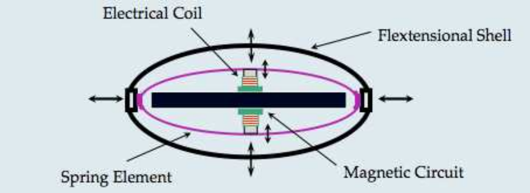
New novel constructions have the potential for reduced "acoustic footprints"

Design Challenges with Marine Vibrators

Want

- High output power
- High efficiency (for used frequencies)
- Exact acoustic signals (linearity, repeatability)

Instead of airguns: Electro-mechanical constructions with well designed useful mechanical resonances



Problems: Backlash, friction, saturation effects, ...

The Control Problem

Input (2):

- Current to coils affecting each side

Measurement sensors (2):

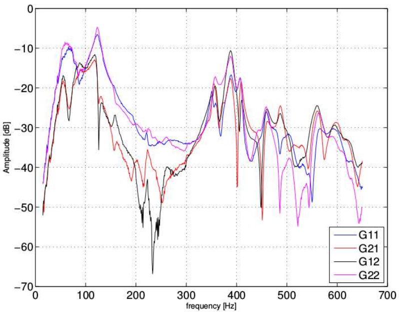
- Accelerometer(s) on shells of vibrator

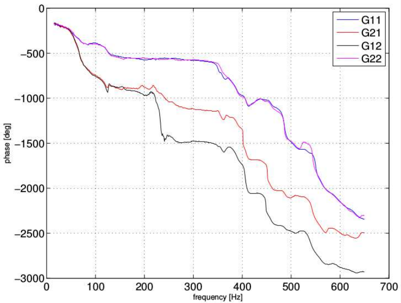
Experiments show that imperfections generate very repeatable errors

Good candidate for iterative learning control (ILC)

Very satisfactory results with ILC

System Identification 2x2 MIMO





Many resonances. Very high system order.

Decided to do ILC in the frequency domain

ILC algorithm, FFT-based

$$u_{k+1}(f) = Q_2(f)u_k(f) + Q(f)G^{-1}(f)(R(f) - Y(f))$$

Wanted reference chosen as

$$R(f) = \mathcal{F}(chirp)$$

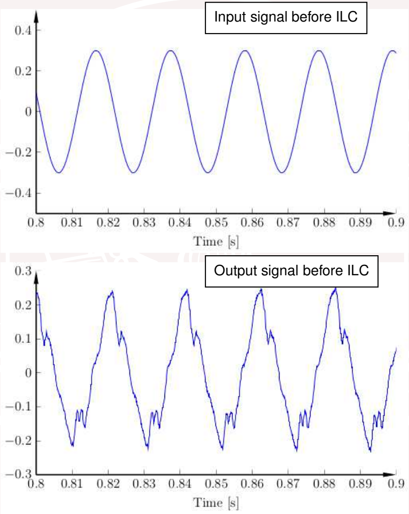
Filters chosen as

$$Q(f) = \begin{cases} 0.1 - 0.5 & \text{for frequencies we want the ILC to be active} \\ 0 & \text{otherwise} \end{cases}$$

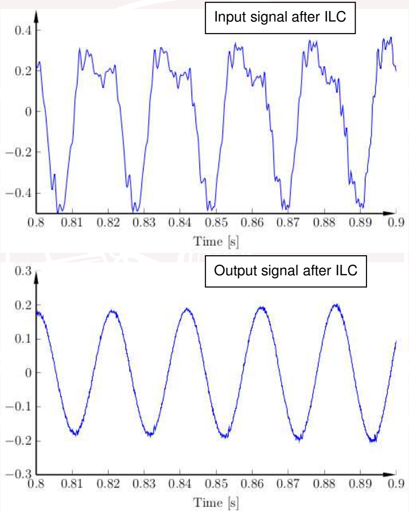
$$Q_2(f) \equiv 1$$

Note: G^{-1} matrix inverse in the 2×2 case

Before ILC


Input signal before ILC
0.4
0.2
0
−0.2
−0.4
0.8
0.81
0.82
0.83
0.84
0.85
0.86
0.87
0.88
0.89
0.9
Time [s]
Output signal before ILC
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
0.8
0.81
0.82
0.83
0.84
0.85
0.86
0.87
0.88
0.89
0.9
Time [s]

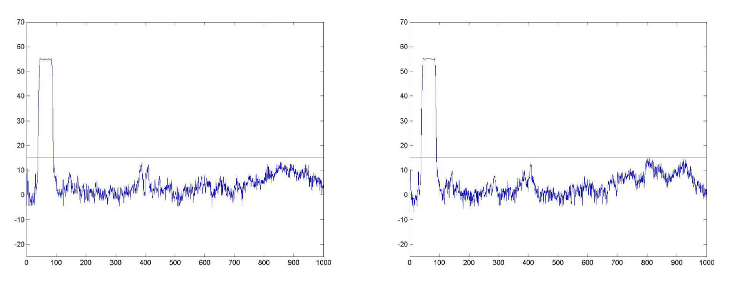
After ILC



Input signal after ILC
0.4
0.2
0
−0.2
−0.4
0.8
0.81
0.82
0.83
0.84
0.85
0.86
0.87
0.88
0.89
0.9
Time [s]
Output signal after ILC
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
0.8
0.81
0.82
0.83
0.84
0.85
0.86
0.87
0.88
0.89
0.9
Time [s]

Spectrograms after ILC - double shell sensor

Output spectra on the shells (ILC active in [30,650] Hz)

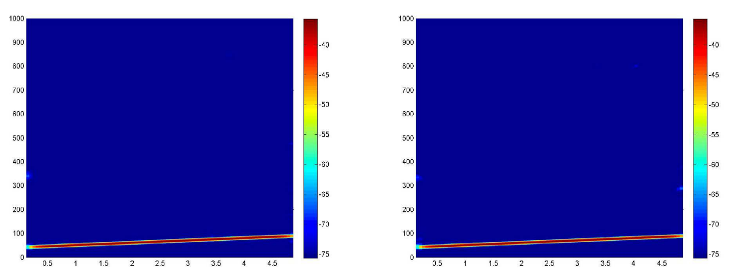


>40dB suppression

Note: Reference = constant amplitude chirp

Spectrograms after ILC - double shell sensor

Spectrum on the accelerometers on the two sides



Both sides move according to wanted reference

40dB suppression

Convergence - double shell sensors

Show
movies/ilc.avi
movies/doubleshell.avi
movies/skal131028spectograms.avi