Iterative Learning Control

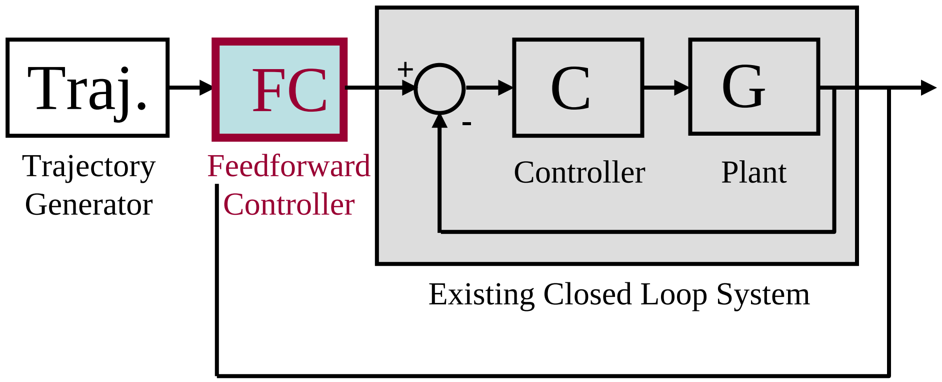
New England Manipulation Symposium

Rafael Quintanilla Escalante

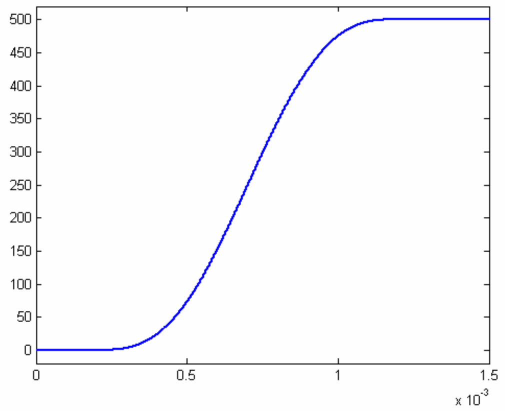
May 25, 2005

Motivation

- Assumes a closed loop controller exists
  - Hardware is complete and in production
  - Some residual oscillations in time response
- We adopt a feedforward control architecture
  - Direct model based inverse
  - Adaptation: runtime
  - Iterative learning control: offline

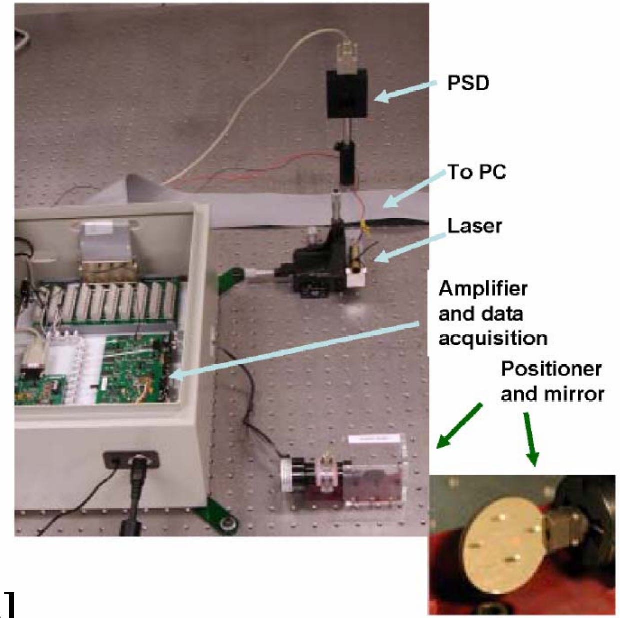


Sinusoidal Motion Trajectory

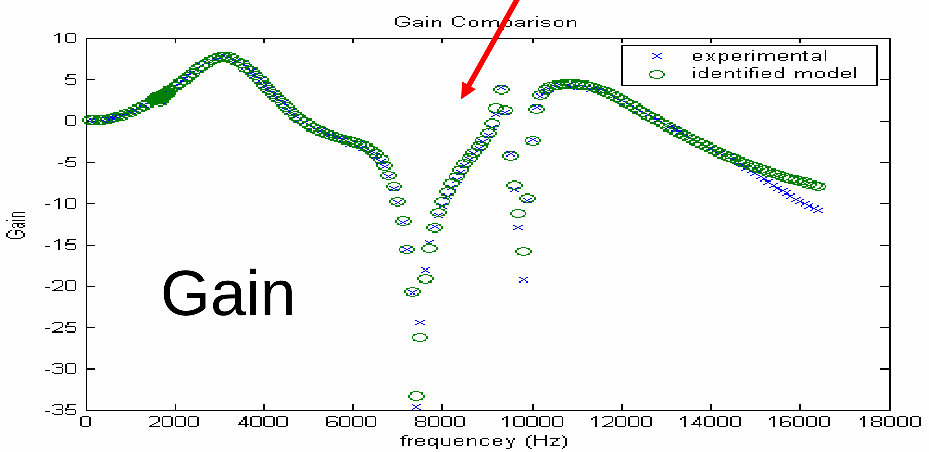


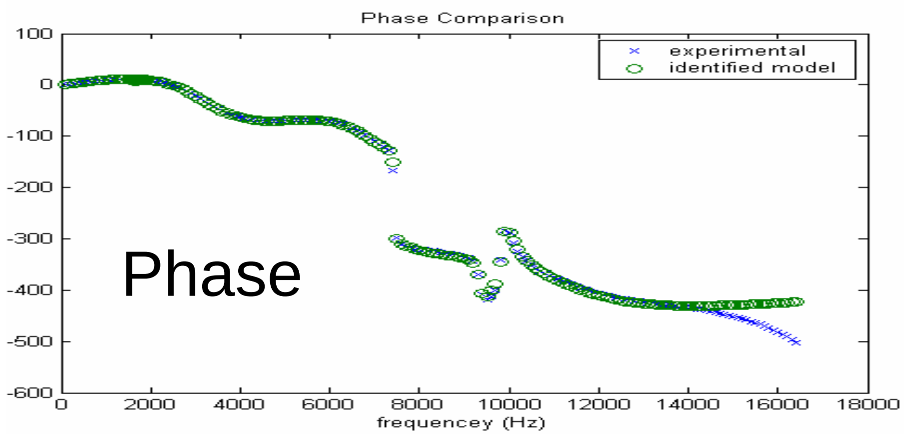
ESI Application

- High precision / high speed application
- Residue vibration at 7 kHz is the limiting factor



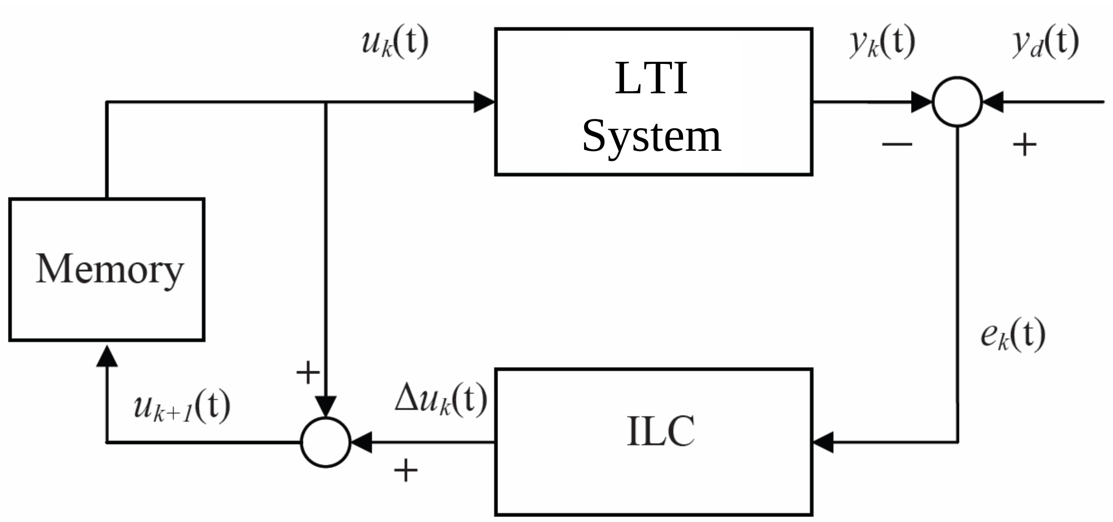
16th order identified model

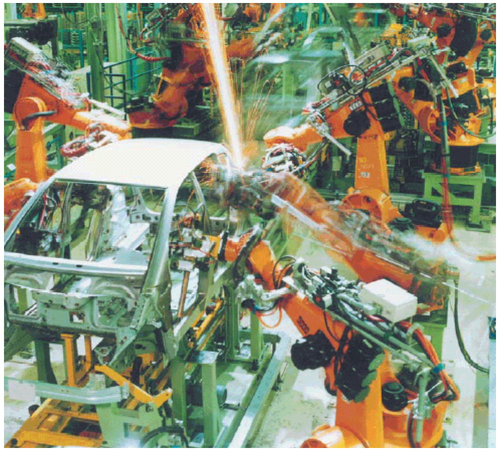




Historical Background of ILC

- Introduced 20 years ago for robot tracking control
- Improves tracking performance of systems operating in a repetitive mode
- $t \in [0 \quad T]$ and k indicates the trial





- Update at time t depends on depends on error at time t from previous iteration

Gradient Based Algorithm

- Let the output tracking error be $V = \frac{1}{2}\left\| G_0(u) - y_{des} \right\|_{L_2^p}^2$

- The change of V in each iteration is approximately (for a small change in u):

$$\Delta V = \left\langle G_0(u) - y_{des}, \Delta y \right\rangle_{L_2^p} = \int_0^T (G_0(u) - y_{des})^* \nabla_u G_0 \Delta u dt$$

where $\nabla_u G_0$ is the Frechet derivative of G_0

- If $\nabla_u G_0 \approx G$ where G is the identified plant, then by choosing

$$\Delta u = -\alpha G^*(G_0(u) - y_{des})$$

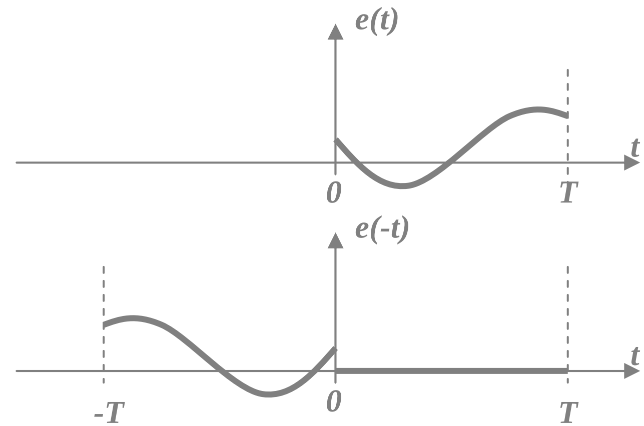
where G^* is the adjoint of G, V will be decreasing.

- Update law: $u_{k+1} = u_k - \alpha_k G^*(e_k)$ where e_k is the tracking error

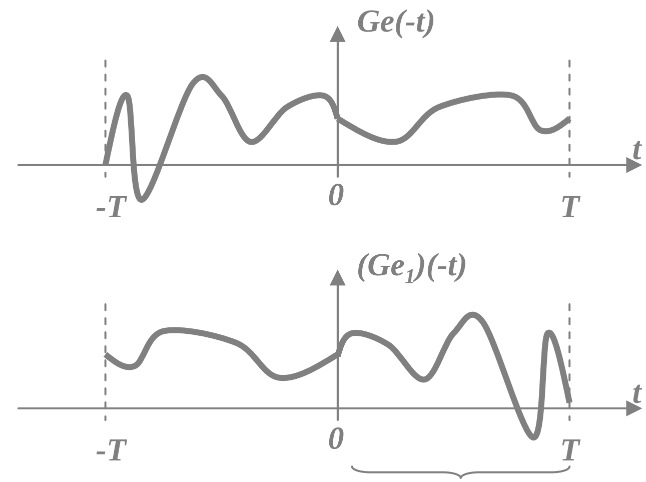
Computation of $G^* e_k$

- $G(s)$ is stable so $G^*(s) = G^T(-s)$ is antistable.

- To implement $G^* e_k$,
  - reverse e_k in time (call it e_{1k})
  - filter e_{1k} by G
  - reverse the results in time



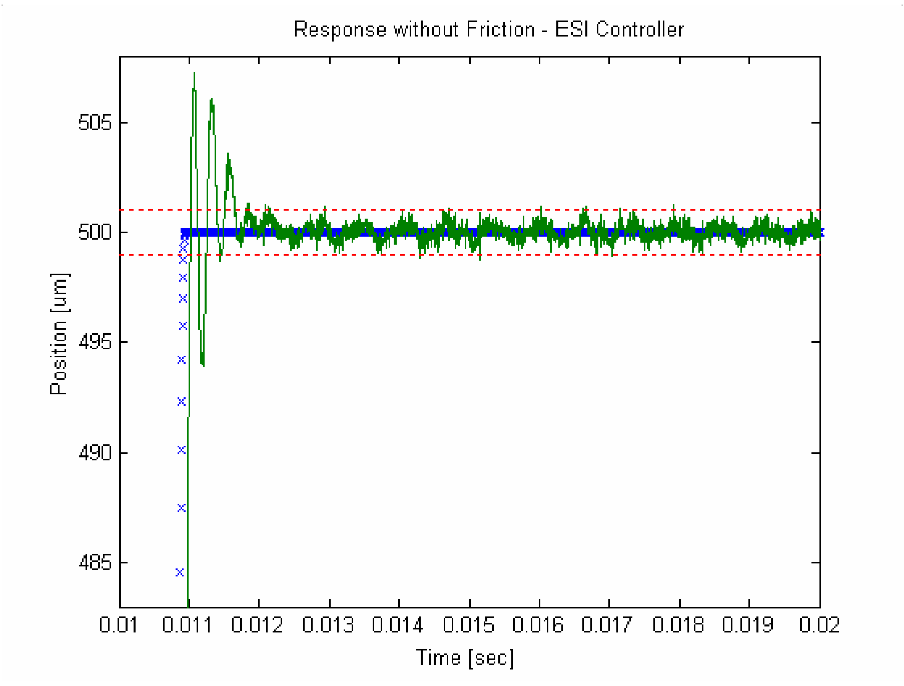
- $G^* e_k$ can be implemented without having an model (G)
  - reverse e_k in time (call it e_{2k})
  - filter e_{2k} by the actual plant (G_0)
  - reverse the results in time

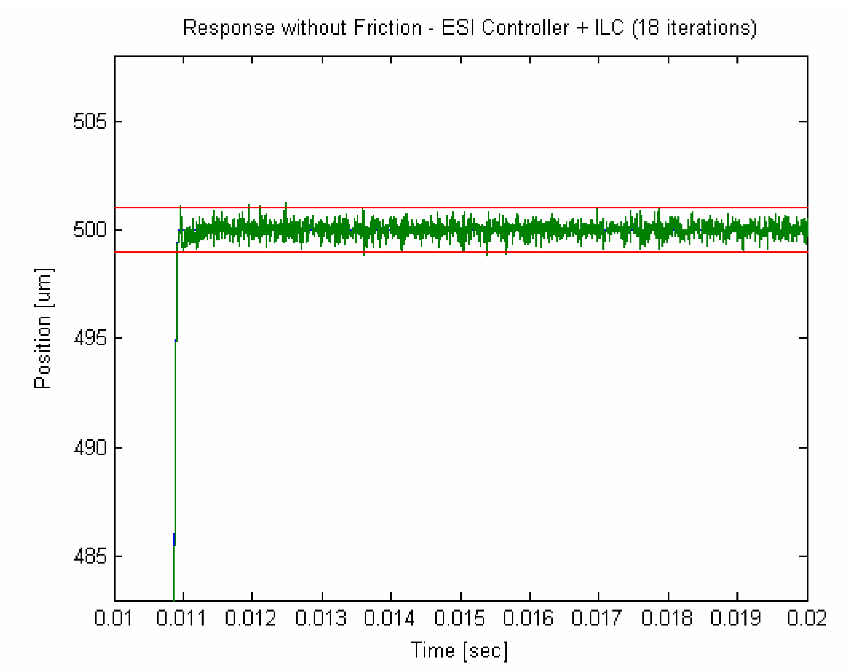


output we need for iterative update

ESI Application

- Experimental results:





500um move before learning

500um after 18 iterations

Research Focus

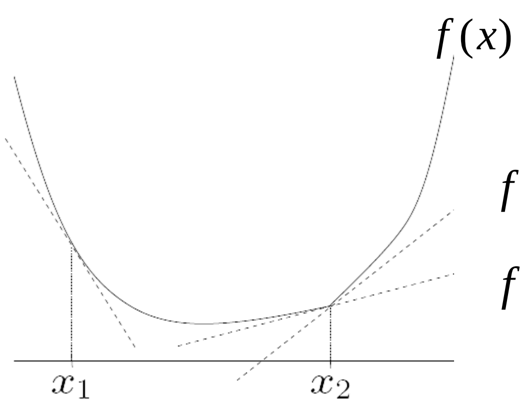
- What if $G_0(u)$ is nonsmooth ? → Gradient is not defined
- Is it possible to use subgradients to get a similar ILC update law ?
- Subgradient of a function:

  g is a subgradient of f (not necessarily convex) at x if

$$f(y) \geq f(x) + g^T(y - x) \qquad \forall y$$

$f(x_1) + g_1^T(x - x_1)$

g_1 is a subgradient at x_1



g_2, g_3 are subgradients at x_2

$f(x_2) + g_2^T(x - x_2)$

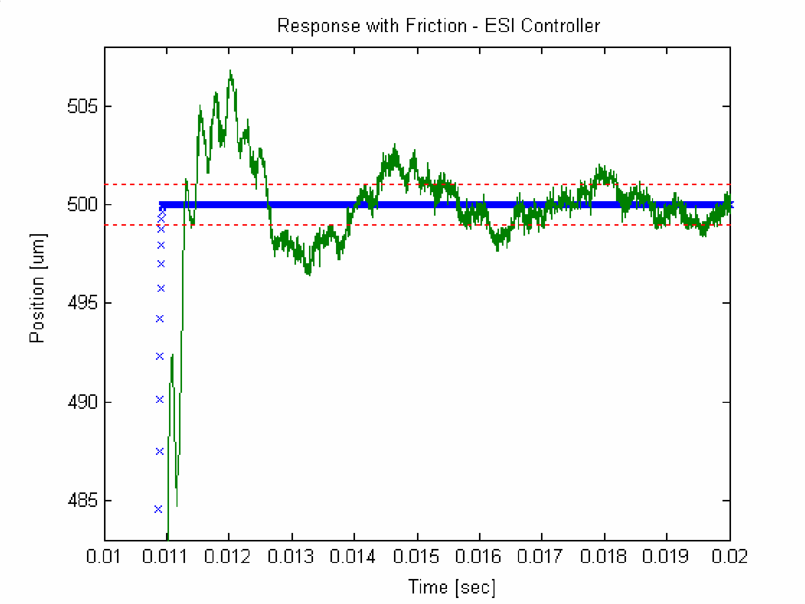
$f(x_2) + g_3^T(x - x_2)$

if f is convex, it has at least one subgradient at every point in dom f

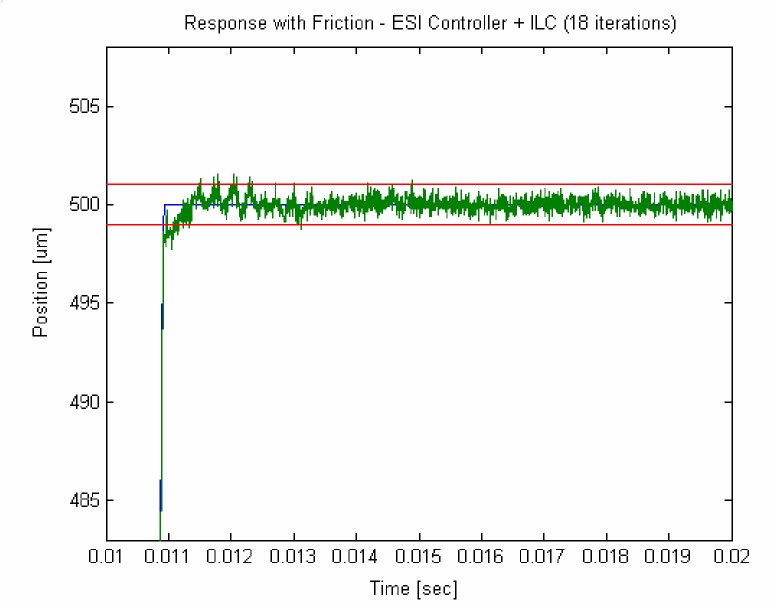
if f is convex and differentiable, $\nabla f(x)$ is a subgradient of f at x

Research Focus

- Extension of subgradient approach to functional space.
- Calculation of functional subgradient (through variational approach.
- Confidence about subgradient approach is partially based on:



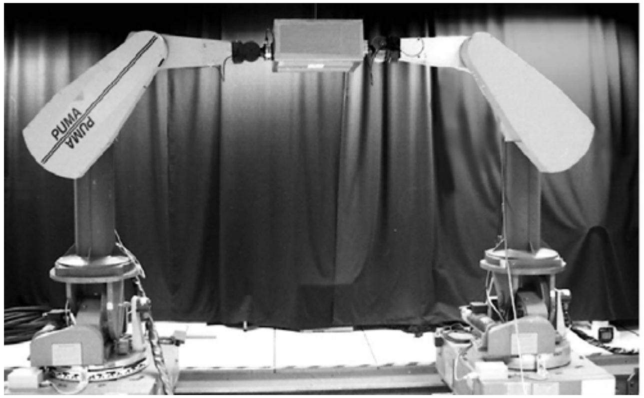
ESI system with Coulomb friction



ILC convergence ignoring Coulomb friction

Conclusions and Future Work

- Given a closed loop system (smooth), iterative refinement (learning control) improves tracking performance.
- For nonsmooth plants experimental results suggest similar results might be obtained if a subgradient approach is used.
- Future work will implement the subgradient based ILC algorithms to different test systems with nonsmooth behavior. i.e.



- *Future work will also investigate the extension of the developed ILC* algorithms to model predictive control (online).

Questions

?