Algorithms, Optimization and Simulation Results for Pulse-to-pulse Feedback in SLC, NLC/JLC, CLIC and TESLA

Linda Hendrickson

Nanobeams, Lausanne

September 2-6, 2002

SLOW FEEDBACK IN PEP2

PEP2 inherited SLC control system, but:

"Slow" feedback not anticipated for PEP2. Added later.

Functions

Stabilize IP collision positions (&angles). Timescale: ~10 seconds.

- Luminosity optimization. Dithering X,Y in turns.
  Closed position bumps at IP using 8 correctors (4X,4Y).

Stabilize orbit at sextupoles & others. Timescale: sec-min.

- BPM-based feedback. Single BPMs, closed corrector bumps.

Global Orbit control. Timescale: seconds-minutes.

- Feedback for both rings at single kick point (X,Y). Many BPMs, control kick at specific location. Not closed. Reject bad BPMs (chi-squared)
- SVD Steering now increasingly automated and frequent (minutes).

Limitations

Deflection feedback not possible due to BPM offset stability.

Intensity normalization not available due to no local networks.

Corrector power supply control slow, non-realtime, unreliable. Etc, etc.

Why pulse-pulse feedback?

- **Operational Benefits (Nan Phinney described)**
- **Luminosity Benefits of pulse-pulse feedback:**

- **Preserve small beams at IP**
  - Linac feedback preserves emittance on medium timescales (seconds-minutes)

    Faster than full steering, much better than nothing.
  - Orbit stabilization at sextupoles needed for small spots and luminosity optimization tuning.
- **Maintain collisions at the IP (beam-beam deflection feedback)**
  - Primary means of maintaining collisions for NLC and CLIC.

    Train is too short to rely on intertrain feedback only
  - Even with long bunch train, pulse-pulse feedback keeps it near the collision point => more optimal bunch-bunch feedback.
  - Optimization of bunch-bunch feedback (setpoints, gain, etc)
  - Keep intertrain actuators in range

Outline:

1) SLC Feedback Algorithms

2) IP Deflection Feedback for NLC,CLIC,TESLA (TRC work)
   1) Simulation Platform
   2) Algorithms and Optimization Methods
   3) Simulation Results

SLC Feedback Algorithms

•**LQG Feedback algorithms (Linear Quadratic Gaussian): Optimal (Modern) Control Theory. State-space formalism, Kalman filter, Predictor-corrector.**

What does this mean to us?

- Optimized: minimizes RMS of signal, given inputs of noise spectrum and plant response.
- Feedback knows about its own actuator movement, so it does not repeatedly try to fix the same error (overcorrection). Feedback responds to UNEXPECTED changes.

SLC Feedback Algorithms, cont’d

Control Design (FDESIGN): done OFFLINE in Matlab.

- **Feedback matrices loaded into realtime database.**
- **No adaptive control (except cascade transport calculations in linac)**
- **Original SLC FDESIGN system was in MatrixX (similar to Simulink).**

  Converted to Matlab m-files to reduce numerical problems and improve maintenance for large machine with diverse loops.
- **Using CONTROL, SIGNAL PROCESSING TOOLBOXES.**

Control Design (FDESIGN) Inputs:

- **Plant noise model:**

  Low-pass, white, harmonic oscillator, bandpass, etc.

  (harmonic oscillator dangerous in simulation)

- **Actuator Response Model:**

  Time delay (N pulses or feedback iterations.)

or Exponential Response (dangerous!)

- **Sensor Noise**
- **Plant Transport Matrices:**

  States => Measurements

  Actuators => States

SLC Feedback Algorithms, cont'd

Typical SLC Steering Feedback Implementation:

- **Plant noise model:**

  Low-pass, white (PINK = low + white)

  Noise model geared for operational characteristics (step response) in addition to measured noise spectrum

  => 6-pulse exponential response.

- **Actuator Response Model:**

  2-pulse Time delay. (But actuators were slower!)

- **Sensor Noise (modeled as negligible in SLC).**

- **Plant Model:**

  Measurements were BPM readings (X and Y beam positions).

  States were positions and angles at specific fit location.

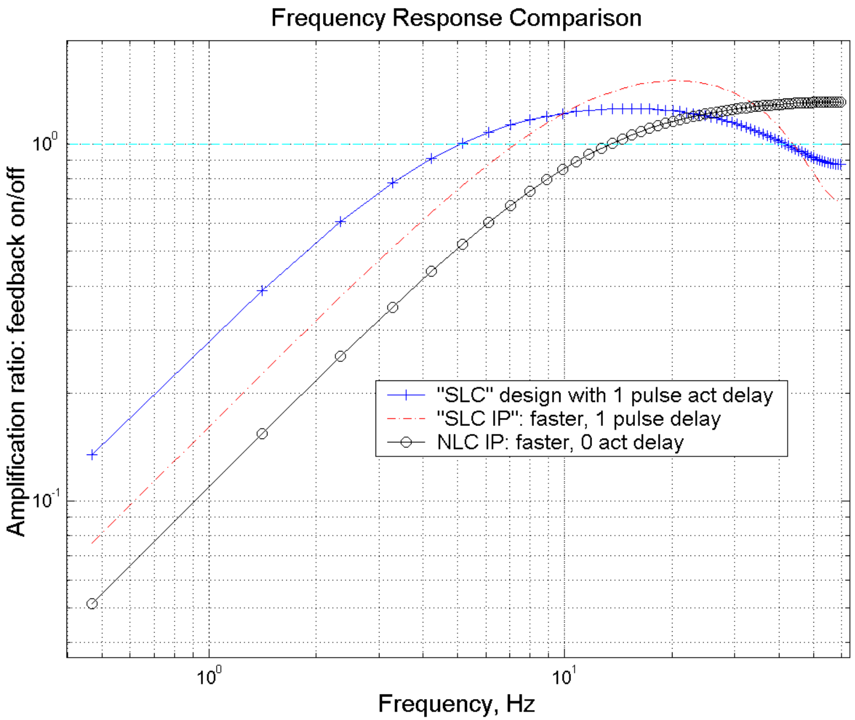
  Actuators were dipole corrector field strengths.

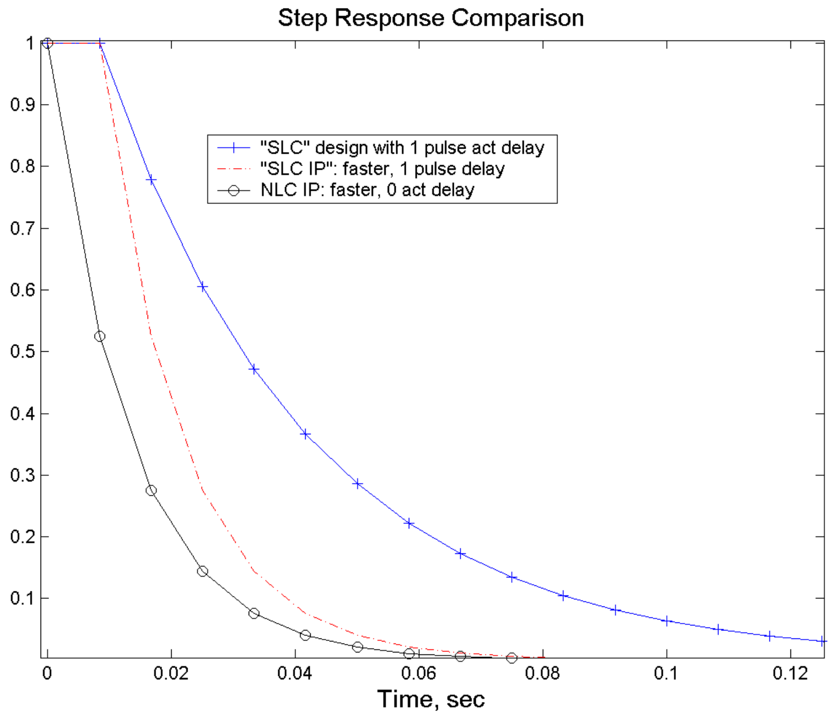
  States => Measurements (from accelerator model)

  Actuators => States (from model, or calibrated with beam)

Feedback timescales: NLC vs SLC feedback design response:

(It helps to assume a faster control system: low-latency BPMs, fast IP kickers/correctors, fast networking)





Simulation Platform for Feedback Systems

MATLAB

- **MATLIAR/DIMAD (MEX) (lattice, realistic ground motion of 2 machines pointing at each other, imperfections, corrector settings => slices => rays)**
- **GUINEA PIG (rays => deflection and luminosity)**
- **FEEDBACK calculations in matlab m-files (deflection and feedback model => corrector settings for LIAR)**

IP Deflection Feedback Simulations

Simulations for NLC (120 Hz), CLIC (200 Hz), TESLA (5 Hz)

Setup:

Start with 100 machines (from Tenenbaum, Seryi, Woodley), misalign and steer to get nominal luminosity. Choose 3 machines for initial simulations.

Feedback Design Considerations:

- **Modeling of Deflection Curve:**
  - ? Linear feedback with fit to linear portion of curve near IP (SLC)
  - ? Linear feedback using a "compromise" slope
  - ? Non-linear fit to measured beam-beam deflection curve
- **Setpoint for beam-beam deflection:**

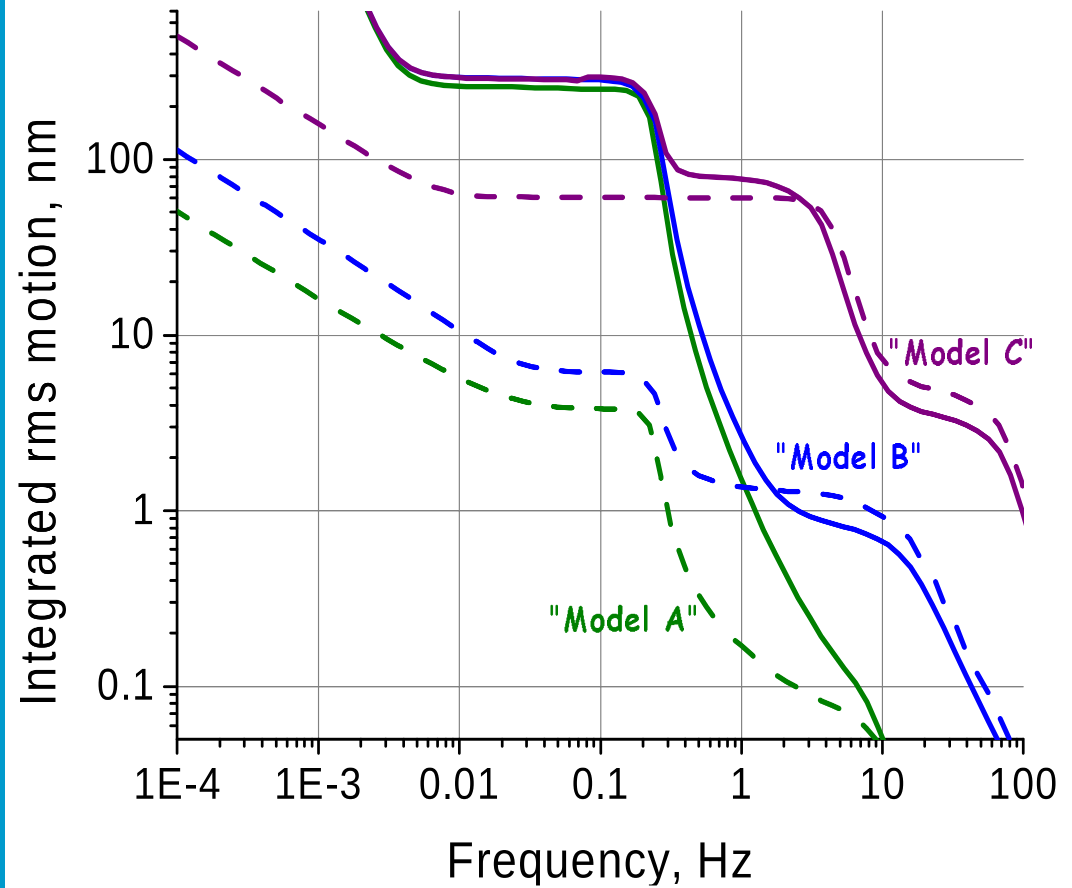
  Should be zero for head-on collisions, but:

  with asymmetric non-gaussian beams, want to maximize luminosity.
- **Time response model for feedback: how aggressive should it be?**

Do we want to optimize these items on the fly?

Ground motion models (Andrei Seryi)

- Based on data, build modeling P(ω,k) spectrum of ground motion which includes:
  - Elastic waves
  - Slow ATL motion
  - Systematic motion
  - Technical noises at specific locations, e.g. FD)



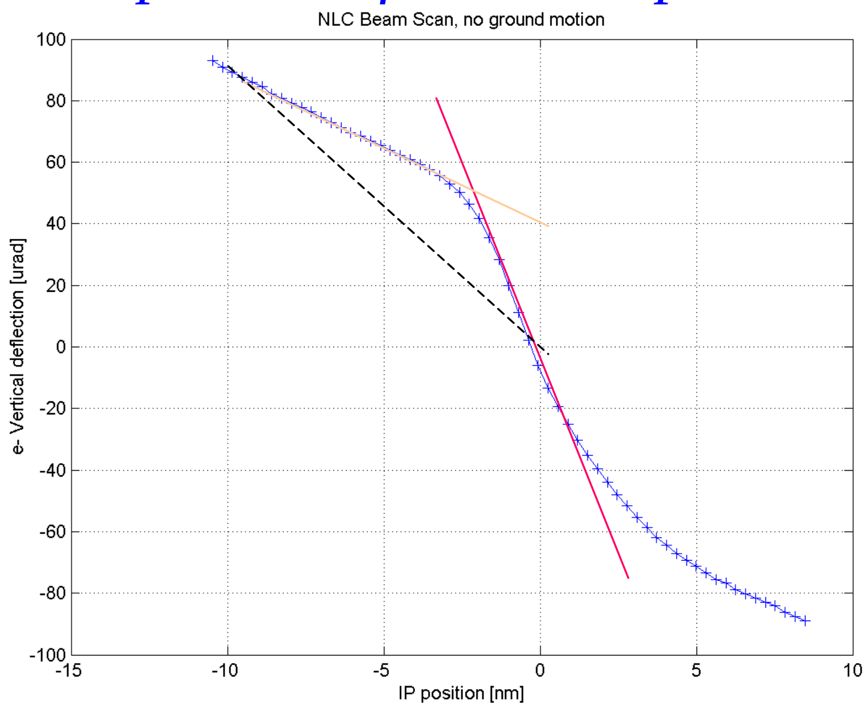
Example of integrated spectra of absolute (solid lines) and relative motion for 50m separation obtained from the models

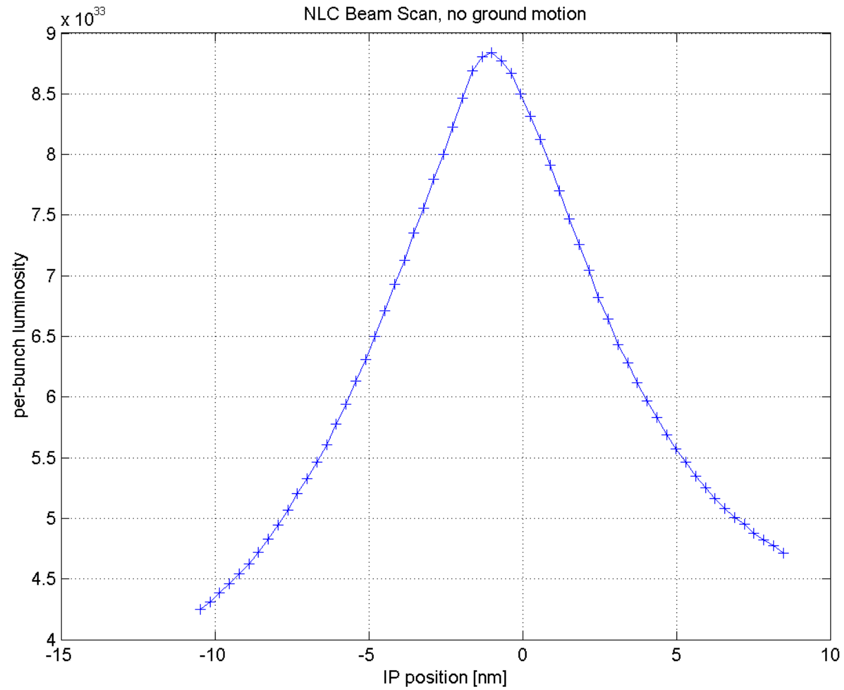
IP Deflection Feedback Simulations

- Scan correctors at IP. (Assume we can take a perfect deflection scan measurement without ground motion!)
- Piecewise linear fit of deflection vs corrector settings.
- Asymmetric gaussian fit of corrector vs luminosity to find position for max luminosity. Piecewise linear fit to find deflection setpoint corresponding to corrector setting. (Not zero!)

Does the deflection curve change with ground motion?

Does optimal deflection setpoint change with ground motion?





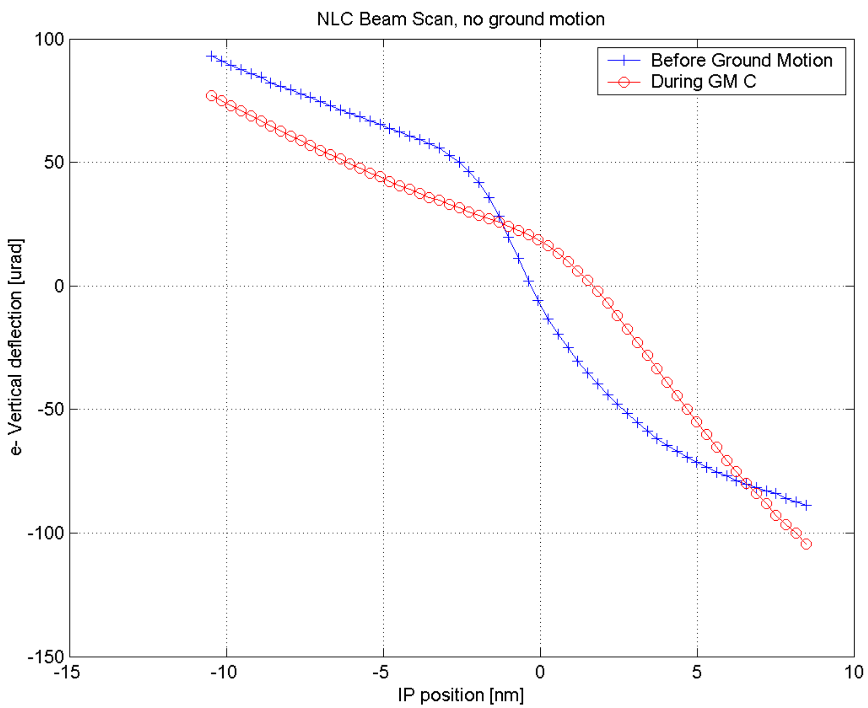
IP Deflection Feedback Simulations

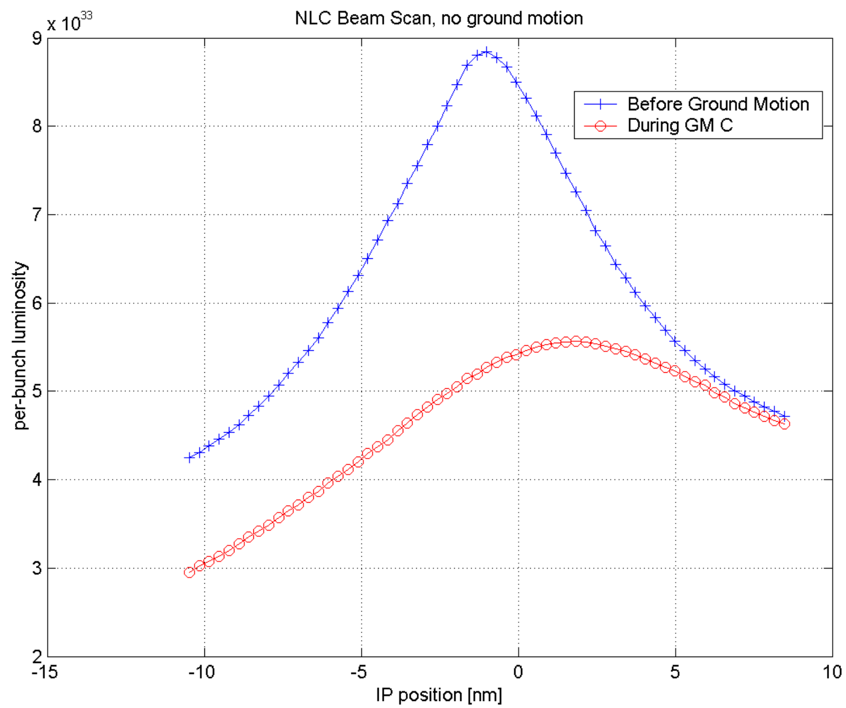
Next Linear Collider

Does the deflection curve change with ground motion? YES, with large ground motion

Does optimal deflection setpoint change with ground motion? YES, with large ground motion

Ground motion C feedback simulations: Before ground motion, and after moving the ground with GM model "C"





Feedback Design: Noise Response

How aggressive should the feedback be?

- **If too aggressive, amplifies the white noise.**
- **If too slow, lose collisions.**

Should we optimize noise response on-the-fly? What if plant noise spectrum changes?

Use LQG feedback design, and just let it find the optimal controller?

Haven't done this, yet. Why not? (besides not having enough time)

- **LQG will want to minimize RMS of IP beam position as a function of time. But: *real goal is: maximize luminosity*.**

  Not necessarily the same thing, depends on ground motion and deflection and luminosity curves.

- ***Might* want a simple way to optimize feedback response with changing noise spectrum. SLC "FDESIGN" matrices were designed in advance. Needs work to get a nice adaptive feedback.**

Feedback Design: Noise Response

Quick-and-dirty solution? For now, convert our SLC "pink noise" matrices to an *equivalent* exponential form in which the time response can be optimized by adjusting one parameter: WEIGHT of previous state estimate compared to new "measured" data.

Sacrifices the power of optimal control theory, but we weren't using it for SLC anyway. Bonus: DC offset in SLC feedback goes away with exponential!

New feedback algorithm:

state_vec = expected_change + weight * (state_vec - raw_state_vec) + raw_state_vec;

delta_act = - nmpt * state_vec;

act_vec = act_vec + delta_act;

expected_change = bmpt * delta_act;

Where: weight is the exponential gain: weight=exp(-1/npulses)

state_vec = estimated state vector (in corrector units)

raw_state_vec = measured X,Y deflections, converted to corrector units

act_vec = actuator vector (X,Y correctors)

nmpt,bmpt are transport matrices (ones in our case)

Optimization testing: Sensitivity of Luminosity to SLOPE (linear model), SETPOINT, and WEIGHT (gain)

For NLC, optimize 3 parameters separately for SMALL, MEDIUM, LARGE ground motion (GM A, B, C)

Method: SCAN over values of each parameter and maximize luminosity.

Timescale for a single ground model:

128 pulses each step, 9 steps, 3 parameters

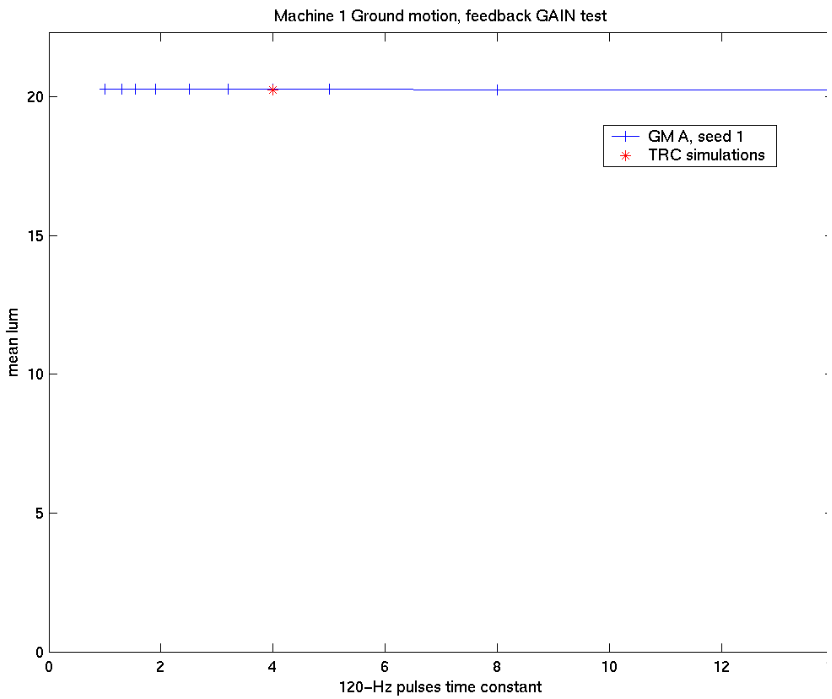
=> ~30 seconds machine time

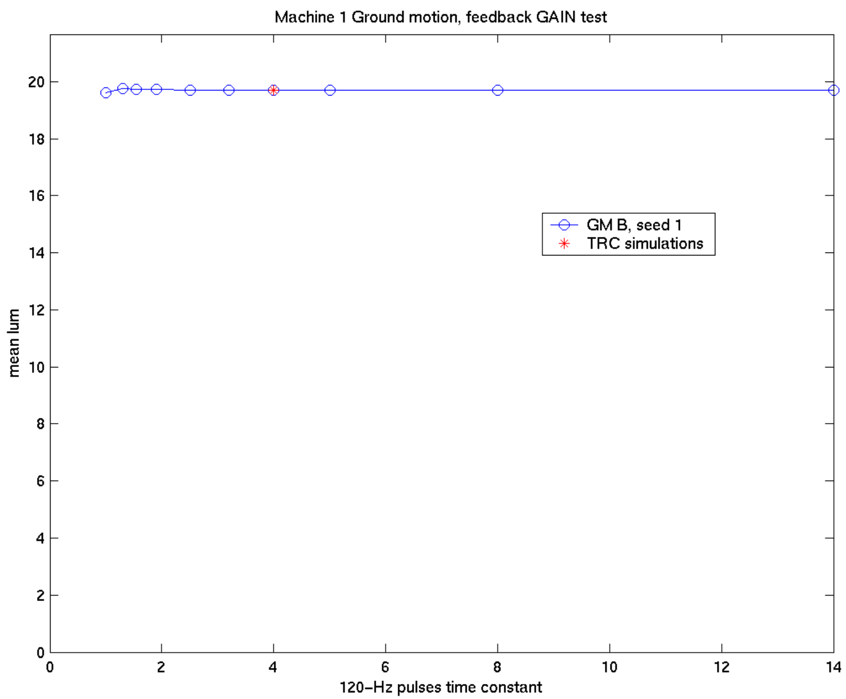
=> ~7 days simulation time, using SLAC Solaris machine

Gain Sensitivity for NLC, GM A,B: Boring!

Luminosity vs feedback time constant (pulses) for SMALL ground motion (GM A): INSENSITIVE!

Luminosity vs feedback time constant (pulses) for MEDIUM ground motion (GM B): INSENSITIVE!



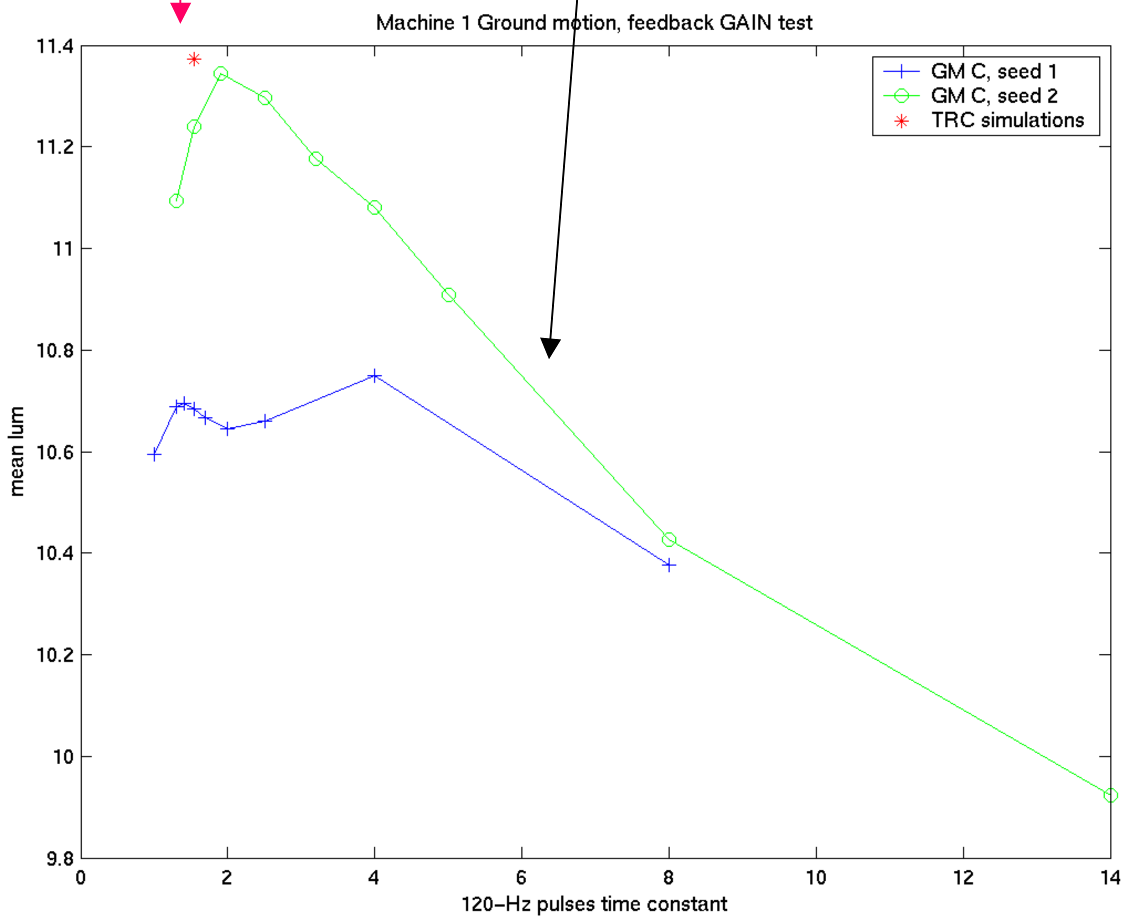


Gain Sensitivity for GM C

TRC: 1.55 pulses

Typical SLC: 6 pulses

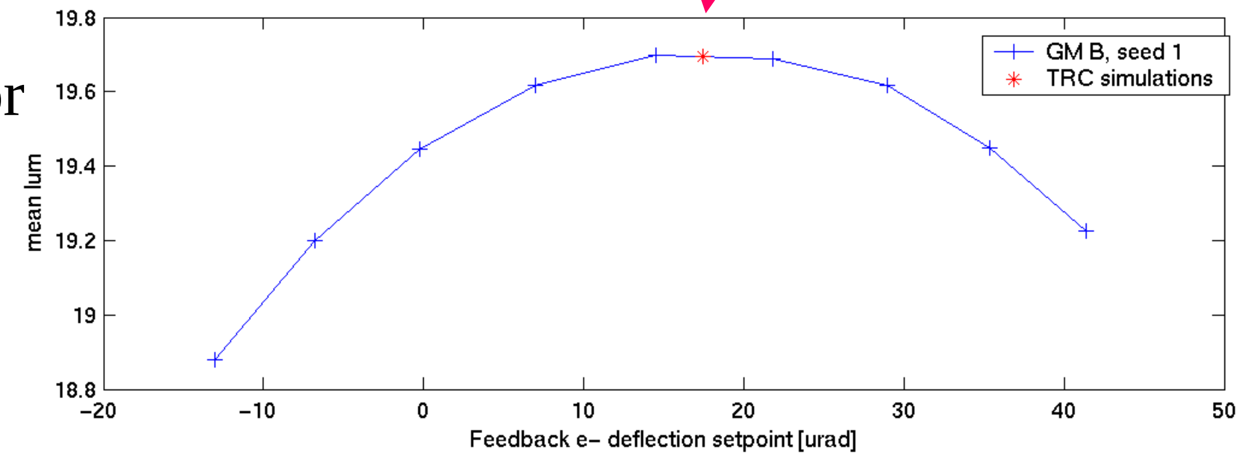
Luminosity vs feedback time constant (pulses) for LARGE ground motion (GM C): Worth Optimizing. But slightly different answers for different random seeds for ground motion model



Setpoint and Slope Sensitivity for GM B

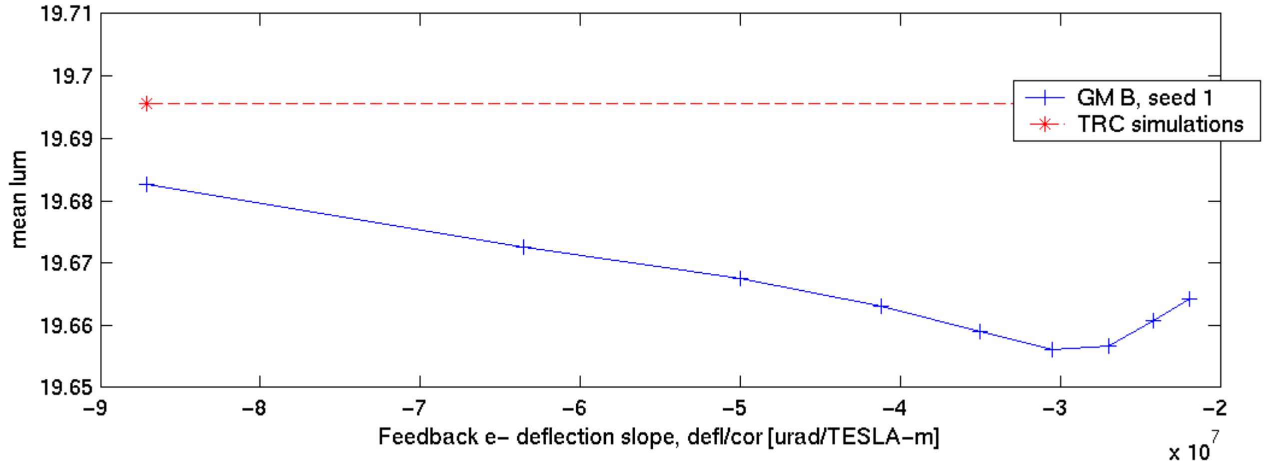
Optimal Setpoint from deflection scan

Luminosity vs deflection setpoint for MEDIUM ground motion (GM B)



Luminosity vs linear deflection slope for GM B. Note: TRC simulations used piecewise linear.

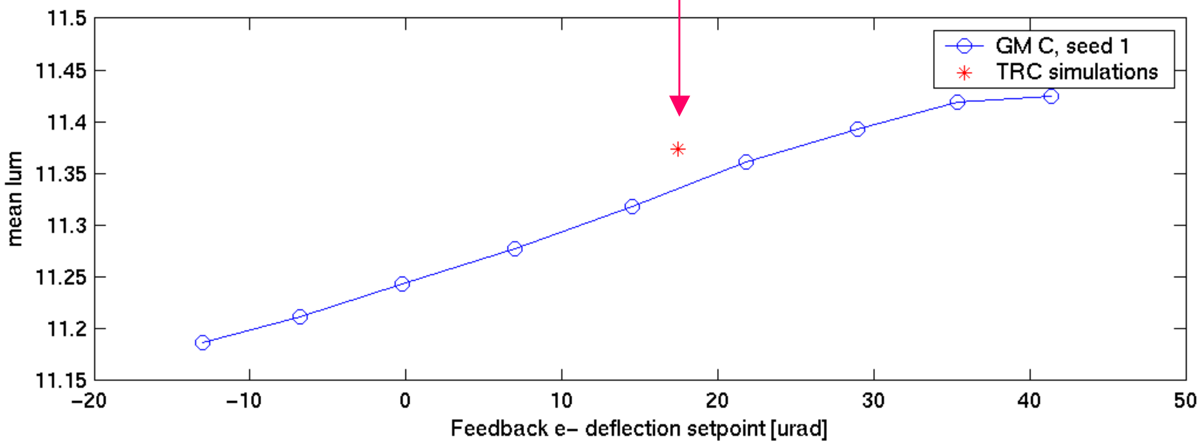
(But note the scale on this plot)



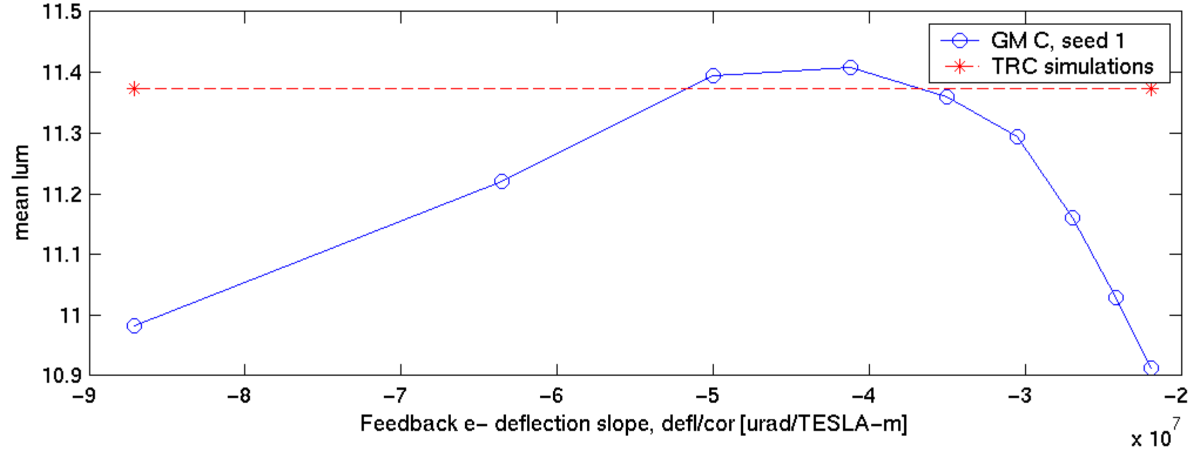
Setpoint and Slope Sensitivity for GM C

Optimal Setpoint from deflection scan

Luminosity vs deflection setpoint for LARGE ground motion (GM C)



Luminosity vs linear deflection slope for GM C. Note: TRC simulations used piecewise linear

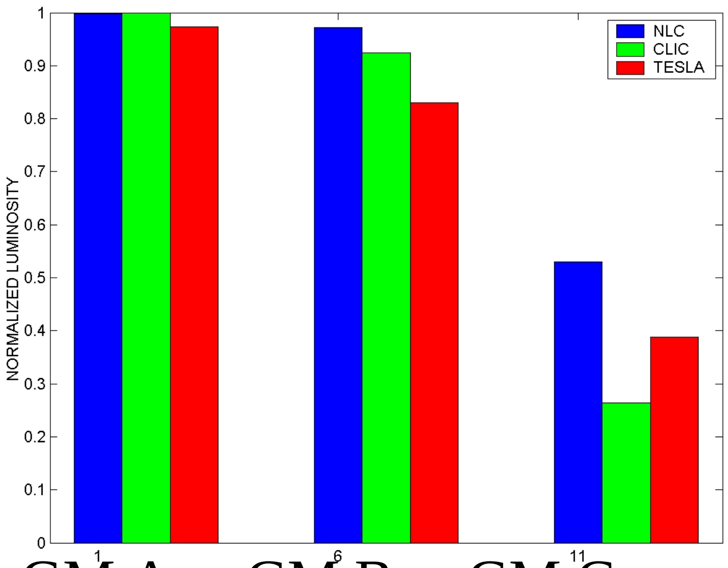


IP FEEDBACK SIMULATIONS for NLC,CLIC,TESLA

Imperfect machines, initial nominal luminosity (for TRC, with Seryi)

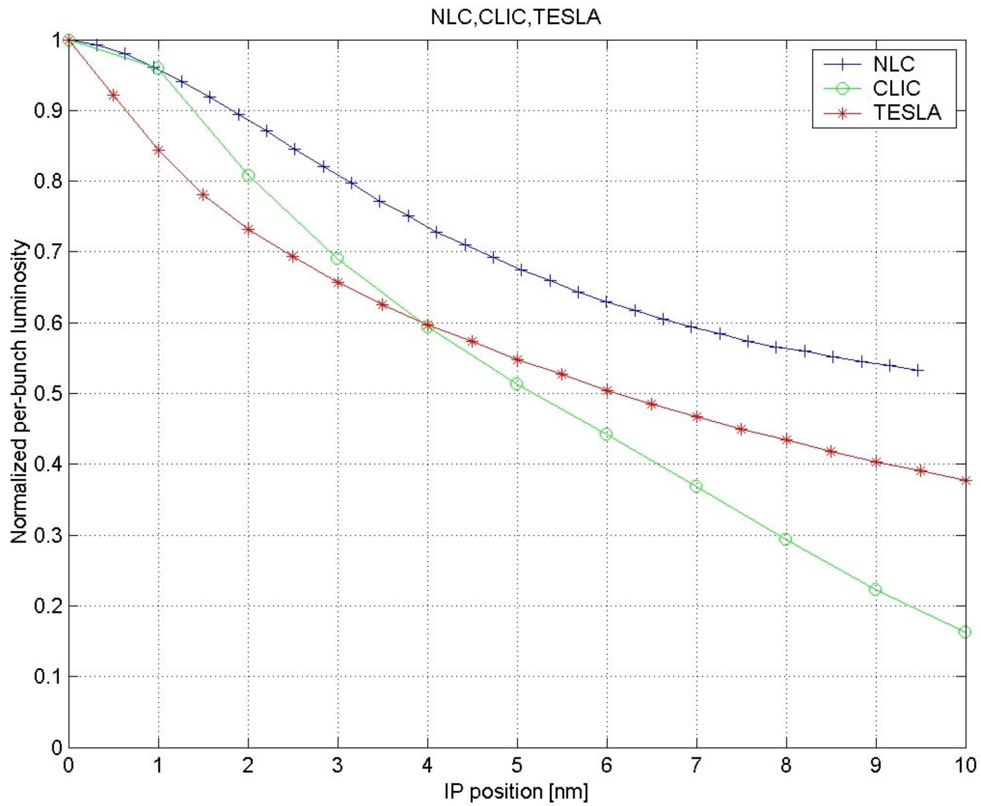
Simulation results for 256 pulses, 3 machine seeds * 3 groundmotion seeds:
Normalized luminosity for each ground motion model

Normalized luminosity as a function of (scanned) offset .



GM A (small) GM B (medium) GM C (large)

❖*(Note for TESLA: ~50 seconds, no angle control)*

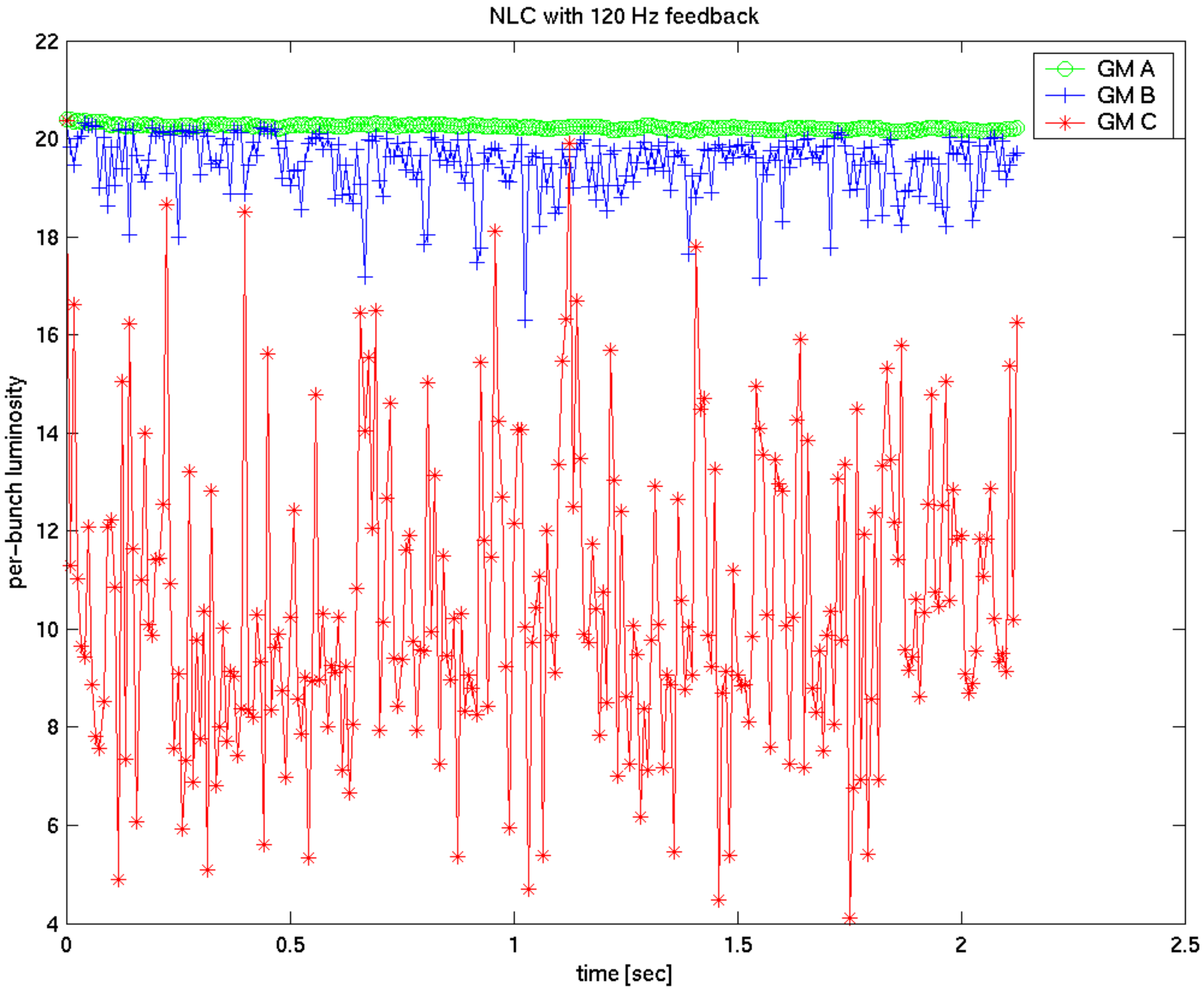


Simulation Results for NLC

With NLC-style IP deflection feedback

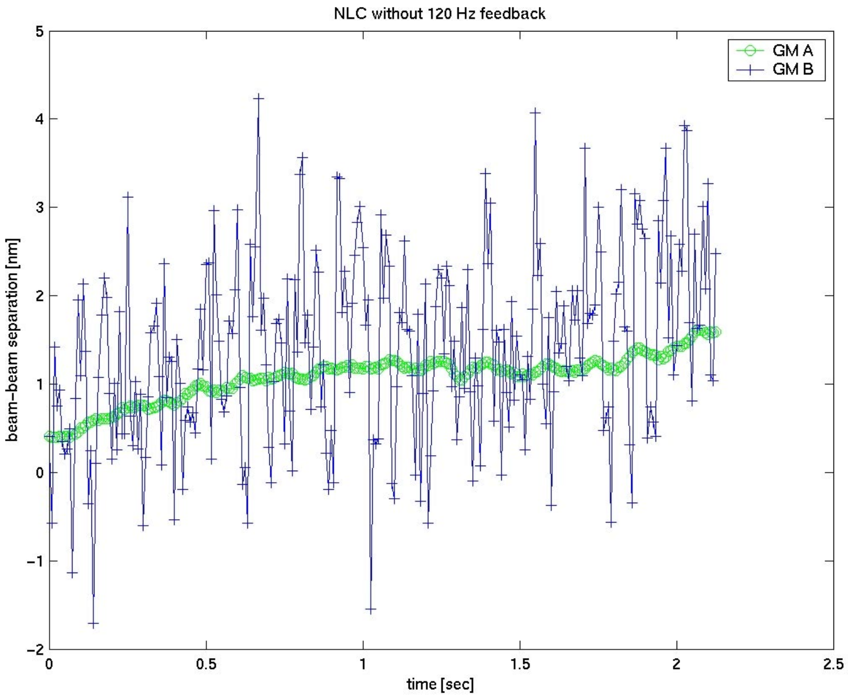
Per-bunch luminosity vs time for NLC feedback with ground motion

A, B, C

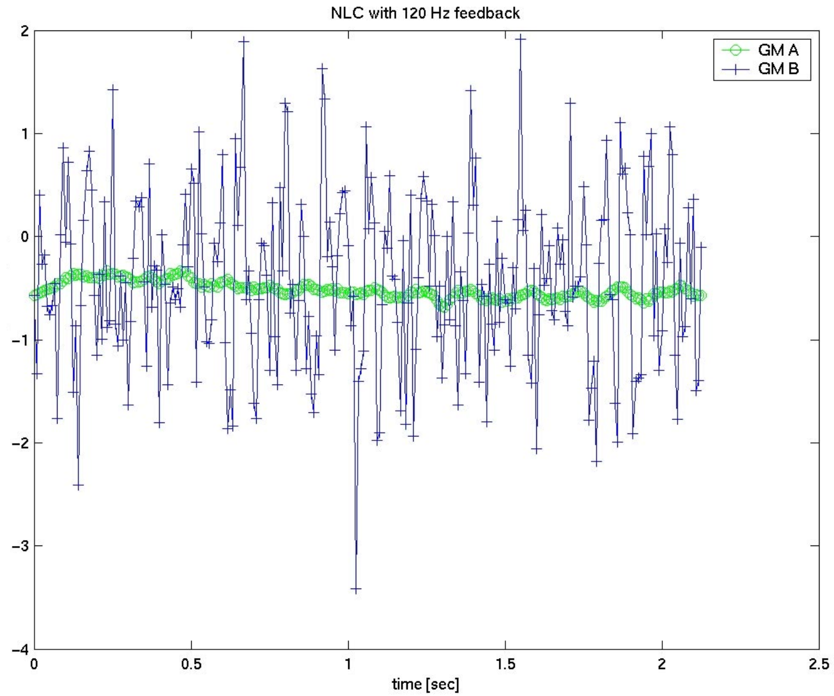


Simulation Results for NLC

IP Position offset vs time with feedback OFF/ON for ground motion A and B



Uncorrected

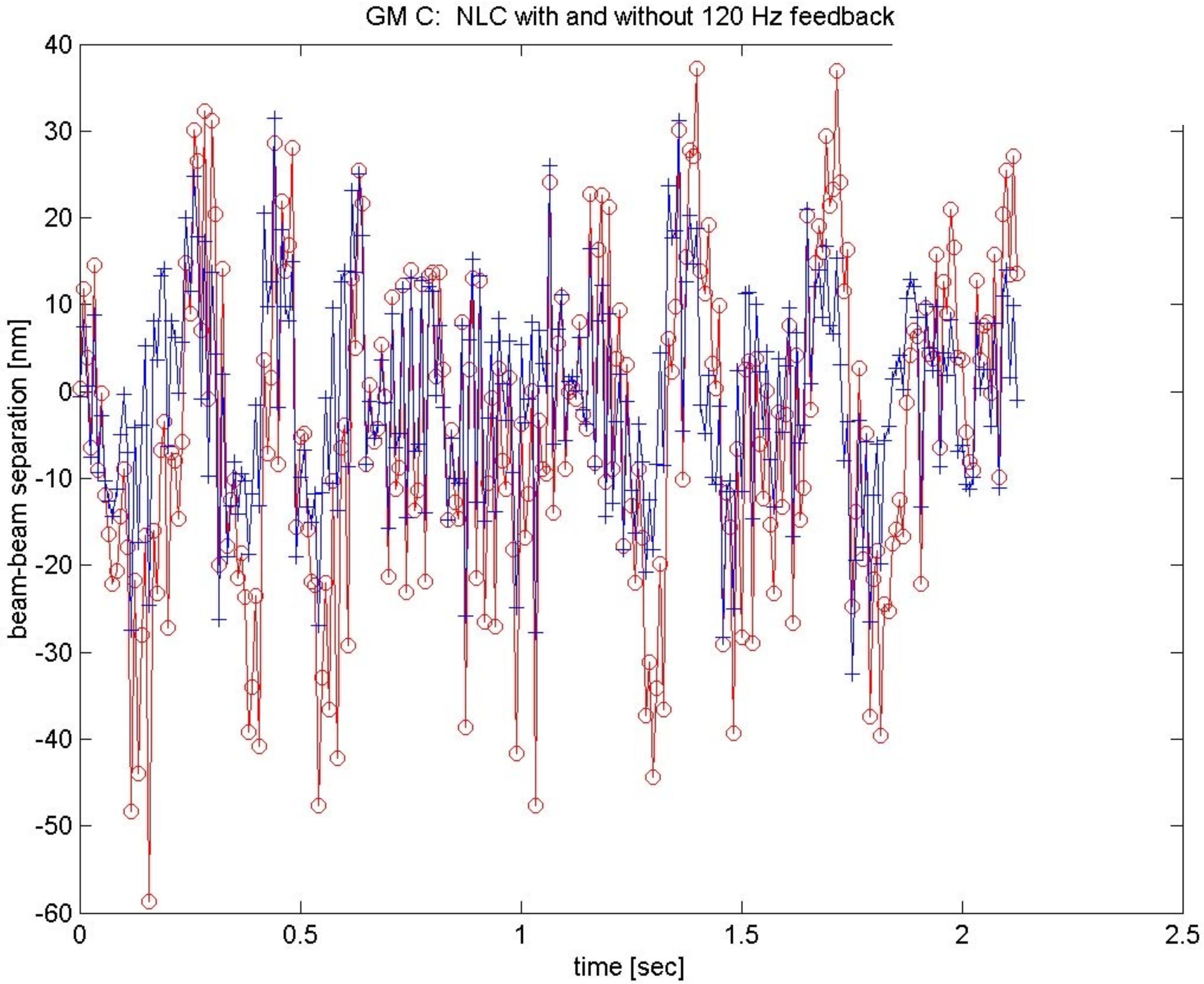


With NLC-style IP deflection feedback

Simulation Results for NLC

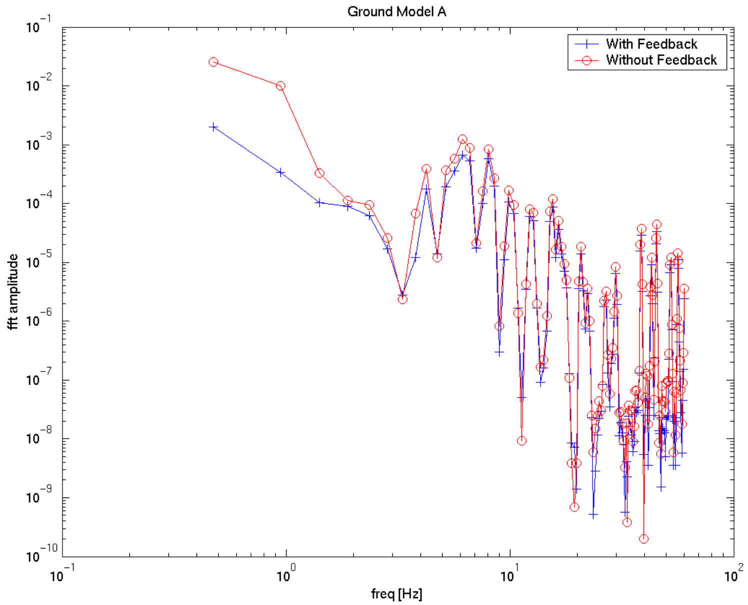
Next Linear Collider

IP Position offset vs time with feedback ON and OFF for ground motion C (large motion)

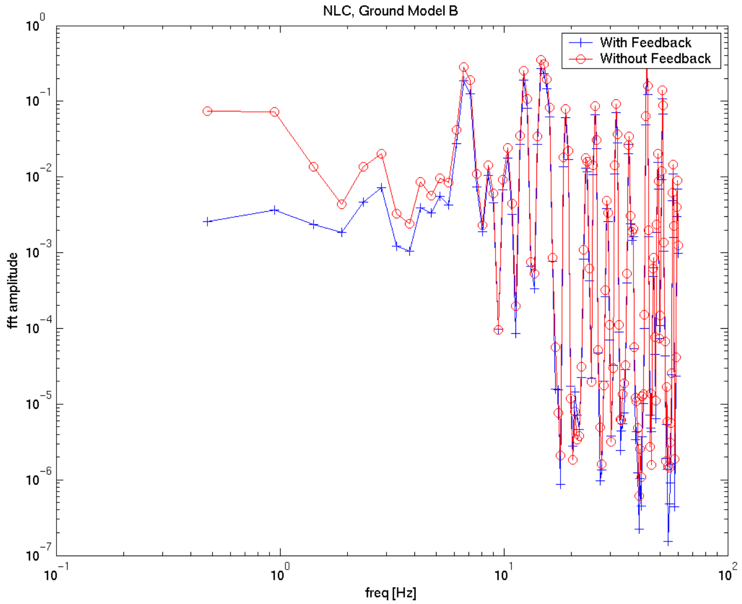


NLC fft's with feedback ON and OFF

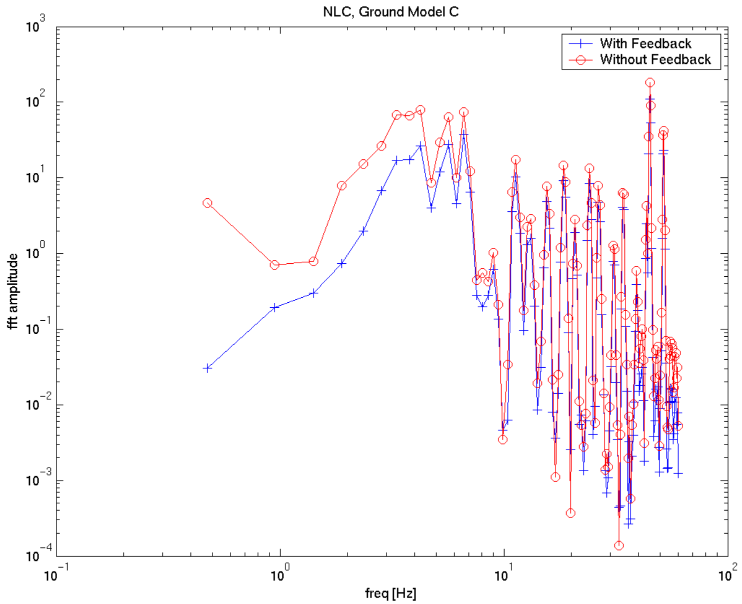
Next Linear Collider



GM A



GM B



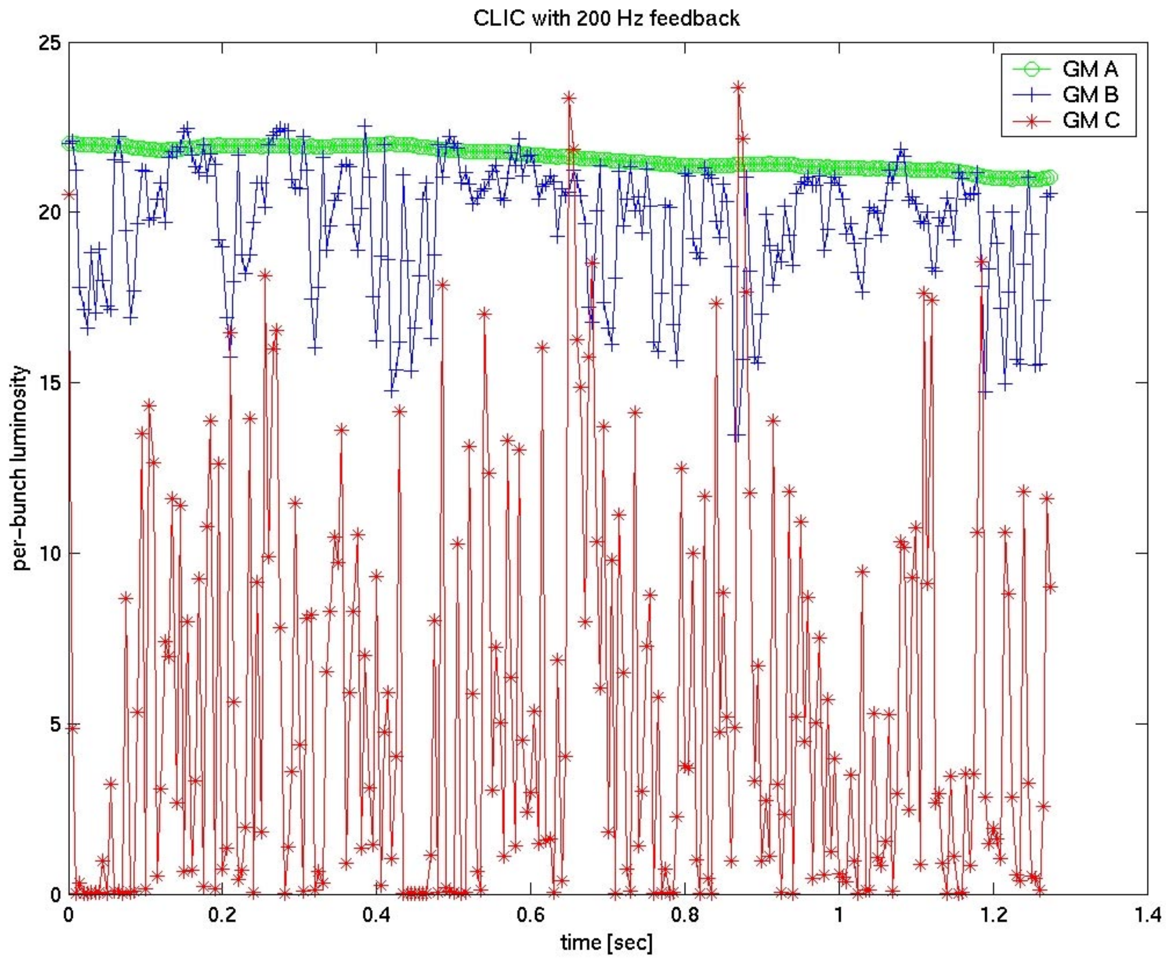
GM C

Simulation Results for CLIC

With NLC-style IP deflection feedback

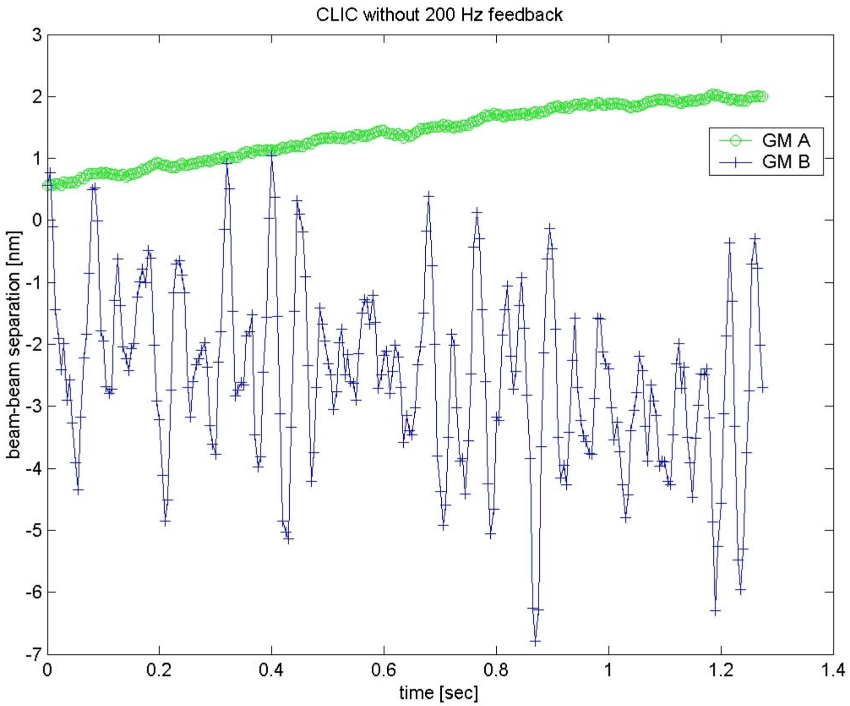
Per-bunch luminosity vs time for CLIC feedback with ground motion

A, B, C

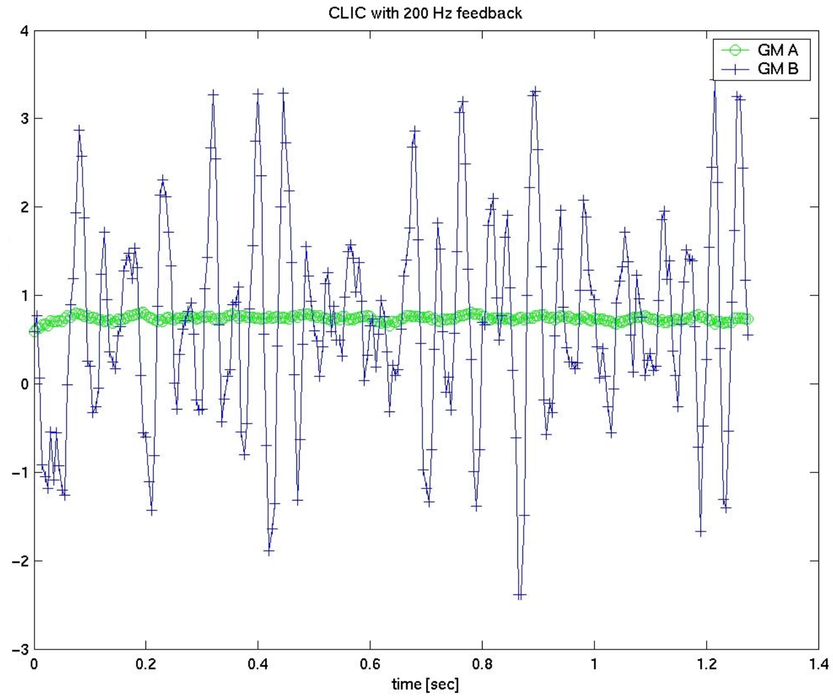


Simulation Results for CLIC

IP Position offset vs time with feedback OFF/ON for ground motion A and B



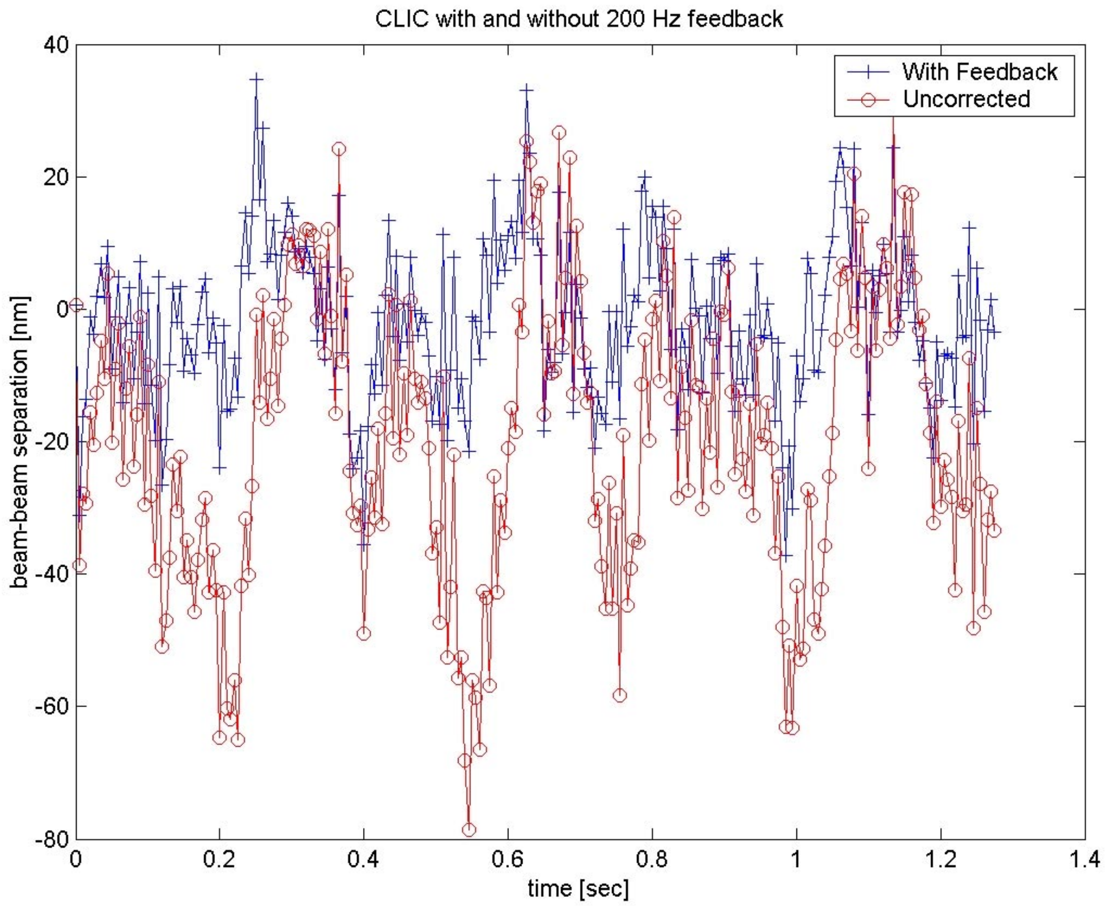
Uncorrected



With NLC-style IP deflection feedback

Simulation Results for CLIC

IP Position offset vs time with feedback ON and OFF for ground motion C (large motion)

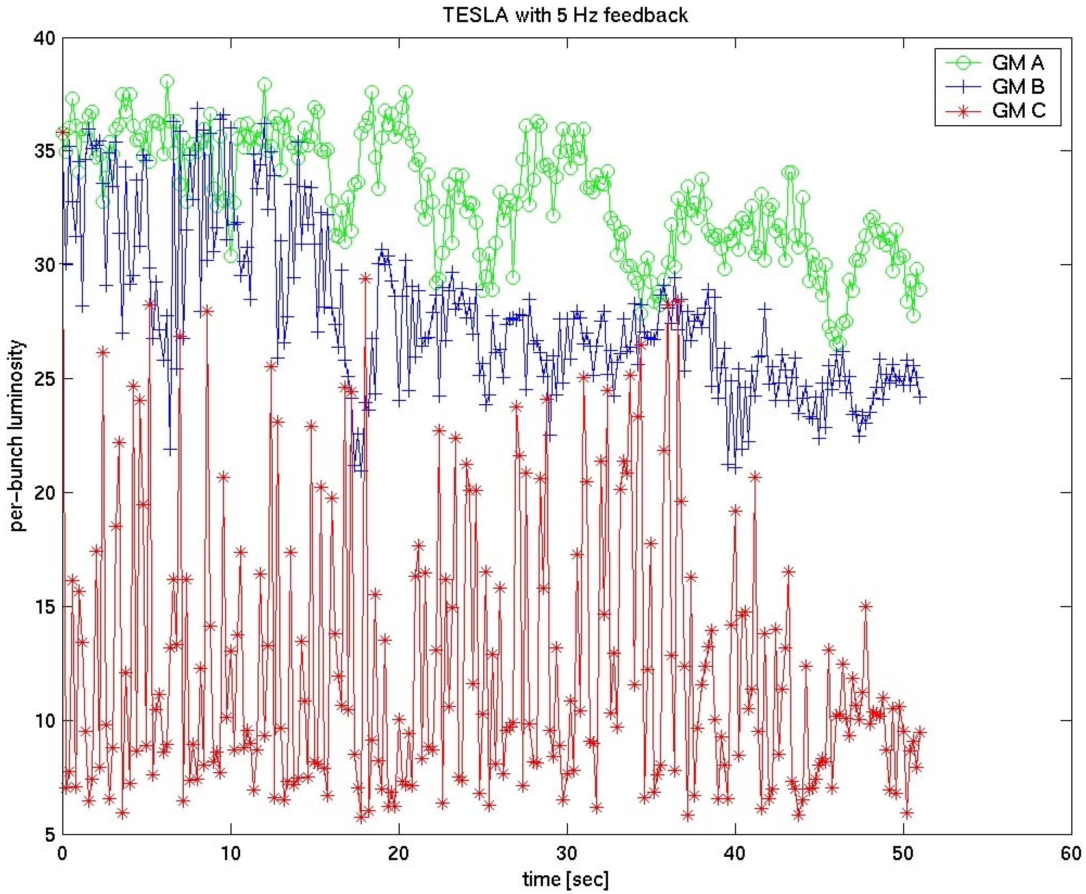


Simulation Results for TESLA

With NLC-style IP deflection feedback

Per-bunch luminosity vs time for TESLA 5-Hz feedback (no multibunch feedback, no angle control) with ground motion

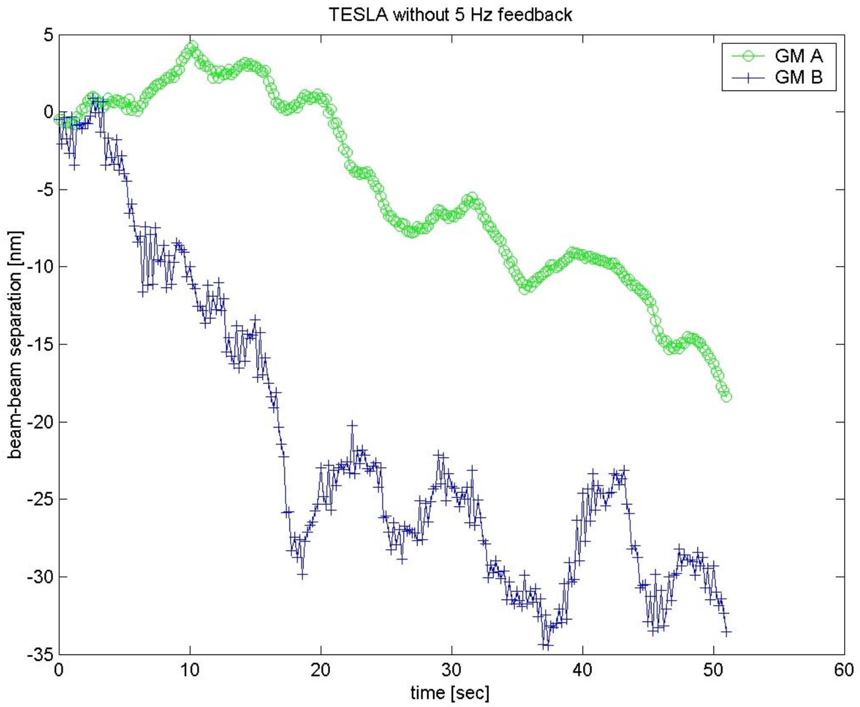
A, B, C

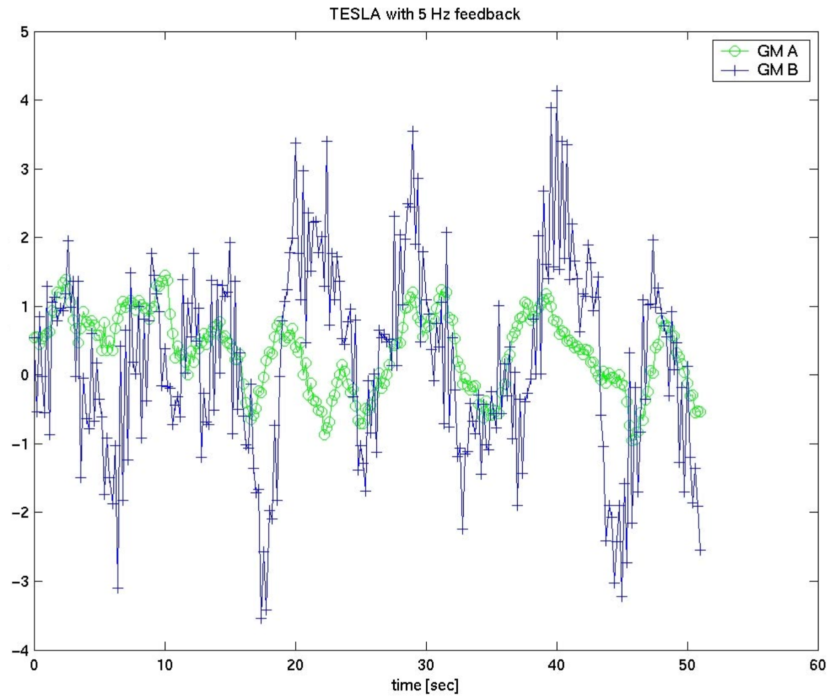


Simulation Results for TESLA

Next Linear Collider

IP Position offset vs time with feedback OFF/ON for ground motion A and B



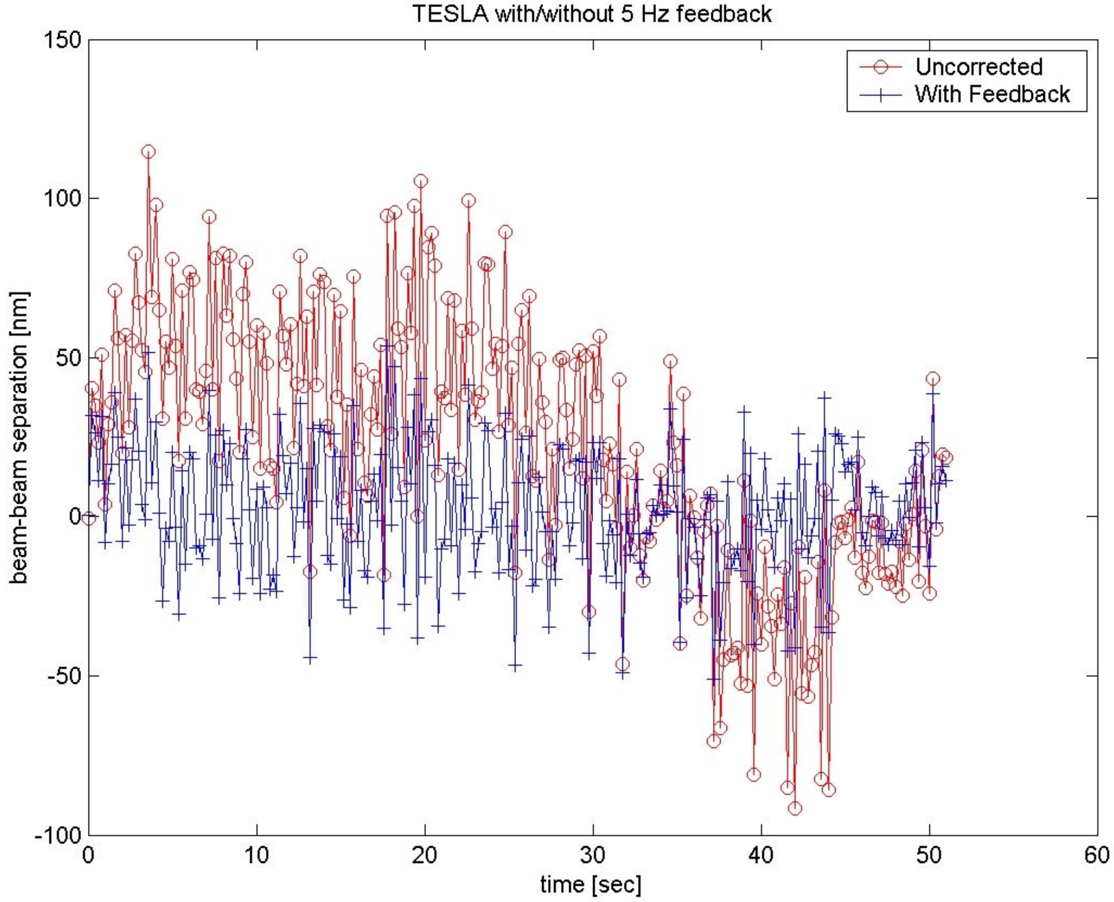


Uncorrected

With NLC-style IP deflection feedback

Simulation Results for TESLA

IP Position offset vs time with feedback ON and OFF for ground motion C (large motion)



CONCLUSIONS?

- SLC feedback experience is a good starting point.

- Feedback response has been improved from baseline design.

- Simple tools and methods for optimizing feedback design have been developed.

Future work for NLC?

Optimization of 120-Hz deflection feedback response for expected ground motion using LQG

More complete simulations of NLC tuning: sextupole orbit correction, optimization with luminosity jitter, realistic imperfections, upstream tuning; IP angle feedback?

Reevaluate linac feedback timescale and interactions with steering, dropped klystrons, etc.

etc…