Hybrid Repetitive Controller Using a Stochastic Evolutionary Search and a Deterministic Iterative Learning Law

B. Ufnalski, M. Małkowski and L. M. Grzesiak

Institute of Control and Industrial Electronics
Warsaw University of Technology, Poland

21st International Conference on Methods and Models in Automation and Robotics
IEEE MMAR 2016
29 October – 1 September, Międzyzdroje, Poland

IEEE Advancing Technology for Humanity

WARSAW UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING
INSTITUTE OF CONTROL AND
INDUSTRIAL ELECTRONICS

Outline

According to Simon Sinek, we should always "Start With Why" to motivate everybody in the house. So, there goes my why.

Why not just the very basic P-type control law?

$$u(p,k) = 1.0u(p,k-1) + k_{\mathrm{RC}}e(p+1,k-1)$$

- Note $p+1$ to reflect the assumed causality of the systems under consideration.

According to Simon Sinek, we should always "Start With Why" to motivate everybody in the house. So, there goes my why.

Why not just the very basic P-type control law?

$$u(p,k) = 1.0u(p,k-1) + k_{\mathrm{RC}}e(p+1,k-1)$$

- Note $p+1$ to reflect the assumed causality of the systems under consideration.
- This control law simply introduces pure integration in the pass-to-pass direction. The internal model principle.

IEEE Advancing Technology for Humanity

According to Simon Sinek, we should always "Start With Why" to motivate everybody in the house. So, there goes my why.

Why not just the very basic P-type control law?

$$u(p,k) = 1.0u(p,k-1) + k_{\mathrm{RC}}e(p+1,k-1)$$

- Note $p+1$ to reflect the assumed causality of the systems under consideration.
- This control law simply introduces pure integration in the pass-to-pass direction. The internal model principle.
- Its left-hand side is of a local type, even if its right-hand side is modified to be of a non-local type (e.g. by using controller errors for more time instances than just $p+1$).

According to Simon Sinek, we should always "Start With Why" to motivate everybody in the house. So, there goes my why.

Why not just the very basic P-type control law?

$$u(p,k) = 1.0\,u(p,k-1) + k_{\mathrm{RC}}\,e(p+1,k-1)$$

- Note $p+1$ to reflect the assumed causality of the systems under consideration.
- This control law simply introduces pure integration in the pass-to-pass direction. The internal model principle.
- Its left-hand side is of a local type, even if its right-hand side is modified to be of a non-local type (e.g. by using controller errors for more time instances than just $p+1$).
- Is inherently unstable in the presence of repetitive disturbance and (logical AND) actuator saturation that prevents at least one error sample along the pass from being driven to zero.

Why not just the very basic P-type control law?

It then has to be modified into

$$u(p,k) = \cancel{1.0}\,\mathbf{Q}\left(z^{-1}\right) u(p,k-1) + \cancel{k_{\mathrm{RC}}}\,k_{\mathrm{RC}}\mathbf{L}\left(z^{-1}\right) e(p+1,k-1),$$

where **Q** and **L** are low-pass filters. This compromises the performance and hence there still is plenty of room for new iterative learning techniques.

Why not just the very basic P-type control law?

It then has to be modified into

$$u(p,k) = \cancel{1.0}\,\mathbf{Q}\left(z^{-1}\right) u(p,k-1) + \cancel{k_{\mathrm{RC}}}\,k_{\mathrm{RC}}\mathbf{L}\left(z^{-1}\right) e(p+1,k-1),$$

where **Q** and **L** are low-pass filters. This compromises the performance and hence there still is plenty of room for new iterative learning techniques.

Some improvement may be possible by replacing $p+1$ with $p+p_{\mathrm{RC}}$, where $p_{\mathrm{RC}} > 1$. For more details please see “B-spline based repetitive controller revisited: error shift, higher-order polynomials and smooth pass-to-pass transition” co-authored by me.

IEEE Advancing Technology for Humanity

How to shape performance according to user-defined objectives?

We can do better than the local control rule. We can go global and let the engineer define the cost function most relevant to his control problem!

Repetitiveness enables iterativeness

Repetitive process $\Rightarrow$ **iterative learning control.** It's so cliché that it's even cliché to say it's cliché.

Repetitiveness enables iterativeness

Repetitive process $\Rightarrow$ iterative learning control. It's so cliché that it's even cliché to say it's cliché.

Evolutionary search algorithm $\Rightarrow$ iterative learning mechanism. Again, it's so cliché that...

Repetitiveness enables iterativeness

Repetitive process $\Rightarrow$ iterative learning control. It's so cliché that it's even cliché to say it's cliché.

Evolutionary search algorithm $\Rightarrow$ iterative learning mechanism. Again, it's so cliché that...

Repetitive controller $\Leftrightarrow$ on the fly optimization problem of shaping control signal. Intriguing?

Repetitiveness enables iterativeness

 **Repetitive process** $\Rightarrow$ **iterative learning control.** It's so cliché that it's even cliché to say it's cliché.

 **Evolutionary search algorithm** $\Rightarrow$ **iterative learning mechanism.** Again, it's so cliché that...

 **Repetitive controller** $\Leftrightarrow$ **on the fly optimization problem of shaping control signal.** Intriguing?

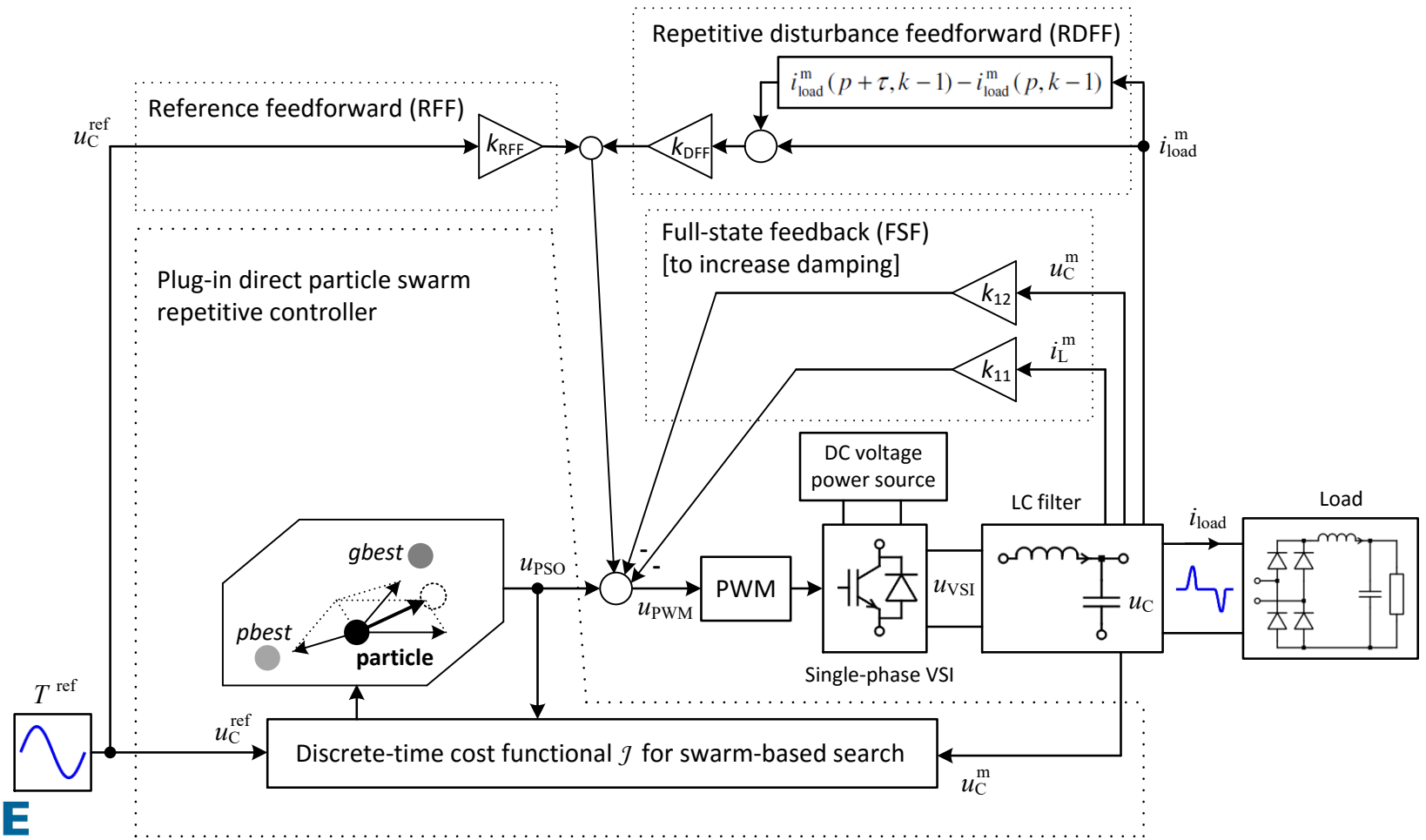
 **Repetitive process control** $\Leftrightarrow$ **dynamic optimization problem (DOP).** Things are getting interesting, aren't they? BTW, why dynamic? Download e.g. my CPE-POWERENG 2016 presentation here.

Direct particle swarm repetitive controller

All in all, the starting point is to note that a physical plant subject to a repetitive process can itself serve as the critic for a population-based evolutionary search, such as PSO, aimed at finding optimal control signal along the entire pass and then constantly updating it on-the-fly in real time according to a given performance index.

WARSAW UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING
INSTITUTE OF CONTROL AND
INDUSTRIAL ELECTRONICS

Direct particle swarm repetitive controller



Optimality vs. robustness

Optimality does not imply robustness!

The control objective function. It is global!

$$\mathcal{J}(k) = \mathcal{J}_0 + \underbrace{\sum_{p=1}^{\alpha} \left(u_{\mathrm{C}}^{\mathrm{ref}}(p) - u_{\mathrm{C}}^{\mathrm{m}}(p,k)\right)^2}_{\text{penalty for control error}} + \underbrace{\beta \sum_{p=1}^{\alpha} \left(u_{\mathrm{PSO}}(p,k) - u_{\mathrm{PSO}}(p-1,k)\right)^2}_{\text{penalty for control signal dynamics to robustify the system}}$$

WARSAW UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING
INSTITUTE OF CONTROL AND
INDUSTRIAL ELECTRONICS

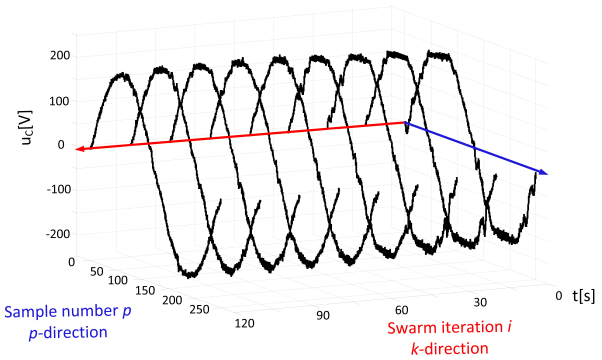
PSO as the control algorithm

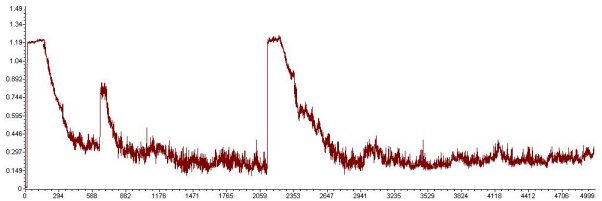
$$\begin{aligned}\Delta\mathbf{u}_{\mathrm{PSO}}(k+1) &= c_1\Delta\mathbf{u}_{\mathrm{PSO}}(k) + \\ &+ c_2 r^{\mathrm{pbest}}\left(\mathbf{q}^{\mathrm{pbest}}(k) - \mathbf{q}(k)\right) + \\ &+ c_3 r^{\mathrm{gbest}}\left(\mathbf{q}^{\mathrm{gbest}}(k) - \mathbf{q}(k)\right) \\ \mathbf{q}(k+1) &= \mathbf{q}(k) + \Delta\mathbf{u}_{\mathrm{PSO}}(k+1)\ ,\end{aligned}$$

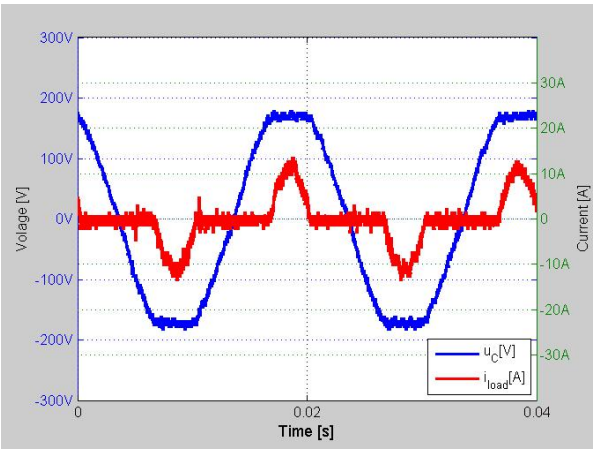
where: $\mathbf{u}_{\mathrm{PSO}}$ is the contributor to the overall control signal, $\mathbf{q}$ is the position of the particle, $\mathbf{q}^{\mathrm{pbest}}$ stores the best solution proposed so far by the particular particle, $\mathbf{q}^{\mathrm{gbest}}$ denotes the best solution found so far by the swarm, r^{pbest} and r^{gbest} are random numbers uniformly distributed in the unit interval.

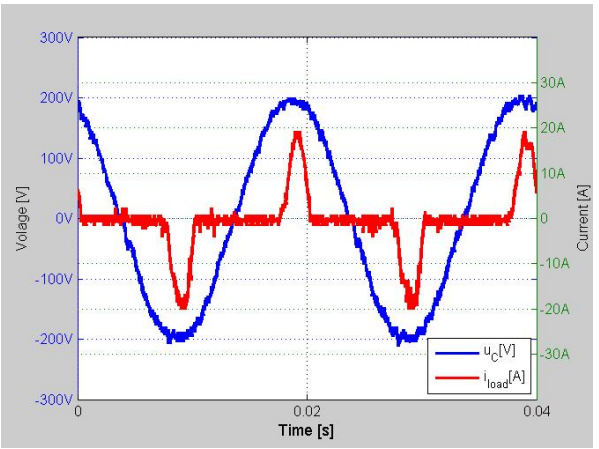
WARSAW UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING
INSTITUTE OF CONTROL AND
INDUSTRIAL ELECTRONICS
IS EP

Direct particle swarm repetitive control algorithm executed in real time on, e.g., TI TMS320F2812 micro? Are you kidding me?!









IEEE Advancing Technology for Humanity

PSO is innately sluggish in comparison to gradient-based search methods

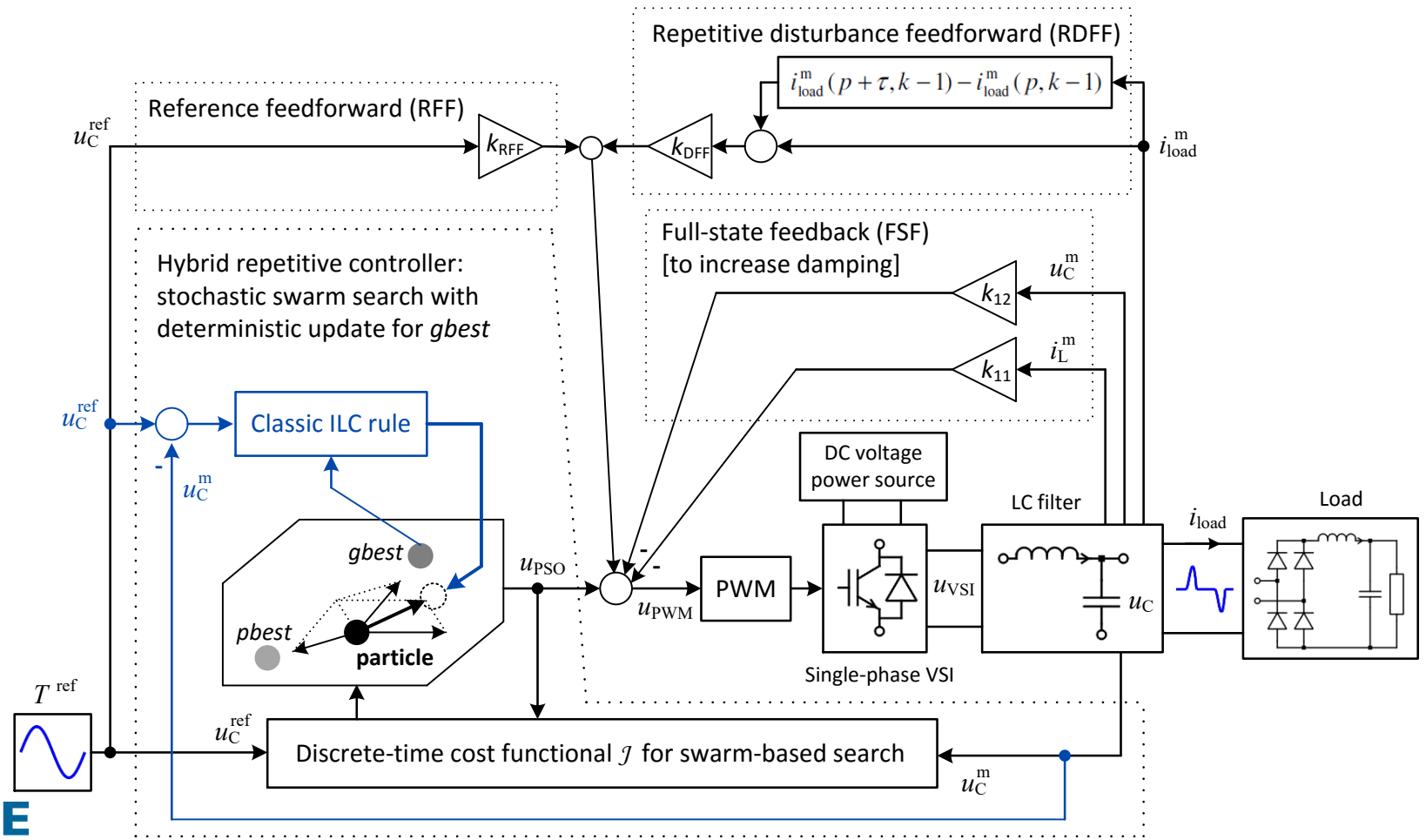
Our currently most successful remedy is to combine the classic deterministic P-type ILC law with the classic stochastic PSO search rule. There are several ways to do this, e.g:

$$\begin{aligned}
\mathbf{q}^{\mathrm{gbest}}(k) &:= \mathbf{q}^{\mathrm{gbest}}(k) + k_{\mathrm{RC}}\mathbf{e}(k-1) \\
\Delta\mathbf{u}_{\mathrm{PSO}}(k+1) &= c_1\Delta\mathbf{u}_{\mathrm{PSO}}(k) + \\
&\quad + c_2 r^{\mathrm{pbest}}\left(\mathbf{q}^{\mathrm{pbest}}(k) - \mathbf{q}(k)\right) + \\
&\quad + c_3 r^{\mathrm{gbest}}\left(\mathbf{q}^{\mathrm{gbest}}(k) - \mathbf{q}(k)\right) \\
\mathbf{q}(k+1) &= \mathbf{q}(k) + \Delta\mathbf{u}_{\mathrm{PSO}}(k+1) \ ,
\end{aligned}$$

The assignment operator(:=) indicates that we calculate the *gbest* attractor using the classic PSO formula and just after doing that we move it using the classic ILC formula. Then we proceed with all the usual PSO calculation.

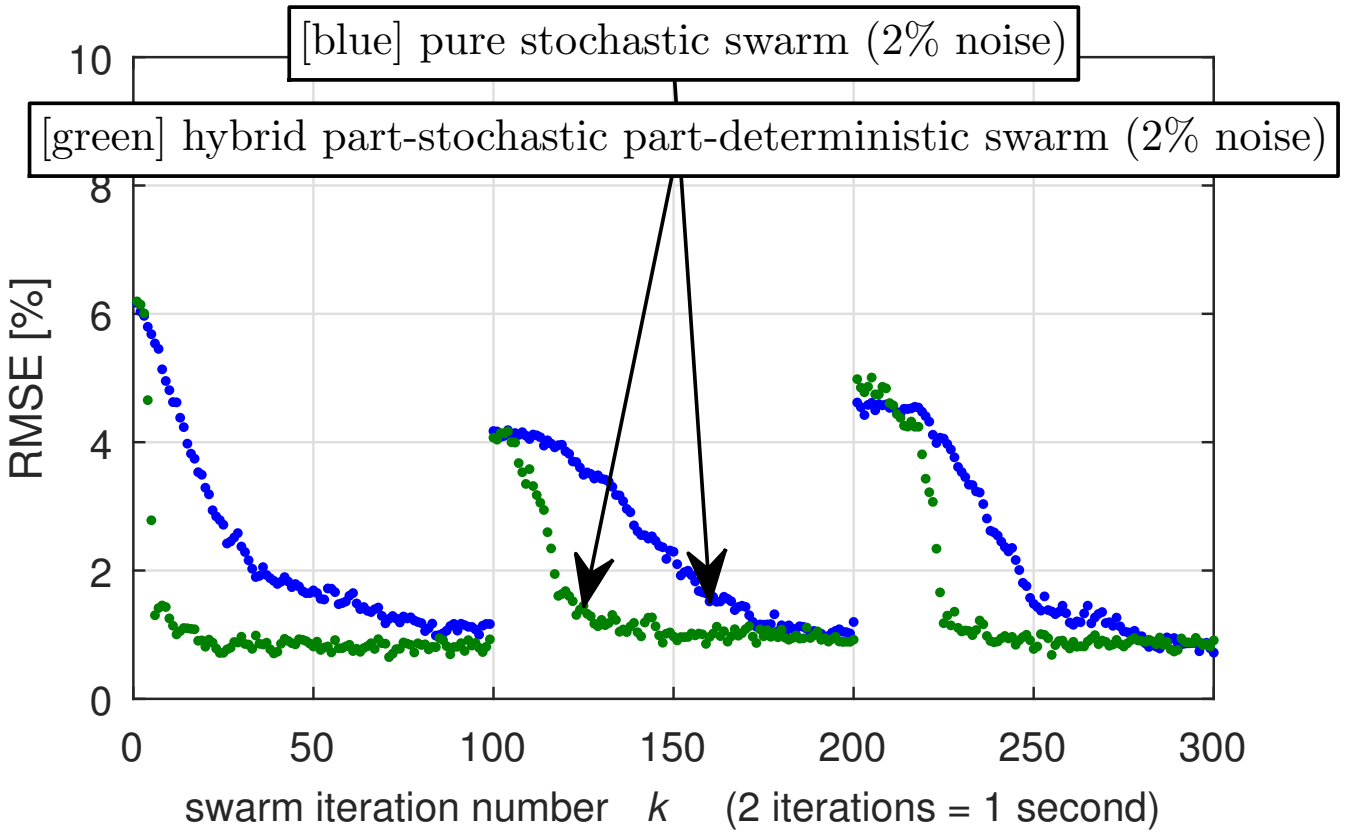
WARSAW UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING
INSTITUTE OF CONTROL AND
INDUSTRIAL ELECTRONICS

Hybrid part-stochastic part-deterministic RC (law no. 1)



WARSAW UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING
INSTITUTE OF CONTROL AND
INDUSTRIAL ELECTRONICS

Responsiveness (convergence rate) comparison



More graphs in the paper!

Hybrid part-stochastic part-deterministic RC (law no. 2)

And here is our second way to combine the deterministic P-type ILC law with the stochastic PSO search rule:

$$\begin{aligned}\Delta\mathbf{u}_{\mathrm{PSO}}(k+1) = {} & c_1\Delta\mathbf{u}_{\mathrm{PSO}}(k) + \\ & + c_2 r^{\mathrm{pbest}}\left(\mathbf{q}^{\mathrm{pbest}}(k) - \mathbf{q}(k)\right) + \\ & + c_3 r^{\mathrm{gbest}}\left(\mathbf{q}^{\mathrm{gbest}}(k) - \mathbf{q}(k)\right) + \\ & + k_{\mathrm{RC}}\mathbf{e}(k-1) \\ \mathbf{q}(k+1) = {} & \mathbf{q}(k) + \Delta\mathbf{u}_{\mathrm{PSO}}(k+1)\ ,\end{aligned}$$

We extend the PSO movement rule with the deterministic P-type update component. Note that this is not equivalent to putting both controllers in parallel and adding their outputs. The part-stochastic part-deterministic search is entirely subject to fitness evaluation according to the user-defined goal. If the penalty for control signal dynamics is included, no overlearning occurs.

Hybrid part-stochastic part-deterministic RC (law no. 3)

And here is our third way to combine the deterministic P-type ILC law with the stochastic PSO search rule:

$$\begin{aligned}\Delta\mathbf{u}_{\mathrm{PSO}}(k+1) = {} & c_1\Delta\mathbf{u}_{\mathrm{PSO}}(k) + \\ & + c_2 r^{\mathrm{pbest}}\left(\mathbf{q}^{\mathrm{pbest}}(k) - \mathbf{q}(k)\right) + \\ & + c_3 r^{\mathrm{gbest}}\left(\mathbf{q}^{\mathrm{gbest}}(k) - \mathbf{q}(k)\right) + \\ & + k_{\mathrm{RC}} r^{\mathrm{IMP}}\mathbf{e}(k-1) \\ \mathbf{q}(k+1) = {} & \mathbf{q}(k) + \Delta\mathbf{u}_{\mathrm{PSO}}(k+1)\ ,\end{aligned}$$

This is a slightly modified control law no. 2. Similarly to the second and third components in the classic particle speed update formula, we introduce randomness to the fourth component. Random numbers r^{IMP} are uniformly distributed in the unit interval.

The takeaways

Everything should be made as simple as possible, but not simpler. Einstein's razor vs. Occam's razor. The classic ILC has its stability/robustness issues. The PSO-based ILC has its sluggishness disadvantage. Hybrid part-stochastic part-deterministic repetitive controllers hold the solution to all these headaches.

Hybrid ILC law no. 1 on MATLAB Central.

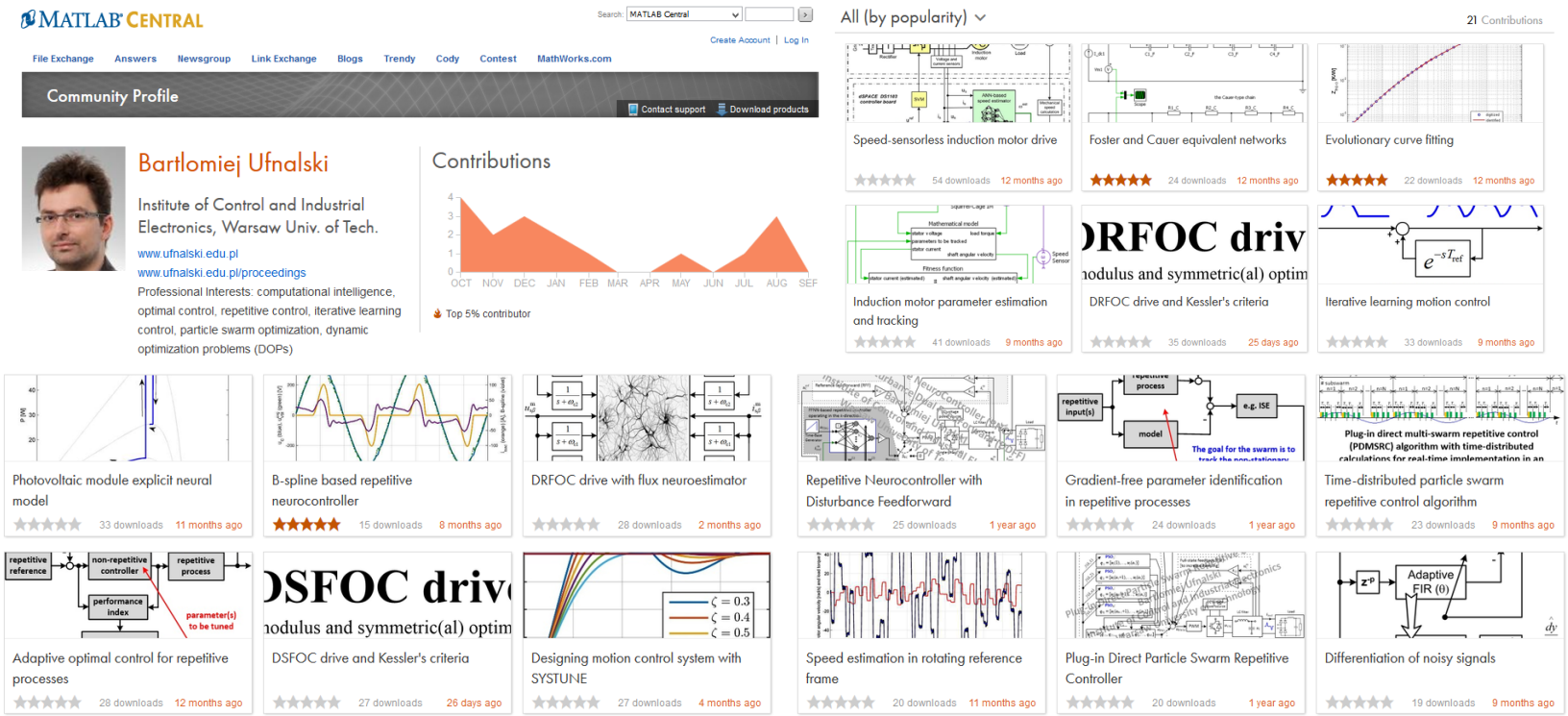
Hybrid ILC law no. 2 on MATLAB Central.

Hybrid ILC law no. 3 on MATLAB Central.

WARSAW UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING
INSTITUTE OF CONTROL AND
INDUSTRIAL ELECTRONICS
P W
IS EP

MATLAB Central

http://www.mathworks.com/matlabcentral/profile/authors/2128309-bartlomiej-ufnalski



Warsaw University of Technology
Faculty of Electrical Engineering
Institute of Control and
Industrial Electronics

Comments? Questions?

Thank you for having me!

A clickable version of this presentation is available at
www.ufnalski.edu.pl/proceedings.

Enjoy your time at MMAR 2016 Conference!

IEEE Advancing Technology for Humanity