Adaptive-Bias-Adaptive-Gain

Sven Schneider

July 19, 2020

1 Overview

Original publication: Franchi, Antonio and Mallet, Anthony *Adaptive closed-loop speed control of BLDC motors with applications to multi-rotor aerial vehicles*, in Proc. IEEE International Conference on Robotics and Automation (ICRA), 2017. [1]

Note: Fig. 4 in this paper shows the controller's step and chirp responses. The plots in the second row from above are labeled x and x^d. This seems to be a typo and instead should read $\dot{x}$ and $\dot{x}^d$.

2 Bias/gain update

- See Figure 1

$\bar{e}$: Given an error signal $\bar{e}$, the update step adapts the output signal z (which can either represent the ABAG's bias or gain).

$\xi_z(\cdot)$: The error signal is thresholded by an arbitrary, user-defined function $\xi_z(\cdot)$ (also called decision map) which describes the *waveform* of the signal z. This thresholding decouples the signal z from the *magnitude* of the error signal! As a result, the overall controller only follows the *trend* of the error.

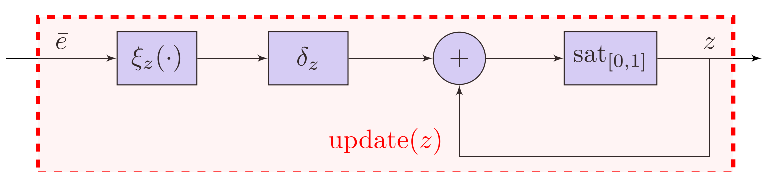

Figure 1: The bias and gain update step

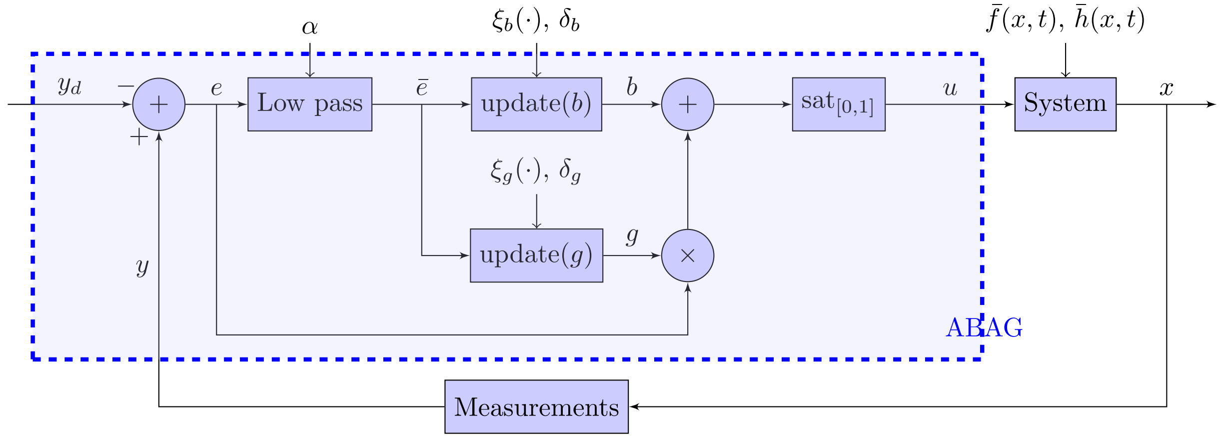

Figure 2: The ABAG controller in a feedback control loop

δ_z: δ_z is the signal adaptation step. It can be seen as a "meta-gain", that determines how the signal (i.e. the bias or gain) should be adapted in the next cycle.

$+$: The previous signal z is fed-back into the current adaptation step and, therefore, is low-pass filtered. As a result, if the error remains zero for a "long" time, also the signal will asymptotically approach zero (note: most likely this also depends on the $\xi_z(\cdot)$ function).

sat$_{[0,1]}$: The signal is thresholded to the range $z \in [0, 1]$. TODO: rationale?

3 ABAG controller

- See Figure 2
- The two update blocks are described in section 2
- Core observations:
  1. The controller is (mostly) "dimension-less" in that most entities that appear in the controller are *relative* to a maximum value.
  2. Due to the low pass filter and the decoupling of waveform and error magnitude (see above), the bias term b "slowly" follows the trend in the error. Also notice, that it directly contributes to the control signal u. Examples of such "slow" disturbances on a system could be gravity or friction.
  3. Just like the bias term, also the gain term g slowly adapts to the trend in the error. However, it acts on the instantaneous error e

(just like the proportional gain acts on the instantaneous error in a P-controller). Therefore, it reacts faster than the bias term.

4. The control signal u is thresholded to the range $u \in [0, 1]$. This means that by purely relying on the ABAG controller, the system will never be (i) actively braked; or (ii) actively controlled in the "inverse" direction. Instead, both of those objectives are left to the system's natural dynamics (e.g. damping via friction or letting gravity "pull" the system back). Thus, the ABAG controller can be seen as a *lazy* control strategy!

- Fig. 4 in the original publication (second last row) shows the contribution of g and b to the overall control signal u. In the case of the step response, one clearly sees that initially both, g and b, drive the system. After some time the immediate control action g decreases and all the control effort is taken over by the bias b.

- Monitoring of gain: We can/should attach a monitor to the gain signal. If this signal remains "active" for longer time, it indicates that the controller is mostly running in feedback mode. In turn, this means that the model (i.e. feed-forward bias term) is not working "correctly". There can be various reasons for this, for instance, the model itself is not good (design issue), the model parameters are not well chosen (e.g. identification issue), or the model is not correct for the current operation state (maybe a higher-level state machine should switch to a different configuration).

4 Choice of decision maps

- Allows definition of deadband/deadzone where the signal is not adapted

- By changing the deadbands' thresholds, the allowable tolerances of the controller can be adapted online. Examples:

  1. One example is a manipulation task where the robot should pick up an object. At the beginning of the motion only the rough direction towards the goal should match, but towards the end of the motion more precision is usually required.
  2. A different example is an object placement task where a goal region is specified. Here the controller's thresholds can be directly infered from the region where the object should be placed.

- The decision map must also determine the desired *direction* and *relative* magnitude of the control action (for this signal, i.e. bias or gain). Note: this is also part of decoupling the error's magnitude from the control signal's magnitude.

- As a result this control approach resembles a *sliding mode control*

- A more generic choice of the decision maps could include the following parameters:
  - The shapes of the waveforms themselves (this may include some of the following parameters)
  - Discretization of the waveform, i.e. the waveform takes discrete steps
  - The saturation limits of each waveform. For instance, the bias may contribute up to 100% while the gain maybe is only allowed to contribute up to 15% (arbitrarily-chosen numbers). This way, the gain can take care of smaller disturbances, while the main effort rests on the bias (i.e. the model).
  - The sampling time of both the bias (ΔT_b) and the gain (ΔT_g) computation. For instance, in the case of a driving vehicle, maybe the model (i.e. the bias) must only be updated every second or so.
  - The deadband when the controller starts reacting to the error.
- The shapes of the decision maps roughly resemble *attractors* in dynamical systems; sin for the bias and −cos for the gain as depicted in Figure 3. (Non-linear) dynamical systems form the basis of various control approaches including:
  - Behaviour modelling as presented in [**?**, **?**, **?**][1]. Can we also exploit this insight toghether with the approach form to
    - control the orientation
    - introduce deadzones by multiplication with a window
    - limit/saturate the signal with a window
  - Sliding mode control [**?**] where *(i)* the sliding surface must be attractive; and *(ii)* the switch to the sliding surface must happen instantaneously.
  - Dynamic movement primitives (DMP) [**?**, **?**][2]

5 Extensions & context robotics

- How does the bias term relate to *Iterative Learning Control* [2, 3, 4, 5]?
- A robotic manipulator potentially runs multiple (ABAG) controllers. Higher-level controllers drive the end-effectors towards Cartesian goals, while lower-level controllers compute torque setpoints for the individual joints. The dynamics solver is located inbetween the task-level and the joint-level controllers.

[1] `https://studywolf.wordpress.com/2016/05/13/dynamic-movement-primitives-part-4-avoiding-obstacles/`
[2] `https://studywolf.wordpress.com/2013/11/16/dynamic-movement-primitives-part-1-the-basics/`

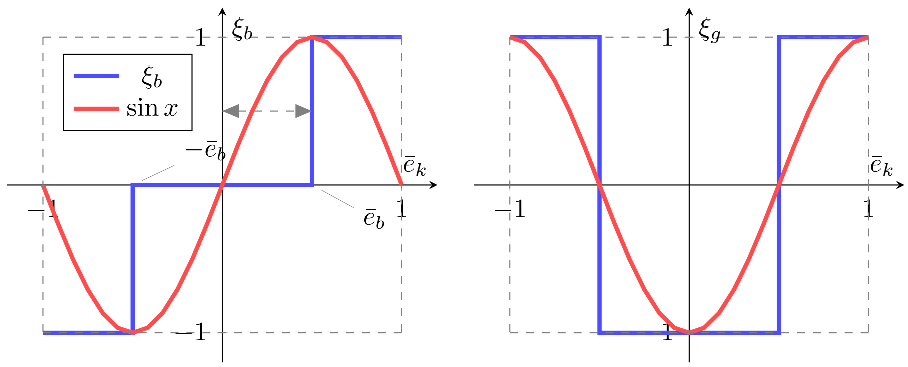

Figure 3: Comparison of discrete decision maps (blue) and continuous trigonometric functions (red) as choice of attractors in dynamical systems. Bias and gain signals are shown left and right, respectively.

- In many applications the control commands to the real system are *dimensional* (e.g. velocity, torque, etc.) in contrast to the adimensional PWM signal that is produced in the original ABAG paper. Hence, at some point the adimensional signal should be scaled. But this is problematic since there are very few signals for which we know the maximum value (e.g. Cartesian velocities, accelerations or forces of a robot's end-effector are dependent on the robot's configuration). An exception are the actuator torque limits. [6] decided to scale the Cartesian quantities with artificial maxima and then propagate them through the kinematic chain with the ACHD solver ("scale-then-solve"). However, would it be possible to postponse the scaling until after the dynamics computations ("solve-then-scale")? This way the torque limits can really be exploited. The approach requires the propagation of individual motion drivers [7].

- The ABAG's gain signal resembles the PID's proportional signal, while the bias signal resembles the integral signal. Superficially, there does not seem to be a derivative-like signal in the ABAG. However, upon closer inspection the low pass filter may play a role similar to the derivative signal. Alternatively, a more explicit and correct (?) approach may be to use the *model-based* feedforward from the dynamics computation as shown in Figure 4 as the derivative signal.

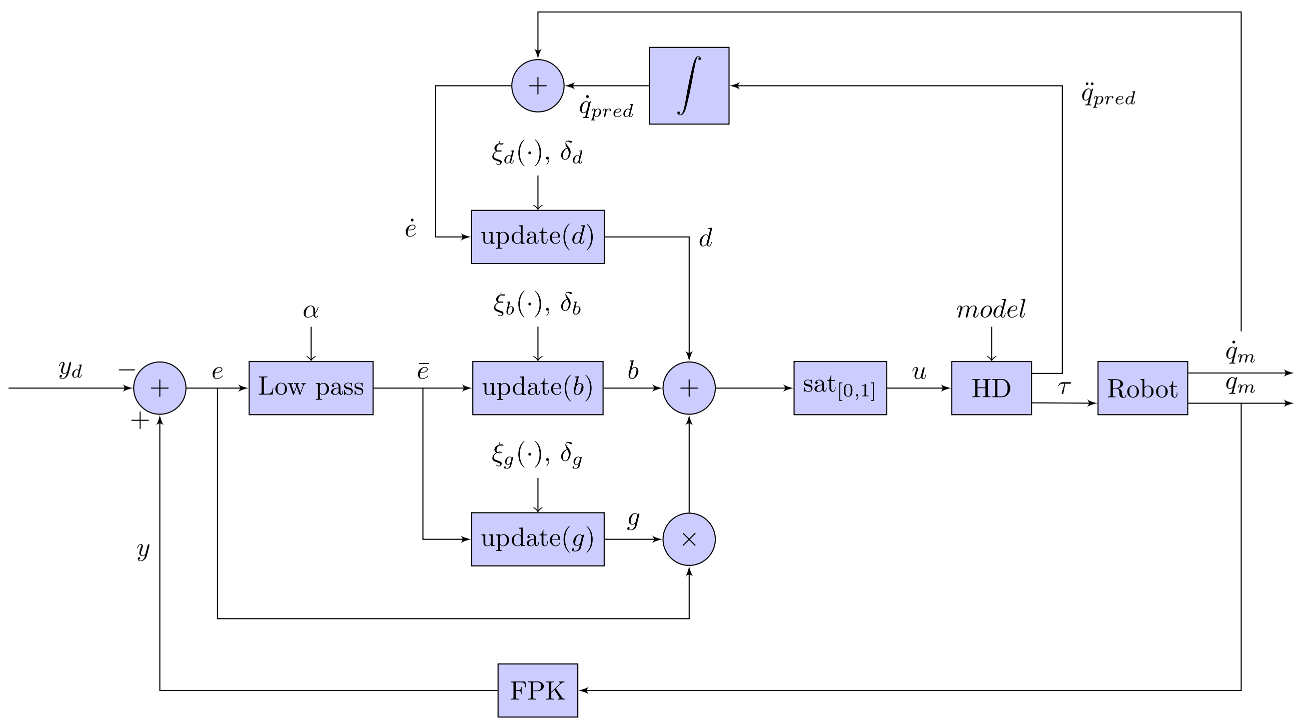

Figure 4: ABAG integrated with the acceleration-constrained hybrid dynamics solver and extended with a “D-like”, model-based term. TODO: It does not work like this yet, since the D-like term originates from joint space while the other terms originate from Cartesian space!

References

[1] A. Franchi and A. Mallet, "Adaptive closed-loop speed control of bldc motors with applications to multi-rotor aerial vehicles," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2017. [Online]. Available: https://hal.laas.fr/hal-01476812/document

[2] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 123–140, 1984.

[3] ——, "Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems," in *Proc. IEEE Conference on Decision and Control (CDC)*, 1984.

[4] M. Schwarz and S. Behnke, "Compliant robot behavior using servo actuator models identified by iterative learning control," in *RoboCup 2013. Lecture Notes in Computer Science*, 2013. [Online]. Available: https://www.ais.uni-bonn.de/papers/RC13_Schwarz.pdf

[5] J.-S. Liu, "Joint stick-slip friction compensation for robotic manipulators by iterative learning," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1994.

[6] D. Vukcevic, "Lazy robot controlby relaxation of motion and force constraints," Master's thesis, Hochschule Bonn-Rhein-Sieg, 2019.

[7] S. Schneider and H. Bruyninckx, "Exploiting linearity in dynamics solvers for the design of composable robotic manipulation architectures," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.