# ICL and Model Predictive Control and its Variants

In the current commercial environment it would be surprising if we had not been asked to consider how ICL could be related to various views of Advanced Process Control (id est Model Predictive Control[1] ). The purpose of this note is to define an Idiomatic perspective for MPC.[2] In so doing it offers a simple example of using an established practice to translate a traditionally implementation oriented control technology into an Intent based perspective naturally fitted to ICL. ICL in its nature is not incompatible with MPC/APC:

- MPC can be an Idiom. The freeing of Idiom support algorithms from a strict single Block context allows any kind of computational control model at the level of implementation.
- MPC can be operated above an ICL application, implemented in a separate environment and passing setpoints to the ICL system. This approach has the advantage of allowing integration with an existing MPC package.

Each of these approaches is an adequate basis for an ICL integration. But in spite of MPC's higher level positioning, it addresses functions already covered by traditional controls represented by normal Idioms. ICL, if successful, will accomplish a revolution in Intent based control design clarity and integration. Because it will then be the basis of a long lived evolution, we need to get the basic issues right. Part of the problem in relating MPC to Idioms, as was the case with traditional controls, is that committed users focus on the appealing implementation technology with only implicit concern for the benefits to be realized. But the whole goal of Idioms is to express designs in terms of their Intent and away from implementation.[3] We need a better definition of the distinct intended MPC benefit over the traditional techniques. This note hypothesizes a likely MPC Intent model only incidentally related to the MPC technology.

The emphasis on dynamic control distinguishes MPC from Linear Programming and other well defined optimization techniques whose role in defining control operating points for lower level control is clear and distinct. The following discussion shows an approach to making MPC more fundamentally operational and compatible with the ICL Idiom paradigm. But first, to set the stage, MPC has the following apparent advantages:

- A self tuned approach to integrated multivariable controls which are otherwise difficult for most people to tune.
- The integration into such an approach of general constraint control.

On one hand my concerns arise partly from knowing both simple and sophisticated means of accommodating the same problems with the traditional simpler technology. But on a more fundamental basis they arise from the following problems in any MPC current implementation and positioning:

- MPC controls the modeled process based on a single centralized mathematical calculation rather than using the normal incremental engineering design approach reflected in the Idiom Loop Statement.[4]
- One of the consequences of this is that an MPC application represents a monolithic design whose detailed application consequences are hard for the operating people or traditional process control

---

[1] Or Dynamic Matrix Control (DMC, originated by Shell), or IdCom (Adersa Gerbios), or Internal Model Control or a number of other variants due to other authors or organizations.

[2] I will use the more specific term MPC over the more ambiguous APC term because "Advanced Control" has been used to mean anything including the feedforward practices covered by the Idioms. At the risk of insulting the purists, by MPC I will mean the general combination of multivariable controls based on:

- Use of convolution sum process models included in the controller structure.
- Control based on a predicted projection from that model.
- Use of that predictive computation to predict and accommodate constraint violations.

[3] Idioms were invented because the author noted this same confusion in traditional control documentation.

[4] Ironically, in my long term work environment the only two of us with the mathematical background to implement MPC, shared the engineering suspicion of it which will be resolved below.

experts to understand.[5] I will argue below that this is a matter of good design rather than simple mathematical ignorance (given that that is operating as well).

- As a consequence, operators tend to deactivate MPC entirely when problems arise (even though it can be operated in a tracking mode which allows it to consistently follow the process and operator actions) and to be turned on (if with deep reservations) at any time appropriate.
- All of this makes it hard for the operating personnel to adjust MPC to changing or partially operational conditions.
- Unlike the practice background of the Idioms, MPC is generally restricted to a linear model, complicating its use in situations with widely varying operating point.
- A powerful technology that we commercialized allowed traditional controls to be adapted on line. The inability of other manufacturers to develop equally effective adaptation (and the original manufacturer to capitalize on its advantage) has lead the industry to go to the kind of off-line tuning which is a natural part of MPC, but which leads to less effective tuning.[6]
- Experience with MPC and its variants often finds its control responsiveness inferior to traditional controls, particularly in disturbance rejection. In a constructive way this is reflected in the practice of using traditional controls under MPC despite the apparent redundancy of control technologies.

Notwithstanding all of this, no process controller designer can reasonably ignore the user interest in MPC. If the user wants MPC he should be able to have it. Successful integration in an Idiom/ICL context still requires a fundamental understanding of a distinctive implied user-added-value.

## MPC Analysis as an Example of Intent/Idiom Modeling

While looking at the immediate problem it is worth considering this as a specific example of the general situation in which control techniques are usually approached in implementation terms to the exclusion of the kind of Intent modeling that makes the Idiomatic notation so much easier to work with. To begin with, the mathematics of MPC that expresses control and constraint targets relates to the end control as the corresponding practices of the PID relates to its nominal purposes. So there is an explicit practice that can be analyzed in terms of Intent. But as with the PID, the practitioners so take the Intent for granted that it are hidden behind the implementation mechanics. In addition practitioners obscure their Intent. For example one user view of the use of MPC was to support the operator function. What does this mean when the operators can't understand it? The problem addressed in this note is to identify a real Intent value provided by those MPC applications, that mixes this technique with the traditional techniques, and suggests how that value can be more operationally accessible and explicitly documented for the user.

## MPC Intent Resolution

The fundamental barrier to operational integration of MPC in the spirit of Idioms is this monolithic character which prevents the operating people from continuing to operate with MPC through different constraint violations and failed process conditions. Because MPC represents a single mathematical approach, it is seen as a single thing, not something easily subject to piecemeal understanding and operation. In the past, as a matter of trying to relate MPC to an appropriately operational alarm strategy, I proposed that MPC be discussed in terms of different classes of application which could be thought of as distinct Idioms supporting distinct appropriate alarm and operating strategies.

[5] MPC practitioners are trained to work with a Convolution sum (Green's function) style of modeling rather than the simple material and energy balance differential equation models more fundamental to chemical engineering. As a result they usually don't have access to the deeper process thinking possible otherwise. A first exposure to effective MPC was an Adersa Gerbios refinery application in Le Havre. The lead engineer was a physicist with no other process control experience. The operators were paid a premium for using the IdCom (their MPC). The application appeared unquestionably successful but better than if based on the process engineering?

[6] My ISA'70 paper, "Adaptive Control Odyssey" shows how to do adaptive control within the usual MPC modeling mathematics! To my knowledge, no one else has accomplished anything similar since.

The recent thought goes further in a way perhaps still more easily engineered. The strict multivariable regulation of MPC corresponds to traditional decoupling. The elaborate multivariable constraint control is more difficult to match traditionally. But operationally it switches the process through different constraint regimes. In order to function effectively operators should be trained to think in terms of the different regimes, to understand what is normal and what is abnormal in each regime, and to operate appropriately in the face of such changes. This requires that the engineers identify the different regimes explicitly (and in the operator interface) rather than let the process slide invisibly from regime to regime as the MPC hits its different constraints.

One of the arguments for traditional constraint control is that, despite the many different possible constraint regimes theoretically possible, most processes normally operate only against one or two of the many potential constraints. But this argument can also simplify the MPC operation because there are then only a small number of constraint regimes that need to be identified and accommodated.

The traditional constraint control is translated in ICL in terms of Hi and Lo Constraint and Limiter Idioms. Thus the identification of MPC regimes is the analog of identifying the different Constraint or Limiter Idioms. It then does not matter (once operators are trained and once the human interface is designed to support regime identification) that the implementation of the MPC regime changes is handled by mathematical "sliding" rather than by the operation of distinct mechanisms.

## An ICL Counterpart Strategy Based on Traditional Control

In principal, traditional controls can implement the equivalent of MPC constraint control by a complex decoupling control which incorporated both regulation and constraints. The result would be complex indeed when combined with the intended adaptive tuning, even though ICL's wider computational freedom could conceal this. But ICL offers another simpler way of recognizing distinct constraint regimes. It is explicitly designed to allow Idiomatic controls to be activated conditionally and to be transitioned bumplessly when a different condition is encountered and a different control structure is invoked.[7]

This leads to a different approach:

- Identify the distinct control/constraint regimes of interest with corresponding operational State names and human interfaces.
- Design and implement a distinct (simple, traditional, optimal, linear or nonlinear) control design for each regime.
- Design and implement the conditional tests which identify each regime and switch in the appropriate control structure.

On one hand this strategy seems to continue the competition between the old and the new. But in fact it does two more important things:

- It poses the question, "What is the real value of MPC?", both in the abstract, and by posing an alternative way for accomplishing the same thing, ensuring that the problem is independent of the solution.
- And it suggests ways of mixing traditional and MPC strategies taking advantage of the strengths of both.[8]

## Conclusions: MPC/APC as High Level Dynamic Constraint Regime Management

The PID and MPC terms both express an implementation not reflecting or limiting Intent use. Regulate and Constraint Idioms have been defined to reflect (usually) PID Intent without precluding other uses. Similarly, it seems reasonable, based on user behavior if not his statements, to postulate Constraint

[7] Traditional blocks implementation implies this possibility but handles bumpless transfer of entire structures, clumsily if at all.

[8] Traditional is simple, while MPC automates multivariable setup and tuning.

Regime Management (CRM) as the principal Intent behind actual MPC applications. Such a positioning, when combined with appropriately identified regimes, human interfaces, and operator training, could overcome the operational concerns of MPC.

The result of such a positioning allows an integrated MPC/ICL positioning based on either of the proposed strategies:

- Operate MPC/APC above ICL as an overall Regime Management tool, specifying high level targets implemented by lower level ICL controls.
- Include MPC/APC in (Large Language) ICL as an (composite CRM) Idiom perhaps coupled with other ICL programming of traditional control based CRM.