Petrocontrol

Advanced Process Control and Optimization

WHAT'S WRONG WITH
UNIT CLOSED LOOP OPTIMIZATION

Y. Zak Friedman, Ph.D
Principal Consultant

34 East 30th Street, New York, NY 10016 • Tel: 212-481-6195 • Fax: 212-447-8756
zak@petrocontrol.com

WHAT'S WRONG WITH UNIT CLOSED LOOP OPTIMIZATION.

On-line, closed loop unit optimization is the latest hype. Models sell for hundreds of thousands of dollars and come accompanied by big claims. Being an independent consultant in the field of advanced control and on-line optimization, I have been asked by several clients to evaluate on-line optimization products. Unfortunately the evaluations have shown those claims to be quite exaggerated. While closed loop optimization can benefit refiners, the products I looked into would not be able to capture those benefits. The purpose of this paper is to highlight the problems, in the hope of helping simulation vendors come up with reasonable on-line optimization products.

CAN A UNIT BE OPTIMIZED IN ISOLATION?

Considering that the global optimization problem encompasses the whole refinery and it usually also spans over time, a single unit optimizer covers only a local subset of the problem. As such it would not necessarily supply a valid answer. Still, there is an argument in favor of unit optimizers: Modeling of the complete refinery is practically impossible. Some units will never have good models, and in any case we would always have a situation where one model out of several is in need of repair. We better come up with a way to take advantage of local optimizers or we have no optimization at all.

How then can we make sure that the local optimizer comes up with a correct solution? There is only one way: Specify the correct unit product prices. Once product prices are known, the unit optimizer can determine how to maximize profits. But unit products are not final refinery products, and specifying their prices is easier said than done. Intermediate product price is a function of quality and flow. It depends also on tankage considerations, final product lifting schedule, on the whole product pool sitting right now in intermediate tankage, and finally on the crudes to be run in the near future. In short, the problem of pricing intermediate products is a scheduling problem.

In discussing these problems with one refinery team, they thought up a possible way to estimate intermediate product prices. Since they do not have a scheduling package they can trust, how about using shadow prices of the refinery LP to define the intermediate product prices? We then went on to test the shadow prices against individual scenarios to check whether this would be a reasonable idea. It turned out that the refinery LP does not take into account the content of intermediate tankage. The planner runs the LP. The scheduler does his best to comply. When he runs into a situation where the intermediate tankage quality is high, the scheduler tries to make more of the high quality product. If there is no demand for the better product he simply goes into a giveaway mode. At times when the blending pool is short on quality, he tries to reschedule the liftings, rearrange crude runs, or purchase a high quality intermediate product. If that cannot be done he can try to change the purchase plan, though practically his influence there is limited.

As long as there are no scheduling problems and the original plan is being followed, LP shadow prices reflect true refinery economics. But when the operation shifts to different product qualities or a different schedule, the shadow prices are not valid. The refinery in question could use LP shadow prices about half the time.

We then went through another mental experiment. We considered the product blending pool at times of scheduling distress. It would be easy enough to come up with a blending LP optimizer that just looks at the gasoline or diesel pool. The intent here is not to physically control the blending operation (though that can also be done), but only to determine the value of blend components and their quality premiums, which would be the shadow prices of the blending pool LP. It turns out that the shadow prices of such an LP would almost always be useful, particularly if we also take into account current rundown flows and qualities, and their ability to relieve the problem or make it worse.

We conclude that the problem of intermediate product economics is not insurmountable. It may take book keeping to determine rundown qualities and blend rundown streams into tanks, but our ability to predict rundown properties would improve with the introduction of models, and once rundown qualities are known, tank quality monitoring becomes a fairly simple piece of software.

We propose that anyone who wishes to implement local unit optimizers first think about how to set the intermediate product prices for these optimizers. Perhaps simulation vendors do not view themselves as suppliers of software for determining these prices, but the fact remains that most of their claims depend on the existence of such software.

RECONCILIATION OF MODEL AGAINST INSTRUMENT READING.

On-line optimization begins with on-line simulation. We will assume for now that the simulation is reliable and can indeed reproduce the unit operating characteristics. Someone need only type in feed properties, set the model in motion, and two minutes later we would see how the simulation can duplicate all of the unit measurements. Or would we?

Assessing the properties of feed to any refinery unit proves a difficult task. To begin with the crude, no one knows precisely what crude is flowing into the crude unit at any given time. Even if people knew roughly what is in the crude tanks, tank layering, natural crude variations and the addition of slops make it infeasible to predict properties. Dynamic prediction of when crude in the header or pipeline will hit the unit is even more difficult.

And is it easy to predict PNA (Paraffin, Naphthene, Aromatic) content and boiling curve of reformer feed? Reformer feed comes from several sources: The crude unit with its unknown crude, perhaps also purchased naphtha of unknown origin, then coker naphtha, hydrocracker naphtha, wild naphtha from hydro treaters, all in varying proportions. “No” must be the answer. It is quite impossible to predict the reformer feed properties.

What about FCC feed? It comes again from the crude unit with its unknown crude, perhaps vacuum gasoil purchases, then coker gasoil, and possibly a variety of other sources. We note that it is rather difficult to characterize FCC feed. Complex measurements of Nickel, Vanadium, Sulfur, Conradson Carbon, boiling curve, UOP K factor and others are needed. Resid FCC feeds are even more complex to characterize. It is improbable that we would ever be able to identify FCC feed in detail enough to match model against instrument readings.

What then would happen to the simulation results if the feed properties are incorrectly assessed? The model will come up with key temperatures and other parameters, different from instrument readings. In that case, how would the operator treat the optimization results? At best he would ignore the optimizer. At worst he would follow the optimization results and make off-specification product.

We draw one important conclusion from this argument: Before optimization begins there must be a comparison of unit instrument readings against model. This comparison must be displayed to the operator.

Better yet, on-line models should come equipped with a mechanism for modifying the feed properties to match model against instruments. Once this principle is accepted, one can think also of other parameters which might be adjusted: catalyst activity, tray efficiency and heat exchanger fouling, particularly if they tend to change quickly.

Of the several on-line optimizers under review, only one has the facility to modify feed properties to better approach instrument reading. Another one has a more limited facility for modifying secondary parameters like: catalyst activity, fouling factors and tray efficiencies to obtain the best fit. We judge this partial approach inadequate because as stated above, the main uncertainty is feed quality.

The rest of the products completely ignore the problem. Some do not even attempt to reproduce the current operation. They simply create an array of setpoints called “optimal” and advise the operator to change from current to optimal.

MATCHING THE DEGREES OF FREEDOM.

Assume for now that we can address the incorrect feed problem and create a match between model and unit instrument readings. Is that enough of a condition for the optimizer to produced a good solution? The optimal solution comes in the way of say ten setpoint changes. Suppose that for some reason one of the setpoints cannot be changed. Should we go ahead and implement all other nine changes? That could lead to a disaster. For example we would increase reactor firing without increasing the feed and cause severe coking.

There are many reasons why one out of ten setpoints may be stuck at any given time. Perhaps a valve is simply stuck and is in need of maintenance. Possibly a piece of equipment is not operating normally, and the operator is afraid of moving anything that would disturb it. What if a valve saturates, instrument tuning causes instability, or a request came in from a downstream unit to keep its feed steady?

To produce a valid optimal solution the optimizer must read and interpret real degrees of freedom on the unit. This involves respecting control modes such as Manual, Automatic, Cascade or Computer. Recognizing a degree of freedom in one direction is also important. If a manipulated variable can only increase, the optimizer should not call for decreasing it. Where advanced controls exists, the optimizer interface may exhibit even more complexity. The advanced control schemes may include inferential controllers, calculated constraints, dynamic considerations, etc. The interface between advanced controls and unit optimizers must be very well designed for the optimizer to work correctly.

It came to us as a surprise, but only one of the products in question attempt to identify which process "handle" is a real degree of freedom and which one cannot move. This optimizer reads control modes of instruments to see whether a secondary controller would accept an external setpoint. The other products do not bother with such details.

With respect to constraints, most models derive them from calculations based on given (hardly known), feed properties and given equipment. Some constraint measurements are being input into the optimizer, but on the whole there is no systematic attempt to verify which equipment is constrained right now.

We must conclude that of the products we have seen, only one attempts to solve the right problem.

THE STEADY STATE WAIT.

All of the models in question employ steady state simulations. To the extent that they need to read unit measurements we must first, clean the measurements from process noise, and second, make sure that these reflect a steady state of the unit. This becomes even more important when we insist that the model agree with measurement readings.

However in reality disturbances come in all the time and no unit is ever at steady state. How then do we create a set of measurements that are in phase enough and would not mislead the unit optimizer? The simulation industry responds to this problem by heavy filtering and long waiting. Complex criteria are checked to determine that the unit has reached steady state, and in a typical situation one must wait three hours until these criteria are met.

This way of making an optimization run once every three or four hours would not be effective. We cannot hope to implement any optimal solution in one sweep of setpoint changes. It would take several runs of the optimizer and many hours to get us there. That would make it impossible to catch up with day - night cycle and normal weather changes, let alone operational switches. The optimizer would always be out of phase with crude runs and intermediate tankage economics.

A question presents itself here: Over the years the advanced control industry has struggled with dynamic control problems. We now have tools for predicting the eventual steady state of a plant from a set of measurements and history of manipulated variable movements. Why is it necessary to wait for some official steady state? We would do much better to feed the optimizer not with current unit measurements but with a predicted set of steady state data. This would permit running the optimizer at any time, even minutes after changing manipulated variables.

THE SOLUTION TRAJECTORY.

Suppose that all of the problems discussed so far were solved. Intermediate product economics are available; Models are able to reproduce reality; Degrees of freedom are correctly identified; Steady state measurements can be forecast; And the optimizer has produced a valid solution. How do we go about implementing this solution? Move the unit in one quick sweep from current to optimal conditions? The operator would not permit sudden stepping of many setpoints.

We must come up with a trajectory, and slowly implement the solution in small steps and in such a way that there would be no constraint violated during that process. Each move on this path should economically improve the unit operation. The optimizer must be able to produce this series of small steps, and then monitor and improve the solution as the unit comes near its optimal operation point.

The simulation products we tested could not however accomplish this feat. First, the optimizers were not searching to produce a path. They aim only to identify the ultimate optimum. Second, there is no constraint monitoring and protection control. Third, these products take so much time to get a set of steady state data that stepwise implementation of the solution is impractical.

The section above has discussed a way to forecast steady state and eliminate the wait. The section below will discuss protection from constraint violations. If we can handle these problems successfully, is there a way to make the optimizer produce a path and gradually improve the operation with each step?

This seems doable. We can limit the range of manipulated variables to small changes from current positions. Solving the optimization problem over a reduced operating envelope is easier than finding the ultimate solution. The optimizer will quickly come up with small improvements every five minutes, and within possibly one hour we would be working very close to the real optimal positions of all manipulated variables.

ADVANCED CONTROL INTERFACE.

We have already considered one of the difficulties associated with steady state unit models. Input data must represent a steady state situation. This section addresses another problem: outputting. Just as the input data must be dynamically manipulated to be in phase, outputting, in the form of changing an array of setpoints, must be timed, taking into account dynamic responses of the unit to different manipulated variables. Otherwise constraints will be broken on the way to optimizing the unit, and this type of optimization would be very costly.

To illustrate a simple example of how this can lead to a disaster, consider an exothermic reactor cooled by a heat exchanger. The optimizer may come up with a correct steady state solution calling for reduction of heat removal, but incorrect timing of that reduction would cause a run away reaction.

We remind the reader that there are devices whose mission is to control the unit just below constraints. Advanced control is the name of these devices. Why not use the advanced controls of a unit to assist bringing it to a new steady state without violating constraints? This approach was adopted by one of the products. It outputs the optimal solution as advanced control targets. The advanced controls would only meet these targets if the unit can be kept within constraints. We consider this the minimum acceptable approach, though would much prefer a step by step implementation of the optimal solution, with a reasonable trajectory as discussed by the previous section.

The other products were much worse. Overlooking all unit dynamics, they simply ramp in a so-called optimal array of setpoints. Once this is done they go to sleep for three hours, waiting for the steady state criteria to read positive, not monitoring any constraints during the waiting time.

Clearly, optimization products that ignore unit dynamics would be too dangerous and unpredictable under closed loop control.

We have asserted that optimizer inputs must come via the advanced controls to make sure they are in phase. The present argument calls for unit optimizers to output their solution not directly to manipulated variables but through advanced constraint control schemes to make sure the outputs are timed correctly. Outputting can be in the form of turning advanced control functions on or off, changing clamp values or setting control targets. The important thing is that optimization and advanced control must work together, not ignore each other.

WHAT IS AN APPROPRIATE LEVEL OF DETAIL FOR THE SIMULATION?

There are gray areas of uncertainty with respect to how detailed a simulation model should be. It is readily accepted that any simulation must predict product qualities, but what about secondary parameters such as pressure drop, heat exchange, hydraulic constraints, etc.? Is it necessary to model these accurately?

The answer depends on whether the effect, to be modeled, influences unit economics. A hydraulic constraint that is not at its limit has no influence on the economics and would be a waste of time to model. But if that constraint stood in the way of increasing throughput - it would become quite important to model correctly. Heat exchange, to the extent that it influences unit efficiency, does not typically need precise modeling, but if the unit is against a furnace constraint, heat exchanger modeling becomes crucial.

In discussing this problem with one of the vendor engineers, he brought an argument in favor of having only a simple representation of secondary effects. Since we are in an on-line environment, he said, we are in a position to measure constraints and there is no need to accurately model them.

To this I have the following answer. Nearly any control logic, including operators working manually, is capable of bringing the unit up against one constraint. The added task of an optimizer is to glide along the multi dimensional constraint envelope in a profitable direction and to meet two or three constraints at a time. Secondary effects, in particular response of constraints to changes in manipulated variables, must be modeled with high accuracy. Otherwise the optimizer would not be able to slide on the constrained surface in the correct direction.

Most optimizers investigated here did not meet the requirement of modeling secondary effects. The best of them had only superficial hydraulic and heat exchange models. Heat exchange models had predetermined fouling factors and they could not correctly balance among unit throughput, product yields and heat recovery.

We assumed at the beginning of this paper that we have perfect models, and here at the end we assert that the models are far from being perfect. Are they still usable in any form? The answer is: yes, somewhat, but do not expect miracles in the way of approaching more than two constraints at a time.

CONCLUSIONS.

We have reviewed several simulation products, being sold under the name: Unit On-line Optimization Models. There are many deficiencies in these models. We can classify on-line optimization difficulties in four categories:

- ➢ Local versus global problem set up;
- ➢ Inability to define the unit feed;
- ➢ Steady state optimization in a dynamic environment;
- ➢ Insufficient level of detail in the models.

The paper has proposed ways to solve the difficulties. In some instances the solutions already exist and all that needs to be done is to incorporate the known technology into the product. In other instances the solutions need to be developed. Of the technology that needs to be developed, two items deserve special attention:

- ⇒ We have yet to find a reliable way to determine intermediate product prices. Without these prices on-line optimization is questionable.

- ⇒ We must develop a method for changing a-priory assumptions of feed characteristics to match model results against key unit measurements. Optimizing the unit cannot begin until we know what feed is being operated on.