Final report - Systematic Creativity and TRIZ basics Online

Project work of Dang Doan

1) Introduction your system (Chapter 2)

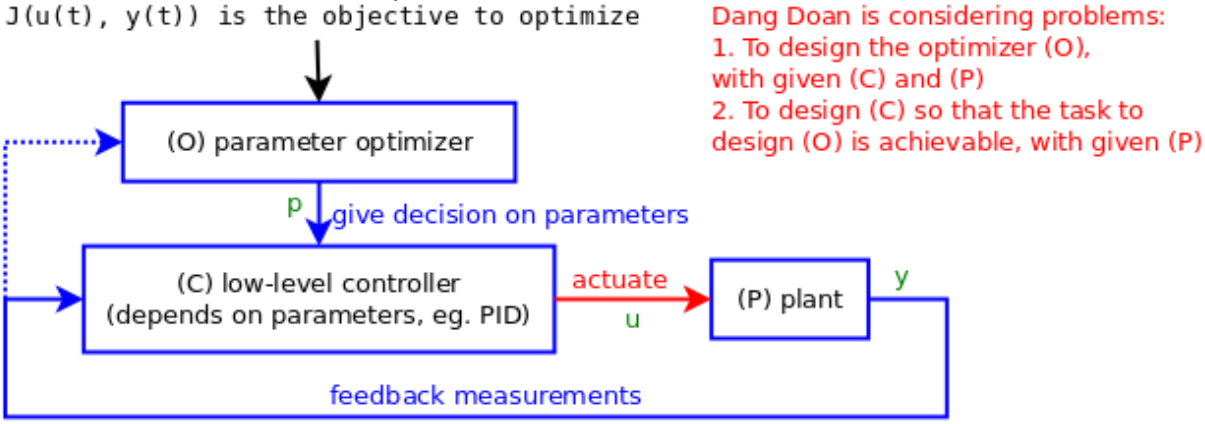
I intend to design a numerical algorithms to solve an (abstract) optimal control problem: the "system" consists of: + a dynamical system, let say we have exact deterministic model written in mathematical formulas + there is a low-level controller attached to that system, for example a proportional feedback controller (with enough sensors and manipulators, ie. hardware is sufficient) + there are some constraints that the controlled system need to satisfy, which will have to be taken into account by the high-level controller + objective: to find the types of the low-level controller so that we can predict how the controlled system function in a near future (called "horizon"), in such a way that constraints could be satisfied in the horizon by solving a TRACTABLE optimization problem at the high-level controller. (in short, with the terms in the field of control that Leonid should be familiar: combine a predictive controller with a low-level controller).

An example: we control a damp by pulling a gate upper (allow more water to pass through in a time unit) or lower (allow less water flow rate), the problem is that somebody has already put a PID controller for the gate, so that we can only tune the Ki, Kp, Kd parameters of that low-level controller. (hence the high-level controller is called "parameter optimizer"). Auto-tuning PID was studied before, however if there are constraints such that the water level (corresponding to accumulated water discharged from the damp) should be bounded at some time window, then the optimization problem could be difficult to solve (we call it "intractable").

2) Patent search for your system (Chapter 3)

I just use patents.google.com to search, putting some keywords and browse the titles of patents to find the ones that are relevant. For example I found a patent on a method from the field of process control: https://patents.google.com/patent/US6038540A/en
It is just abstract description of methodology, even not a software, nor details about how to really implement the "apparatus" as it said: "an apparatus facilitating the implementation of a useful process such as the manufacture of chemicals".
Some other patents on methodology that I found with the same search, they include more details on the methods to be patented:
Adaptive-predictive control and optimization system - https://patents.google.com/patent/US5841652A/en
Strategic planning and optimization system - https://patents.google.com/patent/US6988076B2/en

3) Function definition (Chapter 4)



Dang Doan is considering problems:
1. To design the optimizer (O),
with given (C) and (P)
2. To design (C) so that the task to
design (O) is achievable, with given (P)

Model of (P): y(t) = f(x(t), x_dot(t), u(t)) Note: f(.) is continuous, first-order differential equation
Model of (C): u(t) = g(y(t), p(t)) Note: g(.) is an analytic formula (simple, fast to execute)
Model of (O): p(t) = h(y(t), g(.), f(.)) Note: h(.) is implicit, e.g. by solving optimization problems

Plant (P) is a dynamical system that is governed by differential equations associated with "state" x.
Controller (C) directly controls (P) by giving control signal u in order to change the output y.
Optimizer (O) adjusts the parameter p (which affects how u is computed by the model of (C)),
in order to achieve some optimal value J related to y and u.

Function 1: u changes y via (P)
Function 2: (C) changes u
Function 3: (O) changes p

4) FOS for your system (Chapter 5)

The main challenge of my problem is to organize the procedure for making decision so that the computational task to find the optimal performance of the overall system is considerably low.
The main function should be function 3: change the parameter with knowledge of the other parts of the system.
Leading area (when solving the problem is crucial) should be the area of this problem (control and optimization). So I guess it is difficult to find the solution from other fields.
Within this area, there were effort in process control. However as the patents show, the methods described in those patents are superficial. Often in this type of research, we need to read publications and try to derive new methods / algorithms with the techniques distilled from other papers.
At this moment, I couldn't find good progress from function oriented search.

5) Biomimetics application for your system

I will need to find more about nature behaviours that aim to optimize some parameters of a function.
In the field of optimization algorithms there were some inventions inspired by the nature: Ant colony optimization, partical swarm optimization, bee colony (as given in the lecture).
I found that the growth of trees is also a kind of optimization, or self-control toward the best resources. The trees has some specific bio-chemicals that focus the growth in the direction of sunlight. Or it can be expressed as: the developing cells of the trees are activated by sunlight.

Using this phenomenon, the idea for making a new optimization algorithm vaguely appears: there are some "developing cells" to be used in the algorithm, that show / mark / signal the improvement of the algorithm on the way toward the optimum.

6) IFR concept (amazing concept)

For my system of finding optimal control algorithm: IFR means that there is no system but the function (improving the control performance) is achieved.
To reach the IFR, we can look for the direction "no need for function", then the guideline is: No computation, no need to do optimization, but the controlled plant already has optimal performance.

Another direction to reach the IFR is "the function is performed by the super system". The super system of my system could be the class of control systems that generalizes the structure. In order to "let the super system perform the function of the considered system (optimizing performance)", it may require that: every member of that class is already at the optimal condition. We may go to some abstract thinking: that class is an algebra, in which any single element is "by default optimal", and there are operations using elements in the class that preserve the "by default optimal" property (think of "group" in the mathematical sense); if a system falls into that class, we don't need to do further improving.

7) Contradictions

Technical contradictions:

1. If the robustness of the algorithm is improved, then its speed is lower (worse).

2. If the accuracy of the optimizing task is increased then the performance is improved, but the time needed to solve is longer.

Physical contradiction: The optimization must take place (to achieve good performance) and must not take place (to save time).

Solving the technical contradictions:

1. Search matrix in triz40.com for:
To improve : Reliability
To preserve : speed
-> principles : 21, 35, 11, 28
21: Skipping
35: Parameter changes
11: Beforehand cushioning
28: Mechanics substitution

2. Search matrix in triz40.com for:
To improve : Adaptivity or versatility
To preserve : Loss of time
-> principles : 35, 28

35: Parameter changes
28: Mechanics substitution

Skipping: Conduct a process, or certain stages (e.g. destructible, harmful or hazardous operations) at high speed.
In optimization field: Taking a search with 2nd-order derivative is like skipping

Parameter changes: Change the degree of flexibility.
(no idea yet)

Beforehand cushioning: Prepare emergency means beforehand to compensate for the relatively low reliability of an object.
In optimization field: Tighten the optimization problem so that there is an offset to compensate to the inaccuracy of the solution

Mechanics substitution: Change from static to movable fields, from unstructured fields to those having structure.
In optimization field: problem is changed from without structure into some type of structure (so that the structure can be exploited, e.g. sparsity)

Solving the physical contradiction: using the Separation in Time, the optimizer needs to do optimization (time-consuming action) at the “idle time” (when waiting for the updated value from the controlled system to come in the next sampling time). This idea could be related to a principle in the 40 creative principles: Preliminary action.

8) TESE for your system

Using the trend of increasing dynamicity: any new value of the controlled system will be quickly handled by the whole system. (optimization to happen quickly and incremental).

Using the trend of increasing ideality: the system is closer to the ideality, where there is little need to do any computation but already at optimal performance. (maybe: the computation only needs to be done before the control process starts, and then the performance is optimal as inheritance from the past).