Summary of "A hybrid steepest descent method for constrained convex optimization"

In the paper under discussion the authors have proposed a hybrid steepest descent with progress of time to minimize a convex cost function. Here the convergence to a global minimum is proposed using Lyapunov stability arguments. The purpose of this paper is to give an insight into how to solve the optimization problem with less computational overhead so that it can practically be fed into embedded controllers.

The authors have identified three main approaches to dealing with optimization. Firstly, Repeated Optimization, which ensures a new optimization every time step. Although this approach is suitable for offline optimization, like selecting tuning parameters or designing feedback controller, but definitely not suited for online optimization due to its repetitive nature. Here online optimization refers to a mathematical model used to calculate optimal inputs based on model and recent outputs repeatedly. Secondly, Pre-computed Optimization, in which all the optimization problems are solved offline and stored in a lookup table which can become large. Thirdly, Update Laws, where optimization of the states of a dynamic system are taken into consideration. Although Update Law is computationally less intensive, it is works well in an unconstrained case. Here the author tries to demonstrate a method using the concept of Update Law and dynamical systems, which solves convex cost function with constraints and which is suitable for the discrete-time implementation. Hence, the proposed method is suitable for implementation in the controllers.

In the paper the discussion starts with stating the Ordinary Differential Equation: $\dot{x} = f(x)$ where f is a vector field designed to solve constrained convex optimization problems. Only time invariant problems are discussed in the paper for simplicity. The authors state the standard convex optimization problem: min q(x) s.t g(x) <=0 **…. (1)** with x ∈ R^n, q: R^n → R^n and g: R^n → R^m where g_i is a differentiable convex function with a constraint g(x) <=0. The authors make two assumptions. Assumption 1 states that the feasible set is not empty, i.e. for each x, g(x) < 0 and the optimal value of the cost function lies in the feasible set denoted by q*. Assumption 2 states that q* is finite. The author strives to find a vector function f: R^n → R^n such that the dynamical system $\dot{x} = f(x(t))$ **…. (2)** has three important properties. First is that the value x(t) lying outside a feasible set definitely enters the feasible set at a later time. Second property is that x(t) remains in the feasible set once it enters into it. Third property states that for x(t) in the feasible set the cost function decreases with progress of time until q(x(t)) = q*. These properties state that q* is an equilibrium point which is eventually obtained and hence, finite.

Gradient Descent is an example of Update Law which aims at minimizing the cost function which is unconstrained. In the constrained case the gradient may be projected on the active constraints when the state is on the boundary of the feasible set. This may work in continuous time but difficult to implement for the discrete-time implementation. Therefore, the projection of gradients on constraints is discarded. One of the novelties of this paper is to consider the constraints as a barrier. Now, each constraint forces the trajectory to be in the feasible set and if the trajectory is on the boundary of a feasible set it will be pushed alternatingly by the constraint and gradient. Also, if the gradient and a constraint push in opposite directions the trajectory slides along the boundary of the constraint and hence gives the projection indirectly. Advantage of this method is that the directions of gradient are defined outside the feasible set and works well with time-varying constraints, although this paper confines itself to discussing only time invariant optimization.

The authors proposed a hybrid feedback law: f(x) = { $-\nabla q(x)$ if g_i(x) <=0 ∀j; $-\sum_{i\,\epsilon L(x)} \nabla g_i\,(x)$ Ǝ j g_j(x) >0 with L(x) = l: g_l(x) >0} **…. (3)** It follows from $\dot{x} = \gamma_0(x)\nabla q(x) + \sum_{i=1}^{m} \gamma_i(x)\nabla g_i(x)$ that the trajectory can either follow the boundary between feasible set and infeasible set for a particular value of γ_i(x). It moves towards the feasible set if $\gamma_0(x)$ = 1 and towards infeasible set if $\gamma_j(x)$ = 1 ∀j ϵ L(x). It also follows that if x = p is a stationary point of the dynamical system then it lies on the trajectory. The authors show through theorems that If (1) is feasible then a point p is a stationary point of (2) and (3) if and only if it is an optimal point of (1). This is equivalent to saying from $\nabla q(p) + \sum_{i=1}^{m} \lambda_i \nabla g_i(p) = 0$ …. (4) where λ_i = $\gamma_i(p)/\gamma_0(p)$, the Lagrange multipliers.

Next the authors propose Lyapunov function V(x) = max(q(x), q*) - q* + $\beta \sum_{i=1}^{m} max(gi(x), 0)$ where β is strictly positive value. This function is strictly positive everywhere except at equilibrium point where V(p) = 0. This function is analogous to equation 4 discussed above. The gradient of V(x) is shown to form a convex hull in the vicinity of discontinuity with measure zero. In case of a smooth V(x) it reduces to a time derivative. Having said that, the gradient of V(x) is always negative (feasible set or feasible set) until x(t) reaches q*, the equilibrium where the gradient is 0. This gradient increases past the stable point in the feasible set and decrease in the infeasible set when moving away from the stable point. Large enough value of β determines how much the constraints dominate over the cost function and create a barrier. Also, V(x) is shown to be uniformly asymptotically stable as it has a unique finite stationary point and the gradient descent leads to this point. Hence by putting forth the two assumptions, definition and theorems the author could prove that dynamical system (2) and the hybrid feedback model based on the Lyapunov's function (3) allows to find the stable point to minimize cost.

In order to discretize the system, the sampling is done by approximating the derivative by a forward different in sampling time Δt. This is given by $x_{k+1} = x_k + f(x_k)\,\Delta t$.
A simulation example has been developed to illustrate the system so developed. The problem is as defined:
x = $(x_1\ x_2)^T$; $\min_x$ $-x_1$ subject to $\frac{(x_1-10)^2}{36} + \frac{(x_2-10)^2}{81} - 1 \le 0$ and 10/8x_1 + x_2 – 28 <=0. We have
$\nabla q(x) = (-1\ 0)^T, \nabla g_1(x) = (\frac{(x_1-10)}{18}\ \frac{2(x_2-10)}{81})^T$, $\nabla g_2(x) = (\frac{10}{8}\ \ 1)^T$
MATLAB code for simulating the example given in paper and the results are as follows:

```
x = [0 0]'; % initial points
delta_q = [-1 0]'; %gradient of q
delta_g2 = [10/8 1]'; %gradient of g2
delta_time = 0.5; %time stamp
figure(1)
axis_x = [0 0];
axis_y = [0 0];
% iterations to achieve the optimal value of x
for i = 1:5000
   if g1(x(1),x(2))> 0 || g2(x(1),x(2))> 0
      if g1(x(1),x(2))> 0 && g2(x(1),x(2))<= 0
         [y1, y2] = delta_g1(x(1),x(2));
         f = [y1 y2]';
      elseif g1(x(1),x(2))<= 0 && g2(x(1),x(2))> 0
         f = delta_g2;
      elseif g1(x(1),x(2))> 0 && g2(x(1),x(2))> 0
         [y1, y2] = delta_g1(x(1),x(2));
         f = [y1 y2]' + delta_g2;
      end
   else
      f = delta_q;
   end
   x = x - f*delta_time;
   axis_x(i) = x(1);
   axis_y(i) = x(2);
end
xlim([0 50]);
plot(axis_x,axis_y)
xlabel('x1'); ylabel('x2');
hold on

f = @(x1,x2) (((x1-10)^2)/36+((x2-10)^2)/81)-1;
fimplicit(f, [0, 25, 0, 25])
f1 = @(y1,y2) (10/8)*y1 + y2 - 28;
fimplicit(f1, [0, 25, 0, 25])

figure (2)
t = 1:0.25:500;
disp(t)
plot(t(1:1:1000), axis_x(1:1000));
hold on
plot(t(1:1:1000), axis_y(1:1000));
function g1_result = g1(x1, x2)
   g1_result = (((x1-10)^2)/36+((x2-10)^2)/81) -1 ;
end
function g2_result = g2(x1, x2)
   g2_result = (10/8)*x1 + x2 - 28;
end
function [y1, y2] = delta_g1(x1, x2)
   y1 = (x1-10)/18;
   y2 = 2*(x2-10)/81;
end
```

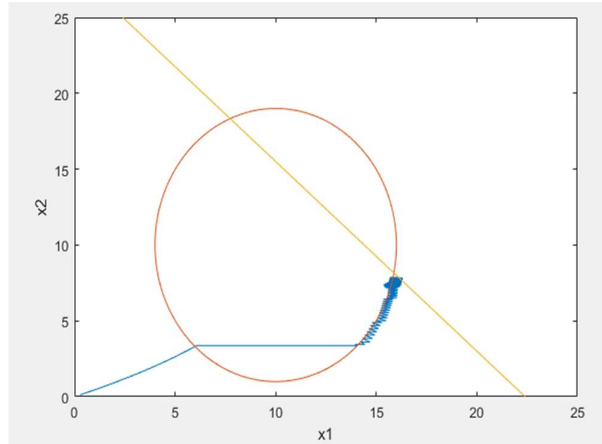

Fig1

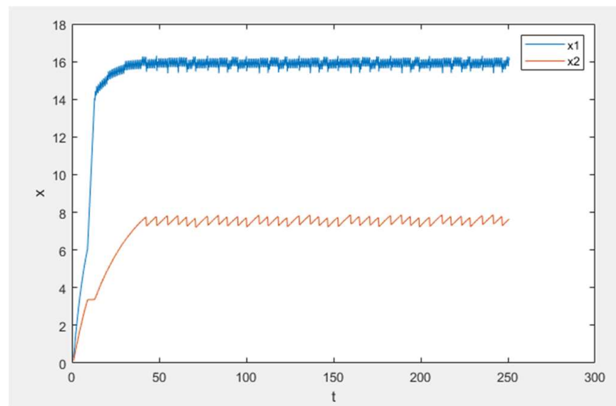

Fig2

The optimization problem given in the paper was solved using the hybrid model provided. The results confirmed the findings in the paper. The problem was to find the min of $-x_1$, which was converted to max x_1. It can be seen how the trajectory enters the feasible set and remains in the feasible set sliding along the boundary of a constraint (Fig 1). Here the time stamp of 0.5 is used. Although the trajectory to optimal point oscillates a lot more that if the time stamp were 0.01, it serves to demonstrate the behaviour of the trajectory clearly. It can be seen from the plot between x and t in Fig 2. That once the optimal point is reached it oscillates around this point till extended period of time. The paper also mentions a method to smoothen the oscillations but at the cost of slight loss of accuracy. The min of x_1 is -15.80 approximately.

In this paper the authors formulated a model to solve a convex optimization problem using concepts of optimization algorithms and dynamical systems model. The methods were verified by various theorems and definitions. By various proofs it was proved that the model achieves the optimal point by converging asymptotically and descending steeply. This hybrid model guarantees the trajectory of the variable to enter the feasible set and converge to the global optimum. An implementation for discrete time has been provided for practical implementations. Thus, it can be observed that the model so developed can easily be implemented in controllers and is computationally efficient.