A Hybrid Strategy with Simultaneous and Sequential Optimizations on an Auto-Clutch System

M.F. Hsieh¹, C.H. Tseng²

Abstract In vehicle design, optimization plays an important role in both mechanical and control design. Mechanical parts and controllers are usually designed and optimized separately and then combined for majority engineers. However, according to Fu and Mills (2002), the simultaneous optimization of both mechanical and control parts always produced a better result than, or at least the same result as, optimizing separately (sequential optimization). However, this cannot guarantee that simultaneous optimization was more practical for engineers than sequential optimization, because the relation between time taken and the end result was never considered. In this paper, the two optimization methods are compared and a new optimization method – hybrid strategy – was proposed with respect to an auto clutch system for Automated Manual Transmission (AMT). The results demonstrated that a better solution can be obtained by simultaneous optimization but a much longer computing time was required, and the hybrid strategy provided a way to obtain the same result as with the simultaneous method, but in a much shorter time.

keyword: optimization, sequential, simultaneous, hybrid, computation, efficiency, efficiency, clutch, vehicle, AMT, shifting.

1 Introduction
Optimization plays an important role in vehicle design, enabling engineers to design more efficiently and accurately. For mechatronic equipment, both control systems and controlled structures dominate performances. However, the optimizations of structure and controller are usually carried out separately, where engineers always design the system by two methodologies: mechanical optimization and active control setting (Wang et al. (1999); Hiramoto and Doki (2004)), which is known as sequential optimization. In mechanical optimization, mechanical design variables are optimized according to the design requirements. Following that, the controller is synthesized according to the mechanical model of the optimization result. However, the performance of the final mechatronic equipment is always worse than the expectation of mechanical optimization, because the added controller changes the system characteristics of the mechanical optimization model. Integrated optimization design (known as simultaneous optimization) for actively controlled mechanisms has received much attention in recent years (Obinata and Zou (1998)), and many papers have introduced it to deal with different problems, e.g. Arkawa and Miyata (1996), Obinata et al. (1999), and Ando et al. (2002). It implements optimization with an integrated system of structure and control components, which coordinates both variables simultaneously to avoid variations of system characteristics after the controller has been added. Fu and Mills (2002) showed that simultaneous optimization will always have a better result than, or at least the same result as, sequential optimization.

However, the efficiency of simultaneous optimization, which is very important in real applications, is never considered. Simultaneous optimization integrates the design variables of both mechanism and control system. Increasing the number of design variables can lead to more time consuming evaluations for derivative information and other optimization information, especially for complex and nonlinear problems (Bredelmans et al. (2001)), which in some cases is not suitable for application (Bredelmans et al. (2001)).

In this paper, sequential optimization and simultaneous optimization are compared using a case study. The result verifies that simultaneous optimization yields better results. However, it also requires much longer computation time, which in some respects is impractical compared to sequential optimization because of the time cost and

¹ NCTU, Hsinchu, Taiwan, R.O.C.
² NCTU, Hsinchu, Taiwan, R.O.C.
optimization advancement. Thus, a new optimization method combining sequential optimization and simultaneous optimization, called hybrid strategy, is introduced. The hybrid strategy is verified in this case to obtain the same results as simultaneous optimization within a much shorter time.

The object for optimization is an auto clutch system used in Automated Manual Transmission (AMT). AMT uses an auto clutch and shifting actuators to simulate Manual Transmission (MT) as an automated transmission system. AMT has become prevalent in recent years because of its high transmission efficiency and low cost (Chang (2003)). However, it has not yet been able to overtake traditional Automated Transmission (AT) because of the discomfort experienced by the user during the shifting process. Clutch control is the key point of this problem. For a more comfortable shifting process, a faster clutch disengaging speed and more stable clutch control, which are the two objectives of the optimization, are required (Chang (2003)).

Using Matlab Simulink, dynamic models of the clutch actuator and a clutch and control model of a commercial PID controller were created. Using the Matlab optimization toolbox, optimizations were implemented with the following strategies:

1. Sequential optimization  
2. Simultaneous optimization  
3. Hybrid strategy

Both optimization strategies make progress in solving the problem of comfort. Under the advancements, a comparison of optimization iteration times, function called times, and CPU time are proposed in this paper.

2 Optimization Problem

2.1 Case Description

The case for the optimization study is an auto clutch system used in Automated Manual Transmission (AMT) as shown in Figures 1 and 2. The most important problem of AMT is the discomfort experienced by the user when shifting gears. In MT, it is common for discomfort to be experienced, due to sudden acceleration and jerky movement when the driver uses the clutch. This phenomenon is particularly pronounced when the driver is inexperienced at clutch control. The same discomfort accompanies the use of AMT. For the shifting of gears to occur without discomfort when using AMT, the driver needs to be experienced at clutch control. The base requirements for such experience are to disengage the clutch quickly and to control it more stably to shorten shifting time and to increase shifting smoothness, which are the two optimization objects in the case.

Dynamic models of the clutch actuator and the clutch and control model of a PID controller for the clutch actuator were created and integrated using Matlab Simulink. The structure of the clutch actuator is shown in Figure 2. Using free body analysis and experiment data, the dynamic models and the control model were created as shown in Figure 3 (Tseng and Hsieh (2004)).
2.2 Cost Function

The objectives are to minimize the disengaging time and to increase the control stability. The problem with multiple objectives is defined as a problem with a single objective by constraining the control stability within a feasible range and setting the time to disengage as the cost function.

2.3 Design Variables

Some dimensions of the clutch actuator and parameters of the PID controller are chosen as design variables.

Taking practical feasibility and optimization sensitivity into consideration, the following parameters are chosen as design variables for the clutch actuator:

- $\lambda$: lead angle of the worm gear.
- $K_{sp}$: spring coefficient of the assist spring.
- $L_{pre}$: predeformation of the assist spring.
- $R$: clutch lever ratio.

The control function for the PID controller is shown below (Astron and Hagglund (1995)):

$$U(s) = K((b + \frac{1}{sT_i}) \frac{1+sT_d}{1+sT_i/N} Y_{sp}(t) - (c + \frac{1}{sT_i}) \frac{1+sT_d}{1+sT_i/N} Y(t))$$

(1)

Where $U(s)$ is the control plant output, $Y_{sp}(t)$ is the set point, $Y(t)$ is the feedback of plant gain output, $b$ and $c$ are setpoint weightings, $N$ is the term for derivative limitation, and $K$, $T_i$, and $T_d$ are the three PID control parameters of Proportion, Integral, and Differential. $K$, $T_i$, $T_d$, $N$, $b$, and $c$ are chosen as design variables for controller optimization.

2.4 Constraints

The optimization constraints restrict all design variables within the practicable range.

Stability is defined as Average Integrated Absolute Error (AIAE) between the clutch control setpoint $y_{sp}(t)$ and feedback of clutch position $y(t)$:

$$AIAE = \frac{\int|Y_{sp}(t) - y(t)|dt}{t}$$

(2)

During optimization, the controller setpoint is a commonly used clutch travel trajectory with weighted time at steady state, as shown in Figure 4.

2.5 Initial Design

The results from simulating the initial design are shown in Figure 5. Where the disengaging time is 0.2236 second and AIAE is 6.794e-4, the AIAE does not have a feasible value. However, by using SQP algorithm, such lack of feasibility can be amended during optimization.
3 Sequential Optimization

3.1 Introduction

For mechatronic systems, sequential optimization deals with the problem by optimizing the mechanism and then synthesizes a controller for the optimized mechanical model. Such an approach is convenient and commonly used by the majority of engineers (e.g. Wang et al. (1999); Hiramoto and Doki (2004)).

However, it cannot guarantee the best solution available, because the synthesized controller can change the system characteristics of the model of the mechanical optimization, and the mechanism is not modified synchronously as the character of the system changes. Further, it has been shown that a better, or at least equivalent, result can be obtained by optimizing the system simultaneously (Fu et al. (2002)).

Since simultaneous optimization can yield a better solution by optimizing controller and mechanism simultaneously, it is also believed that better results can be obtained by several runs of sequential optimization, which regulates control and mechanical parameters sequentially.

3.2 Implementation

Sequential optimizations of the clutch system are implemented in two cases by the following steps:

1. Sequential Optimization (general)
   
   a. Optimize the clutch actuator to be able to disengage clutch in a minimum time (run 1).
   
   b. Synthesize the controller by adding the PID control model and optimize the control parameters (run 1).

2. Sequential Optimization (future runs)
   
   a. Optimize the clutch actuator from the initial state of the previous optimization result (run 2).
   
   b. Optimize the controller from the initial state of the previous optimization result (run 2).
   
   c. Repeat steps a. and b. for further runs (runs 3–6).

3.2.1 Sequential Optimization (general)

The optimization result is shown in Figure 6 and Table 1, with a comparison of initial design and control setpoint. It is obvious that the cost function, the clutch disengaging time, is shorter and the control stability, AIAE, is reduced to a feasible value.

3.2.2 Sequential Optimization with future Runs

Results of the further six runs are shown in Figures 7, 8, and Table 2. The disengaging time still advances with each run but with more general advancements.
### Table 1: Result of sequential optimization

<table>
<thead>
<tr>
<th>λ</th>
<th>$K_p$</th>
<th>$L_{sat}$</th>
<th>R</th>
<th>K</th>
<th>$T_i$</th>
<th>$T_d$</th>
<th>N</th>
<th>b</th>
<th>c</th>
<th>AIAE</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>0.873</td>
<td>5.880</td>
<td>4.000</td>
<td>2.000</td>
<td>3.814</td>
<td>1.341</td>
<td>0.325</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>6.794E-4</td>
</tr>
<tr>
<td>Result</td>
<td>2.585</td>
<td>7.387</td>
<td>7.149</td>
<td>1.567</td>
<td>0.600</td>
<td>3.402</td>
<td>6.273</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1.259E-4</td>
</tr>
</tbody>
</table>

### Table 2: Results of sequential optimization with several runs

<table>
<thead>
<tr>
<th>λ</th>
<th>$K_p$</th>
<th>$L_{sat}$</th>
<th>R</th>
<th>K</th>
<th>$T_i$</th>
<th>$T_d$</th>
<th>N</th>
<th>b</th>
<th>c</th>
<th>AIAE</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>0.873</td>
<td>5.880</td>
<td>4.000</td>
<td>2.000</td>
<td>3.814</td>
<td>1.341</td>
<td>0.325</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>6.794E-4</td>
</tr>
<tr>
<td>Run 1</td>
<td>2.585</td>
<td>7.387</td>
<td>7.149</td>
<td>1.567</td>
<td>0.600</td>
<td>3.402</td>
<td>6.273</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1.259E-4</td>
</tr>
<tr>
<td>Run 2</td>
<td>2.585</td>
<td>7.354</td>
<td>7.169</td>
<td>1.540</td>
<td>0.643</td>
<td>3.499</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>1.256E-4</td>
<td>0.090</td>
</tr>
<tr>
<td>Run 3</td>
<td>2.585</td>
<td>7.351</td>
<td>7.168</td>
<td>1.488</td>
<td>0.621</td>
<td>3.044</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1.214E-4</td>
<td>0.089</td>
</tr>
<tr>
<td>Run 4</td>
<td>2.585</td>
<td>7.354</td>
<td>7.170</td>
<td>1.456</td>
<td>0.583</td>
<td>1.094</td>
<td>2.969</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1.237E-4</td>
</tr>
<tr>
<td>Run 5</td>
<td>2.585</td>
<td>7.553</td>
<td>7.022</td>
<td>1.384</td>
<td>0.130</td>
<td>2.677</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1.251E-4</td>
<td>0.087</td>
</tr>
<tr>
<td>Run 6</td>
<td>2.585</td>
<td>7.552</td>
<td>7.021</td>
<td>1.380</td>
<td>0.128</td>
<td>2.577</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1.258E-4</td>
<td>0.086</td>
</tr>
</tbody>
</table>

### 4 Simultaneous Optimization

#### 4.1 Introduction

Simultaneous optimization deals with the optimization problem integrally. This means that the optimization is implemented with combined models of mechanism and active controller, and design variables are both mechanical parameters and control parameters.

Theoretically, such an algorithm can obtain a better result than, or at least the same result as, sequential optimization (Fu et al. (2002)), because it handles the design variables of both the control and mechanical domains at the same time, and the synchronous modification guarantees the best possible solution for the design variables.

However, the combination of the mechanical and control domains not only increases the number of design variables, but also increases the complexity and nonlinearity for the optimization program to compute both the searching direction and step length information, which in some cases are not suitable for practical application (Bredelms et al. (2001)). Thus, it is believed that simultaneous optimization can result in much longer computation times than sequential optimization.

#### 4.2 Implementation

The mechanical model and the controller model are integrated and design variables of both clutch actuator and controller are selected to implement simultaneous optimization. The optimization result is shown in Figure 9 and Table 3, with comparisons of setpoint and initial design.

![Figure 9: Result of simultaneous optimization](image-url)

### Table 3: Result of simultaneous optimization

<table>
<thead>
<tr>
<th>λ</th>
<th>$K_p$</th>
<th>$L_{sat}$</th>
<th>R</th>
<th>K</th>
<th>$T_i$</th>
<th>$T_d$</th>
<th>N</th>
<th>b</th>
<th>c</th>
<th>AIAE</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>0.873</td>
<td>5.880</td>
<td>4.000</td>
<td>2.000</td>
<td>3.814</td>
<td>1.341</td>
<td>0.325</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>6.794E-4</td>
</tr>
<tr>
<td>Result</td>
<td>2.585</td>
<td>7.551</td>
<td>7.020</td>
<td>1.382</td>
<td>0.942</td>
<td>0.752</td>
<td>1.983</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1.260E-4</td>
</tr>
</tbody>
</table>
Table 4: Process comparison of optimization strategies

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Iteration</th>
<th>Function Called</th>
<th>CPU Time (sec.)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Times</td>
<td>Times</td>
<td>Times</td>
<td></td>
</tr>
<tr>
<td>Initial</td>
<td>Step a Step b sum</td>
<td>Step a Step b sum</td>
<td>current</td>
<td>sum</td>
</tr>
<tr>
<td>Run 1</td>
<td>15 25 40 105 239 344</td>
<td>4.47E4 4.47E4</td>
<td>0.094</td>
<td></td>
</tr>
<tr>
<td>Run 2</td>
<td>27 22 89 245 198 787</td>
<td>7.86E4 1.23E5</td>
<td>0.090</td>
<td></td>
</tr>
<tr>
<td>Run 3</td>
<td>9 19 117 60 124 971</td>
<td>3.01E4 1.53E5</td>
<td>0.089</td>
<td></td>
</tr>
<tr>
<td>Run 4</td>
<td>4 9 130 31 101 1103</td>
<td>2.35E4 1.77E5</td>
<td>0.088</td>
<td></td>
</tr>
<tr>
<td>Run 5</td>
<td>8 15 153 66 117 1286</td>
<td>3.15E4 2.08E5</td>
<td>0.087</td>
<td></td>
</tr>
<tr>
<td>Run 6</td>
<td>11 16 180 91 139 1516</td>
<td>3.77E4 2.46E5</td>
<td>0.086</td>
<td></td>
</tr>
<tr>
<td>Simu.</td>
<td>71 71 1401 1401</td>
<td>2.32E5 2.32E5</td>
<td>0.085</td>
<td></td>
</tr>
</tbody>
</table>

5 Comparison Remark

5.1 Comparison of Sequential Optimization and Simultaneous Optimization

Table 4 shows the comparison between sequential optimization and simultaneous optimization with consideration of optimization iteration times, function call times, and CPU time.

It is clear that simultaneous optimization derives the best solution. However, simultaneous optimization takes much longer for computation (CPU time), almost five times that of sequential optimization, and the improvement over sequential optimization is only 6.6% according to the initial design, as shown in Figure 10.

Figure 10 also shows that the results of the future runs of sequential optimization advance continuously, but the progresses are small after the first run, and Figure 11 shows that for the structure of the clutch actuator, the design variables converge to the result of simultaneous optimization.

However, it may be seen from Table 4 and Figure 10 that these future runs are not worthwhile, because the computation time exceeds that of simultaneous optimization in the 6th and subsequent runs but the best result is still worse.

For optimization processes, simultaneous optimization obtains the solution with much fewer optimization iteration times than the 6th run of sequential optimization.

Nevertheless, the function call times are almost the same, which means that simultaneous optimization needs more function call times to obtain optimization information for optimization iteration.

Figure 10: Comparison of sequential and simultaneous optimization results
5.2 Conclusion Remarks

The following conclusions may be derived and verified from the comparison and reference studies:

1. Simultaneous optimization can obtain the best available solution.
2. The solution for sequential optimization is worse than that for simultaneous optimization, but is much more efficient comparing the optimization advancement and time cost.
3. For sequential optimization, better solutions can be obtained by more future runs. However, it is not worthwhile to use such a method if a result equivalent to that achieved by simultaneous optimization is expected.
4. Simultaneous optimization can reach the target in fewer steps (fewer optimization iterations), but more function call times are required for each iteration to obtain searching information, as a result of which the optimization advances more slowly. In the other word, sequential optimization proceeds faster but need more steps to arrive at the solution point, simultaneous optimization proceeds slower but need fewer steps to arrive at the solution point.
6 Hybrid Strategy

6.1 Introduction

According to the comparison, it may be concluded that simultaneous optimization can obtain the best solution, but it is not efficient. By contrast, sequential optimization (general) is efficient, but cannot guarantee the best solution. For these reasons, an optimization strategy that can obtain the same solution as simultaneous optimization, yet more efficiently, is proposed.

From optimization conceptions, sequential optimization cannot obtain the best result directly because it handles the design variables of the control and mechanical domains separately, where several runs are required to synthesize the two domains to converge gradually to the optimum point, the point that can be reached by simultaneous optimization directly. However, it is very efficient because the fewer design variables of each optimization process, the more efficient is optimization, as shown in the case.

On the other hand, simultaneous optimization can reach the best available result directly, because it handles both domains simultaneously. However, the process is more time consuming because the greater number of design variables, the more function call times are required to obtain optimization information.

To obtain the best solution more efficiently, the hybrid optimization strategy is proposed. The hybrid strategy uses sequential optimization to shorten the searching distance of the simultaneous optimization, which is directly related to optimization iteration times and thus computation time.

First, the hybrid strategy uses sequential optimization to obtain the first point, which is probably close to the solution expected. Second, serving the point as the initial state, it uses simultaneous optimization to compute the final solution. It has been shown in the case and other studies (Fu et al. (2002); Yemc (1992); Begg and Liu (2000)) that sequential optimization can reduce the cost function efficiently and the result converges closely to the best solution. Thus, it is believed that a faster approach to reach the optimum point is accessible by the hybrid strategy.

6.2 Implementation

Steps to optimize the clutch system by hybrid strategy are:

a. Using sequential optimization to optimize the clutch system.

b. Serving the result of step a. as the initial state, using simultaneous optimization to find out the final solution.

The result of implementation of the strategy is shown in Figures 12, 13, and Table 5, with comparisons with initial design, sequential optimization, and simultaneous optimization.

![Figure 12: Optimization of hybrid strategy](image)

![Figure 13: Scaled view of Figure 12](image)

Table 5: Result of hybrid strategy and comparison of other strategies

<table>
<thead>
<tr>
<th></th>
<th>(\lambda)</th>
<th>(K_w)</th>
<th>(L_{init})</th>
<th>(R)</th>
<th>(K)</th>
<th>(T_p)</th>
<th>(T_d)</th>
<th>(N)</th>
<th>(b)</th>
<th>(c)</th>
<th>AIAE</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>0.873</td>
<td>5.880</td>
<td>4.000</td>
<td>2.000</td>
<td>3.814</td>
<td>1.341</td>
<td>0.325</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>6.794E-4</td>
<td>0.223</td>
</tr>
<tr>
<td>Seq.</td>
<td>2.585</td>
<td>7.387</td>
<td>7.149</td>
<td>1.567</td>
<td>0.600</td>
<td>3.402</td>
<td>6.273</td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1.259E-4</td>
<td>0.094</td>
</tr>
<tr>
<td>Simu.</td>
<td>2.585</td>
<td>7.551</td>
<td>7.020</td>
<td>1.382</td>
<td>0.942</td>
<td>0.752</td>
<td>1.983</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1.260E-4</td>
<td>0.085</td>
</tr>
<tr>
<td>Hybrid</td>
<td>2.585</td>
<td>7.550</td>
<td>7.010</td>
<td>1.382</td>
<td>0.942</td>
<td>0.752</td>
<td>1.983</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1.260E-4</td>
<td>0.085</td>
</tr>
</tbody>
</table>
### Table 6: Process comparison of optimization strategies

<table>
<thead>
<tr>
<th></th>
<th>Opt. Iteration Times</th>
<th>Function Called Times</th>
<th>CPU Time (sec.)</th>
<th>Cost Function</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Step a</td>
<td>Step b</td>
<td>sum</td>
<td>Step a</td>
</tr>
<tr>
<td>Initial</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seq.</td>
<td>15</td>
<td>25</td>
<td>40</td>
<td>105</td>
</tr>
<tr>
<td>Simu.</td>
<td>71</td>
<td>71</td>
<td>140</td>
<td>1401</td>
</tr>
<tr>
<td>Hybrid</td>
<td>40</td>
<td>5</td>
<td>46</td>
<td>344</td>
</tr>
</tbody>
</table>

7 Comparison

As shown in Tables 5 and 6, the hybrid strategy achieves the same result as with simultaneous optimization, and much more quickly. Considering the optimization iteration times, the iteration time of simultaneous optimization is substantially reduced from 71 (in original simultaneous method) to 5 (in hybrid method), which is due to the shortened searching distance by sequential optimization in step a. It is clear that even with the CPU time for sequential optimization in step a, the total time taken to optimize using the hybrid method is still much shorter than with the simultaneous optimization method.

A simplified diagram of optimization trajectory with comparisons of sequential optimization, simultaneous optimization, and hybrid strategy is shown in Figure 14.

8 Conclusions

The followings are concluded from the case study and reviewed studies:

1. The sequential optimization method is efficient but cannot guarantee the best solution.
2. Future runs of sequential optimization can obtain better results, but the advancements are slight.
3. Simultaneous optimization can guarantee the acquisition of the best possible solution. However, it is not efficient comparing the advancement from sequential optimization and the required time.
4. Using sequential optimization to shorten the optimization distance and using simultaneous optimization to find out the final solution, hybrid strategy can obtain a result better than sequential optimization and faster than simultaneous optimization.
5. If the best result is expected, hybrid optimization strategy is suggested instead of simultaneous optimization, which can reach the result more efficient than simultaneous optimization.

Acknowledgement The support of this research by the National Science Council, Taiwan, R. O. C., under grant NSC 91-2212-E-009-022 and Industrial Technology Research Institute, Taiwan, R. O. C., under grand C324K21CD0, is gratefully acknowledged.

References:


