III. BASIC THEORY

3.1. Tuned Mass Damper Concept

The tuned mass damper consist of two mass system, that are the mass of the structure itself and the mass of the TMD. The theory of TMD for SDOF systems subjected to harmonic force excitation and harmonic ground motion. The equation of motion can be expressed as below:

The general equation of motion for the primary mass and tuned mass are:

$$[\mathrm{M}]\{\ddot{U}\}+[C]\{\dot{U}\}+[K]\{U\}=-[\mathrm{M}]\{1\}\ddot{u}_g \tag{3-1}$$

with the parameters:

$$\omega^2=\frac{k}{m} \tag{3-2}$$

$$c=2\xi\omega m \tag{3-3}$$

$$\mu=\frac{m_d}{m} \tag{3-4}$$

Where M is defined as the mass ratios, C is defined as damping, K is defined as stiffness, $\ddot{U},\dot{U},U$ are displacement, velocity and acceleration respectively, $\ddot{u}_g$ is ground acceleration, while ξ is the damping factor of the structure and $\{1\}$ equal vector that contain 1 that is $1=[111]^T$. The idea of installing mass damper is to limit the vibration of a structure when subjected to excitation. The mass damper has parameters of m_d, k_d and c_d. The damper is also tuned to the natural frequency of the structure, therefore:

$$\omega=\omega_d \tag{3-5}$$

$$c_d=2\xi\omega_d m_d \tag{3-6}$$

$$k_d = m_d \omega_d^2 \tag{3-7}$$

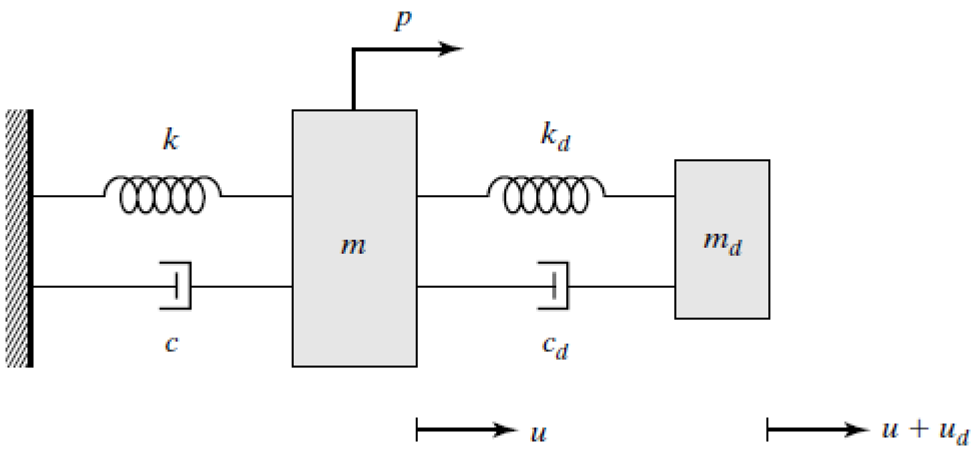

Fig. 3.1 SDOF with TMD system

3.2. <u>Genetic Algorithm</u>

Genetic Algorithm was founded by John Holland and his friends on early 1970 in New York. The basic techniques of the GAs are designed to simulate processes in natural systems necessary for evolution, especially those follow the principles first laid down by Charles Darwin of "survival of the fittest.". Since in nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones.

According to Arfiadi (2000), Arfiadi and Hadi (2011), GAs simulate the survival of the fittest among individuals over consecutive generation for solving a problem. Each generation consists of a population of character strings that are analogous to the chromosome that we see in our DNA. Each individual represents a point in a search space and a possible solution. The individuals in the population are then made to go through a process of evolution. Each individual is coded as a finite length vector of components, or variables, in terms of some alphabet, usually the binary alphabet {0,1}. To continue the genetic analogy these individuals are likened to chromosomes and the variables are analogous to genes. Thus a chromosome (solution) is composed of several genes (variables). A fitness score is assigned to each solution representing the abilities of an individual to `compete'. The individual with the optimal (or generally near optimal) fitness score is sought.

New generations of solutions are produced containing, on average, more good genes than a typical solution in a previous generation. Each successive generation will contain more good `partial solutions' than previous generations. Eventually, once the population has converged and is not producing offspring noticeably different from those in previous generations, the algorithm itself is said to have converged to a set of solutions to the problem at hand. The GA aims to use selective `breeding' (with mutation or recombination) of the solutions to produce `offspring' better than the parents by combining information from the chromosomes.

Real genetic algorithm (RCGA) is used to find the optimize TMD. For the example for the new individual that have four design variable, four random numbers are produced as shown in Figure 3.4. below.

8.2	1.5	23.7	15.8

Figure 3.2. Individual with four design variable RCGA

Mutation and *crossover* procedure that used in this research taken as follows.

For individual G_1 and G_2 used for *crossover*, then produce new generation G'_1 and G'_2 following what is called equal *crossover* as follows:

$$G'_1 = a(G_1 - G_2) + G \tag{3-8}$$

$$G'_2 = a(G_2 - G_1) + G \tag{3-9}$$

where a is random number between 0 and 1. From the equation above, can be seen if the combination of RCGA with *crossover* method the domain that needed for optimization no need to know. C*rossover* has an ability to exploring the unknown domain (Arfiadi and Hadi, 2011). The example below show the designer can guess the early value for the random variable design without affect the final design value.

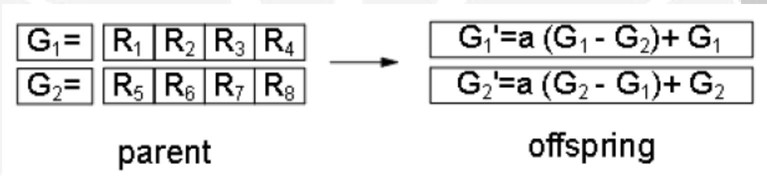

Figure 3.3. Crossover RCGA balance

For mutation, design variable that produced after the simple mutation are as below:

$$G'_p = \lfloor R_1 R_2 ... R'_j ... R_N \rfloor \tag{3-10}$$

$$R'j = \alpha a R_j \tag{3-11}$$

where α > 1 and a = random number between 0 and 1.

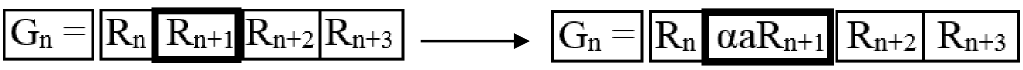

Figure 3.4. Mutation Process

3.3. GA-H_2 optimization

Arfiadi (2000) explained about the objective function of H_2 methods is to minimize the transfer function from external disturbance to the regulated output. The combination of displacements, velocities as well as acceleration of the structure can be included to regulated output. To optimize the TMD parameters, the mass is decided first then optimize the stiffness and damping. After forming the equations of motion then transformed into state equation in term of state vector Z. The regulated output z as the response to be minimized is then chosen in relation with the performance index. The formula is:

Obtain TMD parameters (k_d,c_d)

$$\dot{Z} = AZ + Ew$$

$$z = C_z + Z$$

Such that

$$J = \left[tr\left(C_z L_c C_z^T\right)\right]^{1/2} = \left[tr\left(E^T L_o E\right)\right]^{1/2} \qquad (3\text{-}12)$$

$$AL_c + L_c A^T + EE^T = 0 \text{ or}$$

$$A_l^T L_o + L_o A + C_Z^T C_z = 0$$

The computation of H_2 norm in eq. 3-12 can be obtained easily by using *lyap* commands in the MATLAB Control System Toolbox.