Development of GA-based Control System for Active Shape Control of Inflatable Space Structures

Fujun Peng, Yan-Ru Hu, Alfred Ng

***Abstract*— This paper describes the development of a control system used for the active shape control of inflatable space structures. The genetic algorithm is utilized for the optimization of control variables. A vision system is implemented for the measurement of the structure shape. Shape memory alloy wire actuators are used to exert the obtained optimal tensions. The developed control system is then tested on a 200mm × 300mm rectangular Kapton membrane structure. The membrane is pulled by three tensions along each edge. Different combinations of the tensions produce various wrinkles on the membrane. Test results indicate that the developed control system works very well in improving the structure shape precision.**

I. Introduction

INFLATABLE structures have attracted much interest in the space community due to their unique advantages in achieving low mass and high packaging efficiency.[1,2] Their ultra-lightweight and small-volume properties in turn can potentially reduce the overall space mission cost by reducing the launch vehicle size requirement. Inflatable structures can also reduce total system mass and deployment system complexity, thereby increasing system reliability. This type of structures has been envisioned for many space applications such as large telescopes, antennas, solar sails, sun shields, solar arrays, etc.[1-4]

We are currently working on an in-house R&D project in the development of a large surface area to mass ratio inflatable space structure with possible applications as a Synthetic Aperture Radar (SAR) antenna. The key components of this inflatable structure are inflatable tubes, membrane and the links installed in-between stretching the membrane (Fig.1). It can be rolled into a small volume and fixed on a satellite bus for launching. When the satellite arrives into the orbit , the inflatable tubes are filled with gas and roll out, and the Kapton membrane will be deployed accordingly.

It is expected that the membrane will be subjected to flatness problems during its lifetime in orbit due to the thermal variation in space. A pure passive control method may not be sufficient to maintain the membrane flatness. Hence we are developing an active control scheme to adjust the stretching tensions in real time so that the membrane could be maintained flat. This paper describes the implementation of the active control scheme, which involves a vision system development for the structure shape measurement, the genetic algorithm (GA) implementation for tensions optimization and a sub-controller design for tension exertion using shape memory alloy wire actuators. The developed control system is then tested on a small size membrane structure, and the results demonstrate the effectiveness of the scheme and the developed control system.

Manuscript received February 15, 2005

F. Peng is with the Directorate of Spacecraft Engineering, Canadian Space Agency, Saint-Hubert, Quebec, J3Y8Y9, Canada (tel: +1-450-9266683; fax: +1-450-9264695; e-mail: fujun.peng@space.gc.ca).

Y.-R. Hu is with the Directorate of Spacecraft Engineering, Canadian Space Agency, Saint-Hubert, Quebec, J3Y8Y9, Canada (e-mail: yan-ru.hu@space.gc.ca).

A. Ng is with the Directorate of Spacecraft Engineering, Canadian Space Agency, Saint-Hubert, Quebec, J3Y8Y9, Canada (e-mail: Alfred.ng@space.gc.ca).

II. Genetic Algorithm Implementation

The genetic algorithm is an optimization searching technique derived from the mechanics of natural selection and genetics. This mechanism has been mathematically shown to eventually "converge" to the best possible solution. Compared to traditional search and optimization procedures, the genetic algorithm is robust, and generally more straightforward to use. It is stochastic in nature, thus is capable of searching the entire solution space with more likelihood of finding the global optimum. The genetic algorithm is applicable to both linear and nonlinear systems where little or no *a priori* knowledge of the system is given.[5,6]

To implement the searching of the optimal tensions for the membrane structure, all the parameters (here, they are the amplitudes of tensions) to be optimized are first mapped (coded) into a chromosome. Each parameter corresponds to

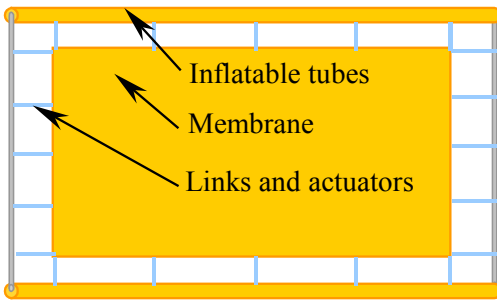


Fig. 1. Sketch of the inflatable structure

one particular portion of the chromosome. The block diagram of GA is shown in Fig. 2. The following steps are executed to search for the best solutions: (1) **[Initialization]** Randomly generate the initial population (possible tension combinations); (2) **[Fitness]** Evaluate the fitness of each individual in the population (larger value means better fit); (3) **[Criteria met?]** Check if the ending condition is satisfied: if yes, stop and return the best solution; if no, generate new population for a repeat execution of the algorithm; (4) **[New population]** Create a new population by repeating the following steps: **[Selection]** Select two parent individuals based on their fitness (the better the fitness, the bigger chance to be selected); **[Crossover]** Using a crossover probability, cross over the parent individuals to form new offspring; **[Mutation]** Using a mutation probability, mutate new offspring at each locus (the position in an individual).; **[Accepting]** Place new offspring in a new population; (5) **[Loop]** Go to step (2). After the best individuals are obtained, decode them to the required parameters, and choose one as the optimized solution.

For numerical simulation of genetic algorithm as shown in Fig. 2, the first loop runs in the order as: initialization → fitness evaluation → criteria judgment. Afterward, the running order becomes: individual selection → crossover → mutation → new population → fitness evaluation → criteria judgment. Its characteristics are: (1) a new population production and its evaluation are completed within the same loop; (2) a new loop always starts with individual selection and ends at criteria judgment; (3) the program stops automatically when the required criteria are met. For the sake of real time control, the genetic algorithm needs to be implemented with hardware in the loop. This requires two-step fitness evaluation, i.e., all individuals of a population have to be converted to control force and output first, and then evaluated in the next loop according to the acquired flatness produced by them. This is implemented as Fig. 3, and its characteristics are: (1) A new population production and its evaluation are completed in two successive loops. (2) A new loop starts with fitness evaluation and ends at data I/O. The whole running order is: fitness evaluation of last loop population → criteria judgment (not met)→ individual selection → crossover → mutation → new population → tension output → data acquisition. (3) If the criteria are met, the genetic algorithm stops the optimization process, but the whole program continues to output the optimal individual to get steady control results. The running order then becomes: fitness evaluation of last loop population → criteria judgment (met)→ optimal individual output → data acquisition. (4) Genetic algorithm initialization also requires two loops. The first loop is: initial population setting → tension output → data acquisition, and the second one is: fitness evaluation of the initial population → individual selection → crossover → mutation → new population → tension output → data acquisition. The difference between

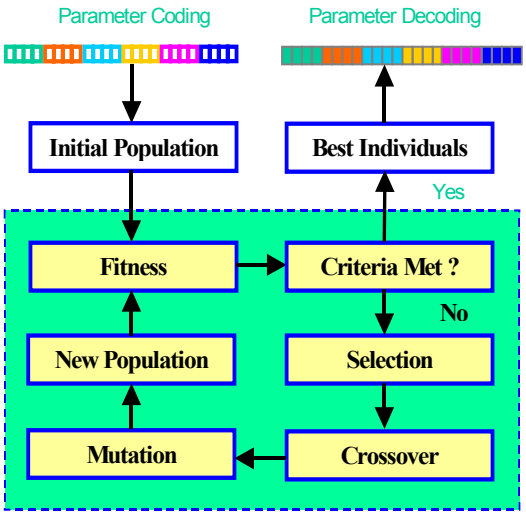


Fig. 2. Block diagram of genetic algorithm

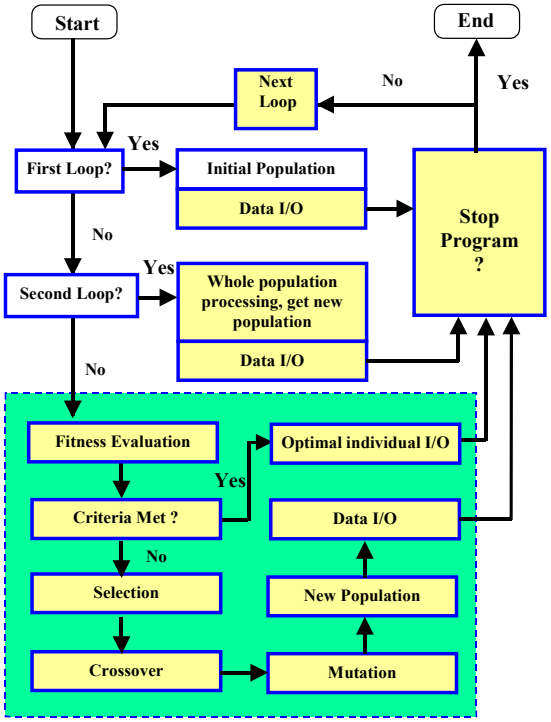


Fig. 3. Block diagram of genetic algorithm implementation with hardware in the loop

the second loop and the loops afterward is that the second loop evaluates all the population individuals, but the loops afterward only evaluates part of them since some best individuals of last loop are transferred to the new population in the present loop without any change. It should be noted that every individual has to be converted to control force combinations before it is output for tension control. It should be pointed out that in Fig. 3, the second step of initialization is simply represented by only two rectangular blocks.

III. VISION SYSTEM IMPLEMENTATION

The vision system consists of a Dell P4 2.4GHz computer, a light projector and a 1300×1000 pixels CMOS camera. Normally, one camera can only capture a 2D image at a time, and cannot give 3D coordinates of points on an object surface by a single image. That is why Photogrammetry technique needs multiple pictures to extract 3D coordinates of points on a physical object. These pictures can be taken either by multiple cameras placed at different locations, or by moving one camera through different locations. Photogrammetry technique also needs to perform referencing to identify which marked point in each image is the same physical point on the object. Manual intervention is usually required to ensure correct referencing, and this may take much time. So the photogrammetry technique is unacceptable for our tests since fast measurement is required. In order to obtain fast measurements, this vision system uses only one image to determine 3D coordinates of points, but with the aid of calibrated light planes projected by a projector. The concept is illustrated in Fig. 4.

The projector projects a light plane at around 45 degree angle onto the object surface, which produces an intersection curve or a straight line if the surface is flat. For any point on the intersection, it is seen by the camera through a straight line radiating from the camera focal point to the selected point. Therefore its location can be determined as the intersection point of the light plane and the radiating line. Its 3D coordinates can be easily calculated if the light plane equation and the radiating line equation are known. Project more light planes to cover the whole area of interest, and choose more points on each intersection curve, we can easily determined the object surface flatness by calculating the 3D coordinates of these selected points.

A. Camera Calibration

To determine the equations of the straight line radiating from the camera focal point to the selected points on the intersection, a camera calibration is required. The calibration procedure involves a small rig that sets a plate at 2 different heights. A target patterns (in our case it is an array of dots of known spacing) is observed at 2 heights. These dots are observed by the camera (*u*,*v* locations) and the mathematics can be written as (in homogeneous units):

$$[x, y, z, 1][M] = [u, v, t] \tag{1}$$

$$U = u / t \tag{2}$$

$$V = V / t \tag{3}$$

where, M is a 4×3 camera calibration matrix, U and V are homogeneous coordinates (pixels). Combine (1) and (2), we have,

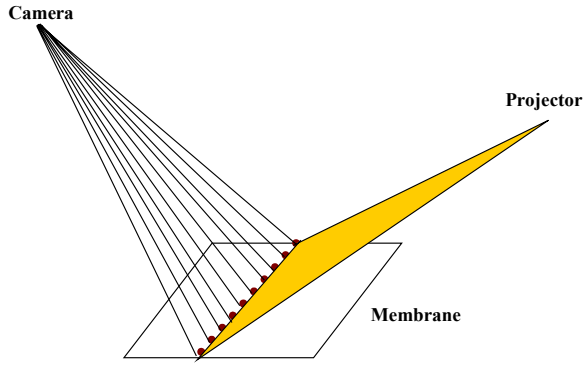


Fig. 4. Principle of vision system

$$[x, y, z, 1]\begin{bmatrix} M_{1,1} \\ M_{2,1} \\ M_{3,1} \\ M_{4,1} \end{bmatrix} - U \cdot [x, y, z, 1]\begin{bmatrix} M_{1,3} \\ M_{2,3} \\ M_{3,3} \\ M_{4,3} \end{bmatrix} = 0 \tag{4}$$

$$[x, y, z]\left(\begin{bmatrix} M_{1,1} \\ M_{2,1} \\ M_{3,1} \end{bmatrix} - U \cdot \begin{bmatrix} M_{1,3} \\ M_{2,3} \\ M_{3,3} \end{bmatrix}\right) = U \cdot M_{4,3} - M_{4,1} \tag{5}$$

Similarly, combine (1) and (3), we have,

$$[x, y, z]\left(\begin{bmatrix} M_{1,2} \\ M_{2,2} \\ M_{3,2} \end{bmatrix} - V \cdot \begin{bmatrix} M_{1,3} \\ M_{2,3} \\ M_{3,3} \end{bmatrix}\right) = V \cdot M_{4,3} - M_{4,2} \tag{6}$$

Substitute the targets coordinates $x_i\ y_i\ z_i$ ($i = 1, \cdots, n$, n is the total number of calibration targets) and corresponding U_i, V_i ($i = 1, \cdots, n$) into (5) and (6) and apply a best fit solution, the camera calibration matrix M can be obtained.

B. Light Planes Calibration

Light planes calibration determines the equations of light planes projected by the projector. The calibration procedure uses the same rig as the camera calibration that sets a plate at 2 different heights (the heights are known). But this time there is no target on the plate. At each height the projector is turned on and multiple lines are projected on the plate. To calibrate a light plane, take multiple points on the two lines projected on the plate by this light plane (at 2 heights). Because the camera calibration has completed, the U and V coordinates for these points and corresponding radiating line equations can be easily determined. Substitute the known z coordinates of these points into the corresponding radiating line equations, the corresponding x and y coordinates are then obtained.

With these obtained 3D coordinates, a light plane

equation can be determined by solving an eigenvalue problem:

$$[W][P] = 0 \tag{7}$$

where,

$$W = \begin{bmatrix} \sum_{i=1}^{N} \bar{x}_i^2 & \sum_{i=1}^{N} \bar{x}_i \bar{y}_i & \sum_{i=1}^{N} \bar{x}_i \bar{z}_i \\ \sum_{i=1}^{N} \bar{x}_i \bar{y}_i & \sum_{i=1}^{N} \bar{y}_i^2 & \sum_{i=1}^{N} \bar{y}_i \bar{z}_i \\ \sum_{i=1}^{N} \bar{x}_i \bar{z}_i & \sum_{i=1}^{N} \bar{y}_i \bar{z}_i & \sum_{i=1}^{N} \bar{z}_i^2 \end{bmatrix} \tag{8}$$

$$\bar{x}_i = x_i - \sum_{i=1}^{N} (x_i / N) \tag{9}$$

$$\bar{y}_i = y_i - \sum_{i=1}^{N} (y_i / N) \tag{10}$$

$$\bar{z}_i = z_i - \sum_{i=1}^{N} (z_i / N) \tag{11}$$

N is the total the number of points taken on the same light plane. Denote the eigenvector of the smallest eigenvalue of matrix W as [a, b, c, d], then the light plane can be expressed as:

$$ax + by + cz = d \tag{12}$$

C. Measurement Procedure

For a specific observed point, its U and V coordinates can be easily identified first. Then the equation of the corresponding radiating line can be expressed as the intersection of 2 planes:

$$[x, y, z]\ [L] = K \tag{13}$$

where,

$$L = \left[\begin{bmatrix} M_{1,1} \\ M_{2,1} \\ M_{3,1} \end{bmatrix} - U \cdot \begin{bmatrix} M_{1,3} \\ M_{2,3} \\ M_{3,3} \end{bmatrix}, \quad \begin{bmatrix} M_{1,2} \\ M_{2,2} \\ M_{3,2} \end{bmatrix} - V \cdot \begin{bmatrix} M_{1,3} \\ M_{2,3} \\ M_{3,3} \end{bmatrix} \right] \tag{14}$$

$$K = [U \cdot M_{4,3} - M_{4,1}, \quad V \cdot M_{4,3} - M_{4,2}] \tag{15}$$

The 3D coordinates of this point can be easily obtained by solving (10) and (13) simultaneously.

D. Membrane Flatness Calculation

With the obtained 3D coordinates of the points distributed on the membrane, the membrane flatness is defined as the standard deviation of these points. The calculation of the standard deviation involves the same procedure as the light plane calibration. Substitute all the obtained 3D coordinates into (8) to (11), the standard deviation is then the square root of the smallest eigenvalue of W divided by the total number of points. The corresponding eigenvector determines the least square best fit plane, given as (12).

IV. SMA Actuator Sub-Controller Design

SMA wire actuators are attractive for inflatable structure shape control due to their unique properties such as high force, long stroke, small size, light weight, etc. Their poor stability and controllability, however, make it almost impossible to exert required tensions to the real structure by simply input a fixed electrical current to SMA actuators. Some attempts have been made to solve this type of problems by adjusting the heating electrical current flowing in the SMA wire. Several control algorithms, such as PID (Proportional Integral Derivative) control, PWM (Pulse Width Modulation) control, optimal control, have been proposed, and Preisach model and neural networks are developed to describe hysteresis property of SMA actuators[7-9]. However, it is hard to model the SMA hysteresis precisely, and consequently the stability of the control system is not guaranteed. This may become an obstacle to use SMA actuators in space missions. Here we propose a simple control strategy based on the idea of adjusting the SMA wire temperature as fast as possible.

The response of a SMA wire actuator is directly related to its temperature, which is mainly affected by the electrical input current and environment cooling condition. It can be expressed as:

$$S(t) = f(C(t), I(t)) \tag{16}$$

where, $S(t)$ is the actuation, which could be displacement or force, $C(t)$ and $I(t)$ are cooling conditions and input current. If we don't use active cooling, the SMA wire actuator response can only be actively adjusted by changing the input current. Consider a simple feedback control, at any time t, the input current could be designed as:

$$I(t) = I(t-1) + \Delta I(t) = I(t-1) + K(t)e(t) \tag{17}$$

where, $K(t)$ is a feedback coefficient and $e(t)$ is the error response denoted by

$$e(t) = S^*(t) - S(t) \tag{18}$$

where $S^*(t)$ and $S(t)$ are the desired response and actual response, respectively. Now the question becomes how to determine $K(t)$ of (2) such that SMA wire temperature changes rapidly and consequently the error response reduces quickly.

Consider $e(t) > 0$, the input current should be given a positive increment, and $K(t)$ should also be a positive value. If $K(t)$ is small, the obtained I(t) may not be large enough to heat the SMA wire very fast, or many more steps are needed to obtain a proper SMA temperature. So it is desirable to have a relatively large value of $K(t)$. But at the same time, the input current should not exceed an upper limit I_{upper}, which is set to avoid burning the SMA wire. To have the fastest response, for $e(t) > 0$, the input current is directly designated as: $I(t) = I_{upper}$.

For $e(t) < 0$, input current increment should be negative

so that SMA wire temperature may decrease and consequently SMA wire actuation becomes smaller. To have the fastest response, input current is set to 0. For $e(t) = 0$, I(t) could either be I_{upper} or 0. The control law can be written as:

$$I(t) = \begin{cases} I_{upper}, & e(t) > 0 \\ 0, & e(t) \le 0 \end{cases} \quad (19)$$

The value of I_{upper} can be determined empirically. It should be large enough to activate full transformation of the SMA wire, but should not burn the SMA wire within an update interval.

V. Control System Integration

The control system is integrated using LabView, Matlab and Automation Manager. LabView code realizes the SMA actuator sub-controller, tension measurement, as well as controls control parameters input, results display and saving and data I/O. Matlab code implements the genetic algorithm. Automation Manager realizes the membrane shape measurement. LabView code is the master, which coordinates the whole system functioning by using ActiveX technique. The whole control system is implemented using three layers of loops, which is illustrated in Fig. 5. Loop 1 receives parameters from GUI (Graphical User Interface), executes GA (genetic algorithm) and then transfers control parameters and a group of candidate tension combinations to Loop 2. Loop 2 realizes the required tension combinations and measures the membrane flatness (standard deviation) corresponding to every tension combination. First, Loop 2 transfers one tension combination and control parameters to Loop 3, which executes the actuator controller and data acquisition. When the desired values of the tension combination are achieved, Loop 3 triggers the vision system to record the corresponding membrane flatness. To obtain steady membrane states, Loop 3 dose not triggers the vision system at once. Instead, it delays a number of iterations after the desired tension are obtained, so that the influence of the tension overshoots can be avoided. This number could be set from GUI. In order for the vision system to have enough time to finish its work, Loop 3 is still running to hold the desired tensions until the vision system competes the measurement. When the vision system and Loop 3 complete an execution, Loop 2 runs again to realize another tension combination and the obtain corresponding membrane flatness. This process is repeated multiple times until all the candidate tension combinations are realized and the associated membrane flatness values are recorded. Loop 1

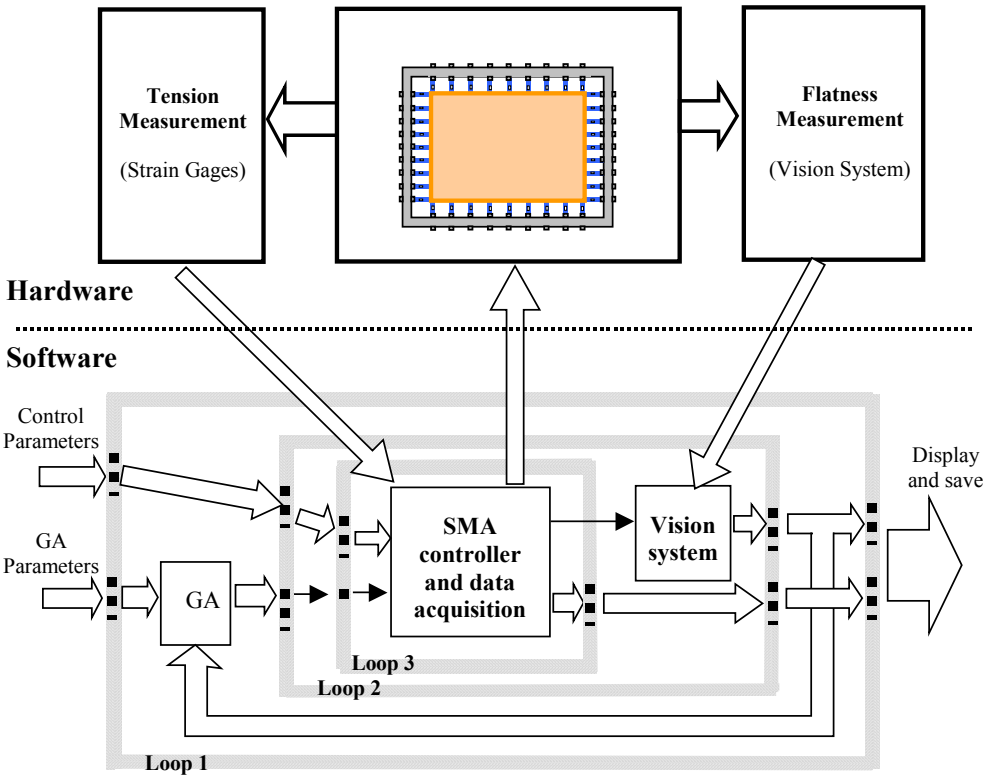


Fig 5. Block diagram of the whole control system implementation

gets back the data obtained by Loop 2, displays, saves to files, and then goes to the next generation. To complete the test of one generation of tension candidates, Loop 1 runs only one circle. The circle number of Loop 2 running is the total number of new individuals in every population. While for Loop 3, the running circles is much more than that of Loop 2, the specific number is always changing depending on the largest difference of the two adjacent tensions. If the difference is large, the actuator then needs more time to change its status from one to the other, and thus more running circles are needed.

VI. Some Tests

The developed control system is tested on a small size membrane structure, which is shown in Fig. 6. Basically, it is a 200mm×300mm rectangular Kapton Membrane, stressed by 12 discrete links installed between the membrane boundaries and aluminum frame. A local thermal load source is placed under the membrane (not visible in Fig. 6). The flatness of the membrane is dependent on the local thermal load and the tension combinations. To realize active control, six shape memory alloy wire actuators (0.25 mm in diameter and 100 mm in length) are installed along the edge as a part of tension links. To monitor the values of tensions, strain gages are glued onto small and thin aluminum strips, which are also a part of tension links. The arrangement of SMA actuator, strain gage and links is sketched in Fig. 7. These tension measurement elements are calibrated using a load cell before tests are performed. In order for the vision system camera to see the projected lines clearly, a very thin coating is put on one side of the membrane. The whole setup including vision system is shown in Fig. 8.

The sub-controller of SMA actuators are tested first. Set the six desired tensions as 3.75N, 3.53N, 3.31N, 3.09N, 2.87N and 2.65N, each corresponding to one actuator, and see how well these tensions can be achieved. The obtained time histories of these six tensions are shown in Fig. 9. It is found that the obtained tensions can track the desired tensions very well. The RMS errors (after 30 seconds) are 0.0072N, 0.0076N, 0.0079N, 0.0057N, 0.0068N, 0.0068N, respectively. The activation voltage is 1.7V for all SMA actuators, and their different electrical resistances lead to different input currents as well as different SMA response speeds. These actuators are then tested using 20 tension combinations, which are generated by GA randomly. These tension combinations are exerted one by one. During the test, the actuators transfer from one desired tension combination to another directly, without cooling down to the room temperature. The desired tensions, measured tensions and the error tensions are shown in Fig. 10. The values of error tensions are given in Table I. We can see that actual tensions achieved are very close to the desired values, and the maximal tension error of these obtained tensions is 0.021N. All the relative errors are less than 0.65%.

To test the vision system, we calibrate the camera and

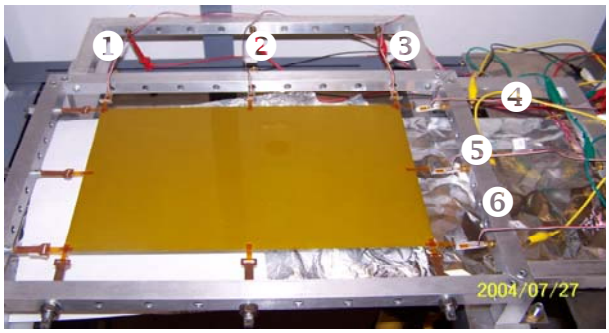


Fig. 6. Picture of Membrane Structure Used for Experimental Study

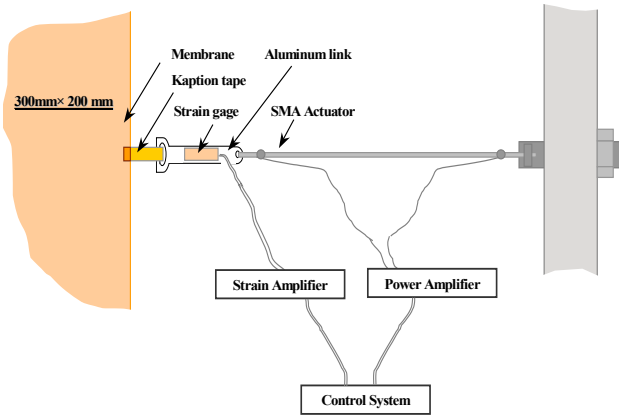


Fig. 7. Arrangement of Actuators And Sensors

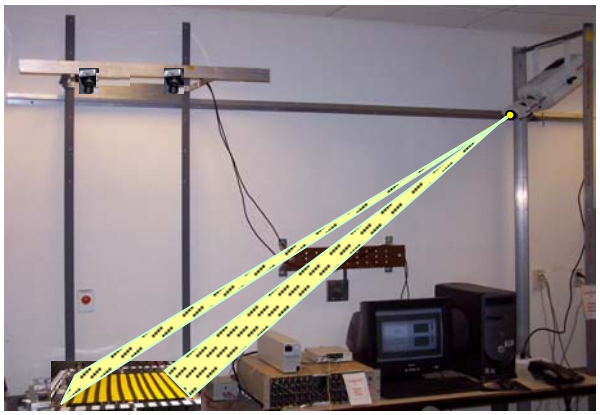


Fig. 8. Picture of the Whole Control System and Setup

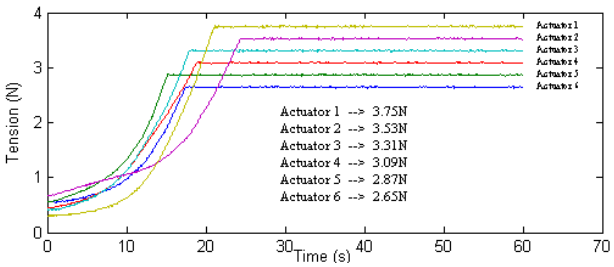


Fig. 9. Time histories of obtained tensions. The controller updating rate is 100Hz

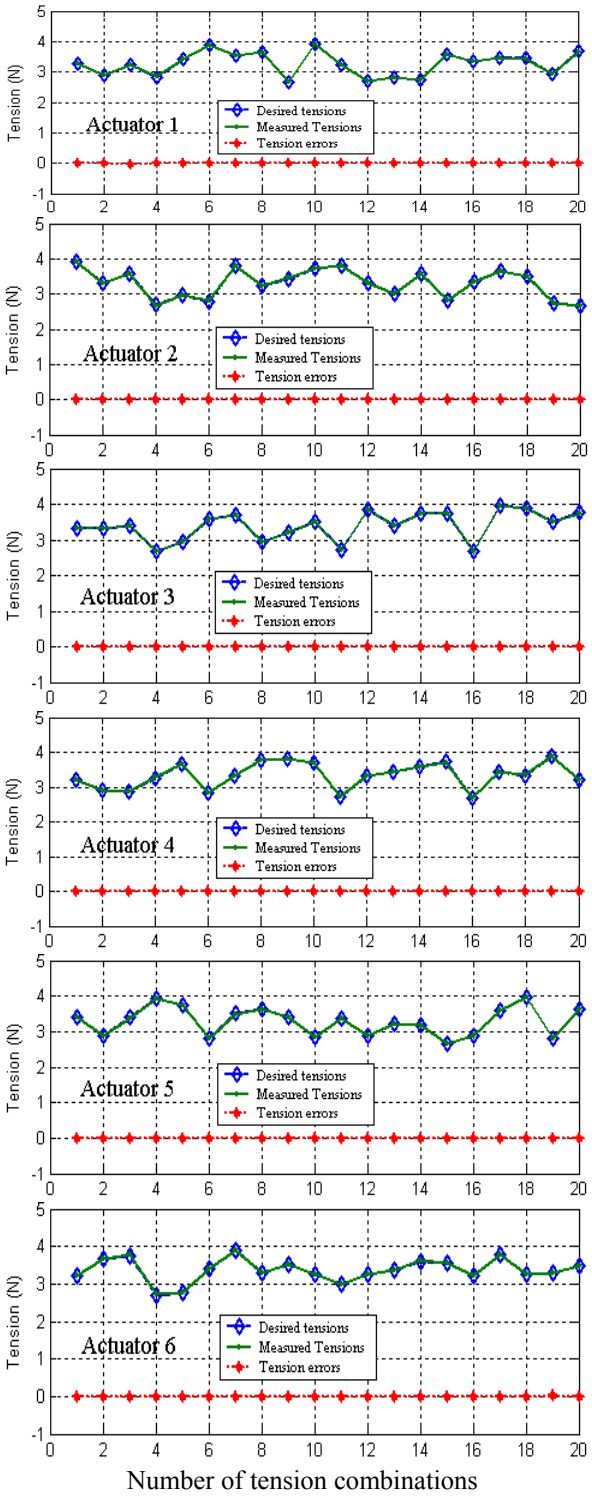
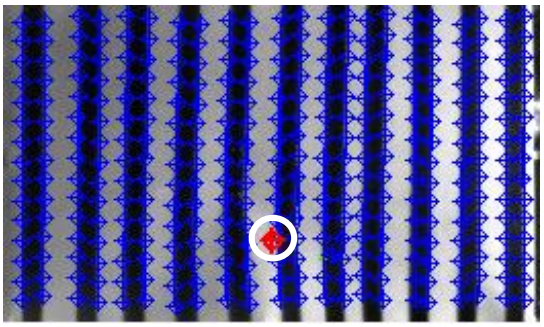


Fig. 10. Test results of 20 tension combinations

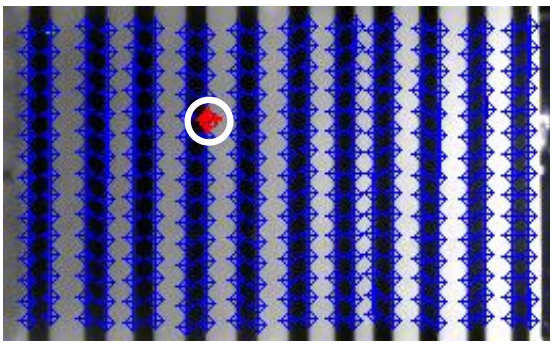
light planes first. Totally 22 light planes (22 edges of the 11 dark strips shined on the membrane) are calibrated. Then the test is performed with 15 points selected at each strip edge. The program gives very quickly all the 3-D coordinate values of these 330 points. The maximal z coordinate, the minimal z coordinate, the best-fit plane of these points and the largest amplitude of the membrane wrinkle are also

TABLE I – TENSION ERRORS (N)

Actuator 1	Actuator 2	Actuator 3	Actuator 4	Actuator 5	Actuator 6
0.0030	0.0096	-0.0019	-0.0058	-0.0127	-0.0051
0.0028	0.0039	-0.0063	-0.0001	-0.0138	-0.0033
-0.0132	0.0028	-0.0019	0.0014	-0.0151	0.0074
0.0035	-0.0084	0.0047	0.0111	0.0017	0.0073
0.0015	-0.0016	-0.0063	-0.0086	-0.0042	0.0025
-0.0032	0.0048	0.0004	0.0003	-0.0117	0.0006
-0.0123	-0.0075	-0.0018	-0.0103	-0.0148	0.0054
-0.0051	0.0052	0.0047	0.0053	0.0112	-0.0019
-0.0072	0.0211	-0.0096	0.0001	0.0005	0.0114
0.0003	0.0029	-0.0030	-0.0028	-0.0069	-0.0054
-0.0049	0.0017	0.0036	-0.0089	0.0036	0.0027
0.0009	0.0091	0.0026	0.0033	-0.0060	0.0070
-0.0007	0.0037	0.0048	0.0014	0.0024	0.0009
0.0002	0.0028	-0.0007	0.0028	0.0055	0.0056
-0.0001	0.0035	-0.0007	-0.0090	0.0028	-0.0062
0.0021	0.0065	0.0014	-0.0011	0.0072	-0.0006
0.0135	0.0002	0.0015	-0.0059	0.0025	-0.0012
-0.0029	0.0001	-0.0018	0.0007	0.0026	-0.0012
0.0065	-0.0031	-0.0030	-0.0003	0.0015	0.0146
-0.0053	0.0060	-0.0085	-0.0032	-0.0005	-0.0007



(a) Measurement 1



(b) Measurement 2

Fig. 11. Picture of membrane with extreme point marked

given. Fig. 11 shows the membrane picture and the 330 points selected. The point with the largest out-of-plane displacement 1.1 mm is marked. Change the tension pulling

the membrane to improve its flatness and perform measurement again, the largest out-of-plane displacement is now reduced to 0.18 mm. It takes only 0.1s to complete one measurement of 330 points coordinates.

After getting the above test results with satisfactory accuracy, we test the genetic algorithm on the flatness control of the membrane. The genetic algorithm generates proper candidate tension combinations, then the SMA actuator sub-controller guides the SMA actuators exerting these tensions. When desired values are obtained, the vision system records the corresponding membrane flatness values. First, tests are performed 20 times at room temperature. Each time the tension combinations converge very fast (with different speeds) to the optimal values. The average membrane flatness of the 20 tests is shown in Fig. 12. The average standard deviation goes down quickly from around 0.22mm to less than 0.05mm. Another 20 tests are then performed with local thermal load applied, which is realized by heating the membrane locally using the heater placed under it. Again optimal tension combinations are found and realized very well. The average membrane flatness of the 20 tests is shown in Fig. 13. The average standard deviation goes down quickly from around 0.33mm to less than 0.05mm.

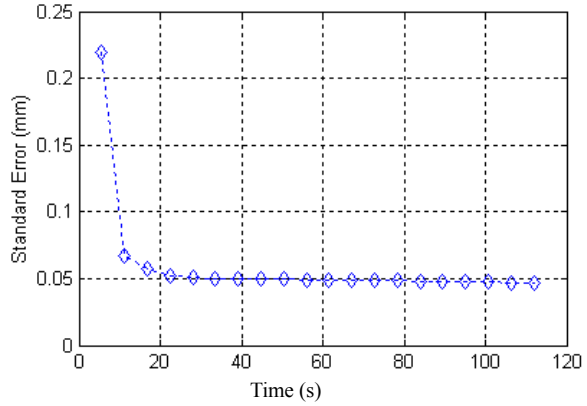


Fig. 12. The average membrane flatness at room temperature

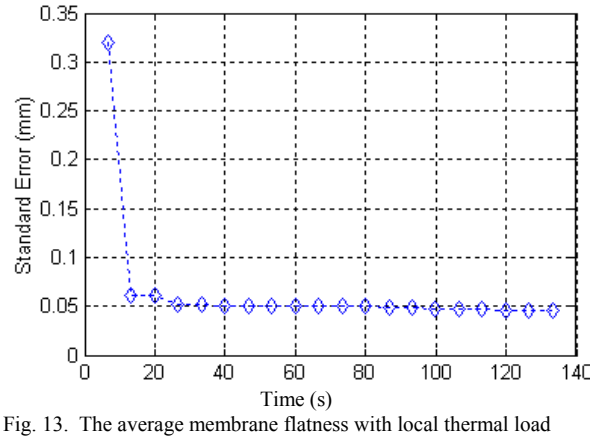


Fig. 13. The average membrane flatness with local thermal load

VII. Conclusion

A control system is developed for the active shape control of inflatable space structures. A sub-controller is designed and implemented to control the SMA actuator, and tests show that the required tensions can be achieved with satisfied accuracy. A vision system is realized to measure the membrane flatness, and test results indicate fast and accurate measurements are obtained. The genetic algorithm is implemented for control purpose with the hard ware in loop. Tests demonstrate that the developed system works very well in achieving good membrane shape.

References

[1] C. H. M. Jenkins (Editor), *Gossamer spacecraft: membrane and inflatable structures technology for space applications.* Progress in Astronautics and Aeronautics, Vol.191, 2001.

[2] D. Cadogan, and M. Grahne, "Inflatable Space Structures: A new paradagm for space structure design," in *the Proc. of the 49th International Astronautical Congress*, Sept. 28 - Oct. 2, 1998, Melbourne, Australia, Paper number: IAF-98-I.1.02.

[3] J. K. H. Lin, and D. P. Cadogan, "An inflatable microstrip reflectarray concept for Ka-band applications," in *the Proceedings of the 41st AIAA/ASME/ASCE/ AHS/ASC Structures, Structural Dynamics and Materials Conference & Exhibit*, April 3-6, 2000, Atlanta, Paper number:AIAA2000-1831.

[4] D. K. Karooka, and D. W. Jensen, "Advanced space structure concepts and their development," in *the Proceedings of the 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference and Exhibit*, April 16-19, 2001, Seattle, Paper number: AIAA-2001-1257.

[5] C. H. Hansen, M. T. Simpson and B. S. Cazzolato, "Genetic Algorithms for Active Sound and Vibration Control," in *the proceedings of the IEE International On-line Conference on Active Control of Sound and Vibration* (IEE Colloquium Inter-Active'99, Web Forum held by the IEE Control System Theory and Design Committee)

[6] L. Chambers (Eiditor), *The Practical Handbook of Genetic Algorithms: Applications*, Second Edition, CRC Press, 2000

[7] G. Song, V. Chaudhry, and V. Batur, "Precision tracking control of shape memory alloy actuators using neural networks and sliding-mode based robust controller," *Smart Materials and Structures*, Vol.12 , pp.223-231, 2003

[8] G. Webb, et al., "Control of SMA actuators in dynamic environments," in *the proceedings of SPIE Conference on Mathematics and Control in Smart Structures*, Vol. 3667, pp.278-289, 1999

[9] O. Heintze, S. Seelecke, and C. Buskens, "Modeling and optimal control of microscale SMA actuators," in *the proceedings of SPIE Conference on Smart Structures and Materials*, pp.495-505, 2003