Assessing 3-D Uncertain System Stability by Using MATLAB Convex Hull Functions

Mohammed Tawfik Hussein
Electrical Engineering Department, Faculty of Engineering
Islamic University of Gaza
Gaza City, Gaza Strip

***Abstract*— This paper is dealing with the robust stability of an uncertain three dimensional (3-D) system using existence MATLAB convex hull functions. Hence, the uncertain model of plant will be simulated by INTLAB Toolbox; furthermore, the root loci of the characteristic polynomials of the convex hull are obtained to judge whether the uncertain system is stable or not. A design third order example for uncertain parameters is given to validate the proposed approach.**

Keywords- Algorithm; 3-D convex hull; uncertainty; robust stability; root locus

I. Introduction

Dealing with higher order system can be considered a challenge and a difficult problem, therefore the contribution of this paper is to the utilization of existence built- in MATLAB Convex Hull algorithm and functions to handle such control problems with less time consuming as will be illustrated throughout this research paper.

A. Motivation and objectives

This paper is dealing with the robust stability of an interval or uncertain system. Developing an algorithm that checks robust stability of third order uncertain system such systems will be an efficient and helpful tool for control systems engineers.

B. Literature Review

The problem of an interval matrices was first presented in 1966 by Ramon E. Moore, who defined an interval number to be an ordered pair of real numbers [a,b], with $a \le b$ [1- 2].

This research is an extension and contribution of previous publications and ongoing research of the author [3]-[9].

C. Paper approach

Three dimension (3-D) convex hull approaches is utilized within MATLAB novel codes that is developed to assess 3-D uncertain system stability, and the algorithm associated is discussed and presented in this paper.

II. Uncertain Systems and Robust Stability

Due to the changes in system parameters due to many reasons, such as aging of main components and environmental changes, this present an uncertain threat to the system, therefore such a system need special type of control system called Robust to grantee the stability to the perturbed parameters. For instances , in recent research robust stability and stabilization of linear switched systems with dwell time[10], as well stability of unfalsified adaptive switching control in noisy environments [11] were discussed.

A. Robust D-stability

Letting D(p,q) denote the uncertain denominator polynomial, then the roots of D(p,q) lie in a region D as shown in Fig. 1 , then we can say that the system has a certain robust D-stability properly**.**

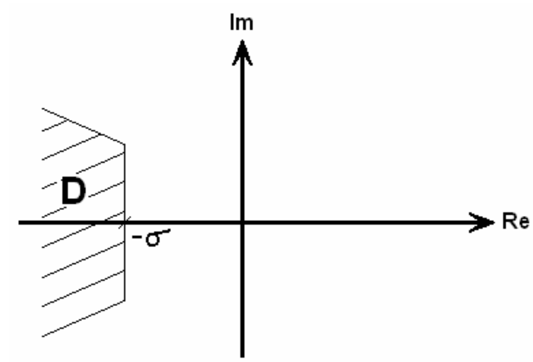

Figure 1. D-region

Definition 1: (D-stability)

Let $D \subseteq C$ and take P(s) to be a fixed polynomial, then P(s) is said to be D-stable if and only if all its roots lie in the region D.

Definition 2: (Robust-D-stability)

A family of polynomials $P=\{p(.,q):q \in Q\}$ is said to be robustly D-stable, if all $q \in Q$, p(.,q) is D-stable, i.e. all roots of p(.,q)lie in D region. For special case when D is the open unit disc, P is said to be robustly schur stable.

B. Edge Theorem

A polytope of a polynomial with invariant degree p(s, q) is robustly D-stable if and only if all the polynomials lying along the edges of the polytopic type are D-stable, the edge theorem gives an elegant solution to the problem of determining the root space of polytopic system [12], [13].

It establishes the fundamental property that the root space boundary of a polytypic family polynomial is contained in the root locus evaluated along the exposed edges, so after we generate the set of all segments of polynomials we obtain the root locus for all the segments as a direct location for the edge theorem.

C. Uncertain 3x3 systems

Third order uncertain systems can take the following general form:

$$A = \begin{bmatrix} [\overline{a}_{11}, \underline{a}_{11}] & [\overline{a}_{12}, \underline{a}_{12}] & [\overline{a}_{13}, \underline{a}_{13}] \\ [\overline{a}_{21}, \underline{a}_{21}] & [\overline{a}_{22}, \underline{a}_{22}] & [\overline{a}_{23}, \underline{a}_{23}] \\ [\overline{a}_{31}, \underline{a}_{31}] & [\overline{a}_{32}, \underline{a}_{32}] & [\overline{a}_{33}, \underline{a}_{33}] \end{bmatrix}$$

It has nine (9) elements which mean *2^9* possible combination of matrix family if all elements were uncertain. Generally, we have *2^n* possible combinations of an uncertain system where n is number of uncertain elements in the system. Characteristic equations for a general 3x3 matrix can be calculated as shown below in equation (1):

$$P(s) = s^3 - [a_{11} + a_{22} + a_{33}]s^2 + [a_{22}a_{33} - a_{23}a_{32} + a_{11}a_{33} + a_{11}a_{22} - a_{21}a_{12} - a_{31}a_{13}]s + \quad (1)$$
$$[-a_{11}a_{22}a_{33} + a_{11}a_{23}a_{32} + a_{21}a_{12}a_{33} - a_{21}a_{13}a_{32} - a_{31}a_{12}a_{23} + a_{31}a_{13}a_{22}]$$

The aim of this paper is to calculate the family of all possible combinations for a 3x3 uncertain matrix and so family of possible characteristic equations can be calculated. Then, using convex hull algorithm we will find exposed edges of calculated polynomials and so the roots of exposed edges to determine region of eigenvalues space for studied system.

D. Computing the Convex Hull of the Vertices

The convex-hull problem is one of the most important problems from computational geometry. For a set S of points in space the task is to find the smallest convex polygon containing all points [14].

Definition 1: A set S is convex if whenever two points P and Q are inside S, then the whole line segment PQ is also in S.

Definition 2: A set S is convex if it is exactly equal to the intersection of all the half planes containing it.

Definition 3: The convex hull of a finite point set S = {P} is the smallest 2D polygon Ω that contains S.

III. Methodology And Algorithm

The main goal of this research is to provide a simple and efficient algorithm to determine the bounds of an interval matrix that represent three dimensional problems, hence assess the stability of such an uncertain system by generating a MATLAB Algorithm for three by three interval matrix. Therefore the methodology and algorithm associated with will be discussed and presented in the following sections.

A. Input data and program call

The developed program takes the nine elements of 3x3 uncertain matrix in a vector form and is called in MATLAB command, and these elements can be either real number for specific elements or interval for uncertain matrix entries.

B. Calculating family of possible matrices

To obtain the family of all possible matrices then the following steps are performed within the function **<afamilynew4.m>**:

- Check for size of each input element to find position of uncertain elements.
- Declare input vector of 18 elements containing upper and lower values of elements, **coeff**, if element is specific then upper and lower values are equal.
- Calculate number of uncertain elements in matrix, **ss**.
- Calculate number of possible combinations, $\mathbf{2^{ss}}$.
- Weighting the coefficients vector by use of the function **<weig2.m>** that assign, at each combination, upper or lower values of elements by making use of the idea if binary numbers in combinations. This is done for all the **512 (2^9)** possible combinations.
- Calculate family of **512** possible matrices, **A**.
- Check for repeated matrices and delete it and make sure that remaining matrices are the $\mathbf{2^{ss}}$ unique matrices, **AA**.
- Calculate $\mathbf{2^{ss}}$ possible characteristic polynomials, **polypointsn**, according to equation 2.1. Note that polypointsn is a matrix and is expected to have the size of **(2^{ss},3**.

C. Find the 3-D convex hull of polynomials

For this purpose, the existence QuickerHull algorithm for convex hulls is utilized and incorporated in the main MATLAB Program [15], [6].

This algorithm has the advantage of being quicker than **convhulln**, the built-in code in MATLAB, as illustrated in Fig. 2.

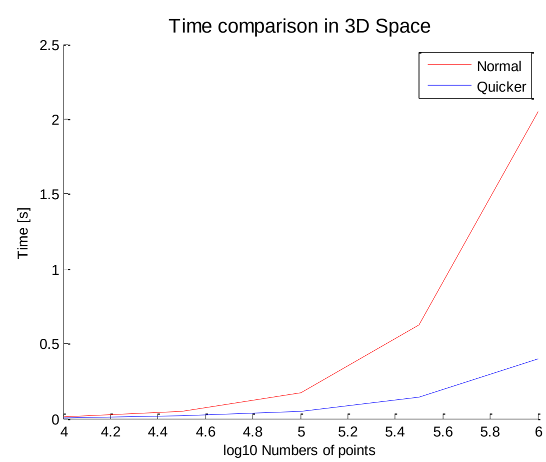

Figure 2. Comparison of processing time in 3D between normal and quicker hull

Then, 3D convex hull of system under study is plotted according to the MATLAB QuickerHull code shown below.

```
function tess=QuickerHull(P)
%  QuickerHull N-D Convex hull.
%    K = QuickerHull(X) returns the indices K of the points in
X that
%    comprise the facets of the convex hull of X. X is an m-
by-n array
%    representing m points in n-D space. If the convex hull has
p facets
%    then K is p-by-n.
%
%    CONVHULLN first attempts to clear points that cannot
be part of the
%    convex hullthan uses Qhull. For dimensions higher than
six and
%    point’s number less than 1000 no filtering is done.
%
%    Example:
%
%          X=rand (1000, 3);
%          tess=QuickerHull(X);
%    See also convhulln, convhull, qhull, delaunayn, voronoin,
%            tsearchn, dsearchn.
 %erroro check

if nargin>1
   error('only one input supported')
end

[N,dim]=size(P);

if dim>1
   if dim>5 || N<1000
      %run the normal convhull for high dimensions and a few
points
       tess=convhulln(P);
        return
   end
```

```
   else
      error('Dimension of points must be>1');
end

%%Filtering points

ncomb=2^dim;%number of combinations among the
dimensions
comb=ones(ncomb/2,dim);%preallocate combination
forbregion=zeros(2^dim,1);

%get all combinations using binary logic
for i=2:dim
   c=2^(dim-i);
   comb(:,i)=repmat([ones(c,1);-ones(c,1)],2^(i-2),1);
end
comb=[comb;-comb];%use  combinations simmetry

%for each combination get forbidden region point
```

```
for i=1:ncomb/2
   vect=zeros(N,1);
    for j=1:dim
      vect=vect+P(:,j)*comb(i,j);
    end
    [foo,forbregion(i)]=max(vect);
     [foo,forbregion(i+ncomb/2)]=min(vect);
end

%get the simplyfied forbidden region
%for each dimension get upper and lower limit

deleted=true(N,1);
for i=1:dim

%get combination with positive dimension
   index=comb(:,i)>0;

%upper limit
   simplregion=P(forbregion(index),i);
   upper=min(simplregion);

%lower limit
    simplregion=P(forbregion(~index),i);
    lower=max(simplregion);
    deleted=deleted & P(:,i)<upper & P(:,i)>lower;
 end

%delete points id that cannot be part of the convhull
index=1:N;
index(deleted)=[];

 %Run QuickHull with the survivors
tess=convhulln(P(~deleted,:));
 %reindex
tess=index(tess);
  end
```

D. *Calculate and sort roots of exposed edges of the convex hull*

First we calculate roots of all exposed edges, and plot the convex hull of all roots of possible characteristic equations.

In order to encircle only imaginary components of roots, we need a special sorting algorithm. We sort values according to real and imaginary axes. Sorting may lead into "zero" values in the imaginary matrix if there are real distinct or repeated roots. So, a process of searching for "zeros" in the imaginary matrix is held by replacing any zero vectors in the imaginary by the following one.

IV. PROGRAM OUTPUTS AND RESULTS

For testing research paper’s program, an example of 3x3 uncertain system of a printer belt-drive system is presented and simulated to validate the proposed approach.

A. Design Numerical Simulation Example

The method proposed in this paper will now be demonstrated using printer belt-drive system that is described mathematically by equation (2) and shown below in Fig. 3.

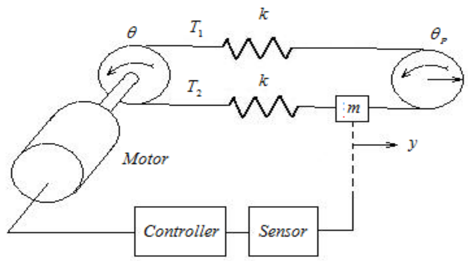

Figure 3. Printer belt-drive system

The system has the following model, see [17]-[19].

$$\dot{x} = \begin{bmatrix} 0 & -1 & r \\ \frac{2k}{m} & 0 & 0 \\ \frac{-2kr}{J} & \frac{-K_m k_1 k_2}{JR} & \frac{-b}{J} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \frac{-1}{J} \end{bmatrix} T_d \qquad (2)$$

Where;

r: radius of pulleys [m].
k:spring constant of the belt.
k_1:light sensor constant [V/m].
b: internal friction of the motor [Nms/rad].
R: coil resistance of motor [Ω].
K_m: motor constant [Nm/A].
J: total inertia of motor and pulleys [kgm^2].
T_d: disturbance torque [Nm].

Values of some parameters of this model can vary in the following manner as shown in Table 1:

TABLE I. MODEL PARAMETERS

Parameters	***Values***
m	0.2
k1	1
k2	0.1
r	0.15
b	[0.1 0.25]
R	[1.5 2.5]
Km	2
J	0.01

Then the A matrix is obtained with both lower and upper parameters values as given below:

$$A = \begin{bmatrix} 0 & -1 & 0.15 \\ [10 \quad 400] & 0 & 0 \\ [-1200 \quad -30] & [-13.33 \quad -8] & [-25 \quad -10] \end{bmatrix}$$

B. Source code

With four uncertain elements as shown in matrix A; thus, we expect a family of 16 matrices. By calling the main MATLAB program with the incorporated Quicker Hull functions as presented below:

```
function main4(a11,a12,a13,a21,a22,a23,a31,a32,a33)

polypointsn =
afamilynew4(a11,a12,a13,a21,a22,a23,a31,a32,a33);
size(polypointsn)
tess=QuickerHull(polypointsn)
figure(4)
trisurf(tess,polypointsn(:,1),polypointsn(:,2),polypointsn(:,3))
r=zeros(length(tess),3);
for n=1:length(tess)
   r(n,:)=(roots([1 tess(n,:)]))';
   if imag(r(n,1))~=0
      temp=r(n,3);
      r(n,2:3)=r(n,1:2);r(n,1)=temp;
   end
end
r
x=real(r)
y=imag(r)

figure(5)
plot(x,y,'b+')
hold on;
k=convhull(x,y);
plot(x(k),y(k),'-r')
grid on

q=length(r)
v=0;
z=zeros(1,3);
for i=1:q-1
   if y(i,:)==z
      v=v+1
           z
      for n=i:q-1
         y(n,:)=y(n+1,:);
         x(n,:)=x(n+1,:);
      end
   end
   v1=v;
   y;
   if i==q-1-v
      break
   end
   for s=1:v
      if y(i,:)==z
```

```
        v=v+1
        for n=i:q-1
          y(n,:)=y(n+1,:);
          x(n,:)=x(n+1,:);
        end
      end
    end
end


yn=y(1:q-v,:)
xn=x(1:q-v,:)
v
length(yn)
length(xn)

xxp=xn(:,3);
yyp=yn(:,3);
xxn=xn(:,2);
yyn=yn(:,2);

kp=convhull(xxp,yyp);
kn=convhull(xxn,yyn);
figure(6)
plot(x,y,'b+')
hold on;
plot(xxp(kp),yyp(kp),'-r'), plot(xxn(kn),yyn(kn),'-r')
hold off
grid on

figure(7)
plot(xxp(kp),yyp(kp),'-r'),hold on, plot(xxn(kn),yyn(kn),'-r')
axis equal
grid on
%plot(real(r),imag(r))
%rtess=QuicherHull(r)
end
```

C. Program output

Our proposed program is supposed to show all of family of possible matrices in addition to roots values of characteristic equations.

Four figures are generated while processing our algorithm, Fig. 4 shows 3D convex hull of polynomials. Fig. 5 shows roots locations on s-plane and encircles them by a convex hull. While Fig. 6 shows encircling only imaginary parts of roots, i.e. an identical polygons are generated on and below real axis and other roots are shown also. Fig. 7 focuses on identical polygons encircle imaginary parts of roots.

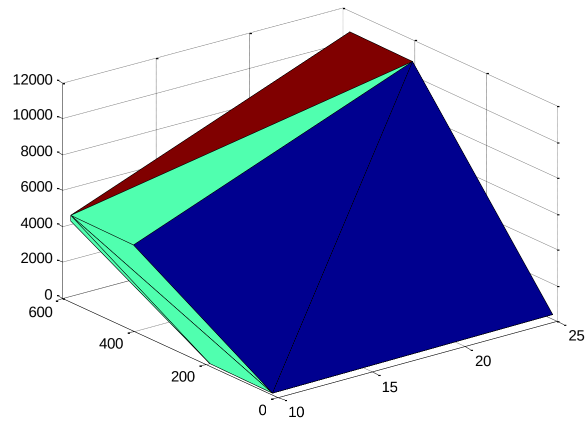

Figure 4. 3D convex hull of system polytopes

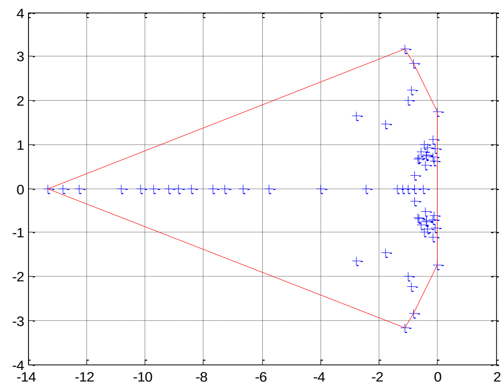

Figure 5. Convex hull of characteristic polynomials roots

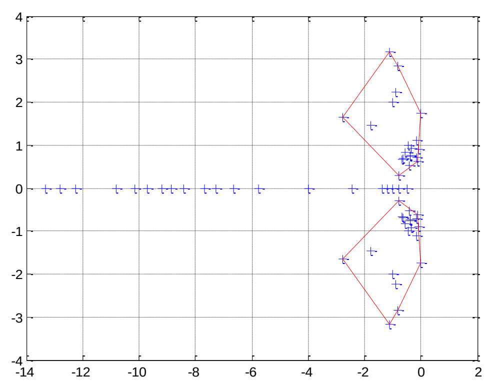

Figure 6. Convex hull of imaginary part of characteristic polynomials roots

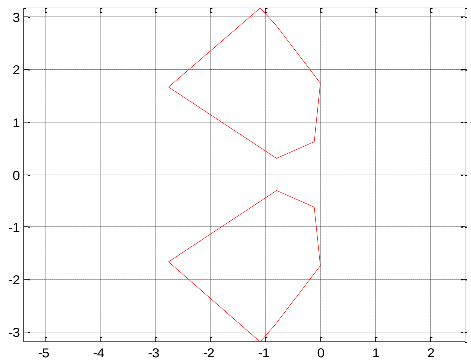

Figure 7. Convex hull of imaginary roots in focus

V. Conclusion And Future Work

In this paper, the stability of uncertain system using convex hull algorithm was tested. And we use MATLAB and INTLAB toolbox to write program that can plot the 3D-convex hull, root loci, step response, and frequency response for any uncertain system. This paper tested the robust stability of an interval 3x3 matrix by the implementation of Printer Belt-Drive System. An efficient and enhanced algorithm was introduced and improved for this purpose.

This algorithm can be easily extended to deal with higher order matrices (n-dimensional system) without a very large increase of processing time.

Acknowledgment

The author wishes to thank his graduate students who enrolled in his course titled “Uncertain Control Systems” during spring 2010 for their inputs, simulations and contribution to this research paper, especially in selecting, testing and running practical engineering examples.

References

[1] R.E. Moore, Interval Analysis. Englewood Cliffs, N.J.: Prentice Hall, Inc., 1966.

[2] L.V. Kolev, “ Interval Mathematics Algorithms for Tolerance Analysis,” IEEE Trans. On Circuits and Systems, 35, pp. 967-975, 1988.

[3] M. T. Hussein," An Efficient Computational Method For Bounds of Eigenvalues of Interval System Using A Convex Hull Algorithm", The Arabian Journal for Science and Engineering, Volume 35, Number 1B, pages 249-263, April 2010,

[4] M.T. Hussein, Lecture Notes for Control of Uncertain Systems Course Faculty of Engineering, IUG, Jan- May 2010.

[5] M.T. Hussein,” Using a Convex Hull Algorithm in Assessing Stability of a DC Motor with Uncertain Parameters: A computational Approach”, IJCSSEIT (ISSN:0974-5807), PP 23-34, Vol. 1, No. 1, January- June 2008.

[6] M.T. Hussein,” A novel Algorithm to Compute all Vertex Matrices of an Interval Matrix: Computational Approach”, International Journal of Computing and Information Sciences, IJCIS (ISSN 1708-0460), 2(2) (2005), pp.137-142.

[7] M.T. Hussein, “A Method to Enhance Tolerance Frequency Analysis of Linear Circuits: Computational Approach”, Al Azher University Engineering Journal, 8(11) (, pp. 373-386, 2005

[8] M. T. Hussein, “A Method for Determination of an Eigenvalue Bounds for an Infinite Family of Interval Matrices: A Computational Approach”, the First International Conference of Science and Development (ICSD-I), March 1-2, 2005.

[9] M. T. Hussein, “Response Analysis of RLC Circuit with an Interval Parameters”, Engineering Week Conference, Gaza, Palestine, December 2004.

[10] L. I. Allerhand and U. Shaked, “Robust Stability and Stabilization of Linear Switched Systems with Dwell Time”, IEEE Trans. on Automatic Control, 56 (2), pp.381-386, 2011.

[11] G. Battistelli, E. Mosca, M. G. Safonov, and P. Tesi, “Stability of Unflasified Adabtive Switching Control in Noisy Envrionments”, IEEE Trans. on Automatic Control,55 (10) ,pp.2424-2429 , 2010.

[12] S.P. Bhattacharyya, H. Chapellat, and L.H. Keel, Robust Control: The Parametric Approach. Prentice Hall, 1995.

[13] S.R. Ross and B.R. Barmish,” A sharp Estimate for the Probability of Stability for Polynomials with Multilinear Uncertainty Structure”, IEEE Trans. on Automatic Control, 53(2), pp.601-606, 2008.

[14] Ketan Mulmuley, Computational Geometry, an Introduction through Randomized Algorithms, Prentice hall, 1994.

[15] INTLAB toolbox User’s Guide Version 3 by math works, http://www.ti3.tu-[harburg.de/rump/intlab/, last visit 20/5/2010.

[16] Luigi Giaccari, http://giaccariluigi.altervista.org/blog/, Created:04/01/2009, last update 08/012009.

[17] C.Bradford Barber, David P. Dobkin, Hannuhuhdanpaa," The Quickhull Algorithm for Convex Hulls", ACM Transactions on Mathematical Software, Vol. 22, No. 4, Pages 469–483, December 1996.

[18] R. C. Dorf, R. H. Bishop, Modern Control Systems, 10th ed., Prentice Hall, Upper Saddle River, NJ 07458, 2005.

[19] K. Ogata, Modern Control Engineering, 4th ed., Prentice Hall, Upper Saddle River, N.J., 2002.

Authors Profile

Dr. Mohammed T. Hussein, an Associate Dean of Engineering for Research and Development and an associate Professor of Electrical Engineering joined the department of Electrical and Computer engineering at Islamic university on August 2003. Dr. Hussein was named Director of e-Learning Center on November 1, 2003. Prior to this appointment he served as a department Head of Engineering Technology in College of Engineering at Prairie View A&M University, Texas. Dr. Hussein earned a Ph.D. degree in electrical engineering from Texas A&M University, College Station, Texas, USA. Dr. Hussein is a registered professional engineer (P.E.) in the State of Texas. Dr. Hussein worked for Motorola Inc., in Tempe, Az., and Oak Ridge National Laboratory in state of Tennessee. His research interests include robust control systems, computer algorithms and applications, and e-Learning. Dr. Hussein holds scientific and professional memberships in IEEE(SM), Eta Kappa Nu, and Tau Beta Pi. He is the recipient of numerous national, state, university, college, and departmental awards including “Who's Who among America’s Best Teachers” on 2000, “Marquis Who’s who among World Leaders” on 2010, and “Teaching Award” in the College of Engineering. Dr. Hussein was nominated and selected on 2003 as an evaluator for Accreditation board for Engineering and Technology (ABET), USA. Dr. Hussein spent summer 2008 as a DAAD visiting Professor at Berlin Technical University, Germany, and on 2009 was selected as academy Fellow, Palestine Academy for Science and Technology.