Estimation Theory
ECEN 5523
Dr. Martin Hagan

Spring 2012

Final Project Report
May 02 2012

Prepared by
Srinivas Sambaraju

Table of Contents

Project Report

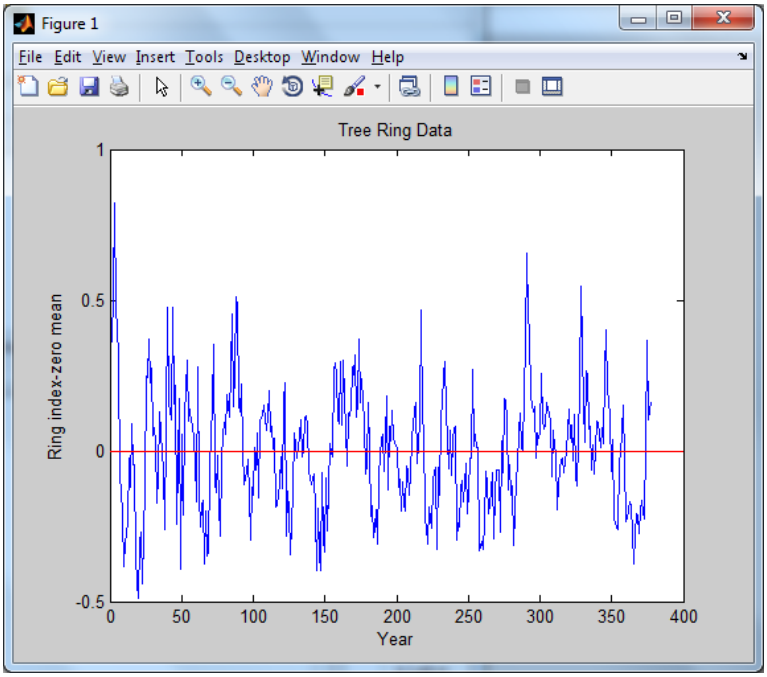
Introduction: An overview of the system identification process; an outline of the report

I have collected data about Tree Ring index
from http://robjhyndman.com/tsdldata/data/indi002.dat
The data has 377 points collected for that many years.

First we need to identify the model that fits the data closely, that is to find if AR, MA or ARMA Process. Once that is identified we need to estimate the parameters (a1, a2,…ana, b1, b2, …bnb). We use Maximum Likelihood estimation for this. Finally need to verify if the estimated parameters fit the confidence interval. If the parameters don't fit the confidence interval, the order of the process should be changed (increased or decreased) and the parameters are re-estimated.

Description of the data set - Plot of the data

The data is initially not zero mean. I subtracted the mean (y-mean(y)) to start the project. Below is a plot of the data.



Descriptions of the types of models that are to be considered

There are three types of models under consideration. They are AR, MA, and ARMA. Based on observation of the Autocorrelation plot and GPAC data, theoretical models can be classified as AR, MA and ARMA. For experimental data the values fluctuate close to zero and are not completely equal to zero as in the theoretical case. Listed below are the observations that can be used for this project.

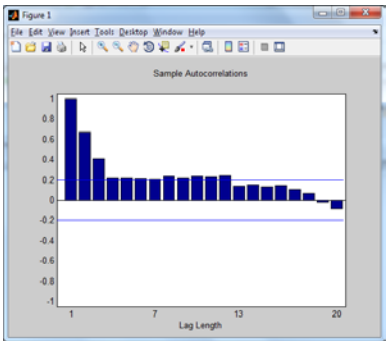
1. MA (b1..bnb) process - ACF goes to Zero after 'bnb' and PACF decays geometrically
2. AR (a1..ana) process - PACF goes to Zero after 'ana' and ACF decays geometrically
3. ARMA (a1..ana, b1..bnb) process - ACF and PACF both decay geometrically.

If the process is ARMA(ana, bnb), ACF and PACF show a geometric decay and it's almost impossible to recognize lag orders from the ACF and PACF only. So a regressive approach will be used in this project.
Below is a mathematical equation to describe the models

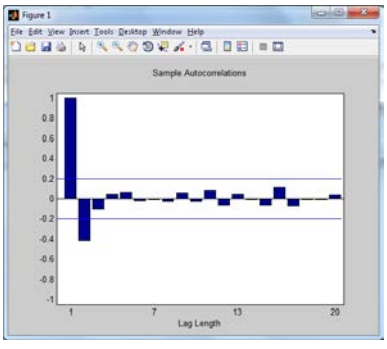
MA : $y(t) = e(t)+b1*e(t-1)+b2*e(t-2) \ldots + bnb*e(t-nb)$
AR : $y(t) = e(t) -a1*y(t-1)-a2*y(t-1)- \ldots -ana*y(t-na)$
ARMA: $y(t) = e(t)+b1*e(t-1)+b2*e(t-2) ..+ bnb*e(t-nb) -a1*y(t-1)-a2*y(t-1)- .. -ana*y(t-na)$

In this project we will use the GPAC instead of the PACF to determine the process and its order. Below are plots for AR, MA and ARMA models based on some simulated data samples. Some knowledge about the measured data vector can be inferred from these plots

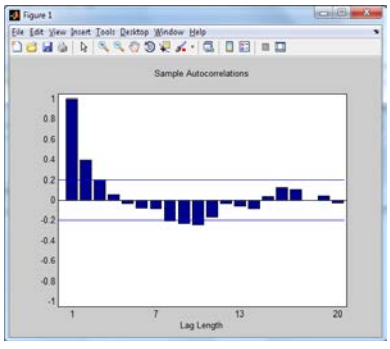
ACF of an AR process



ACF of a MA process



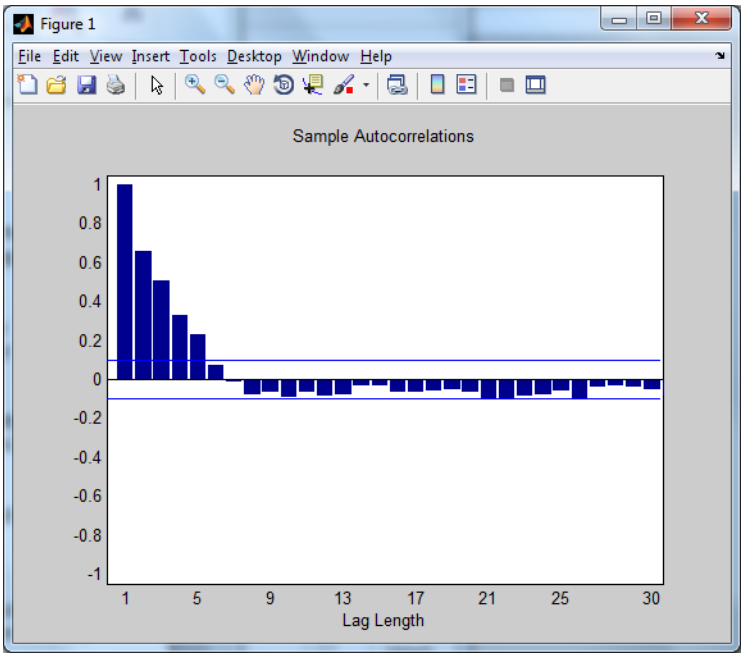
ACF of an ARMA process



Preliminary model development procedures and results

Below is a discussion of the autocorrelation function and the GPAC with a plot of the autocorrelation function and the GPAC table.

Plot of the Auto Correlation function



GPAC of Autocorrelation function of the Tree Ring's data

nb\na	1	2	3	4	5	6	7	8	9	10
0	0.6557	0.1305	-0.0831	0.0091	**-0.1393**	-0.0375	-0.0300	0.0622	-0.0399	0.0346
1	**0.7692**	0.5409	-0.0689	-1.2640	-0.1417	0.0737	-0.1077	0.0430	0.0139	-0.0333
2	0.6475	0.2328	-1.7623	-0.1822	-0.1304	-0.5408	-0.1841	0.0537	0.1528	-0.0982
3	0.7044	3.0260	-1.1710	0.8975	-0.0701	**-0.1705**	-0.0722	-0.0766	-0.1930	0.2280
4	0.3272	-0.2454	-0.3930	0.3554	-1.0791	-0.1501	0.0377	0.3647	-0.3978	-0.1242
5	-0.0745	-0.8118	-0.5404	-0.3654	-0.3677	-0.1824	2.2492	0.6014	-0.2770	0.4648
6	13.302	-0.9263	-0.7463	0.7374	-0.2481	0.7934	-0.6144	-0.0708	0.0296	0.2905
7	0.8006	0.5854	-0.3976	-0.0205	0.5185	-0.2345	-0.7308	-0.3583	0.7977	0.2907
8	1.4878	1.4727	-0.4117	-10.345	0.4898	-2.3282	-0.7526	6.2516	-2.5873	0.2945
9	0.6381	0.8423	0.9026	-0.5380	-2.2291	-0.9143	-0.1765	0.5056	0.4324	0.0851

From the ACF it's not clear if the process is AR or ARMA, though roughly it looks like an ARMA process. Since its not oscillating it might not have complex roots. From the GPAC array of 10 x 10 size, there are at least four 'ana' values that can be considered. They are 0.6557 (na=1, nb=0), 0.7692 (na=1, nb=1), -0.1393 (na =5, nb =0), -0.1705 (na=6, nb=3)

Final model development

Statistics included in the Final Model

- Parameter estimates
- The Standard deviation of the parameter estimates
- The variance of the residuals
- The autocorrelation function of the residuals and
- The GPAC of the residuals

Criteria used to explain why the model is chosen

- Pattern of GPAC
- Variance of residuals
- Chi-square statistic for whiteness of residuals
- Number of parameters

Discussion of maximum likelihood parameter estimation and diagnostic checking

Two different models (ARMA and AR) are tried and compared in this section. First the model is identified, with the number of parameters (theta.na, theta.nb) and estimated using Maximum likelihood process. The parameters are verified using a series of available statistics to make sure that the best possible model order and parameters are chosen.

The maximum likelihood estimation process tries to estimates the parameters that most likely were used in creating the measured data. The Theta vector is initialized to zero at first and then least squares method is used inside the Levenberg- Marquardt algorithm (LM) to gradually improve the estimate. As long as the sum squared error the residuals is smaller than the previous value, 'Mu' value is decreased and we move by a small step in the negative gradient direction towards the solution.

The LM algorithm linearizes the error with respect to the parameters (using A, G matrices) and uses least squares method. It's a trick because for a ARMA model, the error is not a linear function of parameters. Updating 'Mu' based on sum square error(STh) helps in introducing the non-linear least squares approach in the LM algorithm. We use this to gradually converge to the solution.

Criteria used for a good fit and model selection

- Q < ChiSquare value from the table for Alpha = 0.05 and the calculated ChiSqDof
- Parameters should fall in the confidence interval and the better estimate will have the smallest confidence interval or small standard deviation (of the order 1E-02)

- Auto correlation of the residuals should look like an impulse function for white noise. For this experimental data there are small spikes other than the first value. A plot is given in the appendix.
- It's better if the value zero does not fall in the confidence interval. The parameters should not be closer to zero compared to others, should be like 1E-01.
- Follow the pattern in the GPAC and try to see if the 'na', 'nb' values inferred from the 'ana' value reduces the Sum-squared error (STh), Residual variance, reduces the Q value and Standard deviation for the next iteration.

To start with, based on the GPAC of the ACF of the data na=1 and nb=0 is chosen. From here based on GPAC of the ACF of the residuals we proceed to the next values of na, nb and hence try different models.

na=1; nb = 0; GPAC of the ACF of the Residuals

nb(j)\na(i)	1	2	3	4	5	6	7	8	9	10
0	-0.0858	0.1267	-0.0011	0.1007	-0.0536	-0.0534	-0.1057	0.0199	-0.0504	0.0412
1	-1.5515	0.1259	11.393	0.1001	-0.1537	0.0523	-0.1157	-0.2472	-0.0342	0.0300
2	-0.1648	0.9524	0.1312	0.2026	-0.2173	-0.6202	-0.1051	-0.0330	-0.1041	-0.1156
3	-5.3309	0.8034	-1.3222	0.0585	0.0221	-0.1466	-0.0874	0.2543	-0.1791	0.4289
4	-0.6058	-0.5689	-0.9287	0.4133	0.4503	-0.1567	-0.1594	-0.2815	-0.6181	0.2950
5	0.2267	1.1200	-1.0040	-5.7266	-2.3417	-0.1253	-0.3874	0.4557	-0.2887	0.0968
6	7.0160	1.7234	-0.4856	0.3732	-0.4412	1.3226	-0.1764	-0.1869	-0.2073	0.1594
7	-0.3399	0.6584	0.1981	-0.2384	0.4667	0.6427	-0.9394	0.0890	-0.2478	0.2393
8	-2.3710	0.7216	1.3135	0.3972	1.1422	0.9934	-1.2026	-3.7737	-0.2646	0.5001
9	-0.5521	0.3073	2.5420	-9.6733	1.6422	0.2388	-0.3293	-0.1118	0.2192	0.2137

Statistics	
Converged in Iterations	2
SthNu (sum sq error)	8.80840
Residual Variance	0.02343
Q (Sum sq of Rho)	33.33300
ChiSqdof	19
ChiSq Value (Table)	30.14400
Standard Deviation	0.03894
Estimated Parameters	-0.65688
Confidence Intervals	-0.73477
	-0.57899
Parameters	a1

Discussion: Q > ChiSq Value (Table). Fails Chi Square test for Alpha = 0.05

na=1; nb = 1; GPAC of the ACF of the Residuals

nb(j)\na(i)	1	2	3	4	5	6	7	8	9	10
0	-0.0126	0.0818	-0.0369	0.0792	-0.0760	-0.0635	-0.1064	0.0150	-0.0572	0.0384
1	-6.5253	0.0762	0.1387	0.0441	-0.1418	0.0635	-0.1153	-0.3904	-0.0471	-0.0037
2	-0.4709	0.8927	0.0234	0.2421	-0.1588	-0.4514	-0.0974	-0.0145	-0.0418	2.2850
3	-2.2297	0.7603	-7.6471	0.1730	-0.0761	-0.1735	-0.0869	0.2546	-0.1320	0.2856
4	-0.9563	-2.5666	-2.0117	-0.7859	-0.2816	-0.1500	-0.1415	-0.5791	-1.3222	0.2527
5	0.5766	0.7134	-1.2728	3.2096	-1.4126	-0.1140	-0.5268	0.4786	-0.3240	0.1062
6	2.5598	2.0281	-0.6121	0.3617	-0.3722	1.4507	-0.2447	-0.1729	-0.2057	0.1114
7	-0.1653	0.6042	0.0100	-0.1972	0.6660	0.5889	-0.7936	0.2233	-0.2399	0.2087
8	-4.0562	0.6062	12.120	-0.1450	1.1524	1.1220	-1.1653	-1.7301	-0.2722	0.5483
9	-0.5392	0.3483	3.3368	25.336	1.0493	0.0387	-0.3851	-0.1263	0.1974	0.2020

Statistics		
Converged in Iterations	5	
SthNu (sum sq error)	8.69020	
Residual Variance	0.02317	
Q (Sum sq of Rho)	25.74200	
ChiSqdof	18	
ChiSq Value (Table)	28.86900	
Standard Deviation	0.05165	0.07648
Estimated Parameters	-0.74407	-0.15261
Confidence Intervals	-0.84737	-0.30558
	-0.64077	0.00035
Parameters	a1	a2

Discussion: Passed Chi square test, Q is still high, low standard deviation. Parameters are closely in confidence intervals. A decent model but can't confirm that from GPAC of residuals. Patterns are not clear.

na=2; nb = 2; GPAC of the ACF of the Residuals

nb(j)\na(i)	1	2	3	4	5	6	7	8	9	10
0	-0.0003	0.0069	0.0389	0.0371	-0.0212	-0.0923	-0.0794	-0.0051	-0.0454	0.0373
1	-19.749	0.0088	0.0324	0.0593	-0.1822	-0.0741	-0.0736	0.7010	-0.0496	0.0131
2	5.6622	-20.733	0.0297	-0.0005	0.0008	-0.0386	-0.1260	-0.0182	-0.0666	-0.5215
3	0.9507	-1.7260	0.0020	0.0525	-0.0211	-0.0399	-0.1300	0.4553	-0.1155	0.2418
4	-0.5600	-1.7250	41.184	0.1448	-2.6288	-0.0422	-0.1256	-0.4026	-0.8142	0.2189
5	4.3346	-1.7330	-80.764	*-1314.9*	-2.3411	-0.4972	-0.1210	0.0758	-0.1590	0.0070
6	0.8403	-0.7948	2.7806	-8.9381	*28.227*	-6.6333	-0.1324	-0.2255	-0.1563	1.7586
7	0.0779	0.7968	0.1379	-0.2163	0.8009	0.3186	-1.1979	0.1391	-0.1989	-0.0488
8	9.3640	0.7698	1.3262	0.2817	0.8845	3.4747	-1.4956	-2.6306	-0.1896	-1.3331
9	-0.4416	0.2237	2.7356	-9.3054	1.0000	-0.2972	0.0236	-0.4839	0.6754	-0.0442

Statistics				
Converged in Iterations	6			
SthNu (sum sq error)	8.52540			
Residual Variance	0.02286			
Q (Sum sq of Rho)	18.88500			
ChiSqdof	16			
ChiSq Value (Table)	26.29600			
Standard Deviation	0.10593	0.09600	0.11894	0.08606
Estimated Parameters	0.16492	-0.63316	0.75854	-0.03521
Confidence Intervals	-0.04694	-0.82517	0.52065	-0.20732
	0.37679	-0.44116	0.99643	0.13690
Parameters	a1	a2	b1	b2

Discussion: Passed Chi square test, Q is lower than previous case, low standard deviation for a2 and b2 and those parameters are closely in confidence intervals. A decent model but still can't confirm that from GPAC of residuals. Patterns are not clear.

na=6; nb = 2; GPAC of the ACF of the Residuals

nb(j)\na(i)	1	2	3	4	5	6	7	8	9	10
0	-0.0007	0.0008	0.0066	-0.0051	0.0110	-0.0138	-0.0424	0.0558	-0.0334	0.0584
1	-1.1540	0.0065	0.0072	0.0091	0.0046	-0.0475	-0.0606	0.0305	0.0641	0.0483
2	8.6715	-9.6202	0.0079	-0.0009	0.0598	-0.0333	-0.0503	0.0882	-0.0029	0.1157
3	-0.7742	0.9849	0.1226	0.5635	0.0659	-0.0603	-0.0602	0.0914	3.6385	0.1171
4	-2.1599	1.0911	65.835	-8.2291	0.0127	-0.0069	-0.0422	0.0054	-0.0134	0.0563
5	-1.2557	-12.968	-7.2622	-6.6472	-3.5832	-0.0760	-0.0425	-0.0925	0.0095	0.0529
6	3.0674	-3.9072	5.9172	-5.8947	36.408	-20.263	-0.0435	-0.1712	0.8889	0.0554
7	-1.3179	-0.6614	-1.1466	-1.7572	-0.2064	0.4992	-1.9890	0.0210	-0.0976	-0.0264
8	-0.6004	1.2365	0.0634	-1.6626	-4.4802	-0.3435	-2.2246	-10.362	-0.1039	0.7022
9	-1.7217	1.3249	34.290	-1.6718	-7.1194	48.638	-2.4141	0.6728	4.4190	0.1069

Statistics								
Converged in Iterations	20							
SthNu (sumsq error)	8.35090							
Residual Variance	0.02263							
Q (Sum sq of Rho)	14.5570							
ChiSqdof	12							
ChiSq Value (Table)	21.0260							
Standard Deviation	0.33359	0.14373	0.25553	0.07997	0.06048	0.05637	0.33627	0.33202
Estimated Parameters	0.12835	-0.86788	0.11599	-0.00863	0.03609	0.14168	0.71259	-0.29813
Confidence Intervals	-0.5388	-1.15530	-0.39508	-0.16857	-0.08487	0.02893	0.04005	-0.96216
	0.79552	-0.58042	0.62706	0.15131	0.15706	0.25442	1.38510	0.36590
Parameters	a1	a2	a3	a4	a5	a6	b1	b2

Discussion: Passed Chi square test, Q is lower than previous case, low standard deviation for a4, a5, and a6 and those parameters are closely in confidence intervals. A decent model but still can't confirm that na=6 and nb=2 from GPAC of residuals. Patterns are not clear.

na=7;nb=1 is not a better fit than na=7;nb=2; It has higher Q, the parameters are smaller, standard deviation is more and Chi-square alpha value is larger as well. Considering the next option, a better fit obtained by na=7;nb=2; the details are given below.

na=7; nb = 2; GPAC of the ACF of the Residuals

nb(j)\na(i)	1	2	3	4	5	6	7	8	9	10
0	-0.0027	-0.0027	0.0041	-0.0004	0.0140	-0.0115	-0.0259	0.0050	-0.0032	0.0428
1	0.9760	-0.0068	0.0038	0.1444	0.0137	-0.0430	-0.0282	-0.0111	0.0654	0.0395
2	-1.5307	-0.8526	0.0203	0.0116	0.0242	-0.0186	-0.0297	-0.5417	0.2119	0.0505
3	-0.1002	3.6650	-2.0925	-0.0273	0.0173	-0.0405	-0.0474	0.0103	-0.0774	-0.0130
4	-34.300	3.3620	-7.0737	-4.4892	-0.0006	-0.0165	-0.0421	-0.3524	-0.0798	0.4062
5	-0.8292	-2.6042	-1.9947	-4.6510	122.44	-0.0156	-0.0267	-0.0168	-0.0250	0.0685
6	2.2356	-1.4771	2.3652	-2.7581	7.0273	-12.040	-0.0232	0.0209	-0.0567	0.0546
7	-0.2066	0.0721	-1.3997	-0.3054	2.5731	-1.6160	-1.4799	-0.1199	-0.0823	0.0482
8	-0.5881	-4.1315	-1.4248	-10.793	2.4515	-3.6562	-9.7189	6.7246	-0.0506	0.0639
9	-13.564	7.6859	-1.9850	0.3208	1.6911	4.5713	4.3645	2.5521	3.2236	0.0515

Statistics									
Converged in Iterations	31								
SthNu (sumsq error)	8.17390								
Residual Variance	0.02221								
Q (Sum sq of Rho)	7.82080								
ChiSqdof	11								
ChiSq Value (Table)	19.6750								
Standard Deviation	0.05496	0.06837	0.06870	0.07705	0.06769	0.06801	0.05278	0.02204	0.02196
Estimated Parameters	0.83159	-0.03921	-0.72256	-0.16425	0.04603	0.11611	0.18784	1.43220	0.97065
Confidence Intervals	0.72168	-0.17595	-0.85997	-0.31836	-0.08936	-0.01991	0.08229	1.38810	0.92673
	0.94151	0.09753	-0.58515	-0.01015	0.18142	0.25214	0.29339	1.47620	1.01460
Parameters	a1	a2	a3	a4	a5	a6	a7	b1	b2

Discussion: The final model that could be chosen might be an ARMA model with na=7; nb = 2; The reason is this model has the lowest value of "Q" compared to others which means that the Autocorrelation of the residuals is closest to an impulse, and this signifies that the residuals are similar to white noise or random. The standard deviation is lower and hence the confidence intervals are very closer to the parameter values. values of a1, a3, b1, b2 are higher compared to other parameters and the model is well balanced in terms of a' and 'b' parameters.

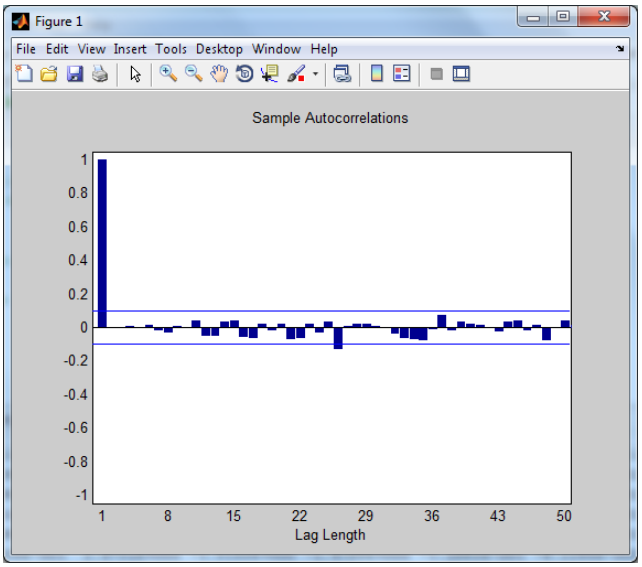
The alternate model to be considered is an AR(5) model given below. It was selected based on the GPAC of the autocorrelation of the data. It has a very good fit with a low standard deviation, small variance. The Q value is small. However the parameter values are not as high as expected, except 'a1' remaining are smaller and 'a3', 'a4' are closer to zero which means that there is a chance that some coefficients may be insignificant compared to the others. Also its selection is not easy to justify from the GPAC as the values up to na=5 are small as well. Also the model converged in just **2 iterations**. The reason is that AR is a linear model and LM algorithm is best suited to this. So the convergence alone cannot be a criteria for model selection.

na=5; nb = 0; GPAC of the ACF of the Residuals

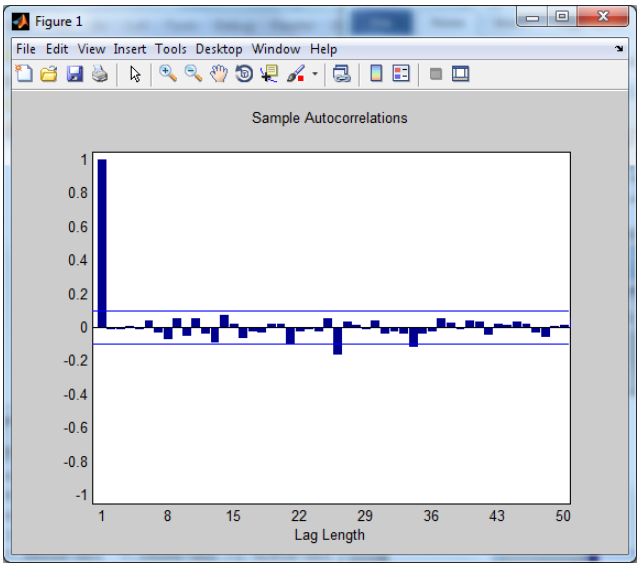
nb(j)\na(i)	1	2	3	4	5	6	7	8	9	10
0	-0.0057	-0.0040	0.0064	-0.0038	0.0384	-0.0242	-0.0639	0.0492	-0.0454	0.0530
1	0.6999	-0.0130	0.0039	0.0604	0.0361	-0.1254	-0.0825	-0.0095	0.0119	0.0268
2	-1.6024	-0.4869	0.0797	0.0207	0.0492	-0.0265	-0.0867	-0.1332	0.0389	0.0639
3	-0.6029	9.0337	-2.3438	-0.1038	0.0451	-0.1533	-0.0888	0.1962	-0.4549	-0.0146
4	-9.9490	5.9750	-14.521	-6.2326	0.0160	-0.0284	-0.0806	-0.3127	-0.5349	4.2251
5	-0.6392	-2.2115	-1.1968	-4.0357	-7.1361	-0.0654	-0.0706	0.0450	-0.1003	0.0998
6	2.5984	-1.7375	5.6457	-3.2898	9.4129	-10.099	-0.0758	-0.1350	-0.0639	0.0566
7	-0.7887	0.0719	-0.1428	-0.3041	1.2278	0.7413	-1.4043	0.0086	-0.1072	0.0694
8	-0.9075	-1.5592	-0.2998	-0.8274	1.3874	3.2379	-1.4839	-18.621	-0.1110	0.1564
9	-1.1664	2.3989	-5.0302	0.0377	1.1364	1.0965	0.8188	0.5450	0.8115	0.1085

Statistics					
Converged in Iterations	2				
SthNu (sum sq error)	8.42060				
Residual Variance	0.02264				
Q (Sum sq of Rho)	16.17600				
ChiSqdof	15				
ChiSq Value (Table)	24.99600				
Standard Deviation	0.05131	0.05943	0.06045	0.05987	0.05198
Estimated Parameters	-0.58336	-0.16250	0.05859	-0.09348	0.14703
Confidence Intervals	-0.68597	-0.28136	-0.06232	-0.21322	0.04308
	-0.48075	-0.04365	0.17950	0.02627	0.25098
Parameters	a1	a2	a3	a4	a5

ACF of Residuals for na=7; nb=2; ARMA(7,2)



ACF of Residuals for na=5; nb=0;AR(5,0)



ARMA(7,2) has less spikes compared to AR(5,0), hence also has lower residual variance and provides a better estimate.

Summary and conclusions

The limitations of the final model and suggestions for other types of models that might improve performance are stated below.

The **ARMA(7,2)** model with **na = 7 and nb = 2** was initially chosen as it best represents the measurement data for this case in terms of statistics but fails in terms of the parameters falling inside confidence intervals. The **AR(5,0)** model with **na= 5 and nb = 0** was also a close choice.

In the simple case with **na=1; nb = 1** we get a very decent fit but still has zero in the con fidence intervals. I was The chi square value is 25.74 which is higher than for **na = 7 and nb = 2;** Overall, I finally tried to get a model where along with all the statistics, there is no zero in the confidence intervals.

Upon observing the **na=1; nb = 0;** and **na=5; nb = 0;** cases closely, I tried the case with **na=2; nb = 0; and I would like to conclude that an AR(2)** model might closely fit the data.

The 'Q' value is more than for the other cases but since all the other criteria are favorable, we can pick this model.

Cov = [0.0026 -0.0017; -0.0017 0.0026]

Variance of parameters = 0.0026

Reasons for selecting the **na=2; nb = 0; AR(2)** model

- Passes chi square test (Q < ChiSquare value from the table for Alpha = 0.05), so the Probability value falls in 95% confidence interval
- Zero not in confidence intervals
- Low residual variance
- Parameters fall inside the confidence interval
- Small standard deviation (0.0512)
- For this experimental data there are small spikes other than the first value.

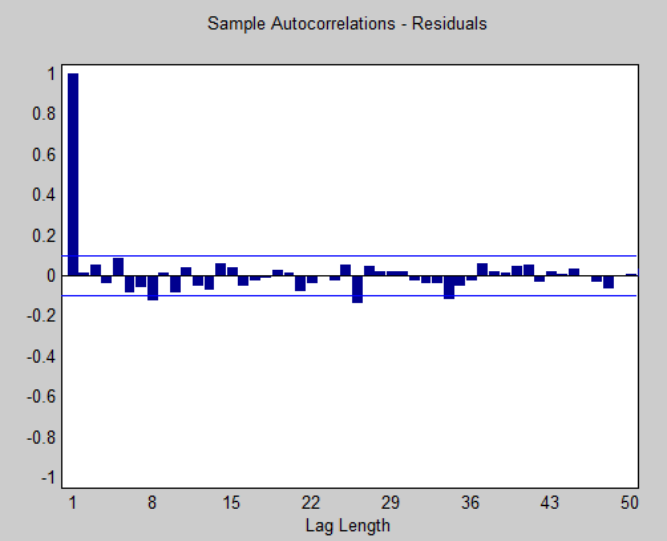
Given below is the GPAC, a table of the statistics and plot of the Auto correlation function of the residuals.

na=2; nb = 0;

nb(j)\na(i)	1	2	3	4	5	6	7	8	9	10
0	0.01042	0.05047	-0.03510	0.08085	-0.07739	-0.06261	-0.10559	0.01293	-0.05899	0.04181
1	4.85330	0.05769	0.08102	0.04748	-0.14241	0.06739	-0.11317	-0.46880	-0.04985	-0.00741
2	-0.67190	1.03390	-0.03084	0.14778	-0.16129	-0.42919	-0.09314	-0.02032	-0.03906	1.18180
3	-2.42360	1.37300	6.80810	0.21888	-0.10916	-0.17128	-0.07656	0.15099	-0.17210	0.28037
4	-0.94664	-2.56950	-2.97970	-1.41480	-0.22017	-0.14196	-0.13668	-0.58480	-1.00980	0.24047
5	0.70748	0.58265	-1.22560	2.04600	-1.14790	-0.10567	-0.61362	0.56535	-0.33199	0.10590
6	2.14650	2.40830	-0.65307	0.36398	-0.35429	1.59840	-0.27390	-0.14335	-0.21094	0.09540
7	-0.13563	0.58444	-0.02603	-0.15861	0.78104	0.61935	-0.73705	0.36190	-0.23458	0.22234
8	-4.71710	0.57987	-3.44580	-0.34262	1.18410	1.18870	-1.25130	-1.32200	-0.27302	0.57837
9	-0.56456	0.36049	4.13010	13.47300	1.00660	0.04291	-0.37511	-0.11957	0.18586	0.19499

Statistics		
Converged in Iterations	2	
SthNu (sum sq error)	8.6579	
Residual Variance	0.0231	
Q (Sum sq of Rho)	23.6632	
ChiSqdof	18	
ChiSq Value (Table)	28.86900	
Standard Deviation	0.0512	0.0512
Estimated Parameters	-0.5711	-0.1308
Confidence Intervals	-0.67350	-0.23330
	-0.46860	-0.02830
Parameters	a1	a2

ACF of Residuals for na=2; nb=0; AR(2)



Suggestions to improve the model

The only issue with the ARMA(7,2) model that is chosen is that 'a2',' a5' and 'a6' have zero in the confidence interval which makes 'a2,'a5' and 'a6' less significant than the others in terms of numeric value. Given the model is approximate it's also possible that they might sometimes be equal zero.

On the other hand the AR(5,0) model has parameters that not significant though other statistics are good. The "Q" value is however significantly bigger the ARMA model which suggests that the AR model has lower residual whiteness than the ARMA model

A bigger measurement vector of the size of at least 1000 points will be very helpful in deducing valuable information from the GPAC about the patterns. As the number of measurements increase, process becomes stationary and ACF declines gradually. An example is provided in the Appendix. A large sample would help to better minimize the least squares or sum squares error and we can get closer parameter estimates.

The ARMA(7,2) could be a better model for this given data but might not be the best. As the order of the ARMA model is high here, though it fits this data set it might not apply well when more measurements become available. Just like a higher order polynomial curve fitting has a limiting point beyond a certain order and becomes unstable, it's better to reevaluate the GPAC and do this iterative process again when the measurements are increased or updated. One of the important drawback is zero inside the confidence intervals.

So may be more iterations on the AR(2) model might give more reasons to improve the model. Also more data will be helpful as well.

Appendix

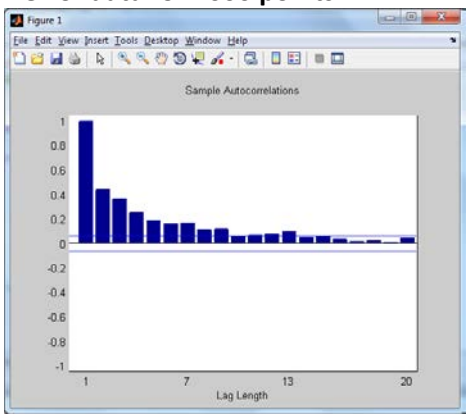
Chi-Square Table reference

http://www.stat.lsu.edu/exstweb/statlab/Tables/TABLES98-Chi.html

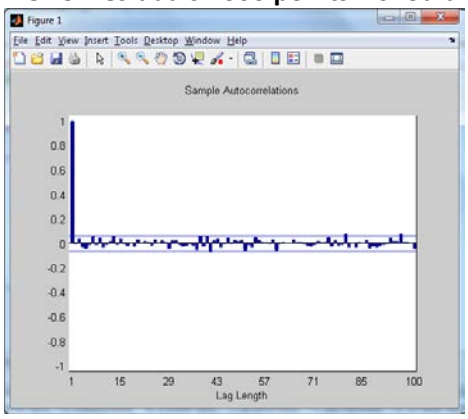
Discussion based on a Test case

A case using 1000 and 100000 points that was used to infer knowledge for this project.

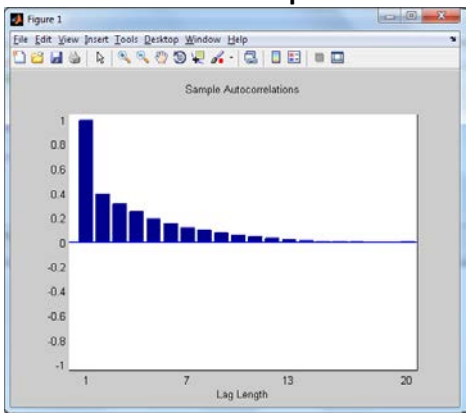
ACF of data for 1000 points



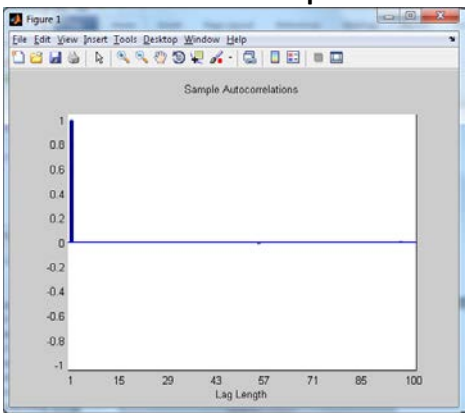
ACF of residuals 1000 points – should be white noise for convergence



ACF of data for 100000 points



ACF of residuals 100000 points - should be white noise for convergence



ACF (100000 points) has less spikes compared to ACF(1000 points), hence also has lower residual variance and provides a better estimate.

Comparison of statistics for the 1000 and 100000 data points		
Converged in Iterations	4	6
SthNu (sum sq error)	995.97	99418.0
Residual Variance	0.99797	0.99420
Q (Sum sq of Rho)	15.64600	13.49200
ChiSqdof	18	18
ChiSq Value (Table)	28.869	28.869
Standard Deviation	[0.03813 0.054805]	[0.00394 0.0056109]
Estimated Parameters	[-0.77974 -0.43742]	[-0.79089 -0.48766]
Confidence Intervals	[-0.85600 -0.54703]	[-0.79876 -0.49888]
	[-0.70347 -0.32781]	[-0.78302 -0.47643]
Parameters	[a1 b1]	[a1 b1]

Conclusions drawn from ACF and GPAC of data and residuals used for project

The GPAC pattern more obvious with 100000 data points, almost similar to theoretical case with ‘ana’ followed by zero or insignificant values to the right in the 'na' direction and ‘ana’ value repeated in the 'nb' direction. With 1000 points the ‘ana’ is repeated in the 'nb' direction but the values are not really insignificant to the right in the in the 'na' direction.

ACF of data is smoother and ACF of residuals is like an impulse and more smoother.
In the case with 100000 data points, iterations to converge increases, Q decreases (whiteness increases, ACF of residuals is an impulse), Residual variance slightly decreases, standard deviation is small or confidence intervals are tighter. SThNu or sum-squared error increases as only the initial value in the residual array greatly dominates the rest.
When we create ARMAseq ‘y’ we use arrays ‘theta.na’, theta.nb’ and ‘e’. eThNu is back calculated using ‘y’ and ‘theta’. So all we get is just white noise or something similar (depending on parameters of na, nb arrays)

In the GPAC of Auto Correlation of residuals the ‘ana’ value is not equal to the ‘ana’ of data. If the na, nb values are correctly determined, this ‘ana’ value in GPAC of residuals makes sense when compared to GPAC of ACF of the data. This value dominates the ‘ana’ position and the rest of the values to the right are all equal to zero or insignificant in the 'na’ direction.

GPAC of data for 1000 data points

nb_j\na_i	1	2	3	4	5	6	7	8
0	0.4402	0.2080	0.0445	0.0158	0.0411	0.0684	-0.0165	0.0270
1	0.8212	0.1180	-0.0291	-0.1003	0.0150	0.0782	0.0953	0.0086
2	0.6969	0.2055	-0.7024	-0.1322	0.3486	0.0809	0.0477	0.1453
3	0.7415	-1.5610	0.1941	0.4574	-0.3355	0.0878	-0.0753	0.1388
4	0.8765	-0.7685	4.8891	1.0923	-0.1006	0.1124	0.3420	0.3217
5	1.0360	2.0622	0.7359	0.4670	0.7536	0.0266	-0.2444	0.1390
6	0.6736	0.7004	-0.3758	-0.2127	0.6139	5.3241	-0.2185	-0.1501
7	1.0373	0.9391	-0.9748	-1.7910	0.9558	0.2977	-0.6058	0.0574

GPAC of Residuals for 1000 data points

nb(j)\na(i)	1	2	3	4	5	6	7	8
0	-0.0065	0.0332	-0.0265	-0.0413	-0.0097	0.0554	-0.0114	0.0336
1	-5.1106	0.0280	-0.0782	-0.0351	-0.2458	0.0534	0.1517	0.0235
2	-0.8104	-2.2044	-0.0537	0.0473	-0.0600	0.0785	0.0334	0.0969
3	1.4784	-0.6300	-0.6610	-0.1128	-0.0133	0.0665	-0.1441	0.0711
4	0.2749	-1.7379	-1.8763	0.1509	-0.4527	0.0621	0.0302	0.1453
5	-4.8715	-1.2159	-1.6193	-13.0080	-1.1370	0.0767	-0.3310	0.1332
6	-0.1987	0.7326	-0.2817	-0.6655	0.1896	1.2292	0.0380	0.0379
7	-3.7350	0.6102	-1.7116	-0.7417	5.2244	1.3172	-1.3281	0.0429

GPAC of data for 100000 data points

nb(j)\na(i)	1	2	3	4	5	6	7	8
0	0.3992	0.1872	0.0885	0.0416	0.0235	0.0113	0.0061	0.0051
1	0.7934	0.0050	0.0011	-0.0083	0.0035	-0.0014	-0.0033	0.0046
2	0.7871	-0.1660	0.0379	-0.0077	0.0007	-0.0108	-0.0062	0.0072
3	0.7854	5.9387	1.1492	-0.0075	-0.0939	-0.0080	0.0049	0.0078
4	0.8014	0.3490	0.0187	-0.4449	-0.0081	-0.0228	0.0276	0.0072
5	0.7926	0.3086	8.1916	-0.4250	1.1434	-0.0046	0.0000	0.0096
6	0.7969	-2.1268	1.1885	0.5568	0.8966	-0.0068	-4.0360	0.0096
7	0.8114	1.3879	2.0402	-3.8646	0.8976	247.7400	1.8779	0.0062

GPAC of Residuals for 100000 data points

nb(j)\na(i)	1	2	3	4	5	6	7	8
0	-0.0004	0.0015	-0.0007	-0.0030	0.0009	0.0004	0.0009	0.0041
1	-4.3606	0.0014	-0.0074	-0.0031	0.0022	-0.0017	-0.0008	0.0036
2	-0.4388	-2.3325	-0.0015	-0.0028	0.0059	-0.0073	-0.0149	0.0036
3	4.3424	-1.8463	3.3669	-0.0032	0.0120	0.0325	0.0164	0.0037
4	-0.2908	-0.1959	-0.4175	-1.5846	0.0031	-0.0004	0.0011	0.0027
5	0.4279	0.3581	0.4183	-1.1690	-0.1504	0.0083	0.0021	0.0037
6	2.49610	-1.05530	2.42180	-1.18310	-3.30850	0.79496	0.01427	0.00305
7	4.53960	21.55300	4.89100	-1.10140	2.66290	-4.80990	1.01950	0.00220

Abbreviations

AR – Auto regressive process
MA – Moving Average
ARMA – Auto regressive Moving Average process
ACF – Auto Correlation function
PACF – Partial Auto Correlation function

Infer Standard deviation and variance from this formula for 95% confidence interval

Confidence interval = Theta_i - 2* sqrt(var_Theta_i) and Theta_i + 2* sqrt(var_Theta_i)
Variance of the parameter estimates = var_Theta_i
Standard deviation of the parameter estimates = sqrt(var_Theta_i)

Software

eResiduals program

```
function [e1] = prj_eResiduals(theta,y)
    %Revert the measurement array using theta.na and theta.nb to get 'e'
    %a(0), b(0) should be  = 1 in this case, when using the filter function
    %y(t) = e(t)+b1*e(t-1)+b2*e(t-2) ... + bnb*e(t-nb)
    %                      -a1*y(t-1)-a2*y(t-1)- ... -ana*y(t-na)}

    %Len = length(theta);
    %Always add 1 at the beginning of a, b for the above sequence when
    %using the filter command
    %split theta to get a, b for filter, theta =[a1..ana b1..bnb]
    %Build a and b based on the theta.na and theta.nb values

    if(theta.na == 0)
        a = [1];
    else
        a = [1 theta.na];
    end
    if(theta.nb == 0)
        b = [1];
    else
        b = [1 theta.nb];
    end

    %simulation of the ARMA-process, y has the ARMA sequence
    e1=filter(a,b,y);
end
```

Autocorrelation program

```
function ry=prj_autocor(y,p,PlotTitle)

  %  Function needs data that is already zero mean
  %  This function computes the acf r(k) k=0:length(x)
  %  CAUTION the first value r(1) is at{gamma}(0) IE the first lag is 0
  %  y = time series vector
  %  p = No of Lags

    [Nr,Nc]=size(y);%row,col
    if Nc > Nr
```

```
        error('x must be a column vector');
    end
    ry=conv(flipud(y),y);
    ry=ry(Nr:end);
    ry=ry/Nr;
    ry=ry/ry(1);
    %plot the autocorrelation function
    figure
    bar(ry);
    line([0 p+.5], (1.96)*(1/sqrt(Nr))*ones(1,2))
    line([0 p+.5], (-1.96)*(1/sqrt(Nr))*ones(1,2))

    % Below code is from Mathworks, used only to represent autocorrelation
    % more effectively in a bar graph form
    % set figure properties
    line_hi = (1.96)*(1/sqrt(Nr))+.05;
    line_lo = -(1.96)*(1/sqrt(Nr))-.05;
    bar_hi = max(ry)+.05 ;
    bar_lo = -max(ry)-.05 ;
    % if rejection lines might not appear on graph
    if (abs(line_hi) > abs(bar_hi))
        axis([0 p+.60 line_lo line_hi])
    else
        axis([0 p+.60 bar_lo bar_hi])
    end

    PlotTitle = strcat('Sample Autocorrelations - ', PlotTitle);
    title({' ',PlotTitle,' '})
    xlabel('Lag Length')
    set(gca,'YTick',[-1:.20:1])
    % set number of lag labels shown
    if (p<28 & p>4)
        set(gca,'XTick',floor(linspace(1,p,4)))
    elseif (p>=28)
        set(gca,'XTick',floor(linspace(1,p,8)))
    end
    set(gca,'TickLength',[0 0])
```

GPAC program

```
function g = prj_gpac(ry,r,c)
    %Prints the GPAC array
    %ry = [ry(0),[ry(1),[ry(2),.. ]
    %r-rows=>  j=0..nb    %c-cols => k =1..na

    phi = [];  %Array to store the GPAC Values
    %Create the denominator index matrix for GPAC coef
    for iter_1= 1:r
        j=iter_1-1;
        for iter_2= 1:c
            DeRy_index = zeros(iter_2,iter_2);%(c,c)
            k=iter_2;kp=k; %k=c
            for m = 1:iter_2%c
                for n = 1:iter_2%c
                    DeRy_index(m,n) = (j-k)+kp;
                    kp=kp-1;
                end
```

```
            kp=k+m;
        end
        %since ry(-1) = ry(1), use absolute values
        DeRy_index_2 = abs(DeRy_index);

        %Create the numerator index matrix for GPAC coef
        NuRy_index = DeRy_index_2;
        jj = j+1;
        for m = 1:iter_2%c
            NuRy_index(m,iter_2) = jj;%m,c
            jj = jj+1;
        end
        %since ry(-1) = ry(1), use absolute values
        NuRy_index_2 = abs(NuRy_index);
        %Add 1 to zero based index to start at 1
        %for matlab compatibility
        DeRy_index_2 = DeRy_index_2 +1;
        NuRy_index_2 = NuRy_index_2 +1;
        %ry = [1.25 0.5 0.4 0.32 0.256 0.2048];
        %copy the ry values at the indexes into Num and Den arrays

        NumArr = ry(NuRy_index_2);
        DenArr = ry(DeRy_index_2);

        tempPhi = det(NumArr)/det(DenArr);
        format('shortE');
        phi = [phi; tempPhi];
    end
end
%create the GPAC Matrix representation
%Convert array to matrix and transpose to get the format
GPAC = reshape(phi,c,r);
GPAC=transpose(GPAC);
g = GPAC;

end
```

Maximum Likelihood program

```
function Theta = prj_MaxLik(y, na,nb)
    %Initialize
    N   = length(y)   %Size of measurement vector
    n = na + nb       %Length of Theta vector, parameters [a1..ana, b1..bnb]

    STh = y'*y  %scalar
    %na = length(a1..ana) = 0 for MA, nb = length(b1..bnb) = 0 for AR
    %na != 0 and nb != 0 for ARMA
    if (na > 0)
        Th.na  = zeros(1,na);
    else
        Th.na  = [0];
    end
    if (nb > 0)
        Th.nb  = zeros(1,nb);
    else
        Th.nb  = [0];
    end
```

```
Eps    = 0.01; % Convergence limit
Mu     = 0.01; %step increase in -ve gradient direction
max_Mu = 10^9; % max value reduces sum square error
max_iter = 25*N;
Converged = false;
%---------------------------------------
%Initialize calculated variables
ThNu  = Th;%Vector to store estimated parameters
Theta = Th; %Vector to store estimated parameters
eThNu = zeros(N, 1); %e sequence updated every step of ThNu is calculated
SThNu = 0;
A = zeros(n, n);
g = zeros(n, 1);
ConvIter = 0; %Iterations to converge
%---------------------------------------
%stage 1
e = y;
[A,g] = GetAg(e,y,N,na,nb,Th);

for iter = 1:max_iter
    ConvIter = iter
    %[A,g] = GetAg(e,y,N,na,nb,Th);
    %stage 2
    [ThNu,DelTh] = GetStage2(A,g,Mu,Th,na,nb);

    eThNu = prj_eResiduals(ThNu,y);
    SThNu = eThNu' * eThNu
    'norm(DelTh)          '
    norm(DelTh)
    %Stage 3
    if (SThNu < STh)
        if(norm(DelTh) < Eps)
            Theta = ThNu;
            STh = SThNu;

            Note='Break..Converged....'
            Converged=true
            break % converged
        end
        Th  = ThNu;
        STh = SThNu;
        Mu = Mu/10.;
        %iter = iter + 1
        if (iter > max_iter)
            Note='Break..max_iter reached....'
            Converged=false
            break %error
        end

        %return to stage 1
        e     = prj_eResiduals(Th,y);
        [A,g] = GetAg(e,y,N,na,nb,Th);
    else
        Mu = 10*Mu;
        %if sum sq err increases
        if (Mu > max_Mu)
            Theta = ThNu;
            STh   = SThNu;
```

```
                Note='Break..Max Mu reached....'
                break %Error
            end
            %return to stage 2
            %[ThNu,DelTh] = GetStage2(A,g,Mu,Th,na,nb)
        end %if (SThNu < STh)

    end %for iter = 1:max_iter
    Note='End of hw6_MaxLik....'
    %Theta = Th;
    Theta
    %if(Converged == true)
    %    Note='Printing the statistics....'
    [Res] = GetStats(y,N,na,nb,eThNu,STh,A,Theta)
    Converged
    ConvIter
end

function [A,g] = GetAg(e,y,N,na,nb,Th)
    %Stage 1 - find A, g
    n=na+nb;
    Delta = 10^-5;
    X = zeros(N, n); %x - Nxn matrix
    for i = 1:n
        NTh = Th;
        if(i <= na)
            NTh.na(i) = NTh.na(i)+ Delta;
        elseif ((i>na) & (i <= (na+nb)))
            NTh.nb(i-na) = NTh.nb(i-na)+ Delta;
        end
        eNu = prj_eResiduals(NTh,y);
        X(:,i) = (e - eNu)/Delta;
    end
    A = X'*X;
    g = X'*e;
    %Done='Completed [A,g] = GetAg()'
end %[A,g] = GetAg(e,y,N,na,nb,Th)

%.............................
function [ThNu,DelTh] = GetStage2(A,g,Mu,Th,na,nb)
    %Stage 2
    I = eye(size(A));
    DelTh = inv(A+Mu*I)*g; %nx1 matrix
    DLen = length(DelTh);

    ThNu.na = Th.na + DelTh(1:na)';
    ThNu.nb = Th.nb + DelTh((na+1):DLen)';

    %Done='Completed [ThNu,DelTh] = GetStage2(A,g,Mu,Th,na,nb)'
end %[ThNu,DelTh] = GetStage2(A,g,Mu,Th,na,nb)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Res] = GetStats(y,N,na,nb,eThNu,STh,A,Theta)
 %1) Residual variance,
 %2) Autocorrelation function of the residuals
 %3) Value of Q
 %4) ChiSqdof
 %5) Chi-square statistic for test of residual whiteness
```

```
%6) Note = 'Printing na, nb, n and Estimated parameters
%7) Standard deviation of parameter estimate
%8) Confidence intervals - Lower and Higher
%9) GPAC of Auto correlation of the residuals

   EstParam = [Theta.na Theta.nb];
   %'Residual variance...'
   ResVar = STh/(N-na-nb)
   %'Co-variance...'
   Cov = ResVar * inv(A)
   Lags = 20 %int32( 0.1*length(EstParam));
   %'Autocorrelation of residuals...'
   PlotTitle = ' Residuals';
   ry_AC = prj_autocor(eThNu,Lags+30, PlotTitle);

   RhoK = zeros(size(ry_AC));
   for i = 2 : N
      RhoK(i-1) = ry_AC(i)/ry_AC(1);
   end

   NRhoK = 0;
   for j = 1: Lags
      NRhoK = NRhoK + RhoK(j)^2;
   end
   Q = N * NRhoK
   ChiSqdof = Lags-na-nb

   %'chi-square statistic for test of residual whiteness...'
   % check value of ChiSqdof vs 0.05 probability in Chi-Square table
   % reject model if value is < Q, eThNu is not white, accept otherwise

   %Calculate the confidence intervals
   IntervSz  = length(Cov);
   ConfInt_1 = zeros(IntervSz,1);
   ConfInt_2 = zeros(IntervSz,1);
   StdDev_param = zeros(IntervSz,1);
   for i = 1:IntervSz
       StdDev_param(i) = sqrt(Cov(i,i));
       ConfInt_1(i) = EstParam(i) - 2* StdDev_param(i);
       ConfInt_2(i) = EstParam(i) + 2* StdDev_param(i);
   end
   Note = 'Printing na, nb, n and Estimated parameters'
   na,nb,na+nb
   Theta.na
   Theta.nb
   EstParam
   Note = 'Standard deviation of parameter estimates'
   StdDev_param'
   Note = 'Confidence intervals - Lower and Higher...'
   ConfInt = [ConfInt_1'; ConfInt_2']
   %ConfInt = [ConfInt_1 ConfInt_2]

   'GPAC of auto cor of residuals...'
   r=10;c=10;
   g = prj_gpac(ry_AC,r,c)

   %'GPAC of auto cor of data...'
   %ry_data = prj_autocor(y,Lags);%Is this the right thing to do?
```

```
    %g = prj_gpac(ry_data,r,c)
    Res='Completed program';
end
```

Maximum Likelihood Driver program

```
%Driver program to run the time series data file and generate the Max
%Likelihood parameter estimates
%Browse the file location and pass it to fopen
%format long
format compact
fid = fopen('C:\Users\Srinivas\Desktop\Srini\Estimation theory\FinalProject\Data
Final\Indi002_Gulmarg_370_0Mean4dig.txt');
disp(fid)
if(fid ~= -1)%If file is opened successfully
    y = fscanf(fid, '%g %g', [1 inf]);
    fclose(fid);
    y=y';
    figure
    plot(y);
    title({' ','Data Sequence - Tree Rings',' '});
    plot(y,'-'); hline = refline([0 0]);set(hline,'Color','r')
    xlabel('Year'); ylabel('Ring index-zero mean');title('Tree Ring Data');
    Lags = 20;

    %Based on the gpac data select na,nb;
    na =5; nb =0;
    %Call the maximum likelihood program
    Theta = prj_MaxLik(y,na,nb);
    PlotTitle = ' Data Sequence';
    ry = prj_autocor(y,Lags,PlotTitle);
    g = prj_gpac(ry,10,10)
end
```

Noise program

```
function [e] = prj_noise(sigma,N,dist)

    stdDev = sqrt(sigma);
    if (strcmp(dist, 'Normal'))
        %Generate values from a normal distribution with mean 1 and std dev 1.
        e = stdDev.*randn(N,1);
        %e = randn(100,1);
    elseif (strcmp(dist, 'Uniform'))
        %Generate values from the uniform distribution on the interval [a, b].
        a=1;b=50;
        e = a + (b-a).*rand(N,1);
    end
    %{
    figure;
    plot(e);
    title({' ','White Noise',' '});
    xlabel('Sequence Length');
```

```
    ylabel('White Noise');
    %e=prj_noise(2,10,'Normal')
    %}
end
```

ARMA Sequence program

```
function [y] = prj_armaseq(b,a,e)
    %Used sequence
    %y(t) = e(t)+b1*e(t-1)+b(2)*e(t-2) ... + b(nb)*e(t-nb)
    %                      -a(1)*y(t-1)-a(2)*y(t-2)- ... -a(na)*y(t-na)}

    %y(t)-0.8y(t-1)=e(t)-0.5e(t-1) a=[-0.8] b=[-0.5]
    %Always add 1 at the beginning of a, b
    %b=[-0.5]; ; a=[-0.8]
    a=[1,a]
    b=[1,b]

    %simulation of the ARMA-process, y has the ARMA sequence
    y=filter(b,a,e);

    %{
    figure
    plot(y);
    title({' ','ARMA Sequence',' '});
    xlabel('Sequence Length');
    ylabel('ARMA Sequence');
    %}
    %a = [1, 0.8]; b = [1, -0.5] ;
    %y = prj_armaseq(a,b,e);
end
```