Analysis of Variance in Micropipelines

Aiguo Xie and Peter A. Beerel

CENG 98-17

Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, California 90089-2562
(213-740-4481
July 1998

Analysis of Variance in Micropipelines †

Aiguo Xie and Peter A. Beerel

Department of Electrical Engineering – Systems
University of Southern California
Los Angeles, CA 90089

Abstract

Classical linear regression methods are applied to estimate the performance of micropipelines. A data set of the system performance measures (i.e., cycle time and latency) is first created by recently developed performance evaluation tool based on Markovian analysis. The performance measures are fitted using stepwise linear regression to the relevant system parameters which are served as independent variables. The resulting models are then simplified through various statistics and hypothesis testing. Finally, the simplified models are validated using another independent data set. The fitted regression models reveal several explicit relationships between the performance measure and the system timing parameters.

1 Introduction

We report an application of the classical linear regression methods on the performance analysis of an important class of electronic systems, i.e., micropipelines [1]. In modern electrical engineering, system performance, e.g., the rate at which a system can process the data, is one of the dominating system design criteria.

†This work is funded in part by a NSF CAREER Award MIP-9502386, NSF Grant No. CCR-9812164, a gift from the Intel Corporation, and a research seed grant from the James H. Zumberge Faculty Research and Innovation Fund at USC.

Most electronic systems such as today's personal computers work at a fixed rate. Specifically, these systems have a built-in *clock* that periodically generates a global signal that controls every other system component so that the entire system marches at the same speed. Because of this, they are called *synchronous systems*. Consequently, their performance such as data processing rate can be easily determined by the rate of the clock. One critical disadvantage of synchronous systems is that the clock cannot run at a rate that is higher than the speed of the slowest system component. Otherwise, the slowest component would not have enough time to finish its current job before a new job arrives. Since a modern electronic system routinely contains thousands of components with very different data processing speeds, the system clock rate is necessarily to be slow, thus limits the system performance. Meanwhile, faster components must wait until next clock signal comes although they may have finished their current job way earlier, resulting a wasting of system resource.

More recently, another type of systems, called *asynchronous* systems, demonstrates many potential advantages over synchronous systems. These systems run without a built-in clock. All system components work at their own speeds except some necessary synchronization between adjacent components when they have to exchange their results. Thus, the systems do not have a fixed data processing rate, and they are expected to have better average performance (over the time) than synchronous ones. Micropipelines belong to this category.

Unfortunately, analyzing the average performance of an asynchronous system is not trivial. Not only do system components work at different speeds, but a given system component may also have different speed depending on the type of data (or job) it currently works on. In other words, the system is stochastic in nature. Many researchers have studied the performance of these systems [2, 3, 4, 5, 6]. A common way to do this is to first model the system as a time-homogeneous finite state Markov chain and then compute its stationary probability distribution from which the system performance measures can be conveniently obtained. However, because the resulting Markov chain may have a huge state space size, these methods are limited to analyzing small systems. Moreover, even it is possible to compute the performance measure using this method in a reasonable amount of time, it is still unclear how the system performance would change if some of the system component were to change (for example, redesigned).

In this work, we are targeted to explore possible direct (explicit) re-

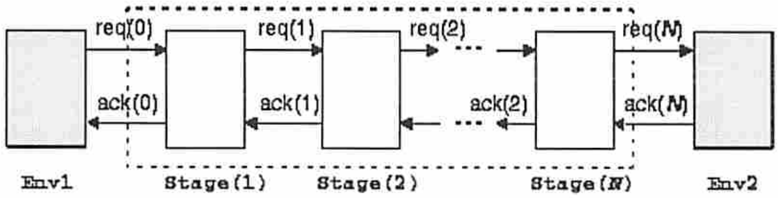

Figure 1: Block diagram of an N-stage micropipeline.

lation between the system performance and the system parameters using linear regression method. If there is any such direct relationship, it is desirable to find a good regression model for it. More precisely, we wish to find regression models that *accurately* estimate/predicate corresponding system performance. For instance, whenever possible we would like to find a regression model with the coefficient of determination R^2 higher than 95% or with the prediction error less than 5%. In particular, we focus on the analysis of micropipelines.

1.1 The mechanism of micropipelines

Figure 1 shows the block diagram of a micropipeline. It consists of $N(N \geq 1)$ data processing units which we call stages. The environment at the left side, labeled as `Env1`, is the source of the data to be processed by the system. It sends data to the first stage. The first stage then processes on that data and when it finishes sends the result to the next stage. The data then flows down the line towards the environment at the right side, labeled as `Env2` which consumes the final result. Once the first stage finishes current data, it can start processing new data if `Env1` has new data ready to send out. This way, there can be multiple data being processed at different stages of the system at a given instant of time. In other words, the `Env1` does not necessarily have to delay sending a new data to the system until the previous data has been consumed by the environment `Env2` at the other side, and thus the name of micropipeline.

When `Stage`(i) $(i = 1, \cdots, n-1)$ finishes processing current data, it issues a signal, labeled as `Req`(i), to `Stage`$(i+1)$, meaning that it has data ready to dispatch and requires processing. If `Stage`$(i+1)$ is current idle, i.e., it is not working on any data, then it takes in the data from `Stage`(i) and issues a signal, labeled as `Ack`$(i+1)$, back to `Stage`(i), meaning that the data has been taken in. At that point, `Stage(`i`)` can start working on new incoming data if there is any. If, on the other hand, `Stage(`$i+1$`)`, is current busy on the previous data, then `Stage(`i`)` has to wait (called *blocked*) until `Stage(`$i+1$`)` finishes the previous data and receives a the signal `Ack(`$i+1$`)` from `Stage(`$i+1$`)`. This type of communications between adjacent stages is sometimes called *handshaking*. The environment `Env1` communicates with `Stage(1)`, and environment `Env2` communicates with `Stage(`N`)` both in a similar way.

1.2 The timing model

We call the time a system component needs to finishing process a given data the *delay* of the component on that data. In practice, a delay can take any non-negative real values. Moreover, a component delay is usually data-dependent, i.e., its value varies with different types of data to be processed, and has a probability distribution over some interval.

In our micropipeline model, we assume the delay of `Stage(`i`)` ($i = 1, \cdots n$) , denoted by $d(i)$, takes values in an interval, $[\delta(i), \Delta(i)]$ with a probability density function $f_{d(i)}$. We further assume that all N stages of the micropipeline are roughly balanced. That is, their delays are independently identically distributed (i.i.d.). Therefore, we just have one common delay interval, denoted by $[\delta, \Delta]$, with a common probability density function f_d for all stages. In this report, we set this delay interval to be $[1, 3]$. Finally, we assume that both environments have a fixed delay of 1.

We define μ to be the mean of the stage delay d, i.e., $\mu = \mathbf{E}\{d\}$, σ^2 to be the variance of d, i.e., $\sigma^2 = \mathbf{E}\{(d-\mu)^2\}$ and m_3 to be the third central moment of d, i.e., $m_3 = \mathbf{E}\{(d-\mu)^3\}$.

The rest of the report is organized as follows. In section 2, we list the important hypotheses that we have tested along with the final regression model we arrived. In Section 3, we described in detail the development of the model using *stepwise regression method* [7] for variable selection. Statistics such as R^2, R^2_{adj} C_p are used to choose a good subset size and thus simplify the model. The models are then validated using an independent data set in Section 4. Conclusions are given in Section 5.

2 The Tested Hypotheses

There are two performance measures that are of primary interest in micropipeline design. They are the average *throughput* and average *latency*.

Average throughput, denoted by λ, measures the expected data processing rate of the system. It is defined as the reciprocal of the expected cycle time T of the system, or equivalently the average frequency at which the environment `Env1` injects data to `Stage(1)`. That is, we have $\lambda = T^{-1}$.

Average latency, denoted by L, measure the expected time a data has to experience in the micropipeline once it starts been processed by `Stage(1)` until it is consumed by the environment `Env2`.

In the sequel, when we say cycle time or latency, we mean their expected values to simplify the description.

Several hypotheses regarding these two performance indices are of great interest.

Hypothesis 1 *Cycle time of our micropipeline model can be adequately estimated by only the number of stages N and the first moment of the stage delay, d. That is, variance and other higher moments of the stage delay do not have significant impact on the cycle time.*

Hypothesis 2 *Cycle time of our micropipeline model can be adequately estimated by the number of stages N, and the first two moments of the stage delay, d. That is, the third and other higher moments of the stage delay do not have significant impact on the system cycle time.*

Similar hypothesis regarding the system latency are:

Hypothesis 3 *Latency of our micropipeline model can be adequately estimated by only the number of stages N, the first moment of the stage delay, d. That is, variance and other higher moments of the stage delay do not have significant impact on the system latency.*

Hypothesis 4 *Latency of our micropipeline model can be adequately estimated by the number of stages N and the first two moments of the stage delay, d. That is, the third and other higher moments of the stage delay do not have significant impact on the system latency.*

Using linear regression, regarding the cycle time, we tested that Hypothesis 1 is false while Hypothesis 2 is true. Regarding the latency, we tested that Hypothesis 3 is true and thus of course 4 is also true since it requires to incorporate more independent variables.

In particular, the cycle time of our micropipeline model can be adequately captured by the following model:

$$T = 2.36 + 0.861\mu + 0.645\log(N)\sigma^2 \tag{1}$$

where N is the number of stages in the system, μ and σ^2 are the mean and the variance of the stage delays, respectively, as defined in section 1.2.

The latency can be adequately captured by the following model:

$$L = 0.271 + 1.62n\mu \tag{2}$$

3 Model Development

In this section, we develop the linear regression model for the cycle time and latency of our micropipeline model. Specifically, based on the data set which we obtained using our recently developed performance analysis tool [4, 5], we first use stepwise linear regression method to find a good model for each subset size. Next, we combine different statics such as R^2, R^2_{adj}, MS(Res) and C_p to determine a good subset size and simplify the model. Testing of various hypotheses will also be given.

3.1 The data set

The data set is derived as follows. Given the number of stages N and a particular probability density function f_d for the stage delay d, we use our Markovian analysis based performance evaluation tool to compute the cycle time and the latency of the micropipeline. In particular, we vary the number of stages N from 1 to 6. For each fixed number of stages, we apply 10 different probability distribution function f_d. Therefore, for both cycle time and latency, we have $6 \times 10 = 60$ data. That is, the data set size $n = 60$.

For clarity, we put the resulting data set in Table 5 of the appendix. Specifically, the independent variables there include N, μ, σ^2, m_3. The

Source of variance	d.f.	SS	MS	F
Total(uncorr)	60	1.2065×10^3		
Mean	1	1.1963×10^3		
Total(corr)	59	10.1898		
Regression	4	9.3019	2.3255	144.1
Residuals	55	0.8879	0.0161	

Table 1: Summary analysis of variance for the simple linear model of cycle time.

dependent variables are T and L.

3.2 Determine the independent variables for regression

Because both dependent variables, especially the cycle time T, could be extremely complicated functions of the system timing parameters such as N, μ, σ^2 and m_3. In particular, this function is unlikely to be linear so that a direct use of linear regression with these independent variables may not be adequate. For this reason, we first check if a simple linear regression model based on these 4 parameters will be sufficient. If not, we will try to added more variables into the data set [1].

Let us first test the adequacy of the following linear model for cycle time:

$$T_j = \beta_0 + \beta_1 N_j + \beta_2 \mu_j + \beta_3 \sigma_j^2 + \beta_4 (m_3)_j + \epsilon_j \tag{3}$$

where $j = 1, \cdots, n$. Or equivalent in the vector form

$$\boldsymbol{T} = \mathbf{X}\boldsymbol{\beta} + \epsilon \tag{4}$$

$$\text{where } \mathbf{X} = \begin{pmatrix} 1 \\ N \\ \mu \\ \sigma^2 \\ m_3 \end{pmatrix} \text{ and } \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix}.$$

[1] Alternatively, one may resort to nonlinear regression methods (see e.g., [8]).

The *coefficient of determination* $R^2 = \frac{SS(Regression)}{SS(Total(corr))}$ computes to 91.3% whereas the adjusted R^2, $R^2_{adj} = 1 - \frac{MS(Regression)}{MS(Total(corr)}$ computes to 90.65%. Both of them do not meet our requirement of 95%. Further, $\hat{\boldsymbol{\beta}}$ is computed to $\begin{pmatrix} 1.768 \\ 0.1234 \\ 0.9620 \\ 0.6227 \\ 0.1240 \end{pmatrix}$. Take $\alpha = 0.05$, we have $t_{\alpha/2,55} \approx 2.009$, which yields the 95% confidence interval for the coefficients as $\boldsymbol{CL(\hat{\beta})} = \begin{pmatrix} 1.768 \pm 0.9615 \\ 0.1234 \pm 0.0193 \\ 0.9620 \pm 0.4770 \\ 0.6227 \pm 0.1880 \\ 0.1240 \pm 0.5762 \end{pmatrix}$ which are rather wide. Therefore, this simple model is not adequate. This means that we need to incorporate more independent variables.

As we discussed earlier, we may not be able to capture the cycle time of the system as linear combination of the system parameters. Instead, it might well be a complicated nonlinear function of the parameters. Therefore, we try to add more variables derived nonlinearly from given independent variables. According to our inspection of the data set and previous experience, we conjecture that the cycle time might be related to following six variables defined as:

$$\begin{aligned} v_1 &\triangleq \log(N)\mu, \\ v_2 &\triangleq \log(N)\sigma^2, \\ v_3 &\triangleq \log(N)m_3, \\ v_4 &\triangleq N\mu, \\ v_5 &\triangleq N\sigma^2, \\ v_6 &\triangleq Nm_3. \end{aligned}$$

Now, the new linear model for the cycle time as given by equation 4 has 10 independent variables [2], i.e., $\mathbf{X} = \begin{pmatrix} 1 & N & \mu & \sigma^2 & m_3 & v_1 & v_2 & \cdots & v_6 \end{pmatrix}^t$ and $\boldsymbol{\beta} = \begin{pmatrix} \beta_0 & \beta_1 & \cdots & \beta_{10} \end{pmatrix}^t$.

[2] The last 6 of these variables are clearly dependent on the first 4 variables. But for the description purpose, We still call them independent variables as in the context of linear regression.

Source of variance	d.f.	SS	MS	F
Total(uncorr)	60	1.2065×10^3		
Mean	1	1.1963×10^3		
Total(corr)	59	10.1898		
Regression	10	10.1359	1.0136	921.5
Residuals	49	0.0539	0.0011	

Table 2: Summary analysis of variance for the new linear model of cycle time with 10 independent variables.

The model with these 10 variables give $R^2 = 99.5\%$, which is sufficiently accurate for our purpose.

A similar test shows that these 10 independent variables are also adequate for the linear regression of latency L. In the remaining of this section, we use further regression methods to eliminate unimportant independent variables from this new model, and find good simple linear models for both cycle time and latency.

3.3 Model simplification using stepwise regression

Let us first focus on regression for the cycle time T. We follow the stepwise regression method described in [7]. The significance levels are chosen to be $SLE = 0.2$ and $SLS = 0.1$ The procedure terminates when the model incorporates 7 independent variables when no more remaining variables can enter and all 7 incorporated variables meet the requirement to stay. Table 3 summarizes the result of the stepwise regression.

3.4 Choosing a good subset size

According to Hocking's argument for estimation purpose [9], the Mallows' C_p statistics [10] should satisfies $C_p \leq 2p' - t$ where t is the number of variables in the full model, in our case, $t = 10$. With this criterion, we are forced to choose the last model with 7 variables in Table 3 which merely satisfies the criterion with $C_p = 6.0 \leq 2 \times (7 + 1) - 10$. All other models in the table would be unsuitable.

subset size	R^2 (%)	R^2_{adj} (%)	C_p	MS(Res)	N	μ	σ^2	m_3	v_1	v_2	v_3	v_4	v_5	v_6
1	58.1	57.4	3824	0.271		x								
2	96.7	96.6	251.6	0.078		x				x				
3	98.0	97.9	131.3	0.060		x				x			x	
4	99.1	99.0	36.5	0.041		x				x			x	x
5	99.2	99.1	25.0	0.038	x	x				x		x	x	
6	99.4	99.4	7.8	0.033	x	x	x			x		x	x	
7	99.5	99.4	6.0	0.032	x	x	x			x		x	x	x

Table 3: Summary of stepwise regression on cycle time.

This result is not satisfactory since the model with 7 variables is too complicated for us to get a good intuition about the relationship of the system cycle time and its parameters. For this reason, we adopted other criteria based on R^2, R^2_{adj} and $MS(Res)$ to choose a good subset size.

Figure 2 plots R^2 and R^2_{adj} for the models listed in Table 3. We see that the model with 2 variables has $R^2 \approx R^2_{adj} = 96.6 > 95\%$, well satisfying our requirement. With more added variables, both R^2 and R^2_{adj} increase marginally. Moreover, this observation is consistent with the MS(Res) statistics plotted in Figure 3 where MS(Res) decrease dramatically from the best 1-variable model to the 2-variable one. After that, it decreases marginally with more variables.

Therefore, we decide to choose the subset size as 2, and the corresponding model for the cycle time is given by Equation 1. For convenience, we replicate it here:

$$T = 2.36 + 0.861\mu + 0.645\log(N)\sigma^2$$

3.5 Hypothesis testing

The four hypotheses we posted in Section 2 related to the adequacy of the resulting models. Ideally, we would like to adopt the F-test on the *mean square of deviation from regression* versus *mean square of experiment error* (pp. 125, [7]), i.e., $F = \frac{MS(deviation)}{MS(exp.err.)}$, to test the adequacy of the model. Unfortunately, our data set does not contain *pure replicate* so that we can

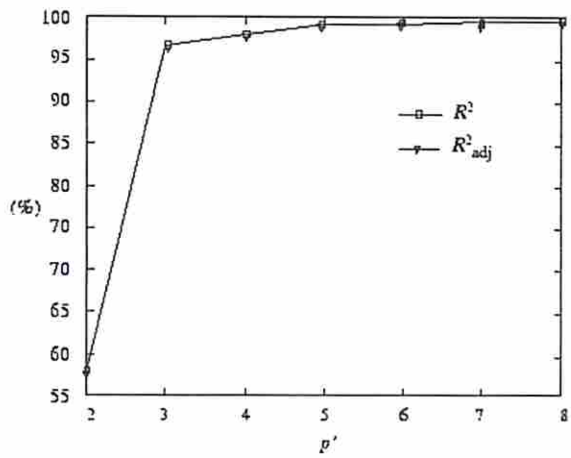

Figure 2: R^2 and R^2_{adj} plot against p' for the model selected by stepwise regression for each subset size.

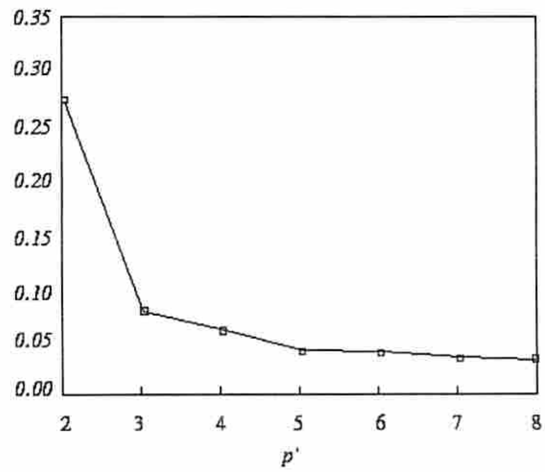

Figure 3: MS(Res) plot against p' for the model selected by stepwise regression for each subset size.

not estimate the quantity of the *pure error* which is required in the above mentioned F-test.

To overcome this difficulty, we combine several other statistics to judge the adequacy of the models.

Hypothesis 1 concerns whether the cycle time can be sufficiently estimated just by the number of stage N and the first moment of the stage delay μ. Let us exam the model that incorporates all independent variables that are related to either N or μ alone or both of them. Thus, in this model, we have $\mathbf{X} = \begin{pmatrix} 1 \\ N \\ \mu \\ \log(N)\mu \\ N\mu \end{pmatrix}$ and $\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix}$.

Variance analysis shows that R^2 and R^2_{adj} of this model are 90.7% and 90.1%, respectively, both of which are not satisfactory to our targeted value of the coefficient of determination. Further, the regression coefficients and their 95% confidence interval are: $\boldsymbol{CL}(\hat{\boldsymbol{\beta}}) = \begin{pmatrix} 1.8377 \pm 0.4388 \\ 0.2578 \pm 0.1114 \\ 1.1318 \pm 0.2137 \\ 0.3104 \pm 0.1095 \\ -0.1736 \pm 0.0664 \end{pmatrix}$.

Although the t-test for the individual $\beta_j (j = 1, \cdots, 4)$ with t-ratios $=$ $(8.493, 4.651, 10.641, 5.695, -5.252)$ all beyond the critical value $t_{0.025,55} \approx 2.009$, meaning all of them are significantly different from zero, the relatively large 95% confidence interval suggest that the model is not sufficient for our estimation purpose.

Combining R^2, R^2_{adj} and 95% confidence interval of regression coefficient, we conclude that the we can not accept Hypothesis 1. In other words, we deduce that the cycle time also heavily depends on the quantities of other higher moments of the stage delay.

Hypothesis 2 concerns whether the cycle time can be sufficiently estimated by just the number of stage N and the first two moments of the stage delay but not other higher moments. It is clear that if the 2-variable model given in Equation 1 is an adequate model, then Hypothesis 2 should be considered true.

Our variance analysis shows that $R^2 \approx R^2_{adj} = 96.6$ in this 2-variable model, well satisfying our requirement of the R^2 value. Further, the re-

	Predicted P	Observed T	Prediction Error $\delta = P - T$
	4.9391	5	-0.0609
	3.7597	3.75	0.0097
	3.8736	3.8923	-0.0184
	4.4446	4.4826	-0.0384
	4.1274	4.203	-0.0756
	4.8160	4.9073	-0.0910
	4.1150	4.1914	-0.0760
	4.0381	4.0005	0.0331
	3.8057	3.8018	0.0007
	4.7976	4.956	-0.1584
Mean	4.2717	4.3192	-0.0475

Table 4: Prediction error for model validation.

gression coefficients has very narrow 95% confidence interval, i.e., $CL(\hat{\beta}) = \begin{pmatrix} 2.36 \pm 0.115 \\ 0.861 \pm 0.055 \\ 0.645 \pm 0.050 \end{pmatrix}$.

These statistics lead us to conclude that Hypothesis 2 is true.

Conducting similar analyses, we found that both Hypothesis 3 and 4 regarding the relationship between the latency and the system timing parameters are true. The result is that system latency depends mainly on the number of stages and the first moment of the stage delay and has little relation with the variance and other higher order of the stage delay.

4 Model Validation

To confirm the effectiveness and confidence of the fitted regression equations, we use an *independent* set of data. This data set which contains 10 very different data from those used in regression is listed in Table 8 of the appendix. We would like to see how good our regression models are.

Table 4 shows some statistics comparing the observed values and the predicted ones for the system cycle time.

The average prediction bias is $\overline{\delta} = -0.0475$; the cycle time is underestimated by approximately 1error is $s^2(\delta) = \frac{1}{10-1}\sum_{i=1}^{10}(\delta(i) - \overline{\delta})^2 = 0.0032$, or $s(\delta) = 0.0568$. The standard error of the estimated mean bias is $s(\overline{\delta}) = s(\delta)/\sqrt{10} = 0.0189$. A t-test of the hypothesis that the bias is zero gives $t = \overline{\delta}/s(\overline{\delta}) = -2.50$ which, with 9 degree of freedom and $\alpha = 0.025$, is slightly beyond the critical value $t = 2.26$.

The mean squared error of prediction is

$$\begin{aligned} MSEP &= \tfrac{\delta'\delta}{10} \\ &= \tfrac{(10-1)s^2(\delta)}{10} + (\overline{\delta})^2 \\ &= 0.0029 + 0.0023 = 0.0052 \end{aligned} \tag{5}$$

Although the bias term contributes about 44% of MSEP, but the square root of MSEP gives only 0.0718 which counts to an approximately 1.66% error in prediction.

In summary, the cycle time is slightly underestimated by the final 2-variable regression model given in Equation 1 by approximately 1%. The model serves very well for prediction purpose with predication error about 1.66%.

Similar analysis for the final regression model for latency (Equation 2) shows that the prediction error of the model is approximately 3% (i.e., $\sqrt{MSEP}/\overline{L}$, where $\overline{L}$ is the sample mean). In particular, the t-test of the hypothesis that the bias is zero gives $t = 1.345 < 2.26$, implying the bias of the prediction is insignificant.

5 Conclusions

We applied linear regression methods to analyze the system performance of micropipelines. Various statistics and hypothesis testing establish several interesting relationship between system performance and system timing parameters. In particular, we find that the cycle time and thus throughput of a micropipeline can not be sufficiently captured by the first moment of stage delays. This essentially invalidates the approach proposed in [6] where the second and all other higher moments of the timing parameters are ignored. Instead, we see that the cycle time of a micropipeline can be adequately described by the number of stages and the first two moments of stage delays. In particular, the variance of stage delay affects the cycle time through the logarithm of the number of stages. On the other hand, the latency of a micropipeline can be capture by the first moment of stage delay, demonstrating a rather strong insensitivity to the higher moments. These results promise many interesting applications in both analysis and optimal design of micropipelines.

Acknowledgment

We wish to thank Professor J.-P. Dion at the Department of Mathematics of University of Southern California for his discussions and encouragement on applying linear regression methods throughout this work.

References

[1] I. E. Sutherland. Micropipelines. *Communications of the ACM*, 32(6):720–738, 1989.

[2] M. R. Greenstreet and K. Steiglitz. Bubbles can make self-timed pipelines fast. *Journal of VLSI Signal Processing*, 2(3):139–148, November 1990.

[3] J. Ebergen and R. Berks. Response time properties of some asynchronous circuits. In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC)*, pages 76–86. IEEE Computer Society Press, 1997.

[4] A. Xie and P. A. Beerel. Symbolic techniques for performance analysis of asynchronous systems based on average time separations of events. In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC)*, pages 64–75. IEEE Computer Society Press, 1997.

[5] A. Xie and P. A. Beerel. Accelerating Markovian analysis of asynchronous systems using string-based state compression. In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC)*, pages 247–260. IEEE Computer Society Press, 1998.

[6] P. Kudva, G. Gopalakrishnan, E. Brunvand, and V. Akella. Performance analysis and optimization of asynchronous circuits. In *Proc.*

International Conf. Computer Design (ICCD), pages 221–225, October 1994.

[7] J. O. Rawlings. *Applied Regression Analysis: A Research Tool.* Wadsworth & Brooks, 1988.

[8] A. R. Gallant. *Nonlinear Statistical Models.* New York: Wiley, 1987.

[9] R. R. Hocking. The analysis and selection of variables in linear regression. *Boimetrics*, 32:1–49, 1976.

[10] C. L. Mallows. Some comments on C_p. *Technometrics*, 15:661–675, 1973.

Appendix. The Data Sets

N	μ	σ^2	m_3	T	L
1	2	0.2	0	4	3
1	2	0.6	0	4	3
1	2	0.4	0	4	3
1	2	0.8	0	4	3
1	1.3	0.4	0.504	3.3	2.3
1	1.6	0.64	0.432	3.6	2.6
1	1.9	0.69	0.108	3.9	2.9
1	2.1	0.69	-0.108	4.1	3.1
1	2.4	0.64	-0.432	4.4	3.4
1	2.7	0.41	-0.504	4.7	3.7
2	2	0.2	0	4.18	6.18
2	2	0.6	0	4.42	6.42
2	2	0.4	0	4.32	6.32
2	2	0.8	0	4.48	6.48
2	1.3	0.4	0.504	3.55	4.85
2	1.6	0.64	0.432	4	5.6
2	1.9	0.69	0.108	4.35	6.25
2	2.1	0.69	-0.108	4.55	6.65
2	2.4	0.64	-0.432	4.8	7.2
2	2.7	0.41	-0.504	4.95	7.65

Table 5: The data set for regression.

N	μ	σ^2	m_3	T	L
3	2	0.2	0	4.237	9.45
3	2	0.6	0	4.59	9.941
3	2	0.4	0	4.451	9.729
3	2	0.8	0	4.679	10.085
3	1.3	0.4	0.504	3.734	7.953
3	1.6	0.64	0.432	4.224	8.754
3	1.9	0.69	0.108	4.547	9.719
3	2.1	0.69	-0.108	4.721	10.303
3	2.4	0.64	-0.432	4.922	11.066
3	2.7	0.41	-0.504	5.077	11.491
4	2	0.2	0	4.32	12.709
4	2	0.6	0	4.675	13.506
4	2	0.4	0	4.531	13.174
4	2	0.8	0	4.776	13.744
4	1.3	0.4	0.504	3.86	10.334
4	1.6	0.64	0.432	4.351	11.969
4	1.9	0.69	0.108	4.648	13.24
4	2.1	0.69	-0.108	4.803	14.003
4	2.4	0.64	-0.432	4.957	14.992
4	2.7	0.41	-0.504	4.998	15.626

Table 6: The data set for regression (continued)

N	μ	σ^2	m_3	T	L
5	2	0.2	0	4.35	16.022
5	2	0.6	0	4.729	16.893
5	2	0.4	0	4.581	16.643
5	2	0.8	0	4.829	17.441
5	1.3	0.4	0.504	3.948	13.154
5	1.6	0.64	0.432	4.429	15.218
5	1.9	0.69	0.108	4.706	16.792
5	2.1	0.69	-0.108	4.848	17.735
5	2.4	0.64	-0.432	4.978	18.934
5	2.7	0.41	-0.504	5.001	19.624
6	2	0.2	0	4.412	19.328
6	2	0.6	0	4.761	20.716
6	2	0.4	0	4.616	20.134
6	2	0.8	0	4.863	21.162
6	1.3	0.4	0.504	4.011	15.997
6	1.6	0.64	0.432	4.481	18.49
6	1.9	0.69	0.108	4.744	20.369
6	2.1	0.69	-0.108	4.877	21.494
6	2.4	0.64	-0.432	5	22.89
6	2.7	0.41	-0.504	5.045	23.648

Table 7: The data set for regression (continued)

N	μ	σ^2	m_3	T	L
2	3	0	0	5	8
2	1.5	0.25	0	3.75	5.25
3	1.4	0.44	0.408	3.892	7.946
3	1.8	0.76	0.264	4.483	9.463
4	1.6	0.44	0.192	4.203	11.688
4	2.4	0.44	-0.192	4.907	14.798
5	1.5	0.45	0.3	4.191	14.341
5	1.7	0.21	-0.084	4.005	14.521
6	1.2	0.36	0.576	3.805	15.032
6	2.5	0.25	0	4.956	22.696

Table 8: The data set for model validation.