3-24-2016

A Response Surface Validation of a Quantum Key Distribution Model

Jacob M. Ehrlich

A Response Surface Validation of a Quantum Key Distribution Model

THESIS

Jacob M. Ehrlich, Second Lieutenant, USAF

AFIT-ENS-MS-16-M-104

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-MS-16-M-104

A RESPONSE SURFACE VALIDATION OF A QUANTUM KEY DISTRIBUTION MODEL

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Opertaions Research

Jacob M. Ehrlich, BS

Second Lieutenant, USAF

MARCH 2016

AFIT-ENS-MS-16-M-104

A RESPONSE SURFACE VALIDATION OF A QUANTUM KEY DISTRIBUTION MODEL

THESIS

Jacob M. Ehrlich, BS
Second Lieutenant, USAF

Committee Membership:

Raymond R. Hill, PhD
Chair

Douglas D. Hodson, PhD
Member

AFIT-ENS-MS-16-M-104

Abstract

The need for secure communication in the presence of an adversary introduced the field of cryptology – the practice and study of techniques for secure communication. A common method to secure communication is to distribute a secret key among authorized parties so they can encrypt and decrypt messages between each other. By doing so, ideally, any messages intercepted by a third party are meaningless. An innovative technique to distribute a shared key is Quantum Key Distribution (QKD). QKD uses laws of quantum mechanics to generate and distribute such keys. The purpose of this thesis is to validate an existing mathematical model that is abstract enough to model the essential characteristics of a wide range of QKD system designs. The current model is based on a set of coupled equations. Equation coupling is high as many output variables for a specific phase are inputs for other equations. Because of this, the model output response function is complex, motivating the use of experimentation and response surface modeling to characterize and understand the relationship between inputs and outputs. The mathematical model was designed to capture the essential details associated with a wide variety of system configurations (i.e., designs). Surfaces representing the relationships between inputs and outputs are plotted and used with subject matter experts (SME's) to validate model behavior. After validation, a genetic algorithm is used to optimize the estimated surface. Our findings confirm the complexity of the model and indicate the presence of extreme outliers.

Key words: model validation, design of experiments, response surface methodol–ogy, quantum key distribution, genetic algorithm.

Acknowledgements

I would like to express my sincere gratitude to my advisor, Dr. Ray Hill, for his continuous patience, knowledge, and support during this effort. Without his guidance, this endeavor would not have been possible.

Jacob M. Ehrlich

Table of Contents

Page

List of Figures

List of Tables

A RESPONSE SURFACE VALIDATION OF A QUANTUM KEY DISTRIBUTION MODEL

I. Introduction

1.1 Secure Communication and Cryptology

The need for secure communication arises when two or more entities wish to communicate without a third party "listening in" and understanding the communications. This concern has existed in human society since the dawn of mankind. Ensuring an unauthorized third party is not intercepting and understanding messages may be impossible or may require resources the authorized parties are unwilling or unable to commit to information security; hence the need for cryptology emerged. According to Rivest "Cryptology is about communication in the presence of adversaries" [7]. The basic concept is the authorized parties come up with a technique to encrypt and decrypt messages. Encrypted messages can be sent out in the presence of unauthorized parties as the messages are meaningless without the ability to decrypt it.

Evidence of cryptology exists throughout history. One of the earliest uses of cryptology is the invention of a cipher. An example of a cipher comes from the ancient Greeks. The Spartan military, used a tool called a scytale to encrypt messages during their military campaign. As seen in Figure 1, a scytale consists of a rod with a strip of parchment wrapped around it

The strip alone is a jumble of characters, however when wound around a rod of a specified diameter it reveals the correct order of characters. If a message "Help me I am under attack" needed to be sent one writes across the parchment shown more

Figure 1. Example scytale

clearly in Figure 2. The resulting cipher text becomes "HENTEIDTLAEAPMRC–MUAK".

```
_____________________________
    |   |   |   |   |   |   |
    | H | E | L | P | M |   |
  __| E | I | A | M | U |__|
 |  | N | D | E | R | A |
 |  | T | T | A | C | K |
 |  |   |   |   |   |   |
_____________________________
```

Figure 2. Example Encryption for a scytale

As technology advanced so did cryptology moving from these simple ciphers to complex electromechanical cipher machines. One such machine, the Enigma machine seen in Figure 3, was used by Nazi Germany in World War II [5].

This cipher technology was recently made famous by the movie "The Imitation Game" [4]. The movie notes the brilliance of the Enigma cipher stating "There are 159 million, million, million possible Enigma settings. All we had to do was try each one. But if we had 10 men checking one setting a minute for 24 hours every day and seven days every week, how many days do you think it would take to check each of the settings? Well, it's not days; it's years. It's 20 million years. To stop an incoming

Figure 3. Enigma Machine

attack, we would have to check 20 million years worth of settings in 20 minutes" [4].

1.2 Cryptosystems and the one-time pad

Cryptosystems contain two basic components: an algorithm and one or more keys. "The algorithm is the mathematical transformation used to encrypt and decrypt mes–sages and the key(s) are parameters used in the encryption and decryption processes" [5]. One such cryptographic algorithm is the one-time pad (OTP). Up until its in–vention, by Gilbert Vernam in 1917, any cryptographic algorithm could theoretically be decrypted given enough time and resources (e.g., computational resources, cipher text, etc.) [6].

Kerchoff's principle says that security of a cryptosystem relies on the security of the key rather than the security of the algorithm [9]. The OTP is the only mathemat–ically proven cryptographic algorithm to be unconditionally secure [7]. Unconditional

security means there is no way for an outside agent to intercept the encrypted mes–sage without identifying his presence to both the receiving and send parties. It is a secret-key cryptosystem where a randomly generated key(s), of equal size to the message being sent, is used with the message to encrypt it; and then discarded after use never to be reused again. In Cryptology it is common to refer to the party sending the message as Alice, while the party receiving the message is referred to as Bob. [5].

The third unauthorized party, or eavesdropper, hereafter named Eve, must inter–cept both the key and message. Further Eve must intercept the key without Alice or Bob's detection. Quantum Key Distribution (QKD) provides a means to distribute a key in a secure manner. To understand QKD we reference classic cryptology. As an example, in older times one would have a unique wax seal used to seal messages en–suring the recipient knows whether or not the message was intercepted. QKD applies this concept to modern cryptology by exploiting the laws of quantum mechanics to ensure the privacy of the key being distributed.

1.3 Quantum Key Distribution (QKD)

QKD began with the idea of quantum coding introduced by Stephen Wiesner. Quantum coding is the storage of information on polarized photons using conjugate base pairs [10]. In 1984 a protocol called BB84 was created by Charles Bennett and Gilles Brassard to use QKD [1]. The protocol requires connecting Alice and Bob through both a classic public channel and quantum channel. To ensure security, QKD leverages four concepts from quantum mechanics:

1. Heisenberg Uncertainty Principle states that the exact position, energy, and time of two polarized photons cannot be measured at the same time.

2. Entanglement states physical properties of photons pairs are correlated.

3. Schrödinger's paradox states that observation of a quantum state collapses it.

4. No cloning theorem states perfectly cloning any unknown quantum state is forbidden [9].

The BB84 uses these principles in four main steps:

1. Quantum Exchange,

2. Sifting,

3. Information Reconciliation, and

4. Privacy Amplification. [5]

Step 1 starts with Alice randomly generating a bit (0 or 1) to code onto a photon and a conjugate basis to polarize the photon (rectilinear or diagonal). The rectilinear basis uses 0 and 90 degree polarizations while the diagonal uses 45 and 135 degrees. Alice then sends the encoded polarized photon to Bob. Bob guesses as to the basis used. Assuming a perfect environment, if Bob correctly guesses the basis, he can read the bit encoded onto the photon with 100% accuracy. If, however, Bob picked the wrong basis we know from quantum mechanics that the data is destroyed and a random bit value is measured resulting in a 50% probability the bit Bob measures corresponds to the bit sent by Alice.

Step 2 consists of Bob sharing his random guesses for the basis using the classic public channel, and Alice acknowledging which are correct [5]. The assumption that Bob reads the bit from a correctly guessed basis with 100% accuracy means that, theoretically, Alice and Bob should agree on the bit value for each correctly guessed basis resulting in an identical subset of the original key.

Step 3 involves Bob revealing a subset of the measured bit values from the correctly guessed basis over the public channel. Alice then confirms the values. Comparing

values allow for calculation of an error rate. Errors can result due to environmental noise, non-ideal equipment, and/or eavesdroppers. To meet the guaranteed security requirement, it is assumed that all errors are due to the eavesdropping; however, a practical preset error rate threshold can be set. So long as the error rate measured falls below the threshold the process can continue. In this case, any errors that exist are corrected using an information reconciliation protocol [5].

Step 4 limits Eve's knowledge of the sifted key to a negligible amount by using a random universal hash function from a publically known set of functions. The error rate determined in the previous step and the sifted key are input into the function and a final key is produced that is shortened based on how much knowledge Eve has of the original sifted key. If an error threshold is exceeded, calling the security of the key into question, then the entire process is discarded and Alice starts the process over again with a new key.

1.4 QKD in practice

Limitations of a quantum environment exist and include: the inability of Alice to code and emit one photon at a time, photons become damaged/corrupted/lost in the channel between Alice and Bob, Bob's inability to perfectly measure the photon with his detectors, and imperfect basis alignment between Alice and Bob. This results in errors in the sifted key despite the fact that an eavesdropper may not be present. To counter this in practice, a threshold for accepting the sifted key is set by the users based on the risk tolerance of the individuals and the importance of the messages needing to be sent [5]. Further, if Eve is active enough to intercept a large portion of the messages, her interception action would suppress communication between Alice and Bob. It is thus assumed that Eve is a passive listener.

1.5 Thesis Motivation

A detailed mathematical model of a wide range of QKD system configurations was created by Cernera [2] based on a coupled set of equations. The model was validated by comparing the output of the model to a few specific configurations found in literature [3]. This thesis aims to provide a mechanism for subject matter expert (SME) validation over a wide set of parameters and ranges. Using response surface methodology a series of three dimensional graphs of the model are created and analyzed by SME's.

1.6 Overview

In chapter 2 the complexity of the existing model will be described using graphics and equations as well as identify and describe the input and output variables. Chapter 3 first applies design of experiments to optimize memory allocation (a model input) for the remainder of the analysis. Second a response surface is created for a wide range of system configurations. The model produced is then graphed for use in SME validation. Finally, the result of this surface characterization allows for use of a genetic algorithm to estimate an optimal response for the model.

II. Model Description

2.1 Model Source

The model examined was created by Cernera as part of a Master's thesis at the Air Force Institute of Technology and models a QKD system in terms of various phases [2]. The intended purpose of the model is to allow for the study of critical system parameters, to identify and demonstrate potential bottlenecks that affect the *total system runtime*, which is the sum of time it takes to complete each phase, and conduct analysis of system design trade-offs [2]. The message size was held constant for the entirety of the study at 1Mbit allowing final key generation rate to be the meassure of interest.

2.2 Model Description

The developed model is deterministic and provides a baseline model that can encompass a wide range of QKD systems. Previously QKD was described as using four separate phases [8]. This model uses eight phases to allow for a higher level of detail as depicted in Figure 4. Each one of the eight phases in the model has its own unique set of input and output variables. A list of these parameters is provided in Table 4.

The user first specifies a set system parameters, and inputs for each of the eight phases. The model provides an independent view of the system resource costs as it relates to time, computational workload, and memory consumption. Cernera [2] provides a complete description of the model.

The parameters of the model are listed in Table 1, along with a description.

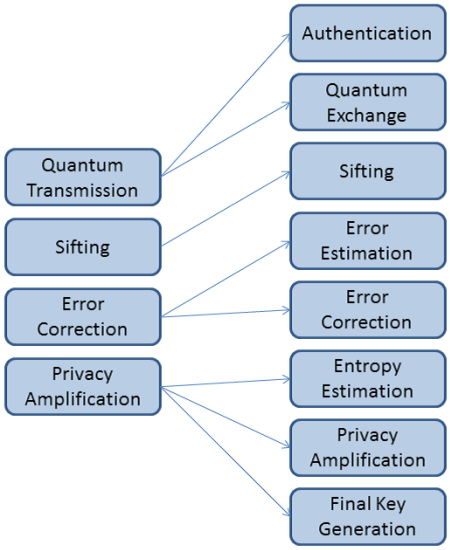


Figure 4. Phase Relationship: mapping the usual four phases in QKD to the eight used in the model

2.3 Equations in Model

The current model is based on a coupled set of equations compiled by Cernerea [2] as extracted from his thorough review of QKD models. The output variables unique to each phase are listed in Table 4. A description of these variables can be seen in Table 2 and 3. Equation coupling is high as many output variables for a specific phase are inputs for other equations. This causes the model output response function to be complex, motivating the use of experimentation and response surface modeling to characterize the output response function.

All transmission times across the classical channel are calculated using Equation 1. Transmission time in seconds is a function of message size, bandwidth, the number of transactions, and the propagation delay of the classical channel.

$$transmission\_time(msg_{size}, bandwidth, num\_trans) = \frac{msg_{size}(bits)}{bandwidth(Mbits/sec)} * num\_trans + t_{class\_prop\_delay}(sec) \quad (1)$$

Table 1. System characterization parameters

Component	Parameter name	Description (units)
	dist_btwn_Alice_Bob	the distance between Alice and Bob (km)
Classical Channel	delay_per_unit_length	The delay per kilometer incurred as a result of propagation (sec/km)
	bandwidth	Information capacity of the classical channel (Mbits/sec)
Quantum Channel	delay_per_unit_length	The delay per kilometer incurred as a result of propagation (sec/km)
	loss_per_km	The average amount of loss, in dB, experienced per kilometer (dB/km)
Alice	pulse_rate	The pulse rate of Alice's laser (Mhz)
	MPN	The Mean Photon Number (unitless)
	$\text{signal}_{percent}$	The percentage of signal states present in Alice's transmission (%)
	$\text{Alice}_{total\_memory}$	The total amount of memory allocated in Alice for QKD (bytes)
	Alice_{cpu_power}	The computational power of Alice's classical processor (unit/sec)
Bob	dB_loss_Bob	The loss as a result of propagation through Bob (dB)
	$\eta_{detector}$	The efficiency of Bob's detectors (unitless)
	t_{dead}	The dead time of Bob's detectors (sec)
	$\text{Bob}_{total\_memory}$	The total amount of memory allocated in Bob for QKD (bytes)
	Bob_{cpu_power}	The computational power of Bob's classical processor (unit/sec)

Table 2. Model Input descriptions

Input variable name	Description (units)
Desired Final Key Bits	The number of final key bits (Mbit)
$\text{auth\_reservoir}_{size}$	The number of bytes in the authentication reservoir that can be used to authenticate a message (bytes)
auth_key_req	The number of bytes required to authenticate a message (bytes)
AB_avg_msg_size	The average size of a message passed between Alice and Bob that is sent from Alice (bytes)
BA_avg_msg_size	The average size of a message passed between Alice and Bob that is sent from Bob (bytes)
AB_num_trans	The number of transactions between Alice and Bob that are initiated by Alice (unitless)
BA_num_trans	The number of transactions between Alice and Bob that are initiated by Bob (unitless)
mem_req_pulse	The amount of memory required to store information on each pulse (bytes)
sifting_eff_frac	The approximate ratio of detections to corrrect basis measurements, as defined by the protocol (unitless)
bits_sacrificed_pct	The percentage of the sifted key buffers sacrificed to provide an error estimate (%)
num_bits_sacrficed	The number of bits sacrificed for each Error Reconciliation routine (unitless)
block_{size}	The required input block size to perform an Error Reconciliation routine (bits)
pct_entropy_loss_QBER	The percentage of key lost due to entropy loss on the quantum channel (%)
pct_entropy_loss_public	The percentage of key lost due to entropy loss durring Error Reconciliation (%)
pct_entropy_loss_multi_photon	The percentage of key lost due to multi-photon pulses (%)
pct_entropy_loss_safety	the percentage of key lost due to arbitrary safety margin (%)
min_num_req_bits	The minimum number of required bits necessary to perfom Privacy Amplification (bits)
num_auth_reservoir_bits	The number of bitys reserved for the Authentication reservoir (bits)

Table 3. Model Output descriptions

Output variable name	Description (units)
T_Auth	The time required to complete a single authentication (sec)
auth_reservoir_remaining$_{size}$	The number of bytes remaining in the authentication reservoir (bytes)
T$_{QE}$	The time required to complete a single iteration of Quantum Exchange (sec)
A_raw$_{buffer}$	Ther size of Alice's memory buffer after completion of QE (bytes)
B_raw$_{buffer}$	The size of Bob's memory buffer after completion of QE (bytes)
Alice_candidate_key_bits	The number of bits Alice possesses at the end of QE with the potential to be final key bits (bits)
Bob_candidate_key_bits	The number of bits Bob possesses at the end of QE with the portential to be final key bits (bits)
T$_{sift}$	The time required to complete the sifting process (sec)
B_sift$_{buffer}$	The size of Bob's sifted key buffer (bytes)
A_sift$_{buffer}$	The size of Alice's sifted key buffer (bytes)
T$_{ErrEst}$	The time required to complete Error Estimation (sec)
A_ErrEst$_{buffer}$	The size of Alice's Error Estimated key buffer (bytes)
B_ErrEst$_{buffer}$	The size of Bob's Error Estimated key buffer (bytes)
T$_{ErrRec}$	The total time required to complete a round of Error Reconciliation (sec)
A_ErrRec$_{buffer}$	The size of the Error Reconciled buffer (bytes)
B_ErrRec$_{buffer}$	The size of the Error Reconciled buffer (bytes)
num_err_rec	The total number of Error Reconciliation routines required to process the input buffer (unitless)
T$_{EntEst}$	the time required to complete Entropy Estimation (sec)
A_EntEst$_{buffer}$	The size of the Entropy Estimated buffer (bytes)
B_EntEst$_{buffer}$	the size of the Entropy Estimated buffer (bytes)
N$_{secure}$	The number of bits that can be saved after Privacy Amplification (bytes)
T$_{PrivAmp}$	The time required to complete Privacy Amplification (sec)
A_PrivAmp$_{buffer}$	The size of Alice's Privacy Amplified buffer (bytes)
B_PrivAmp$_{buffer}$	The size of Bob's Privacy Amplified buffer (bytes)
T$_{FKG}$	The time required to complete Final Key Generation (sec)
A_FKG$_{buffer}$	The size of Alice's Final Key Generation buffer (bytes)
B_FKG$_{buffer}$	The size of Bob's Final Key Generation buffer (bytes)

Table 4. Model Input and Output

Phase	Input variable name	Output variable name
Desired vs Actual Performance	Desired Final Key Bits	
Authentication	auth_reservoir$_{size}$ auth_key_req AB_avg_msg_size BA_avg_msg_size AB_num_trans BA_num_trans	T_Auth auth_reservoir_remaining$_{size}$
Quantum Exchange	mem_req_pulse	T$_{QE}$ A_raw$_{buffer}$ B_raw$_{buffer}$ Alice_candidate_key_bits Bob_candidate_key_bits
Sifting	sifting_eff_frac AB_avg_msg_size BA_avg_msg_size AB_num_trans BA_num_trans	T$_{sift}$ B_sift$_{buffer}$ A_sift$_{buffer}$
Error Estimation	bits_sacrificed_pct AB_avg_msg_size BA_avg_msg_size AB_num_trans BA_num_trans	T$_{ErrEst}$ A_ErrEst$_{buffer}$ B_ErrEst$_{buffer}$
Error Reconciliation	num_bits_sacrficed block_size AB_avg_msg_size BA_avg_msg_size AB_num_trans BA_num_trans	T$_{ErrRec}$ A_ErrRec$_{buffer}$ B_ErrRec$_{buffer}$ num_err_rec
Entropy Estimation	pct_entropy_loss_QBER pct_entropy_loss_public pct_entropy_loss_multi_photon pct_entropy_loss_safety AB_avg_msg_size BA_avg_msg_size AB_num_trans BA_num_trans	T$_{EntEst}$ A_EntEst$_{buffer}$ B_EntEst$_{buffer}$ N$_{secure}$
Privacy Amplification	min_num_req_bits AB_avg_msg_size BA_avg_msg_size AB_num_trans BA_num_trans	T$_{PrivAmp}$ A_PrivAmp$_{buffer}$ B_PrivAmp$_{buffer}$
Final Key Generation	num_auth_reservoir_bits AB_avg_msg_size BA_avg_msg_size AB_num_trans BA_num_trans	T$_{FKG}$ A_FKG$_{buffer}$ B_FKG$_{buffer}$

Equation 2 calculates the total number of pulses that Alice sends as a function of the amount of memory allocated to Quantum Exchange, and the amount of memory required to send a pulse.

$$num\_pulses\_sent = \frac{A_{mem\_avail}}{mem\_req\_per\_pulse} \tag{2}$$

Equation 3 calculates the number of detections that will actually be received by Bob as a function of efficiencies of the channel and Bob's hardware, the probability that a pulse contains a photon(s), the chosen signal percentage, and the total number of pulses sent by Alice.

$$num\_det\_at\_Bob = num\_pulses\_sent * Pois(X \geq 1) * sig_{percent} * \eta_{channel} * \eta_{Bob} * \eta_{det} \tag{3}$$

Equation 4 calculates the total amount of time (in seconds) required for a single authentication as a function of time it takes to transmit data between Alice and Bob and the time required to perform computations.

$$\begin{aligned} &T_{Auth} = \\ &transmission\_time(avg\_msg\_size_{\{AB,BA\}}, bandwidth, num\_trans_{(AB,BA)}) + \\ &\qquad\qquad \frac{\{A,B\}_{workload_{Auth}}}{\{A,B\}_{cpu_{power}}} \end{aligned} \tag{4}$$

Equation 5 calculates the amount of authentication key (in bytes) remaining in the authentication reservoir after the phase has completed execution as a function of the number of bytes in the authentication reservoir that can be used to authenticate a message and the number of bytes required to authenticate a message.

$$auth\_reservoir\_remaining_{size} = (auth\_reservoir_{size} - auth\_key\_required) \tag{5}$$

Equation 6 calculates the time (in seconds) required to complete a single iteration of Quantum Exchange (QE) as a function of either the total time it takes for Alice to transmit all pulses required to fill her memory or the amount of time it takes to complete the classical processing necessary to process the information associated with each pulse, whichever takes longer.

$$T_{QE} = max(\frac{num\_det\_at\_Bob}{actual\_det\_rate} + t_{quant\_prop\_delay}, \frac{\{A,B\}_{workload_{QE}}}{\{A,B\}_{cpu_{power}}}) \tag{6}$$

Given that Alice and Bob must store information about each pulse during QE, Equation 7 and Equation 8 calculate the size (in bytes) of the raw memory buffers after completion of QE for Alice and Bob, respectively as a function of the number of pulses sent (Alice) and detections (Bob) times the memory required for each pulse.

$$A_{raw_{buffer}} = num\_pulses\_sent * mem\_req\_pulse \tag{7}$$

$$B_{raw_{buffer}} = num\_det\_at\_Bob * mem\_req\_pulse \tag{8}$$

Equation 9 and Equation 10 calculate the number of candidate bits at the end of QE (i.e., bits that have the potential to become final key bits) as a function of the number of pulses sent (Alice) and detections (Bob), respectively.

$$A_{QE\_candidate\_key\_bits} = num\_pulses\_sent = \frac{A_{mem\_avail}}{mem\_req\_per\_pulse} \tag{9}$$

$$\begin{aligned} &B_{QE\_candidate\_key\_bits} = num\_det\_at\_Bob = \\ &num\_pulses\_sent * Pois(X \geq 1) * sig_{percent} * \eta_{channel} * \eta_{Bob} * \eta_{det} \end{aligned} \tag{10}$$

Equation 11 calculates the time (in seconds) required to complete the sifting pro–

cess as a function of the time it takes to transmit required data across the classical channel and the computational time required.

$$T_{Sift} = transmission\_time(avg\_msg\_size_{\{AB,BA\}}, bandwidth, num\_trans_{\{AB,BA\}}) + \frac{\{A,B\}_{workload_{Sift}}}{\{A,B\}_{cpu_{power}}} \quad (11)$$

Equation 12 calculates the size of Alice and Bob's sifted key buffer (in bytes) and is the result of removing both the unnecessary information about each pulse and mismatched basis measurements between Bob and Alice.

$$A_{sift_{buffer}} = B_{sift_{buffer}} = sift\_eff\_frac * \frac{B_{raw_{buffer}}}{mem\_req\_pulse} \quad (12)$$

Equation 13 calculates the time (in seconds) required to complete Error Estimation as a function of the time it takes to transmit required data across the classical channel and the computational time required.

$$T_{ErrEst} = transmission_time(avg\_msg\_size_{\{AB,BA\}}, bandwidth, num\_trans_{\{AB,BA\}} + \frac{\{A,B\}_{workload_{ErrEst}}}{\{A,B\}_{cpu_{power}}} \quad (13)$$

Equation 14 and Equation 15 calculate the size of Alice and Bob's error estimated key buffer (in bytes), respectively as a function of percentage of bits that are saved multiplied by the size of the sifted key buffer.

$$A_{ErrEst_{buffer}} = (1 - bits\_sacrificed_{pct}) * A_{Sift_{buffer}} \quad (14)$$

$$B_{ErrEst_{buffer}} = (1 - bits\_sacrificed_{pct}) * B_{Sift_{buffer}} \tag{15}$$

Equation 16 calculates the total time (in seconds) required to complete a round of Error Reconciliation as a function of the time it takes to complete a single block of input key multiplied by the number of total blocks in the error estimated input buffer. If the number of blocks do not divide the error estimated buffer evenly, the last block is padded in order for the algorithm to run successfully.

$$\begin{aligned} T_{ErrRec} = ceil(\frac{\{A,B\}_{ErrEst_{buffer}}}{block_{size}}) * \\ (transmission\_time(avg\_msg\_size_{\{AB,BA\}}, bandwidth, num\_trans_{\{AB,BA\}}) + \\ \frac{\{A,B\}_{workload_{ErrRec}}}{\{A,B\}_{cpu_{power}}}) \end{aligned} \tag{16}$$

Equation 17 and Equation 18 calculate the size of the error reconciled buffer (in bytes) as a function of size of the error estimated buffer and the number of bits sacrificed during all iterations of the algorithm (if any).

$$\begin{aligned} A_{EntEst_{buffer}} = A_{ErrRec_{buffer}} = \\ A_{ErrEst_{buffer}} - (num\_bits\_sacrificed * num\_err\_rec) \end{aligned} \tag{17}$$

$$\begin{aligned} B_{EntEst_{buffer}} = B_{ErrRec_{buffer}} = \\ B_{ErrEst_{buffer}} - (num\_bits\_sacrificed * num\_err\_rec) \end{aligned} \tag{18}$$

Equation 19 calculates the total number of error reconciliation iterations as a function of the number of blocks contained in the error estimated buffer.

$$num\_err\_rec = ceil(\frac{\{A,B\}_{ErrEst_{buffer}}}{block_{size}}) \tag{19}$$

Equation 20 calculates the time required to complete Entropy Estimation in sec–

onds as a function of the time it takes to transmit required data across the classical channel and the computational time required.

$$T_{EntEst} = transmission\_time(avg\_msg\_size_{\{AB,BA\}}, bandwidth, num\_trans_{\{AB,BA\}}) + \frac{\{A,B\}_{workload_{EntEst}}}{\{A,B\}_{cpu_{power}}} \tag{20}$$

Equation 21 calculates the number of bits that can be saved during the upcoming Privacy Amplification phase as a function of the percentage of the key remaining after losses and the size of the error reconciled buffer.

$$N_{secure} = (1 - total\_ent\_loss\_pct) * \{A,B\}_{ErrRec_{buffer}} \tag{21}$$

Equation 22 calculates the time (in seconds) required to complete Privacy Am–plification as a function of time it takes to transmit required data across the classical channel and the computational time required.

$$T_{PrivAmp} = transmission\_time(avg\_msg\_size_{\{AB,BA\}}, bandwidth, num\_trans_{\{AB,BA\}}) + \frac{\{A,B\}_{workload_{PrivAmp}}}{\{A,B\}_{cpu_{power}}} \tag{22}$$

Equation 23 shows the size (in bytes) of the privacy amplified buffer is equal to the number of bytes that can be saved as a result of entropy estimation.

$$A_{PrivAmp_{buffer}} = B_{PrivAmp_{buffer}} = N_{secure} = (1 - total\_ent\_loss\_pct) * \{A,B\}_{ErrRec_{buffer}} \tag{23}$$

Equation 24 calculates the time (in seconds) required to complete Final Key Gen–

eration as a function of the time it takes to transmit required data across the classical channel and the computational time required.

$$T_{FKG} = \\ transmission\_time(avg\_msg\_size_{\{AB,BA\}}, bandwidth, num\_trans_{\{AB,BA\}}) + \frac{\{A,B\}_{workload_{FKG}}}{\{A,B\}_{cpu_{power}}} \quad (24)$$

Equation 25 and Equation 26 calculate the size (in bytes) of the final key buffer for Alice and Bob, respectively, as a function of the size of the privacy amplified buffer and the need to reserve bits for the authentication reservoir.

$$A_{FKG_{buffer}} = A_{PrivAmp_{buffer}} - num\_auth\_reservoir\_bits \quad (25)$$

$$B_{FKG_{buffer}} = B_{PrivAmp_{buffer}} - num\_auth\_reservoir\_bits \quad (26)$$

2.4 Implementation

Equations 1 to 26 are coupled together and implemented using Excel functions and Excel VBA. The model is split into two sheets. The first sheet, shown in Figure 13 contains inputs for all high-level system parameters.

All equations are confirmed to be accurately coded into the model. If the inputs contain a problem that must be corrected the cells will turn red to signify that. Green signifies a valid configuration. The second sheet contains the mathematical description in an Input - Output style, subdivided by the eight phases and can be seen in Figures 14 to 20.

III. Response Surface of Model Output

3.1 Overview

We are interested in characterizing the quantum key distribution model output. Given the highly coupled nature of the model, the output response is hypothesized as quite complex, likely highly nonlinear. We employ response surface methods to accomplish the characterization. We consider the model a "blackbox", present inputs, and collect outputs. Inputs are listed and described in Table 5. Three outputs are collected: the number of bits in the final key, the final rate of bits processed measured in bits/sec, and total system runtime. The first two outputs matter only in terms of identifying the third, the time it will take to complete the entire quantum key exchange. Each phase takes a certain number of iterations to complete the QKD process. The total time is calculated by determining the amount of time it takes to complete an iteration, times the number of iterations added for each of the phases. The limiting factor is the time it takes to complete the process and is used as the only response variable for the purpose of the study. The desired final key bits is expected to overshadow other variables and is known to greatly affect the response. To gain a deeper understanding of the model the message size was held constant for the entirety of the study at 1Mbit.

The first objective of the study is to determine an optimal setting of memory allocation while holding all other variables constant. The second is to characterize the relationship of the remaining inputs as related to the output. The third objective is to apply a heuristic search method to the response function to locate optimal settings to the QKD model.

Table 5. Factors and Response

Inputs/Factors	Description
Aauth	Percent memory allocation assigned to Authentication for Alice
AQE	Percent memory allocation assigned to Quantum Exchange for Alice
Asift	Percent memory allocation assigned to Sifting for Alice
AErrEst	Percent memory allocation assigned to Error Estimation for Alice
AErrRec	Percent memory allocation assigned to Error Reconciliation for Alice
Aent	Percent memory allocation assigned to Entropy Estimation for Alice
AprivAmp	Percent memory allocation assigned to Privacy Amplification for Alice
AFKG	Percent memory allocation assigned to Final Key Generation for Alice
Bauth	Percent memory allocation assigned to Authentication for Bob
BQE	Percent memory allocation assigned to Quantum Exchange for Bob
Bsift	Percent memory allocation assigned to Sifting for Bob
BErrEst	Percent memory allocation assigned to Error Estimation for Bob
BErrRec	Percent memory allocation assigned to Error Reconciliation for Bob
Bent	Percent memory allocation assigned to Entropy Estimation for Bob
BprivAmp	Percent memory allocation assigned to Privacy Amplification for Bob
BFKG	Percent memory allocation assigned to Final Key Generation for Bob
dist	The distance between Alice and Bob, designated in model: dist_btwn_Alice_Bob
loss	The average amount of loss, in dB, experienced per kilometer, designated in model: loss_per_km
MPN	The Mean Photon Number
signal	The percentage of signal states present in Alice's transmission, designated in model: $\text{signal}_{percent}$
etadetect	The effeciency of Bob's detectors, designated in model: $\eta_{detector}$
tdead	The dead time of Bob's detectors, designated in model: t_{dead}
Amem	The total amount of memory allocated to Alice for QKD, designated in model: $\text{Alice}_{total\_memory}$
Bmem	The total amount of memory allocated to Bob for QKD, designated in model: $\text{Bob}_{total\_memory}$
Output/Response	
Runtime	Amount of time it take to complete the QKD process, designated in model: Total System Runtime

3.2 Statistical Methods

Characterizing the relationship between the inputs and outputs is of interest. We apply mathematical and statistical methods to understand the nature of the relationship. We call the inputs to the "black box" factors and the output a response, y.

In general a response y of interest is influenced by several input variables, or factors, that are controlled to levels specified by an experimental design. The response is a function of n factors plus some uncontrollable noise or error term assumed to follow a normal distribution centered at zero with constant variance; this is seen in Equation 27.

$$y = f(x_1, x_2, ...x_n) + \epsilon \tag{27}$$

However, because this is a deterministic computer model, there is no variability in the model. Denoting the expected response by $E(y) = f(x_1, x_2, ..., x_n) = \eta$ the surface represented by $\eta = f(x_1, x_2, ..., x_n)$ is called a response surface. This response surface characterizes the relationship between the factors and the response. The model that relates the response to the factors is estimated using least-squares multiple linear regression methods. All factors in Table 5 are continuous except for the total amount of memory allocated to Alice and Bob for QKD. Memory is inherently discrete and is determined by 2^z, where z is an integer. When a design is chosen to collect the data the variables in the natural space are converted into a coded space where all factors range from -1 to 1.

Although there is no random error in our sampling, the resulting analysis does contain an error component which is generally called the lack of fit (LOF) component. The LOF component represents all potential model terms excluded from the model.

Exclusion of a parameter assumes the parameter is insignificant and therefore equal to zero, and the estimate is noise. The LOF component is used to assess parameter significance in subsequent analyses.

3.3 Memory allocation: screening phase

The lack of variability in the black box model removes the need for replications. The study began with the desire to screen the variables designated to memory allo–cation for each of the eight phases for both Alice and Bob. The allocation of memory is defined as a percentage of the total memory. For Alice and Bob the sum of the percentage of memory allocated across each of the eight phases cannot exceed 100% of the memory available. Percentages are nonnegative and each phase must have a nonzero percentage to complete. Defining an appropriate range of values for each factor is important to ensure proper memory allocation. The high level was set to 12% and the low level was set to 5%.

The relationship between the memory allocation and the response is unknown; however, knowledge of the model leads to a hypothesis that in addition to main effects, factor interactions between phases may be significant, especially given the level of coupling among the model equations. Selecting a design that identifies the main effects and two-level interactions for the 16 factors resulted in 144 runs.

The data was fit to a first order model with two-level interactions and statisti–cally significant parameters were identified. A list of significant parameters sorted according to significance are shown in Figure 5.

The only significant parameter identified was AQE, the memory allocation as–signed to Alice for quantum exchange. The distribution of responses is seen in Figure 6 and shows the negligible effect of all factors except for the significant parameter already identified. This results in Figure 6 having only three points, one point for

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
AQE	-8.0032	0.268488	-29.81	<.0001*
Aauth*BErrRec	-0.015376	0.282104	-0.05	0.9581
AErrEst*BQE	0.0146131	0.294726	0.05	0.9618
AQE*AErrEst	0.014538	0.301614	0.05	0.9629
Bent	-0.012664	0.274101	-0.05	0.9644
Aent	0.0140571	0.307235	0.05	0.9648

Figure 5. Sorted parameters for memory allocation

each level of the parameter identified (low or -1, middle or 0, high or 1)

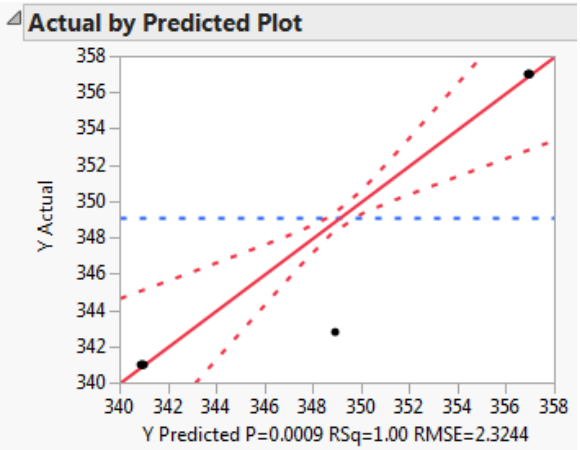


Figure 6. Memory allocation response distribution

After observing a single significant factor for memory allocation all nonsignificant factors for Alice's memory allocation were set to the low level of 5% to allow for a larger exploratory range in the significant factor for the remainder of the analysis. Factors for Bob's memory allocation remain at the midpoint.

3.4 Response Surface Equation

The next step is to characterize the relationship of the remaining factors found in Table 5 in relation to the response. The levels for the factors are in Table 6. The default settings are used as the center point of the design space.

Bob's memory allocation did not effect the response time. It is reasonable then to assume that the total memory available to Bob is insignificant over the range of

Table 6. Factors Levels

Factors (units)	Low level (-1)	High level (1)
AQE(%)	0.05	0.35
dist (km)	14	32
loss (dB/km)	0.1	0.3
MPN	0.5	0.7
signal (%)	0.6	0.9
etadetect	0.05	0.15
tdead (μs)	15	25
Amem (GB)	2	16

memory considered. A quick screening test was run using all values in Table 6 that confirmed this hypothesis as well as ensure all remaining effects significantly explain variability in the response in some way. Therefore Bmem is removed as a factor in the response surface model.

Since the model response is expected to be complex no model form is assumed *a priori*. The existence of discrete variables limits the use of some space filling designs, so a grid approach was used. The size of the grid determines the largest order of a model that can be estimated. A grid size of five levels in the seven continuous factors based on a step size of 0.5 was selected. Practically it was determined that the values for memory were 2, 4, 8, 16 based on current industry values for available memory. This grid permits estimation of up to a fourth order response surface model. This yielded 312,500 design points which provide a high enough resolution for the purpose of identifying a response surface. The grid was centered on the default settings. The resulting ANOVA is in Figure 7, while the actual model parameters are in Figure 8 through 11.

The model includes the significant parameters, those with an α value less than 0.05. The model contains polynomials to the third degree in all variables except the percentage of signal states present in Alice's transmission (*signal*) which is quadratic, and the dead time of Bob's detectors (*tdead*) which is linear. Interaction terms contain

Summary of Fit

RSquare	0.858855
RSquare Adj	0.858793
Root Mean Square Error	1051.15
Mean of Response	1433.562
Observations (or Sum Wgts)	312500

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	138	2.1001e+12	1.522e+10	13773.12
Error	312361	3.4513e+11	1104915.8	Prob > F
C. Total	312499	2.4452e+12		<.0001*

Figure 7. ANOVA from Grid design

up to six variables in all variables except *tdead*. Significant interactions with *tdead* are those between both the average amount of loss, in dB, experienced per km (*loss*) and the efficiency of Bob's detectors (*etadetect*), and the interaction between all three.

The coupling of equations results in a complex response surface. Figure 26 shows a three dimensional representation of the response surface in the percent memory al–located to quantum exchange (AQE) and the effeciency of Bob's detectors (*etadetect*). However, while the interactions are significant and included in the model, Figure 47 shows that the response is primarily linear in terms of the dead time of Bob's detectors (*tdead*). A complete set of three dimensional graphs are found in the appendix.

3.5 Response Surface Optimization

The estimated response function was optimized using a Genetic Algorithm (GA). A GA is a search procedure inspired by natural selection. Solutions to problems are considered members of a population. Each specific population member is an encoded solution (set of input values) for the model. Each member, solution, has associated with it a value equating to the strength of that member; we use the objective function value as the strength value. We then provide a biased reproduction process where

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	-851.2664	31.72015	-26.84	<.0001*
AQE	-1558.929	9.959763	-156.5	<.0001*
dist	995.40745	9.959763	99.94	<.0001*
loss	1358.9281	9.959763	136.44	<.0001*
MPN	-344.136	6.513737	-52.83	<.0001*
signal	-557.9831	6.513737	-85.66	<.0001*
etadetect	-1272.558	9.959763	-127.8	<.0001*
tdead	9.2492636	2.659222	3.48	0.0005*
Amem	1209.9516	47.77563	25.33	<.0001*
AQE*dist	843.96495	9.211815	91.62	<.0001*
AQE*loss	1145.207	9.211815	124.32	<.0001*
dist*loss	-64.77811	9.211815	-7.03	<.0001*
AQE*dist*loss	-315.8645	13.02747	-24.25	<.0001*
AQE*MPN	-329.498	9.211815	-35.77	<.0001*
dist*MPN	148.25474	9.211815	16.09	<.0001*
AQE*dist*MPN	200.79627	13.02747	15.41	<.0001*
loss*MPN	203.91164	9.211815	22.14	<.0001*
AQE*loss*MPN	275.35265	13.02747	21.14	<.0001*
dist*loss*MPN	-96.10957	13.02747	-7.38	<.0001*
AQE*dist*loss*MPN	-90.38852	18.42363	-4.91	<.0001*
AQE*signal	-535.8262	9.211815	-58.17	<.0001*
dist*signal	240.80876	9.211815	26.14	<.0001*
AQE*dist*signal	325.83942	13.02747	25.01	<.0001*
loss*signal	327.20598	9.211815	35.52	<.0001*
AQE*loss*signal	444.58324	13.02747	34.13	<.0001*
dist*loss*signal	-150.5425	13.02747	-11.56	<.0001*
AQE*dist*loss*signal	-148.4125	18.42363	-8.06	<.0001*
MPN*signal	-100.2427	9.211815	-10.88	<.0001*
AQE*MPN*signal	-129.5914	13.02747	-9.95	<.0001*
dist*MPN*signal	74.039245	13.02747	5.68	<.0001*
AQE*dist*MPN*signal	80.851858	18.42363	4.39	<.0001*
loss*MPN*signal	103.91023	13.02747	7.98	<.0001*
AQE*loss*MPN*signal	110.75667	18.42363	6.01	<.0001*

Figure 8. Parameter values part 1

the ability of a members value to reproduce is dependent upon the members' fitness. Starting with a population of solutions, better solutions create more influence on future solutions. Over a number of generations the evolution process ideally yields an optimal solution from within the population.

The response function obtained is optimized using Matlab's genetic algorithm function "*ga*". The inputs to the function are

1. Objective function,
2. Number of variables,
3. Upper and Lower bounds on each variable, and
4. Integer Constraints where applicable.

Parameter Estimates

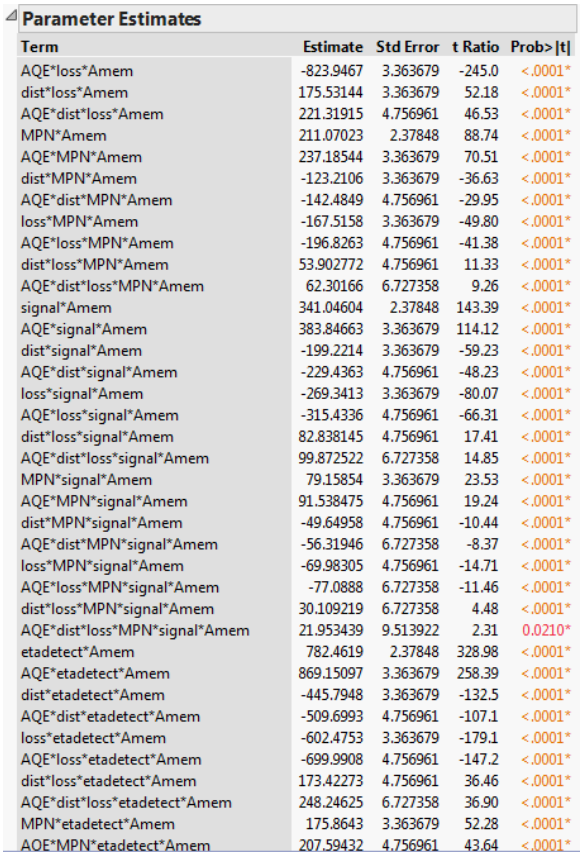
Term	Estimate	Std Error	t Ratio	Prob>|t|
dist*loss*MPN*signal	-45.03321	18.42363	-2.44	0.0145*
AQE*dist*loss*MPN*signal	-37.30989	26.05495	-1.43	0.1522
AQE*etadetect	-1207.514	9.211815	-131.1	<.0001*
dist*etadetect	516.63635	9.211815	56.08	<.0001*
AQE*dist*etadetect	714.90532	13.02747	54.88	<.0001*
loss*etadetect	707.74128	9.211815	76.83	<.0001*
AQE*loss*etadetect	979.74994	13.02747	75.21	<.0001*
dist*loss*etadetect	-325.44	13.02747	-24.98	<.0001*
AQE*dist*loss*etadetect	-366.8836	18.42363	-19.91	<.0001*
MPN*etadetect	-216.3122	9.211815	-23.48	<.0001*
AQE*MPN*etadetect	-293.7724	13.02747	-22.55	<.0001*
dist*MPN*etadetect	159.48219	13.02747	12.24	<.0001*
AQE*dist*MPN*etadetect	183.72708	18.42363	9.97	<.0001*
loss*MPN*etadetect	217.60169	13.02747	16.70	<.0001*
AQE*loss*MPN*etadetect	258.90331	18.42363	14.05	<.0001*
dist*loss*MPN*etadetect	-86.5821	18.42363	-4.70	<.0001*
AQE*dist*loss*MPN*etadetect	-112.8856	26.05495	-4.33	<.0001*
signal*etadetect	-350.745	9.211815	-38.08	<.0001*
AQE*signal*etadetect	-481.3987	13.02747	-36.95	<.0001*
dist*signal*etadetect	268.88449	13.02747	20.64	<.0001*
AQE*dist*signal*etadetect	304.66205	18.42363	16.54	<.0001*
loss*signal*etadetect	358.64128	13.02747	27.53	<.0001*
AQE*loss*signal*etadetect	437.80201	18.42363	23.76	<.0001*
dist*loss*signal*etadetect	-144.1898	18.42363	-7.83	<.0001*
AQE*dist*loss*signal*etadetect	-187.1895	26.05495	-7.18	<.0001*
MPN*signal*etadetect	-110.9202	13.02747	-8.51	<.0001*
AQE*MPN*signal*etadetect	-126.0654	18.42363	-6.84	<.0001*
dist*MPN*signal*etadetect	68.808832	18.42363	3.73	0.0002*
AQE*dist*MPN*signal*etadetect	74.047893	26.05495	2.84	0.0045*
loss*MPN*signal*etadetect	101.94716	18.42363	5.53	<.0001*
AQE*loss*MPN*signal*etadetect	103.49545	26.05495	3.97	<.0001*
dist*loss*MPN*signal*etadetect	-51.60618	26.05495	-1.98	0.0476*
AQE*Amem	1083.6927	2.37848	455.62	<.0001*
dist*Amem	-547.7208	2.37848	-230.3	<.0001*
AQE*dist*Amem	-606.3855	3.363679	-180.3	<.0001*
loss*Amem	-741.2709	2.37848	-311.7	<.0001*

Figure 9. Parameter values part 2

The objective function is the regression model from Figure 8 to Figure 11, with minor modifications to account for the discrete variable *Amem*. The values *Amem* can take on are 2,4,8,16. A simple transformation is made from *Amem* to *Amemtrans* using the relationship seen in Equation 28. Applying a lower bound of 1 and an upper bound of 4 in addition to an Integer constraint on the transformed variable *Amemtrans* ensures the optimized solution found using GA uses one of the required discrete values in *Amem*.

$$Amem = 2^{Amemtrans} \tag{28}$$

After the transformation of *Amem* remaining variables and bounds are listed in Table 6. The estimated optimal settings obtained using the GA are seen in Table 7.



Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
AQE*loss*Amem	-823.9467	3.363679	-245.0	<.0001*
dist*loss*Amem	175.53144	3.363679	52.18	<.0001*
AQE*dist*loss*Amem	221.31915	4.756961	46.53	<.0001*
MPN*Amem	211.07023	2.37848	88.74	<.0001*
AQE*MPN*Amem	237.18544	3.363679	70.51	<.0001*
dist*MPN*Amem	-123.2106	3.363679	-36.63	<.0001*
AQE*dist*MPN*Amem	-142.4849	4.756961	-29.95	<.0001*
loss*MPN*Amem	-167.5158	3.363679	-49.80	<.0001*
AQE*loss*MPN*Amem	-196.8263	4.756961	-41.38	<.0001*
dist*loss*MPN*Amem	53.902772	4.756961	11.33	<.0001*
AQE*dist*loss*MPN*Amem	62.30166	6.727358	9.26	<.0001*
signal*Amem	341.04604	2.37848	143.39	<.0001*
AQE*signal*Amem	383.84663	3.363679	114.12	<.0001*
dist*signal*Amem	-199.2214	3.363679	-59.23	<.0001*
AQE*dist*signal*Amem	-229.4363	4.756961	-48.23	<.0001*
loss*signal*Amem	-269.3413	3.363679	-80.07	<.0001*
AQE*loss*signal*Amem	-315.4336	4.756961	-66.31	<.0001*
dist*loss*signal*Amem	82.838145	4.756961	17.41	<.0001*
AQE*dist*loss*signal*Amem	99.872522	6.727358	14.85	<.0001*
MPN*signal*Amem	79.15854	3.363679	23.53	<.0001*
AQE*MPN*signal*Amem	91.538475	4.756961	19.24	<.0001*
dist*MPN*signal*Amem	-49.64958	4.756961	-10.44	<.0001*
AQE*dist*MPN*signal*Amem	-56.31946	6.727358	-8.37	<.0001*
loss*MPN*signal*Amem	-69.98305	4.756961	-14.71	<.0001*
AQE*loss*MPN*signal*Amem	-77.0888	6.727358	-11.46	<.0001*
dist*loss*MPN*signal*Amem	30.109219	6.727358	4.48	<.0001*
AQE*dist*loss*MPN*signal*Amem	21.953439	9.513922	2.31	0.0210*
etadetect*Amem	782.4619	2.37848	328.98	<.0001*
AQE*etadetect*Amem	869.15097	3.363679	258.39	<.0001*
dist*etadetect*Amem	-445.7948	3.363679	-132.5	<.0001*
AQE*dist*etadetect*Amem	-509.6993	4.756961	-107.1	<.0001*
loss*etadetect*Amem	-602.4753	3.363679	-179.1	<.0001*
AQE*loss*etadetect*Amem	-699.9908	4.756961	-147.2	<.0001*
dist*loss*etadetect*Amem	173.42273	4.756961	36.46	<.0001*
AQE*dist*loss*etadetect*Amem	248.24625	6.727358	36.90	<.0001*
MPN*etadetect*Amem	175.8643	3.363679	52.28	<.0001*
AQE*MPN*etadetect*Amem	207.59432	4.756961	43.64	<.0001*

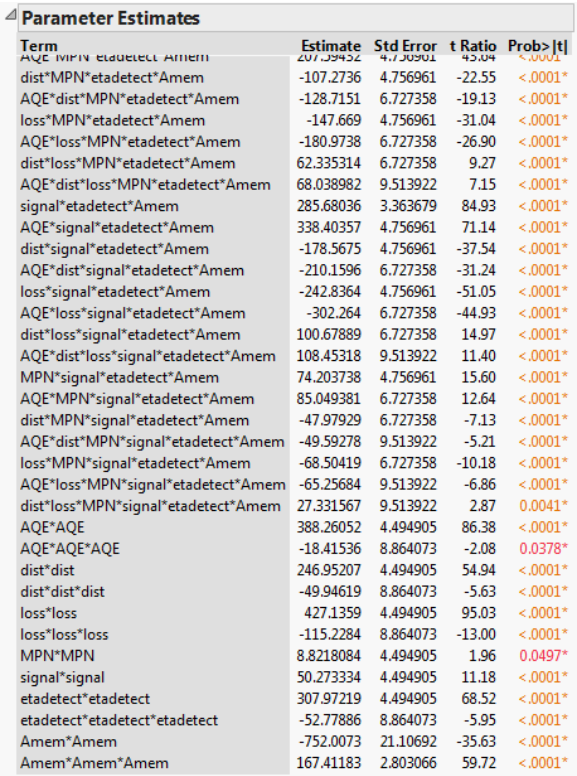
Figure 10. Parameter values part 3

Applying these settings to the Excel model results in a response value of 7749.082661.

3.6 SME Validation Method

The response surface characterizes model performance. Subject matter expert (SME) input is employed to determine if the results obtained using the response surface makes sense. The variables expected to have positive first order correlation (that is as the value for the variables increase the response is expected to increase) are dist, loss, tdead and Amem. The remaining variables (AQE, MPN, signal, etadetect) all are expected to produce negative first order effects. Figure 8 confirms the accurate signs for the first order effects in the model for all variables.

Observation of the surfaces in Figures 22 through 49 reveal unexpected behavior



Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
AQE*MPN*etadetect*Amem	207.39452	4.756961	43.64	<.0001*
dist*MPN*etadetect*Amem	-107.2736	4.756961	-22.55	<.0001*
AQE*dist*MPN*etadetect*Amem	-128.7151	6.727358	-19.13	<.0001*
loss*MPN*etadetect*Amem	-147.669	4.756961	-31.04	<.0001*
AQE*loss*MPN*etadetect*Amem	-180.9738	6.727358	-26.90	<.0001*
dist*loss*MPN*etadetect*Amem	62.335314	6.727358	9.27	<.0001*
AQE*dist*loss*MPN*etadetect*Amem	68.038982	9.513922	7.15	<.0001*
signal*etadetect*Amem	285.68036	3.363679	84.93	<.0001*
AQE*signal*etadetect*Amem	338.40357	4.756961	71.14	<.0001*
dist*signal*etadetect*Amem	-178.5675	4.756961	-37.54	<.0001*
AQE*dist*signal*etadetect*Amem	-210.1596	6.727358	-31.24	<.0001*
loss*signal*etadetect*Amem	-242.8364	4.756961	-51.05	<.0001*
AQE*loss*signal*etadetect*Amem	-302.264	6.727358	-44.93	<.0001*
dist*loss*signal*etadetect*Amem	100.67889	6.727358	14.97	<.0001*
AQE*dist*loss*signal*etadetect*Amem	108.45318	9.513922	11.40	<.0001*
MPN*signal*etadetect*Amem	74.203738	4.756961	15.60	<.0001*
AQE*MPN*signal*etadetect*Amem	85.049381	6.727358	12.64	<.0001*
dist*MPN*signal*etadetect*Amem	-47.97929	6.727358	-7.13	<.0001*
AQE*dist*MPN*signal*etadetect*Amem	-49.59278	9.513922	-5.21	<.0001*
loss*MPN*signal*etadetect*Amem	-68.50419	6.727358	-10.18	<.0001*
AQE*loss*MPN*signal*etadetect*Amem	-65.25684	9.513922	-6.86	<.0001*
dist*loss*MPN*signal*etadetect*Amem	27.331567	9.513922	2.87	0.0041*
AQE*AQE	388.26052	4.494905	86.38	<.0001*
AQE*AQE*AQE	-18.41536	8.864073	-2.08	0.0378*
dist*dist	246.95207	4.494905	54.94	<.0001*
dist*dist*dist	-49.94619	8.864073	-5.63	<.0001*
loss*loss	427.1359	4.494905	95.03	<.0001*
loss*loss*loss	-115.2284	8.864073	-13.00	<.0001*
MPN*MPN	8.8218084	4.494905	1.96	0.0497*
signal*signal	50.273334	4.494905	11.18	<.0001*
etadetect*etadetect	307.97219	4.494905	68.52	<.0001*
etadetect*etadetect*etadetect	-52.77886	8.864073	-5.95	<.0001*
Amem*Amem	-752.0073	21.10692	-35.63	<.0001*
Amem*Amem*Amem	167.41183	2.803066	59.72	<.0001*

Figure 11. Parameter values part 4

of the response due to higher order effects. In example Figure 29 shows the marginal effect of dist and loss on the response variable. The expectation here is that as both dist and loss increase so should the runtime. Figure 29 shows opposite behavior. Further, the settings expected by SMEs to produce an optimal point results in a value of 68,422 seconds indicating the existence of extreme outlier points in unexpected locations.

3.7 Data distribution

The distribution of Runtime can be seen in Figure 12. Roughly 80% settings produce Runtimes in hundreds as expected; however, the right tail of the distribution is very thick indicating the existence of extreme outliers that effect the model.

Figures 55 through 82 overlay the actual observed data points on the surface

Table 7. Optimal Settings

Factors	Setting (coded space)	Setting (natural space)
AQE	1	35 %
dist	1	32 km
loss	-1	0.1 dB/km
MPN	1	0.7
signal	1	90 %
etadetect	1	0.15
tdead	1	25 μs
Amem	0.5	8 GB

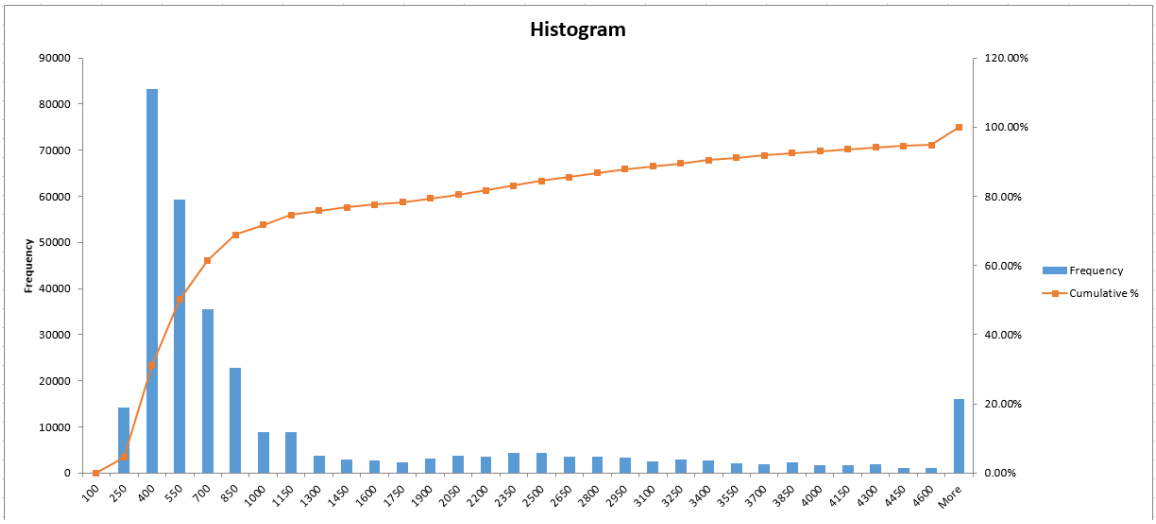


Figure 12. Response variable distribution

profiles. The large amount of data points create a practically significant amount of noise that make it difficult to observe a clear pattern in many figures.

3.8 Residual Analysis to understand data distribution

The thick right tail largely consist of outlier, or influential data points. To identify these points in the data we take the absolute value of the studentized residuals for each observation. Typically any value larger than three is considered to be an outlier. As seen in Table 8 using a residual value threshold results in 5655 outliers. Using this residual value threshold there are data points in both the left and the right tail. To

Table 8. Outlier Analysis

Residual Value Threshold	Number of Outliers	Noticable pattern
3	5655	no data points remain where AQE is equal to 12.5%
4	2957	values for etadetect are non-negative in coded space (≥ 10)
5	1702	values for AQE are non-negative in coded space ($\geq 20\%$)
6	967	AQE is positive in coded space ($\geq 27.5\%$), no data points remain where Alice's memory equals 4 GB
7	639	values for etadetect are positive in coded space (≥ 12.5)
8	433	no data points remain where Alice's memory equals 2 GB
9	337	no data points remain where loss is equal to 0.25 dB/km Alice's memory equals 16 GB for all remaining points
10	247	values for loss are all negative in coded space (≤ 0.15)
11	179	AQE equals 35% for all remaining points no data points remain where signal equals 60%
12	133	no new pattern noticable
13	100	values for dist are non-positive in coded space ($\leq 23km$) values for signal are non-negative in coded space ($\geq 75\%$)
14	75	etadetect equals 0.15 for all remaining points
15	45	loss equals 0.1 dB/km

isolate only those points in the right tail we raise the residual value threshold to four, resulting in 2957 data points. To help characterize and identify the similarities in the outliers we systematically raise the residual value threshold and observe noticeable patterns. The complete results are seen in Table 8.

IV. Conclussions

4.1 Summary and Conclusion

The study verified the correct coding of equations 1 through 26 into the QKD model. The analysis began with a screening experiment to determine the significant memory allocation factors for both Alice and Bob. The significant parameters from the screening experiment combined with the system level parameters seen in Table 6 were used to create a mathematical 138 parameter metamodel used to plot response surfaces used for SME validation. The response surfaces revealed unexpected behavior and the identification of extreme points which lead to an analysis of the distribution of the data and outlier analysis. Overlaying the actual data points on the surface profiles seen in Figures 55 through 82 indicate an immensely complicated surface, difficult to model.

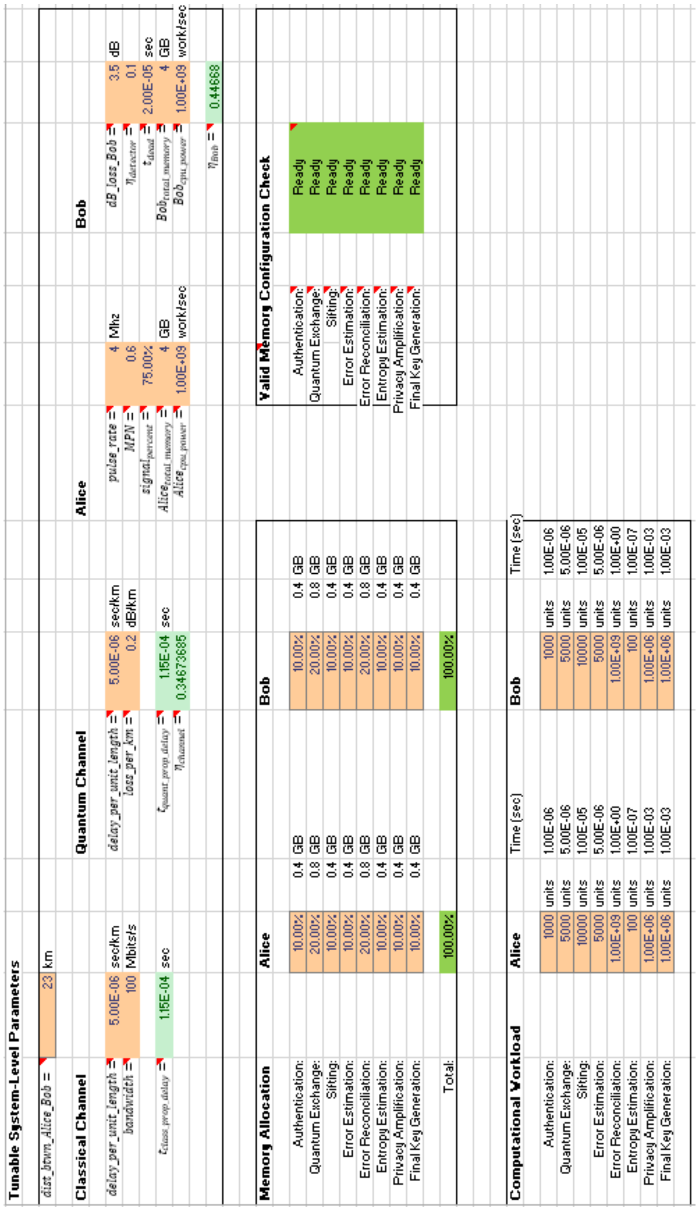
The model was validated for data points surrounding initial settings provided by Cernera [2]; however, extreme points were observed specifically when Alice's memory was large (16 GB) and the memory allocated to Quantum Exchange for Alice was large (35%). The settings indicate an edge of the intended design range for the QKD model.

4.2 Future Analysis

Three approaches are recommended for future research. First: After initial screening of memory allocation to each of the eight phases for both Alice and Bob the parameters were removed from the study. A look into the relevance of the parameters at a lower percent of memory allocation will likely show significant effect on the Run–time. Second: Transforming the response variable would allow for an examination of other surfaces that may be more fitting for the model. Third: Phase specific inputs

went unexplored in this study. A study involving these inputs and the outputs for each phase may lead to identification of which phase is causing bottlenecks creating large Runtimes.

Appendix A. Excel Model



Tunable System-Level Parameters

dist_btwn_Alice_Bob =	23	km

Classical Channel			Quantum Channel			Alice			Bob		
delay_per_unit_length =	5.00E-06	sec/km	delay_per_unit_length =	5.00E-06	sec/km	pulse_rate =	4	Mhz	dB_loss_Bob =	3.5	dB
bandwidth =	100	Mbits/s	loss_per_km =	0.2	dB/km	MPN =	0.6		$\eta_{detector}$ =	0.1	
						$signal_{percent}$ =	75.00%		t_{dead} =	2.00E-05	sec
$t_{class\_prop\_delay}$ =	1.15E-04	sec	$t_{quant\_prop\_delay}$ =	1.15E-04	sec	$Alice_{total\_memory}$ =	4	GB	Bob_{total_memory} =	4	GB
			$\eta_{channel}$ =	0.34673685		$Alice_{cpu\_power}$ =	1.00E+09	work/sec	Bob_{cpu_power} =	1.00E+09	work/sec
									η_{Bob} =	0.44668	

Memory Allocation	Alice			Bob		
Authentication:	10.00%	0.4	GB	10.00%	0.4	GB
Quantum Exchange:	20.00%	0.8	GB	20.00%	0.8	GB
Sifting:	10.00%	0.4	GB	10.00%	0.4	GB
Error Estimation:	10.00%	0.4	GB	10.00%	0.4	GB
Error Reconciliation:	20.00%	0.8	GB	20.00%	0.8	GB
Entropy Estimation:	10.00%	0.4	GB	10.00%	0.4	GB
Privacy Amplification:	10.00%	0.4	GB	10.00%	0.4	GB
Final Key Generation:	10.00%	0.4	GB	10.00%	0.4	GB
Total:	100.00%			100.00%		

Valid Memory Configuration Check	
Authentication:	Ready
Quantum Exchange:	Ready
Sifting:	Ready
Error Estimation:	Ready
Error Reconciliation:	Ready
Entropy Estimation:	Ready
Privacy Amplification:	Ready
Final Key Generation:	Ready

Computational Workload	Alice		Time (sec)	Bob		Time (sec)
Authentication:	1000	units	1.00E-06	1000	units	1.00E-06
Quantum Exchange:	5000	units	5.00E-06	5000	units	5.00E-06
Sifting:	10000	units	1.00E-05	10000	units	1.00E-05
Error Estimation:	5000	units	5.00E-06	5000	units	5.00E-06
Error Reconciliation:	1.00E+09	units	1.00E+00	1.00E+09	units	1.00E+00
Entropy Estimation:	100	units	1.00E-07	100	units	1.00E-07
Privacy Amplification:	1.00E+06	units	1.00E-03	1.00E+06	units	1.00E-03
Final Key Generation:	1.00E+06	units	1.00E-03	1.00E+06	units	1.00E-03

Figure 13. Implementation of model using Excel

Figure 14. Implementation of model: Authentication Phase

Figure 15. Implementation of model: Quantum Exchange

Figure 16. Implementation of model: Sifting

Figure 17. Implementation of model: Error Estimation

Figure 18. Implementation of model: Error Reconciliation

Entropy Estimation

Inputs

A_ErrRec_{buffer} =	42206.59212	bytes
B_ErrRec_{buffer} =	42206.59212	bytes
$pct\_entropy\_loss\_QBER$ =	12.00%	
$pct\_entropy\_loss\_public$ =	12.00%	
$pct\_entropy\_loss\_multi\_photon$ =	12.00%	
$pct\_entropy\_loss\_safety$ =	4.00%	
$AB\_avg\_msg\_size$ =	0	bytes
$BA\_avg\_msg\_size$ =	0	bytes
AB_num_trans =	0	
BA_num_trans =	0	

Intermediate Calculations

$$time\_btwn\_sent\_msg = \frac{msg\_size(bits)}{bandwidth\left(\frac{bits}{s}\right)}$$

$time\_btwn\_sent\_msg_{AB}$ =	0.00E+00	sec
$time\_btwn\_sent\_msg_{BA}$ =	0.00E+00	sec
$total\_entropy\_loss\_pct$ =	40.00%	

Figure 19. Implementation of model: Entropy Estimation

Figure 20. Implementation of model: Privacy Amplification

Figure 21. Implementation of model: Final Key Generation

Appendix B. Surface profiles

Graphics for all combinations of two variables in a 3D plot can be seen below:

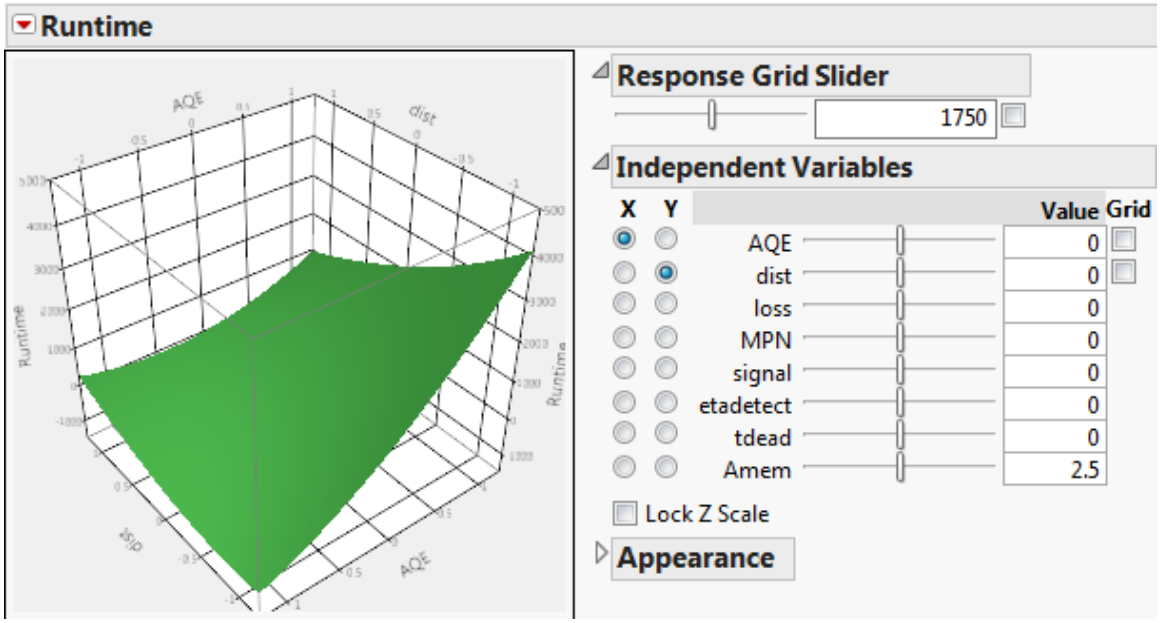


Figure 22. Surface profile of AQE vs. dist

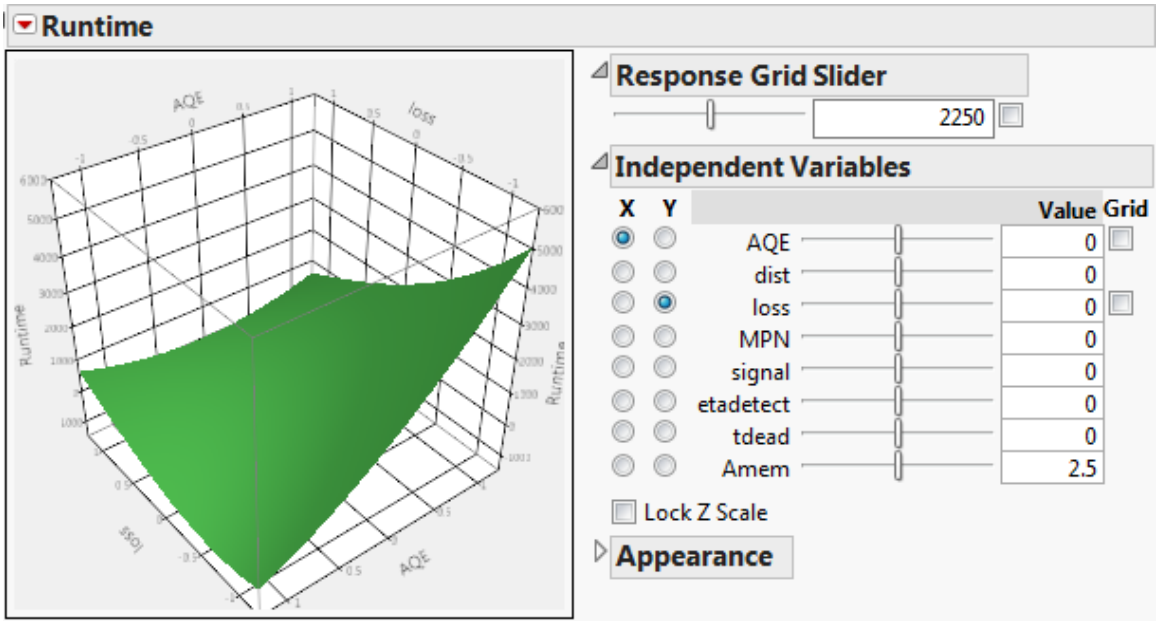


Figure 23. Surface profile of AQE vs. loss

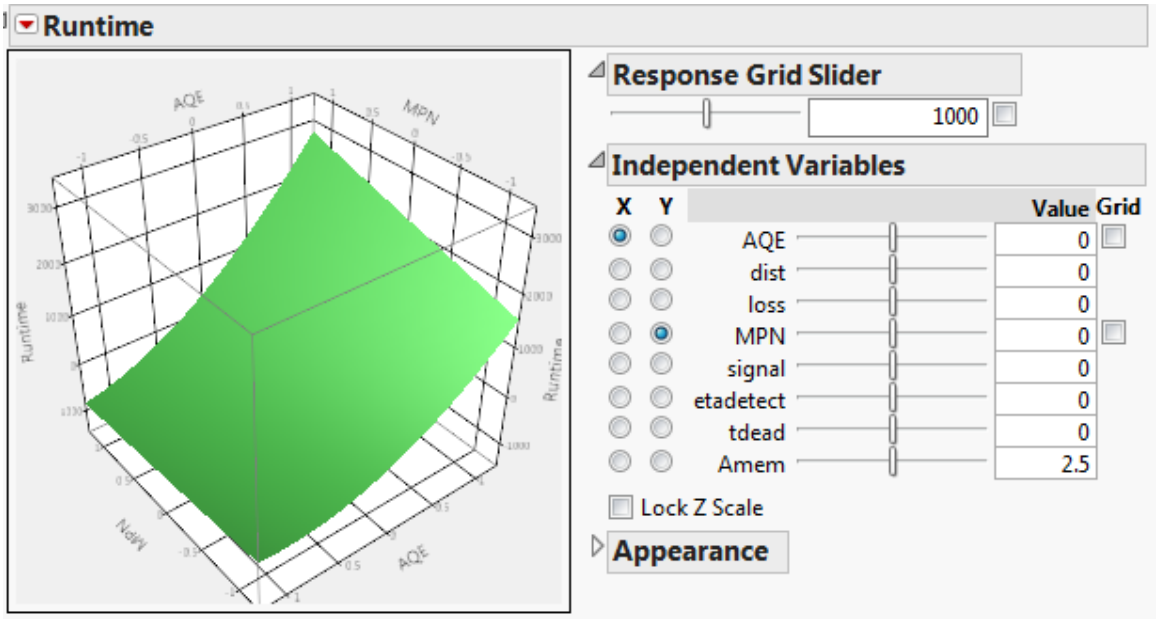


Figure 24. Surface profile of AQE vs. MPN

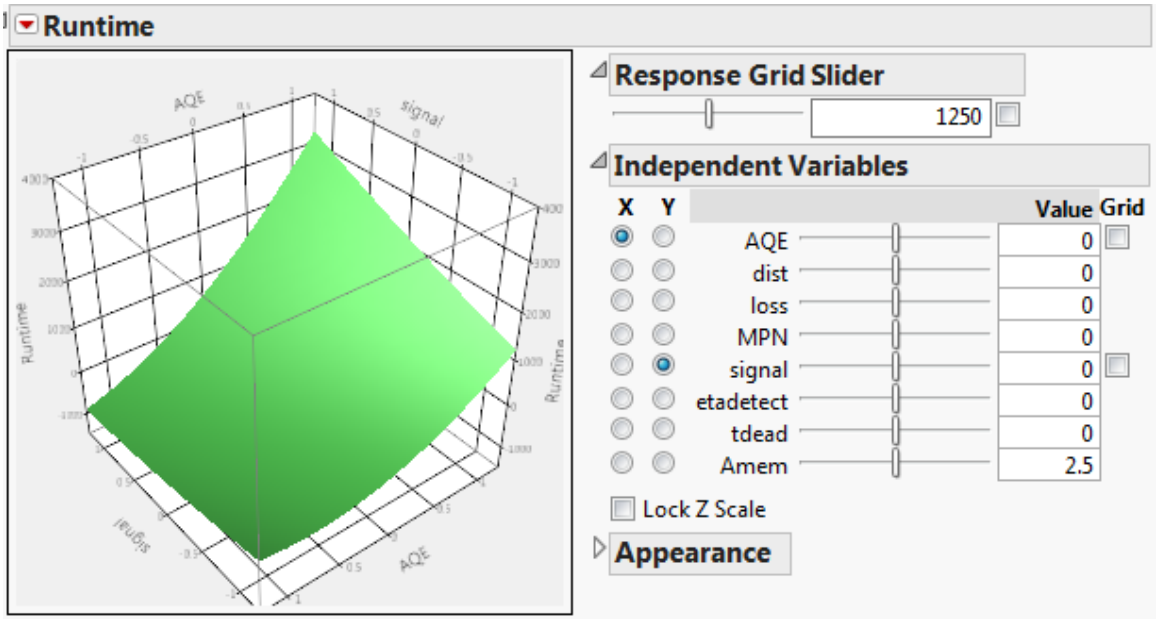


Figure 25. Surface profile of AQE vs. signal

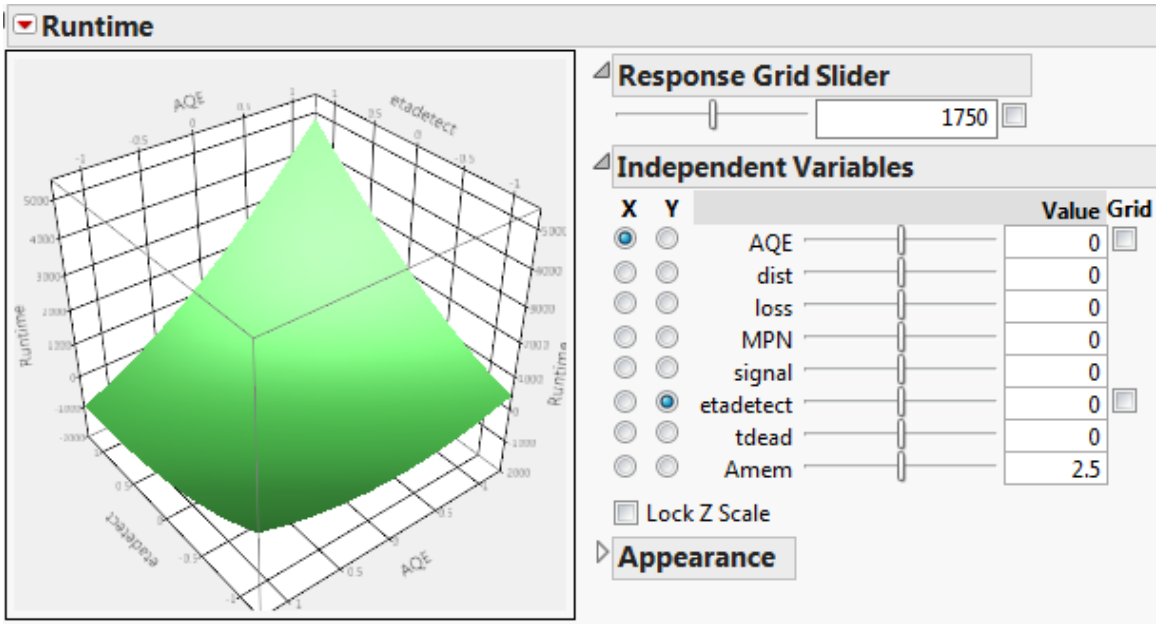


Figure 26. Surface profile of AQE vs. etadetect

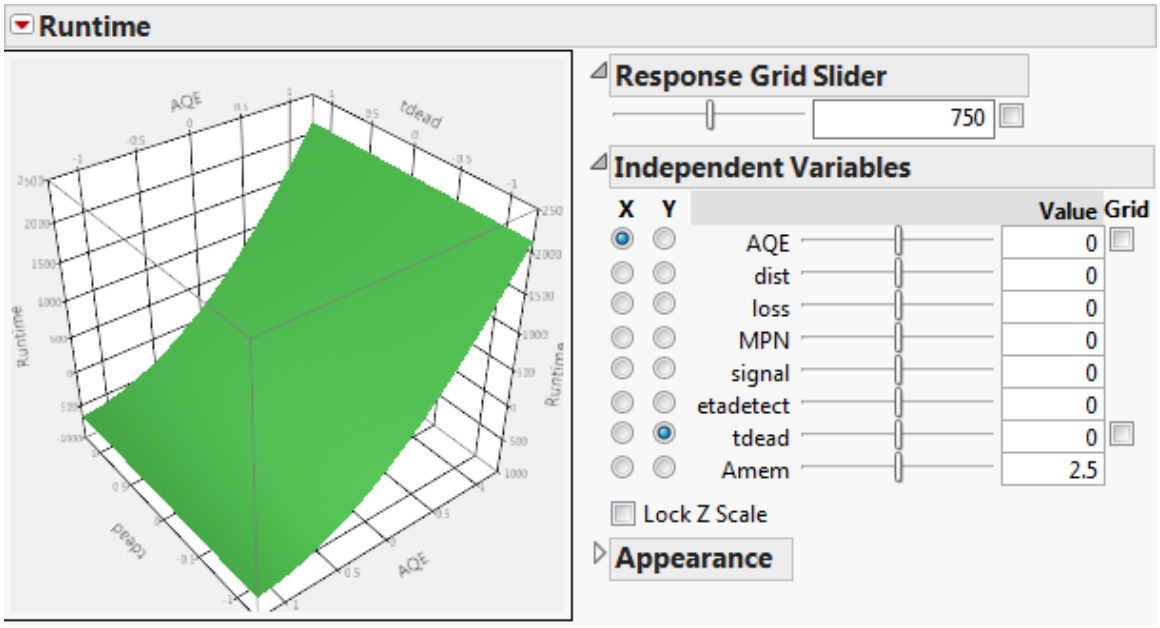


Figure 27. Surface profile of AQE vs. tdead

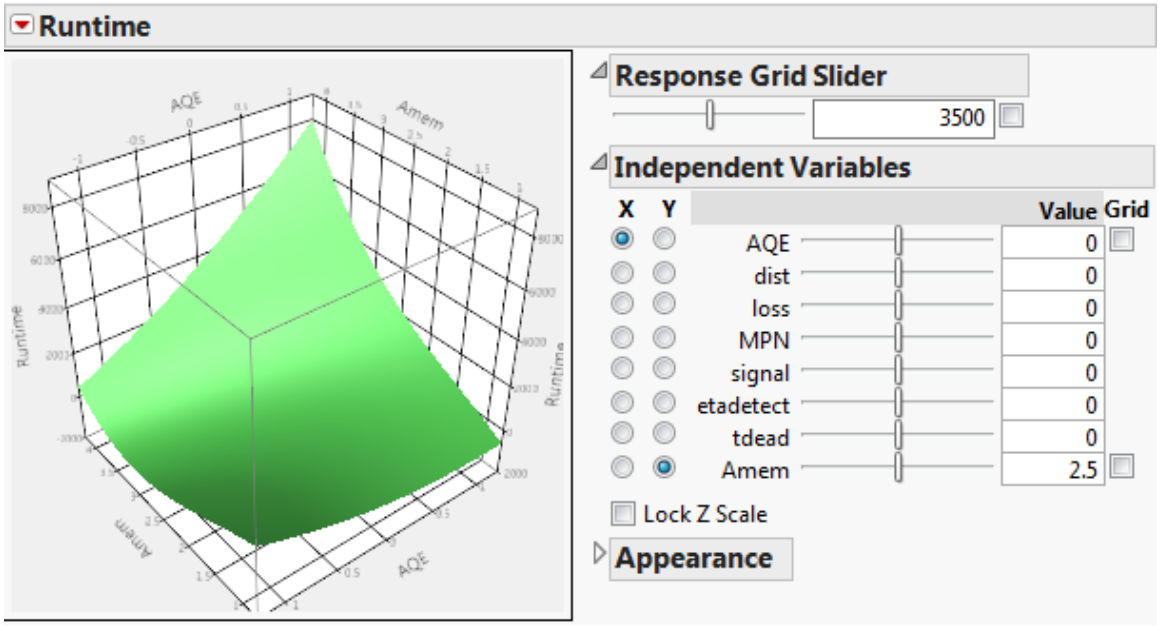


Figure 28. Surface profile of AQE vs. Amem

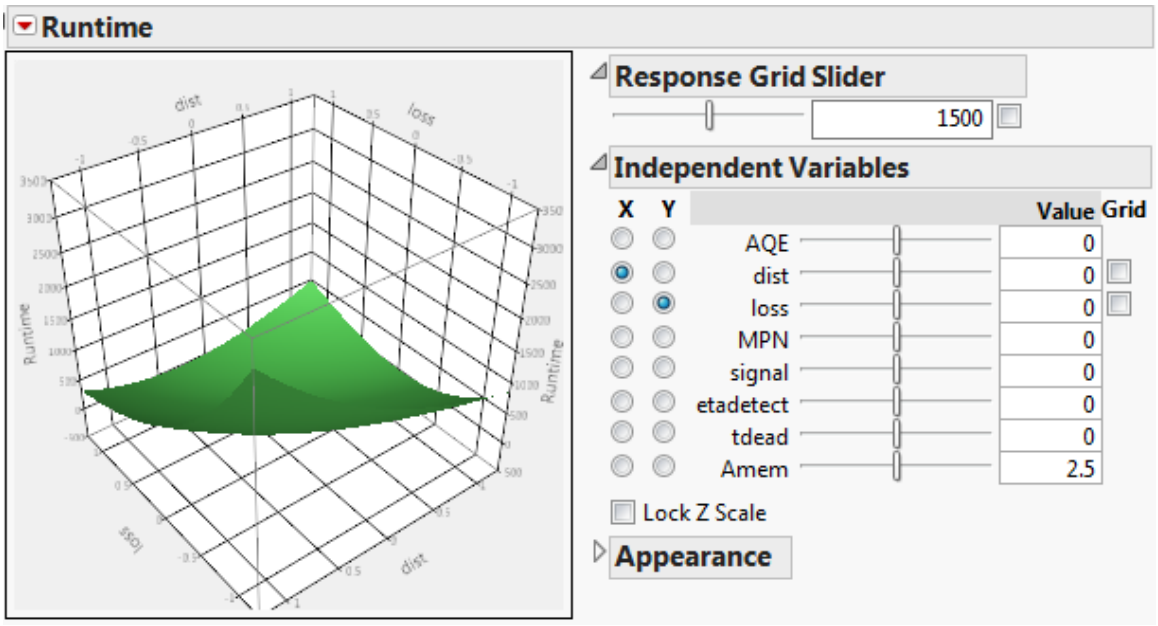


Figure 29. Surface profile of dist vs. loss

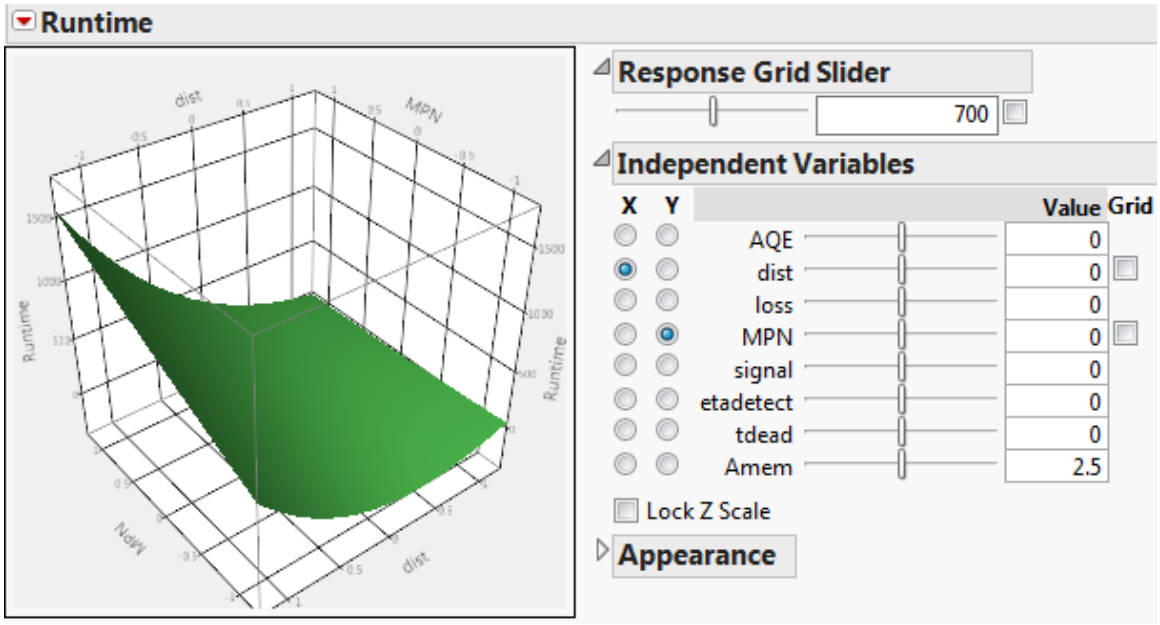


Figure 30. Surface profile of dist vs. MPN

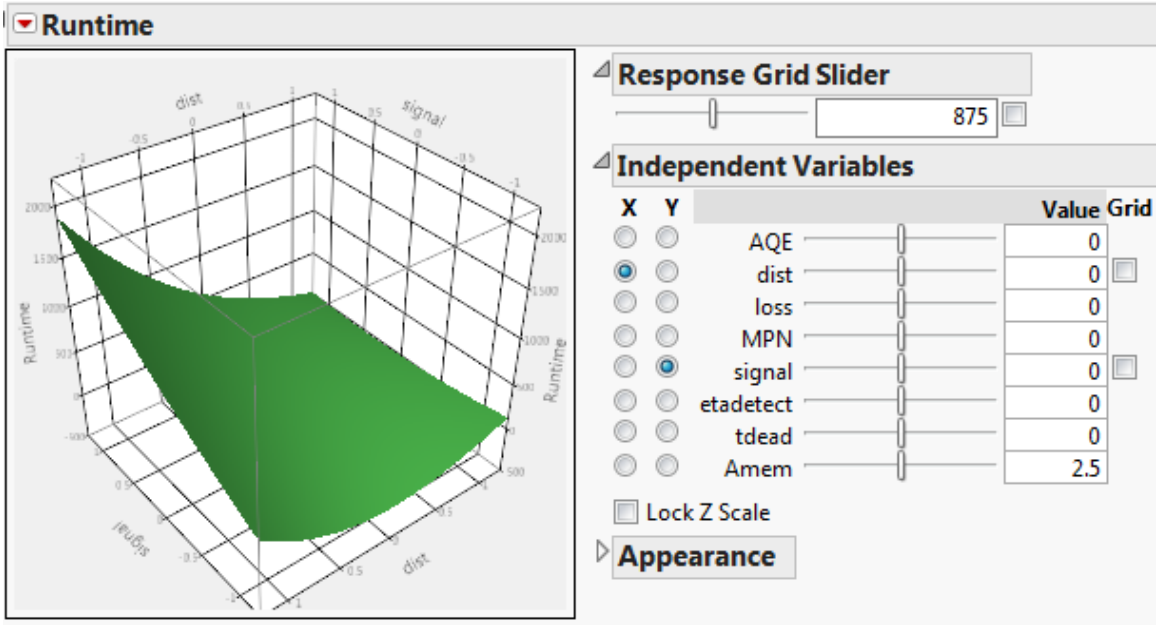


Figure 31. Surface profile of dist vs. signal

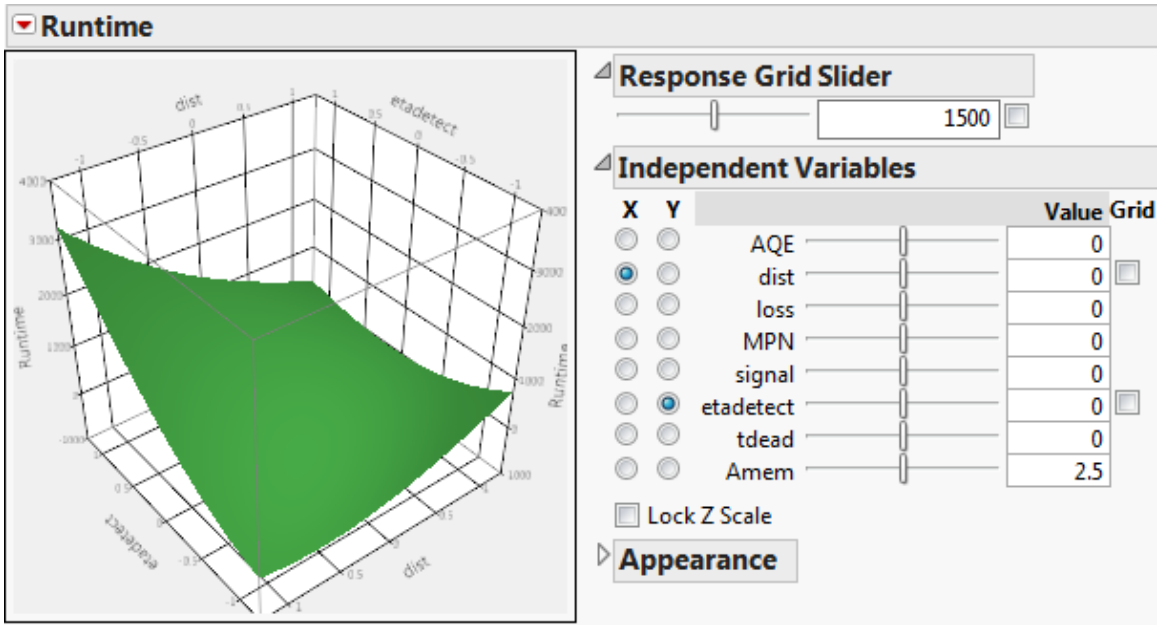


Figure 32. Surface profile of dist vs. etadetect

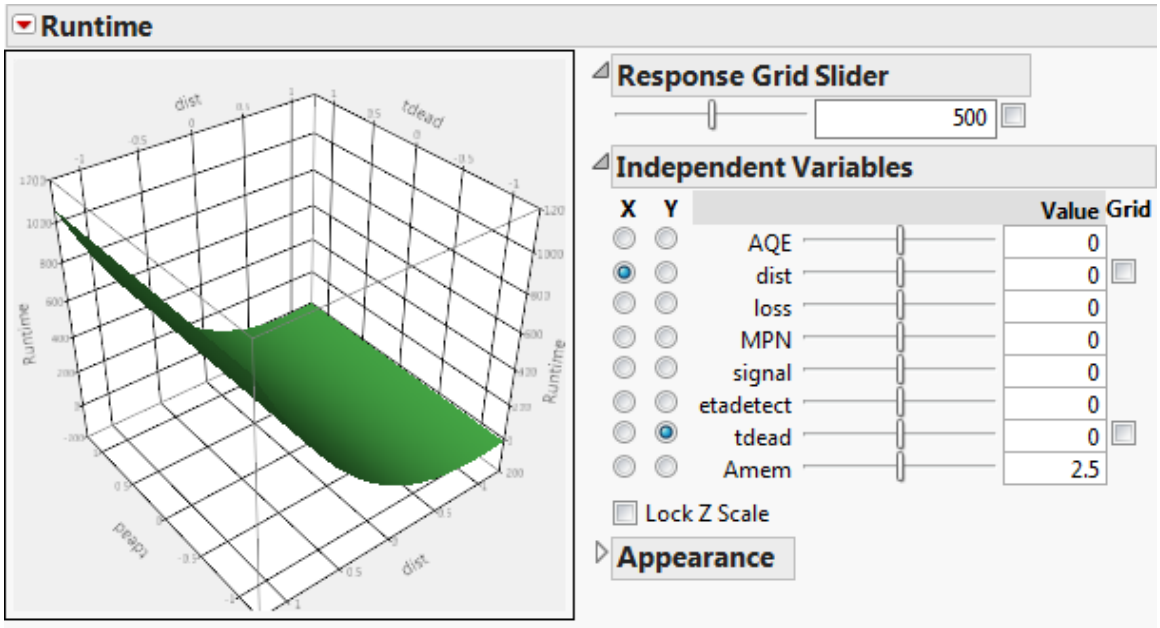


Figure 33. Surface profile of dist vs. tdead

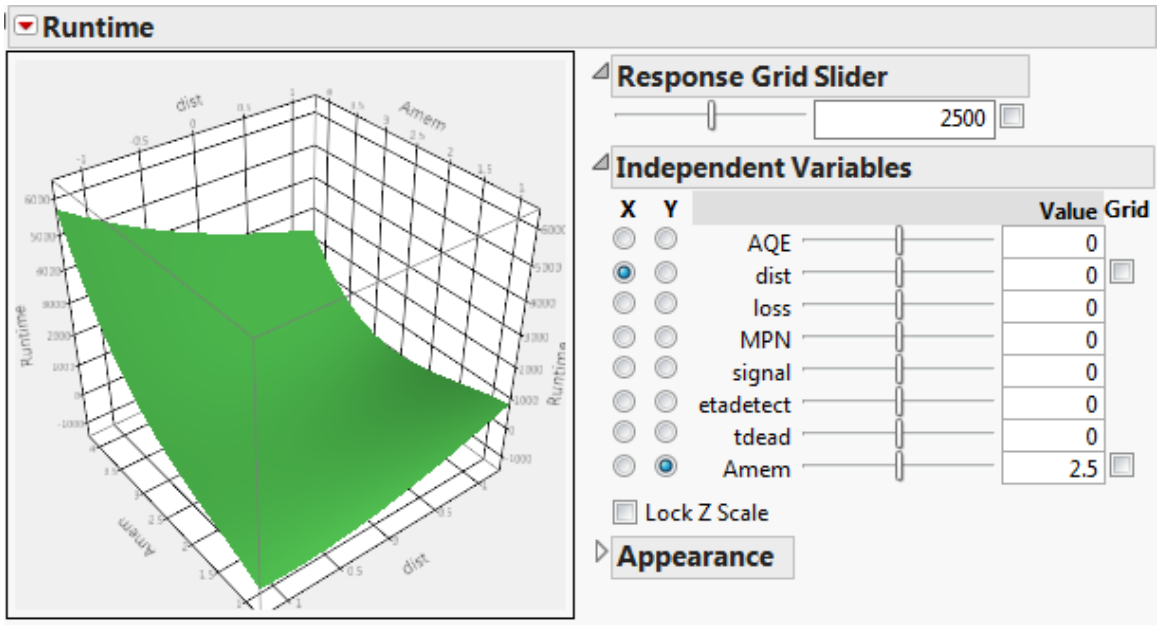


Figure 34. Surface profile of dist vs. Amem

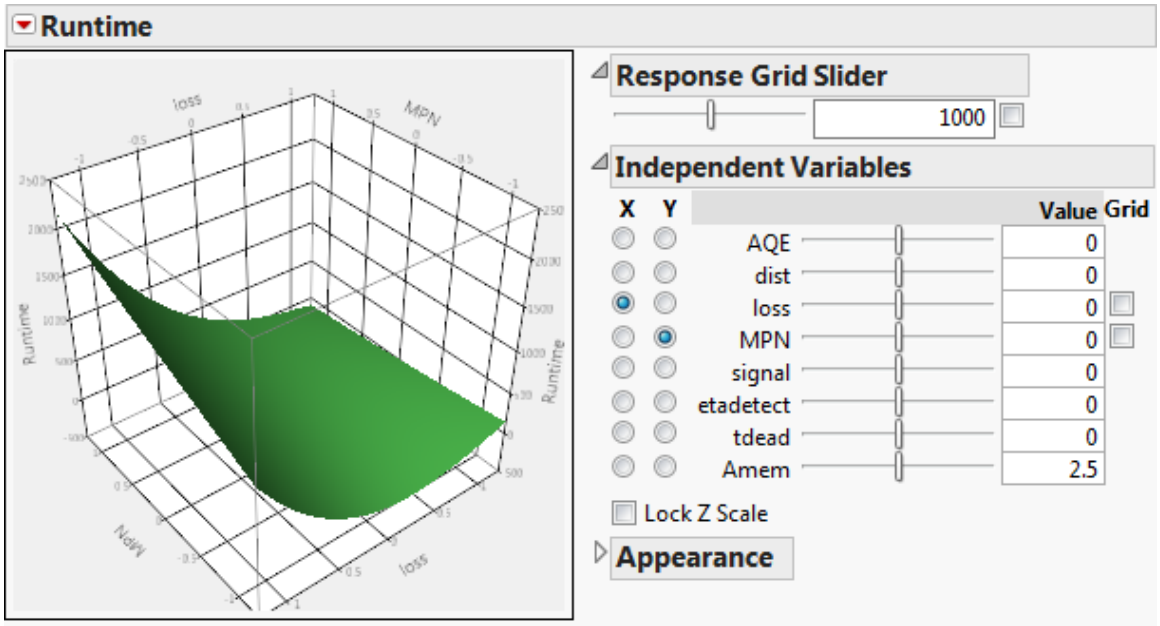


Figure 35. Surface profile loss vs. MPN

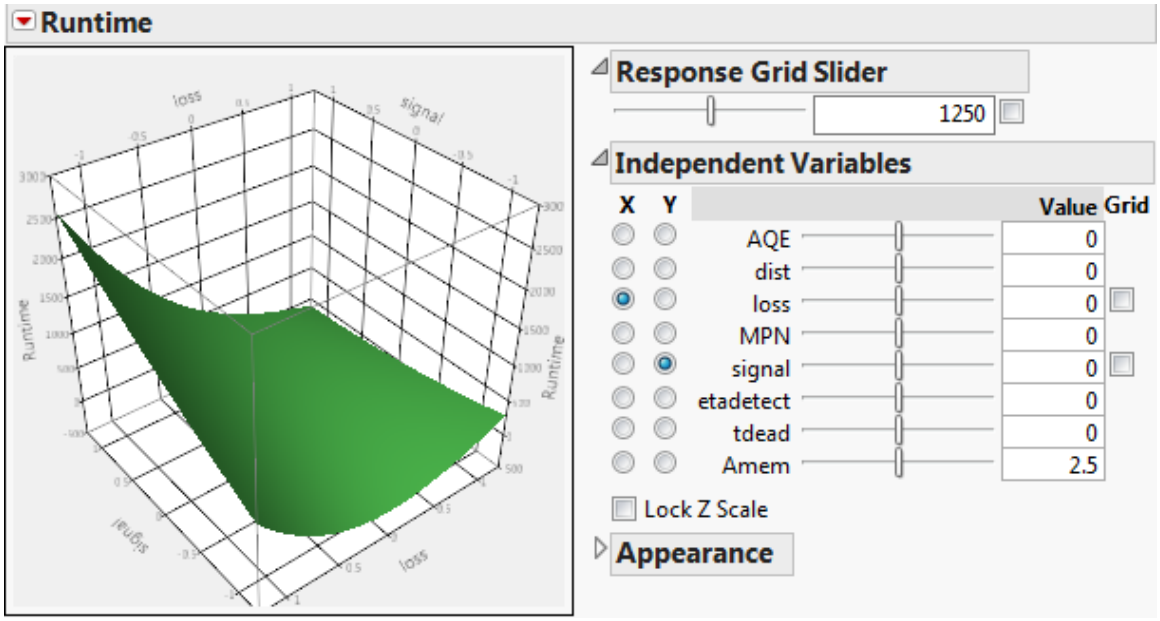


Figure 36. Surface profile of loss vs. signal

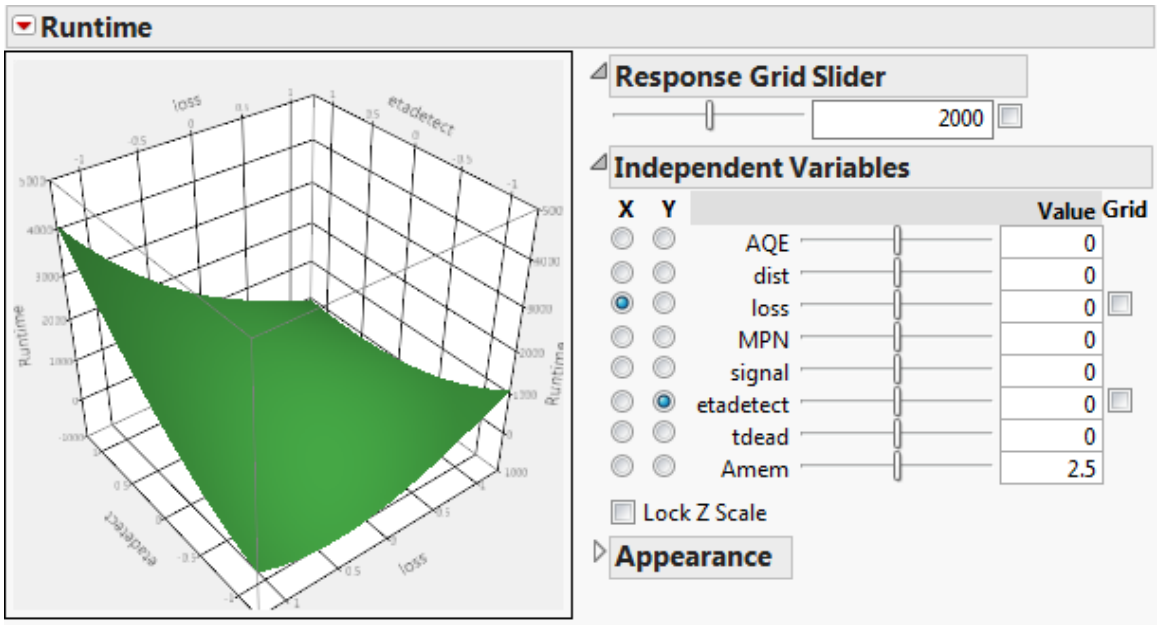


Figure 37. Surface profile of loss vs. etadetect

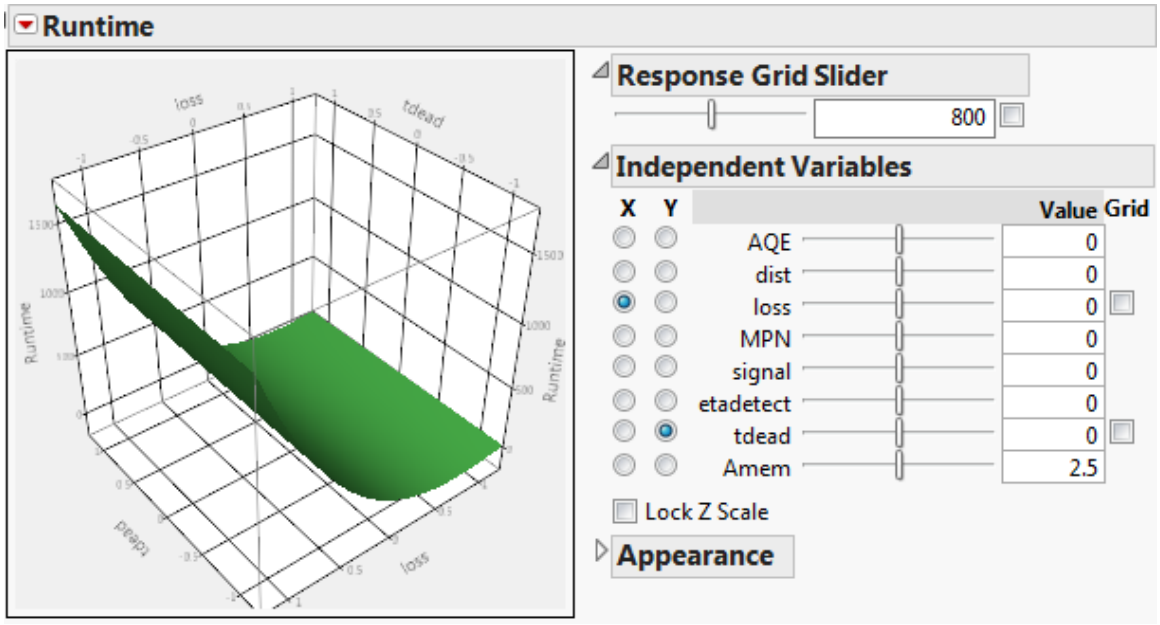


Figure 38. Surface profile of loss vs. tdead

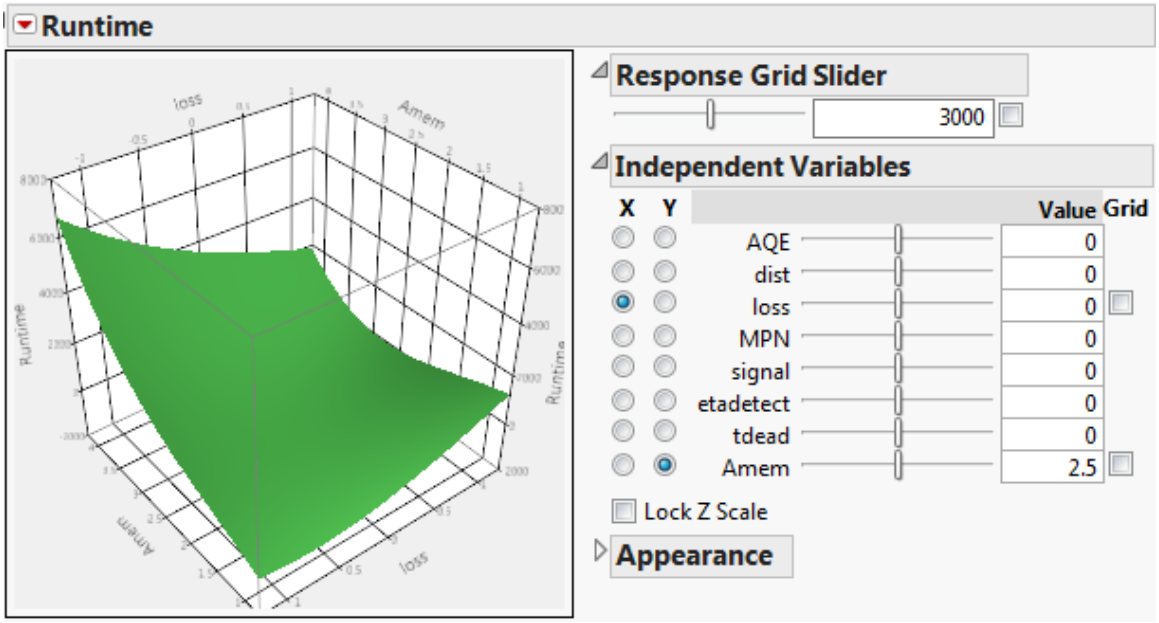


Figure 39. Surface profile of loss vs. Amem

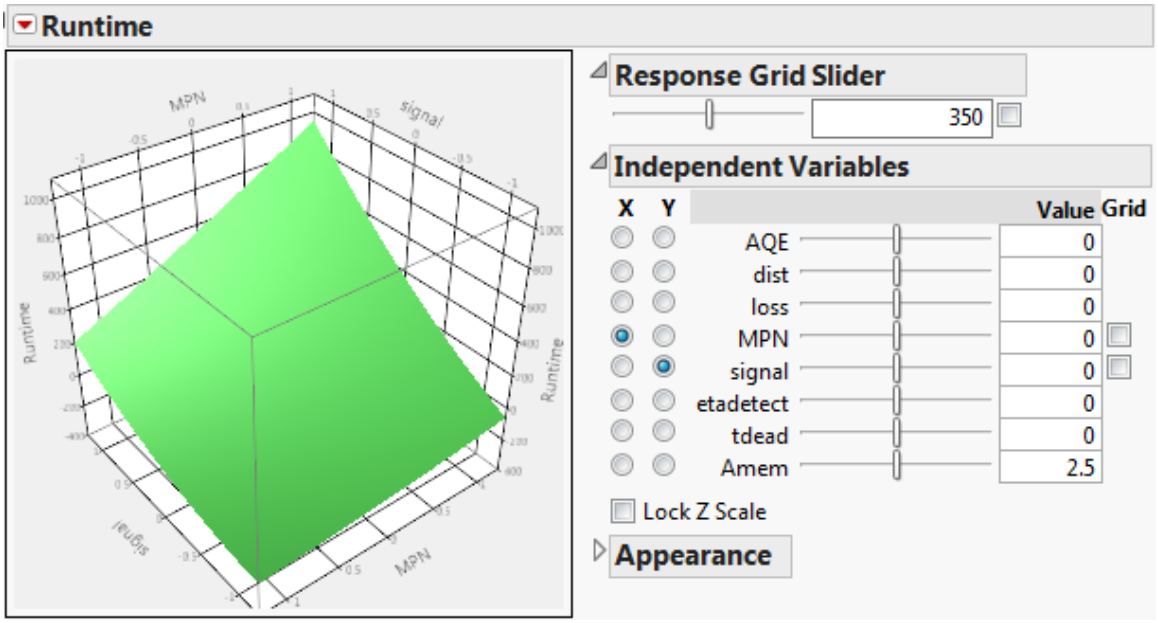


Figure 40. Surface profile of MPN vs. signal

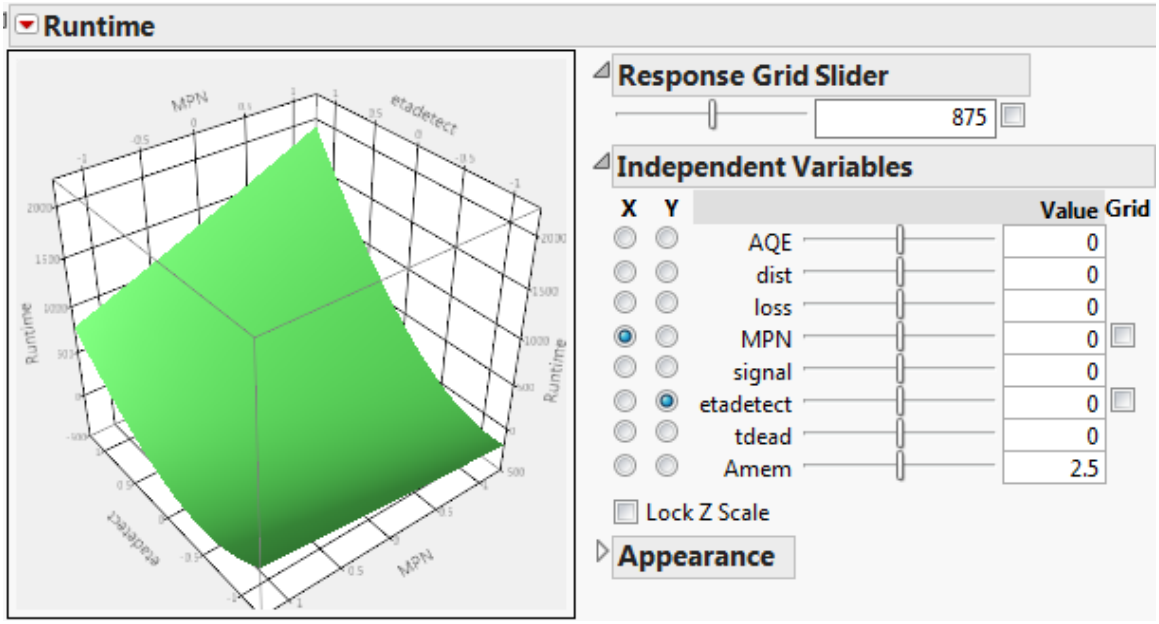


Figure 41. Surface profile of MPN vs. etadetect

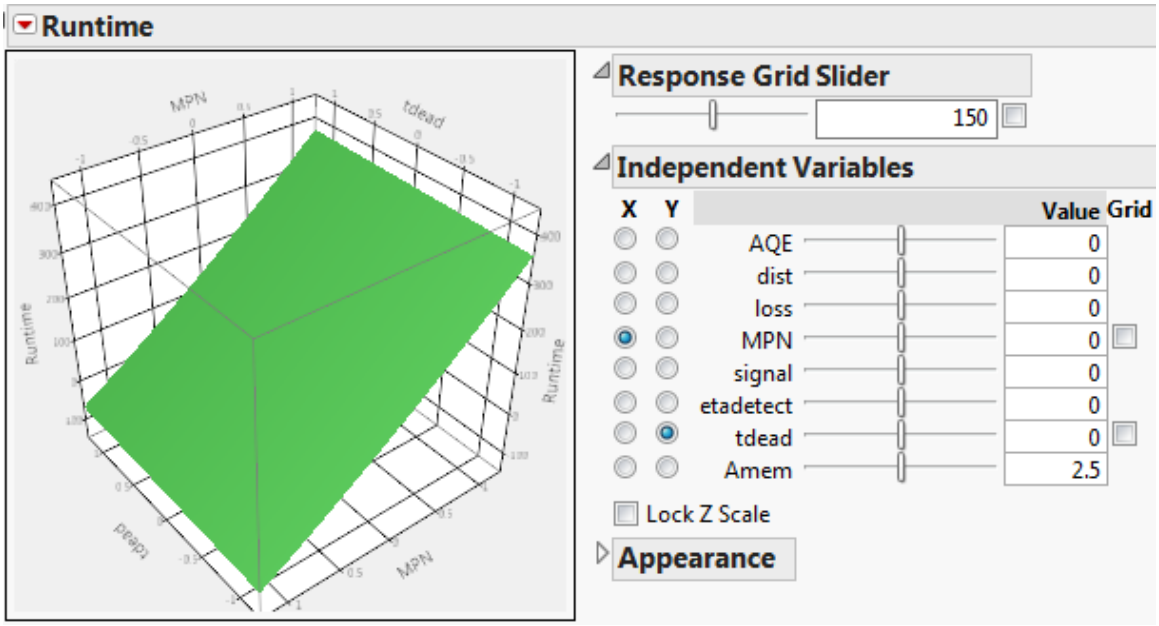


Figure 42. Surface profile of MPN vs. tdead

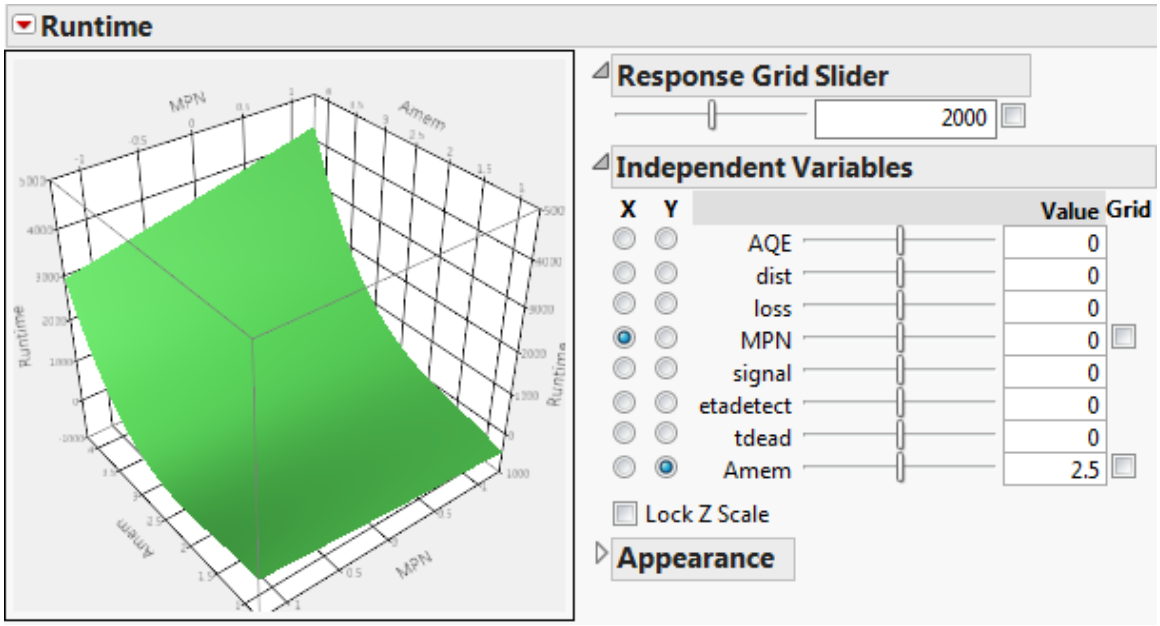


Figure 43. Surface profile of MPN vs. Amem

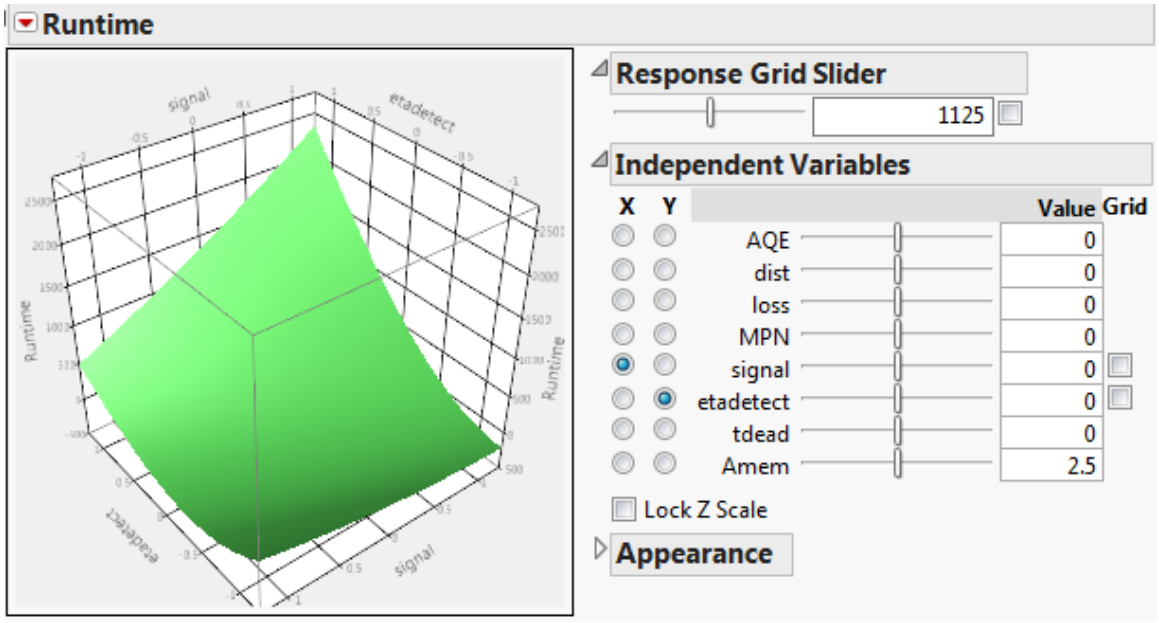


Figure 44. Surface profile of signal vs. etadetect

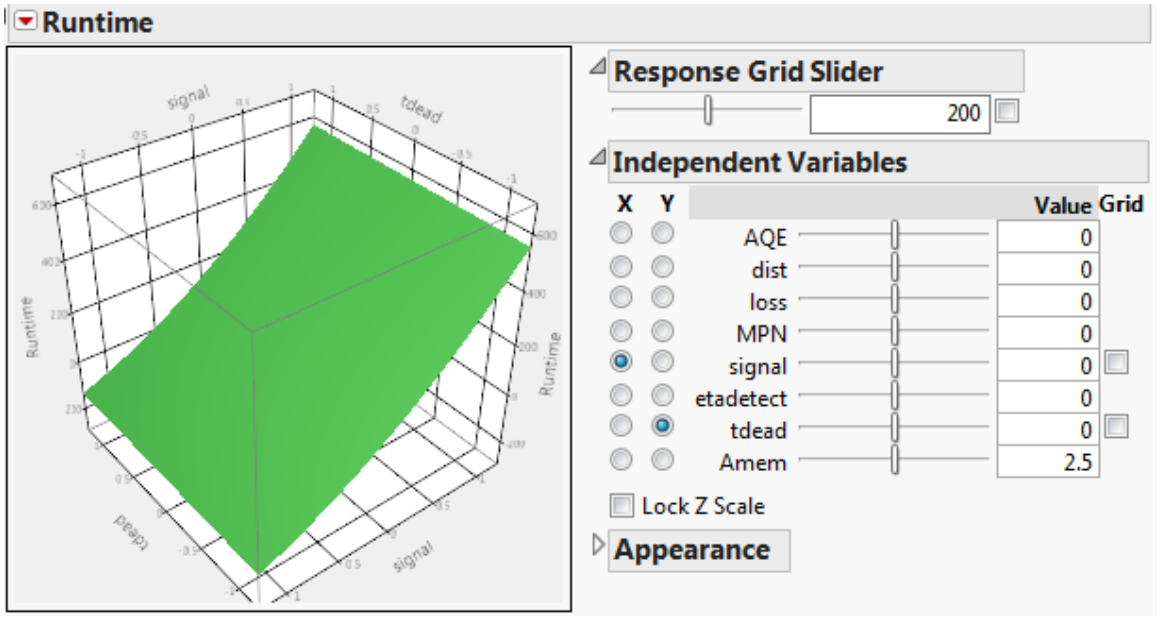


Figure 45. Surface profile of signal vs. tdead

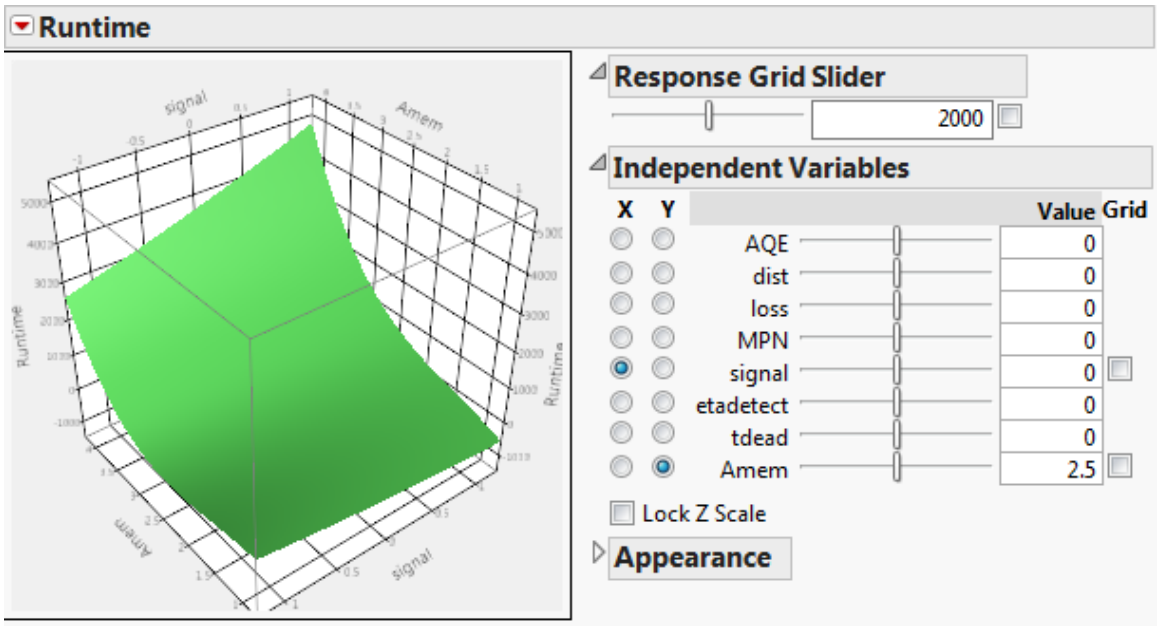


Figure 46. Surface profile of signal vs. Amem

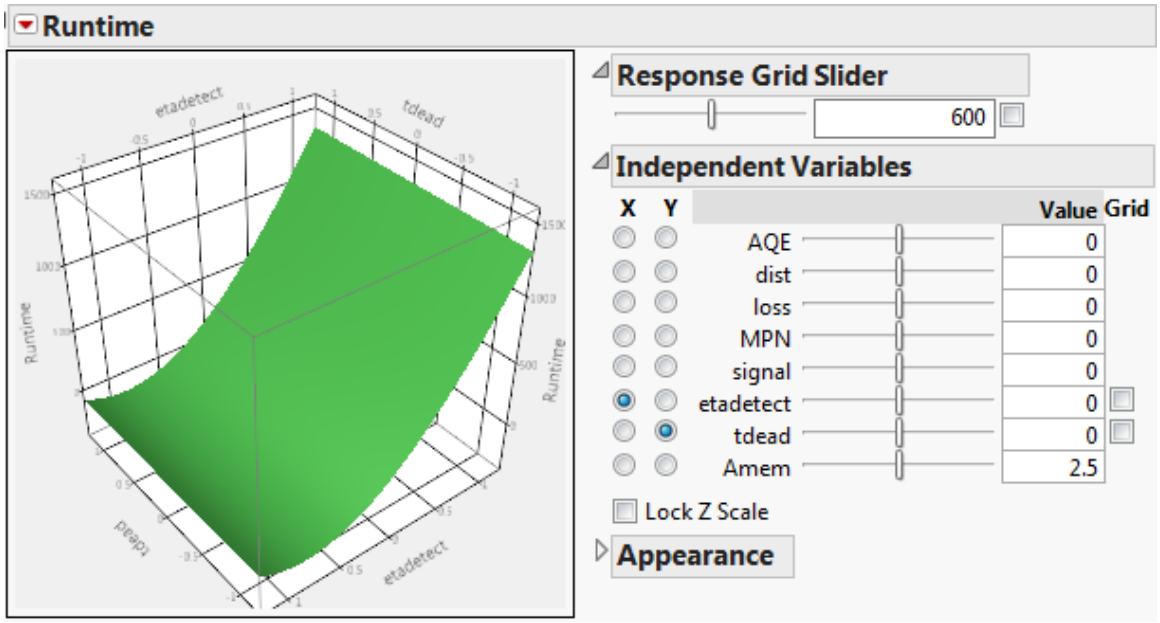


Figure 47. Surface profile of etadetect vs. tdead

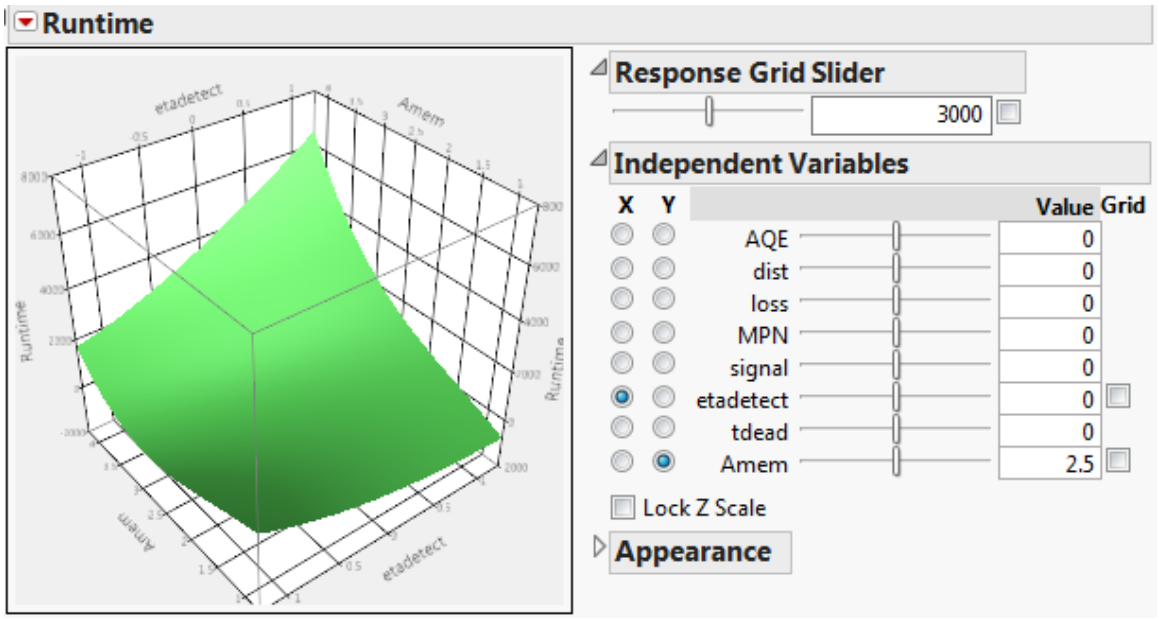


Figure 48. Surface profile of etadetect vs. Amem

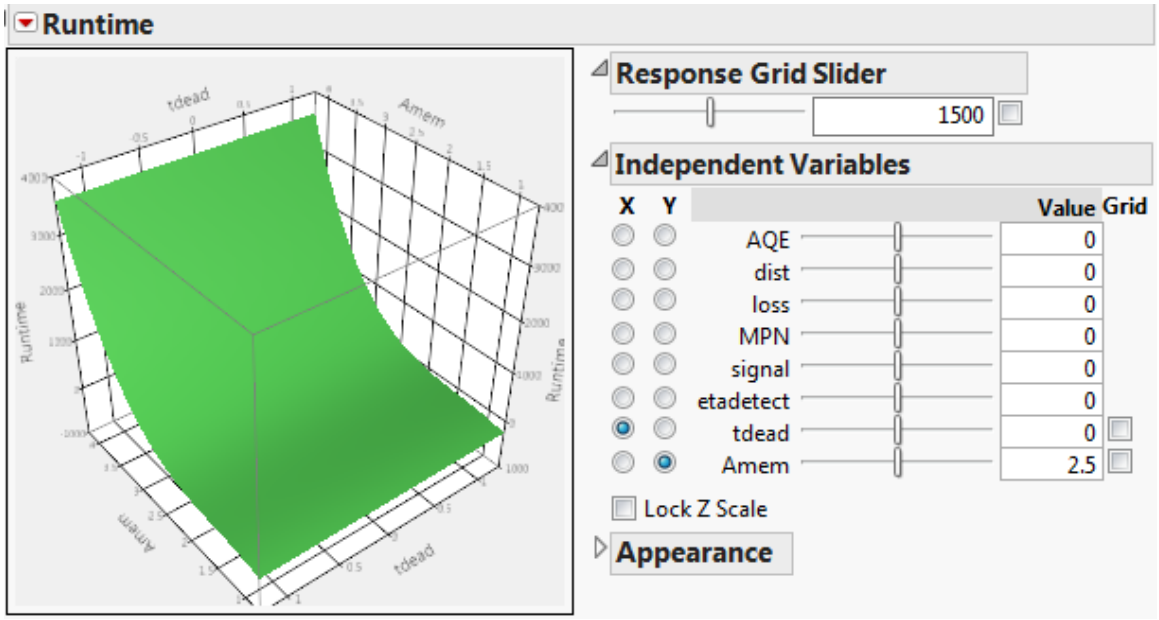


Figure 49. Surface profile of tdead vs. Amem

Appendix C. Residual plots

Residual analysis is conducted using the studentized residuals and can be seen in Figure 50 through Figure 54.

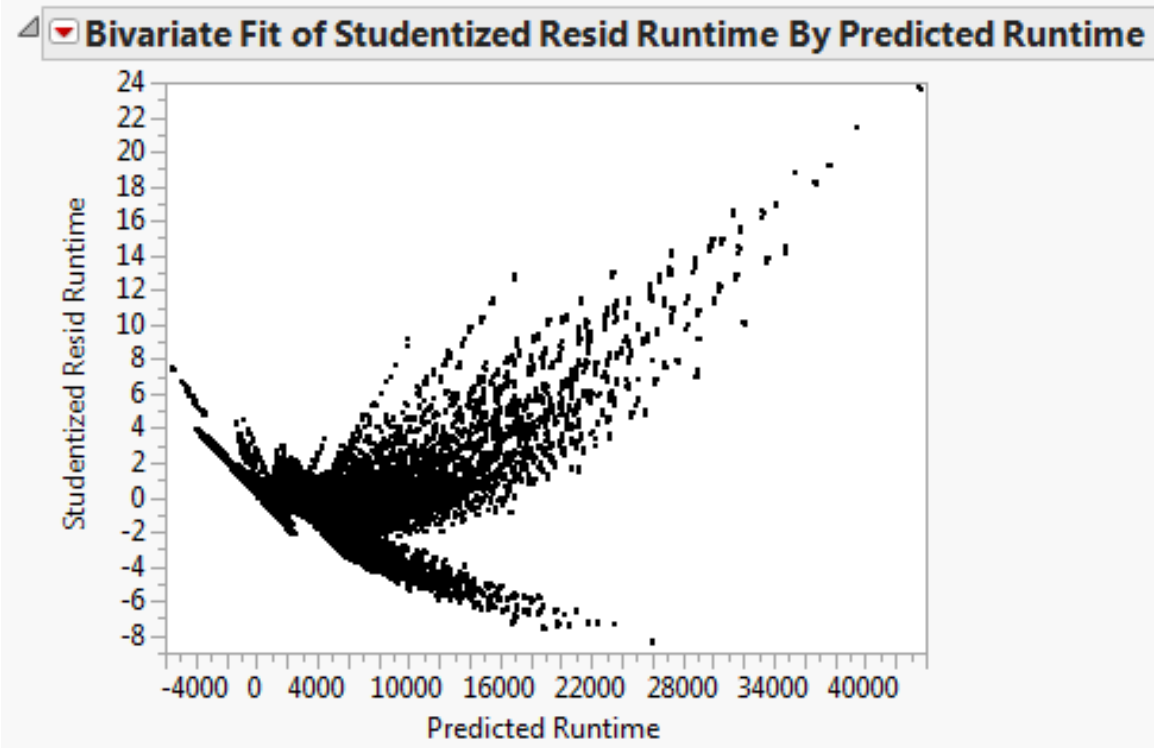


Figure 50. Residuals vs. Predicted

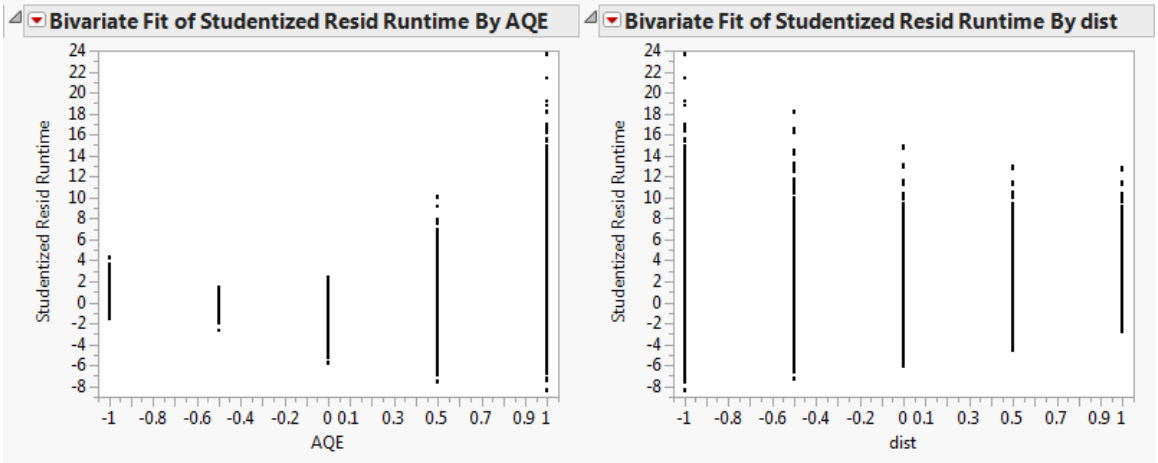


Figure 51. Residual distribution for AQE and dist

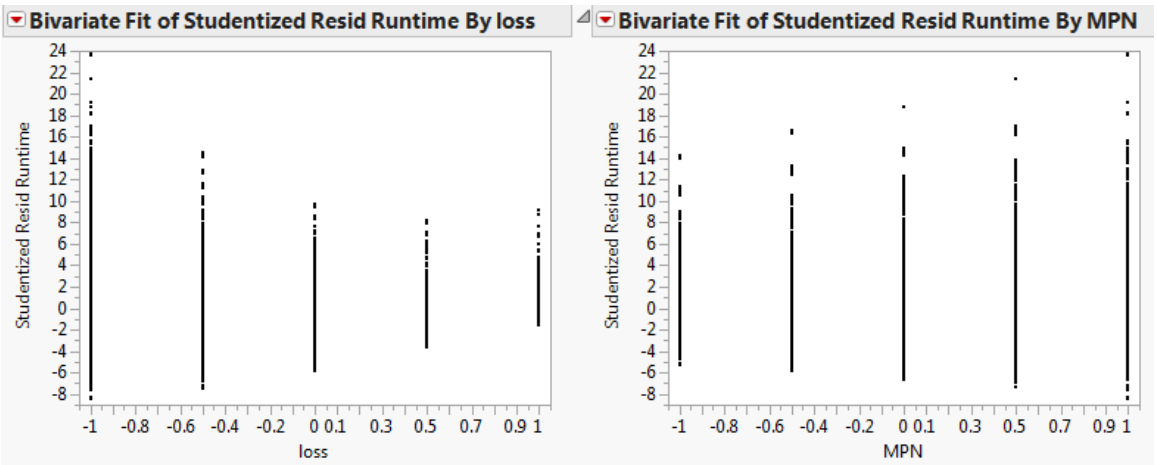


Figure 52. Residual distribution for loss and MPN

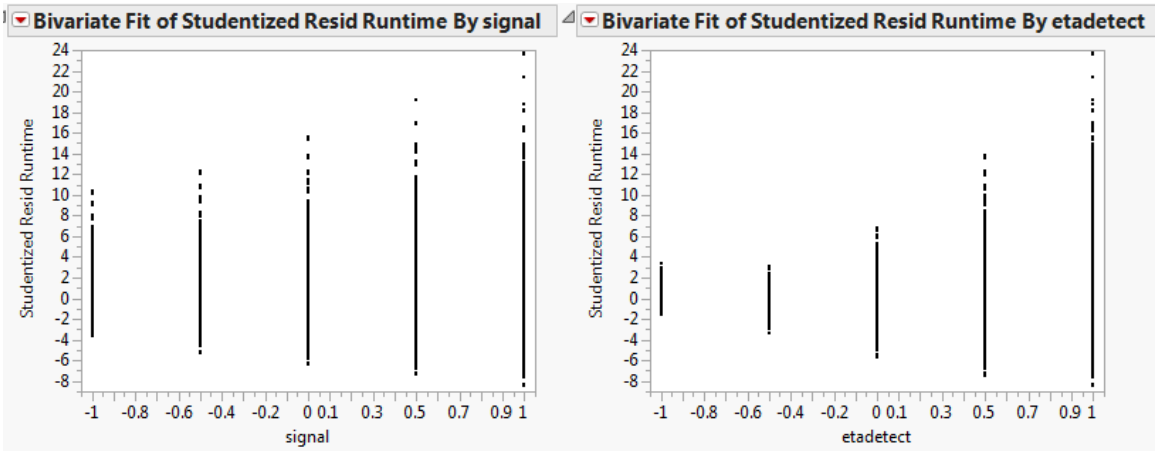


Figure 53. Residual distribution for signal and etadetect

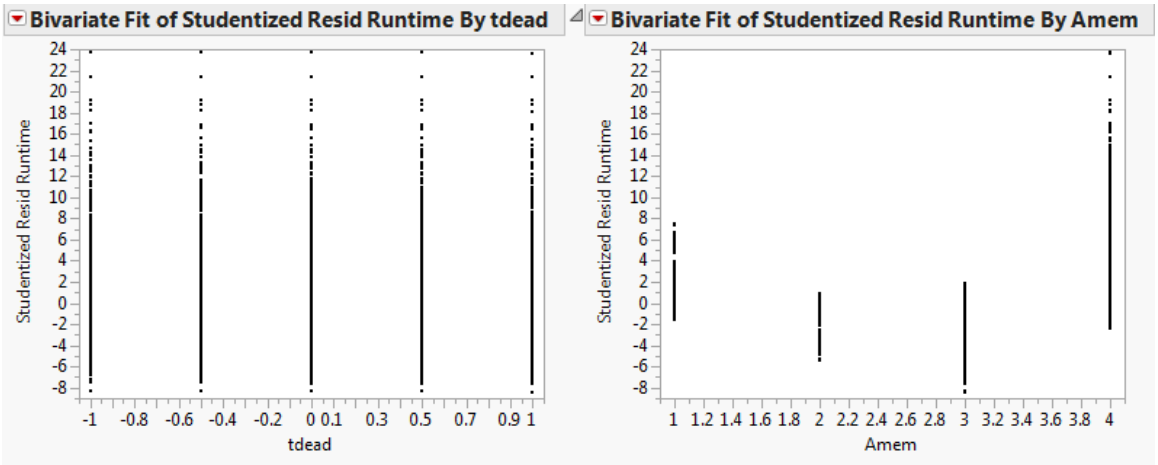


Figure 54. Residual distribution for tdead and Amem

Appendix D. Data overlay on Surfaces

Surface profiles with actual observed data points overlayed on surface:

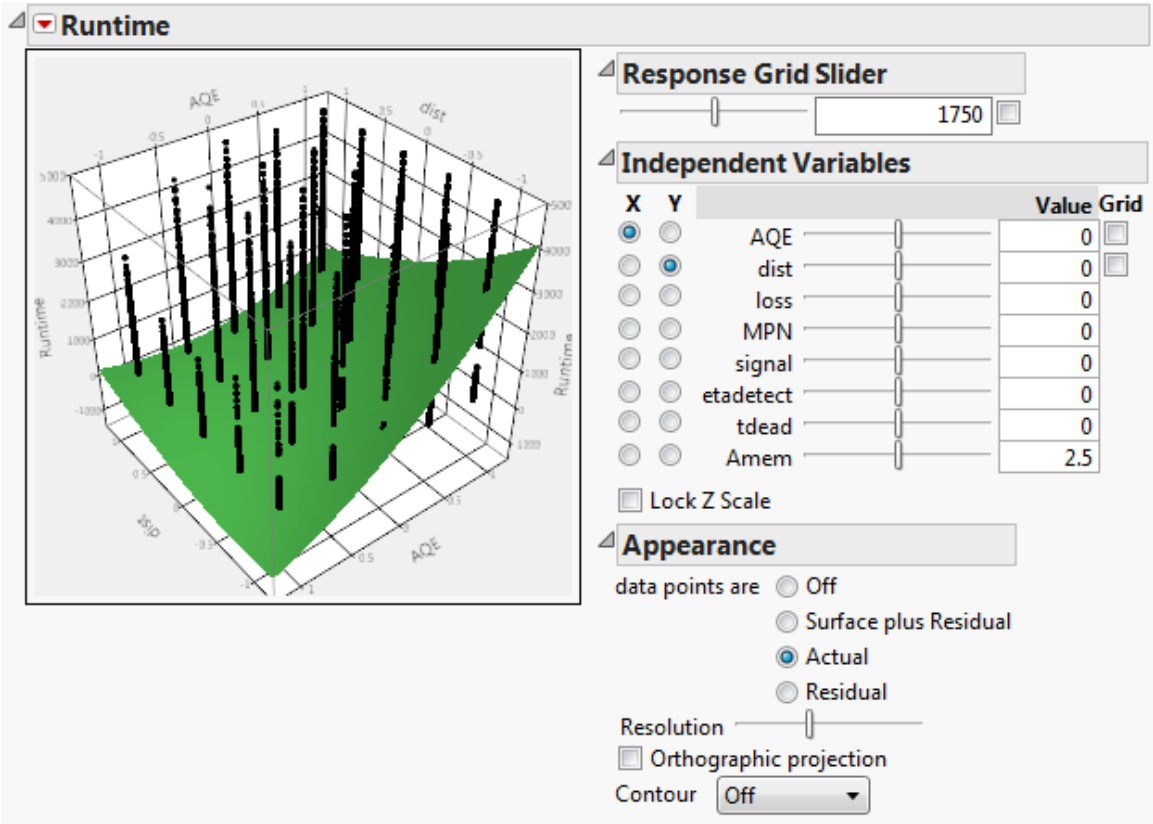


Figure 55. Surface profile of AQE vs. dist with actual

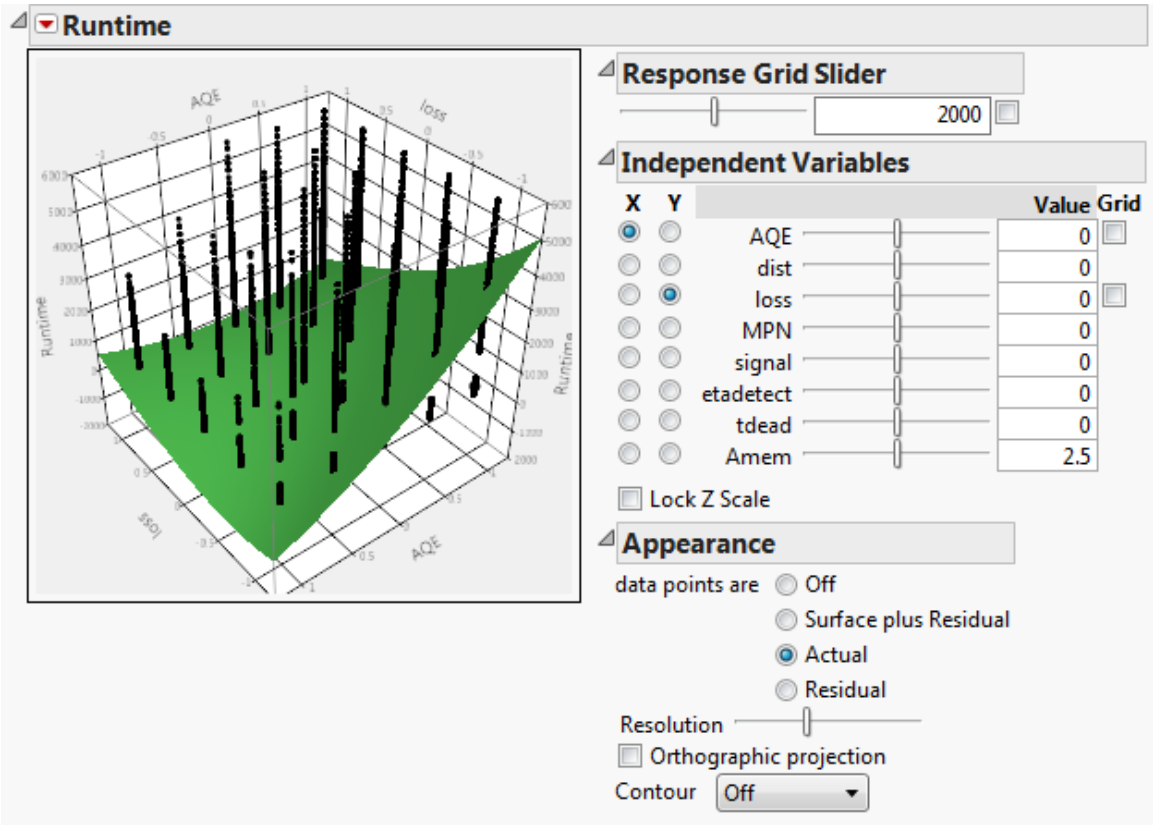


Figure 56. Surface profile of AQE vs. loss with actual

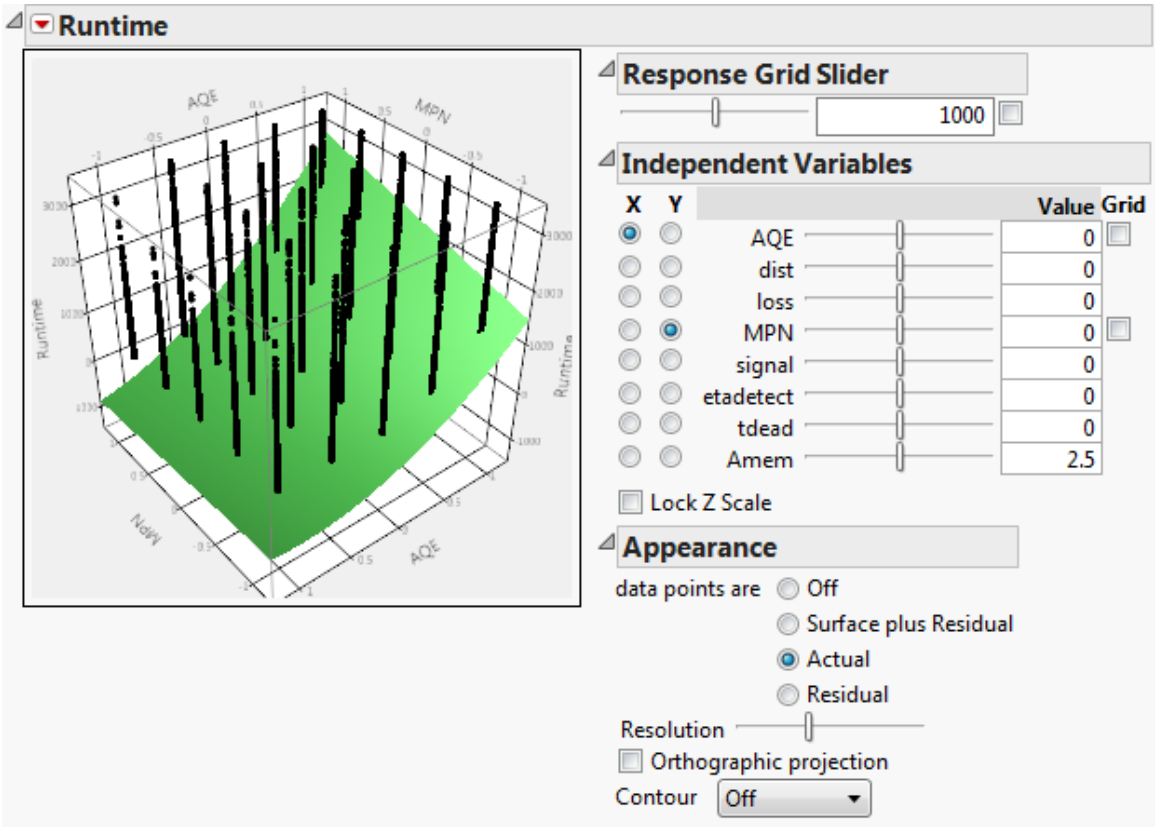


Figure 57. Surface profile of AQE Vs. MPN with actual

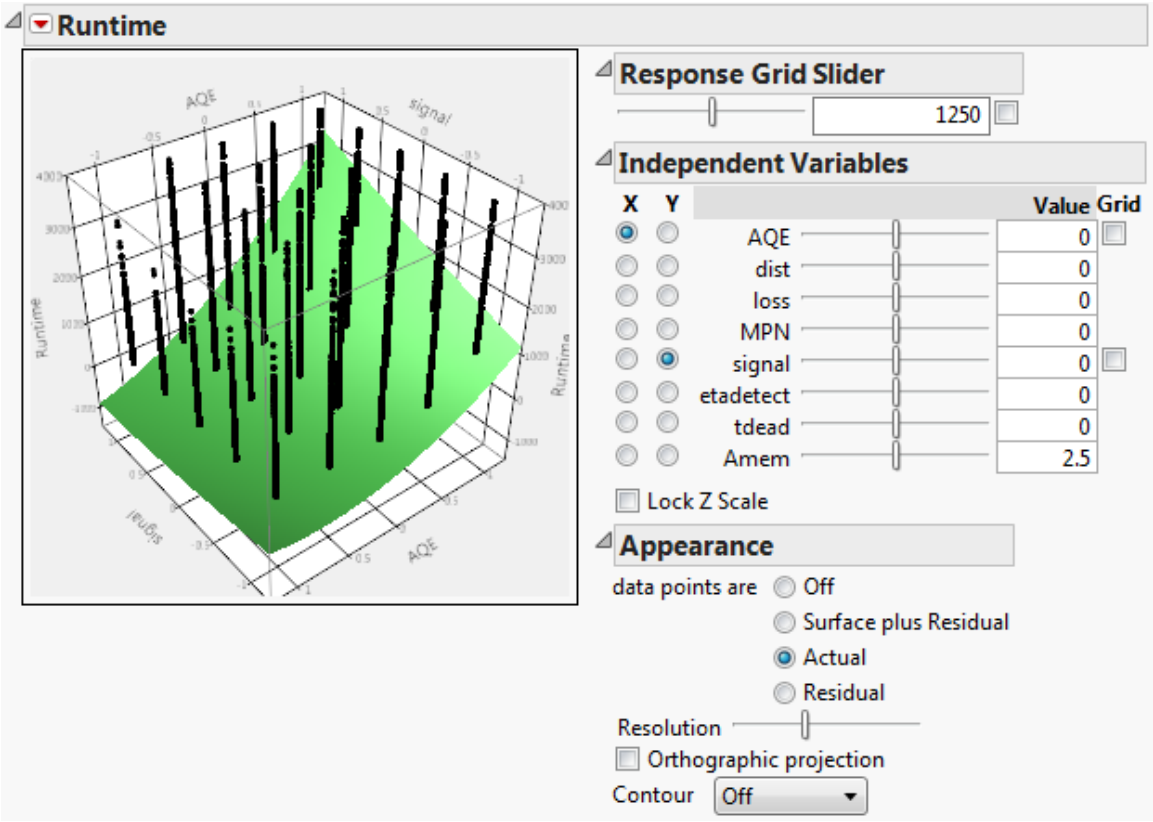


Figure 58. Surface profile of AQE vs. signal with actual

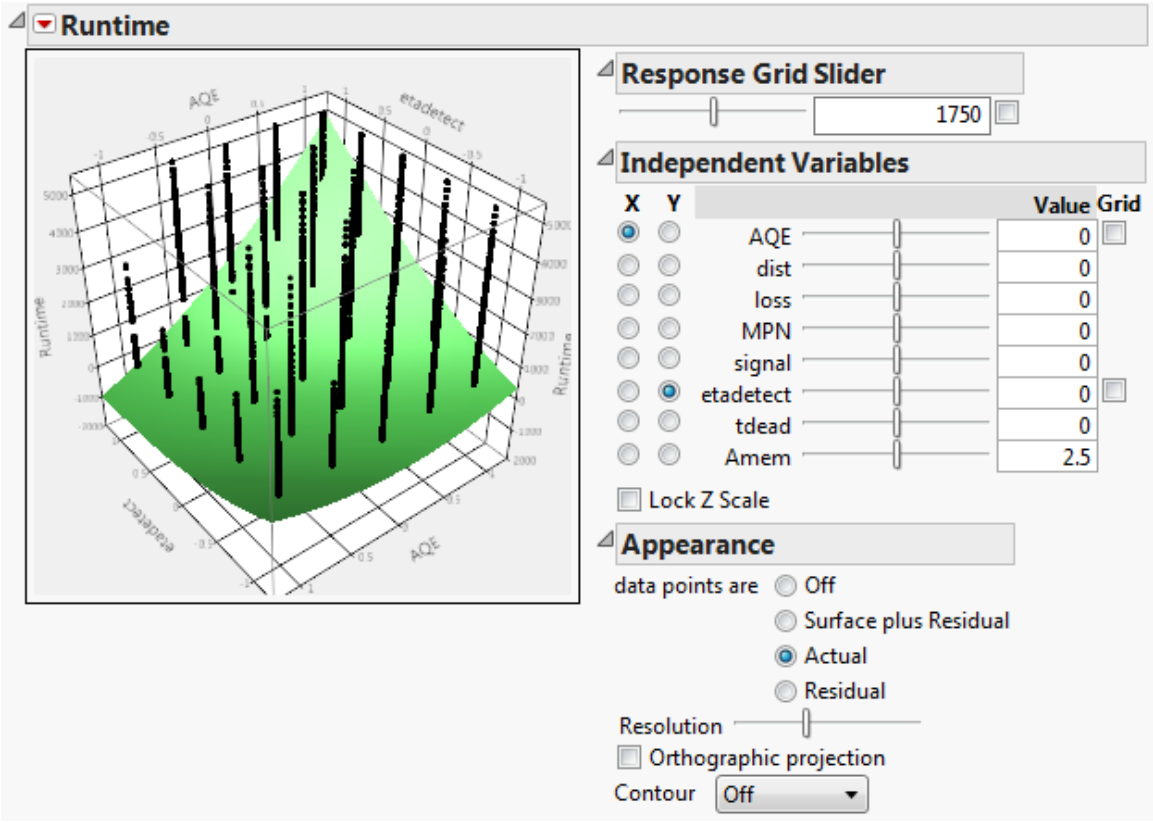


Figure 59. Surface profile of AQE vs. etadetect with actual

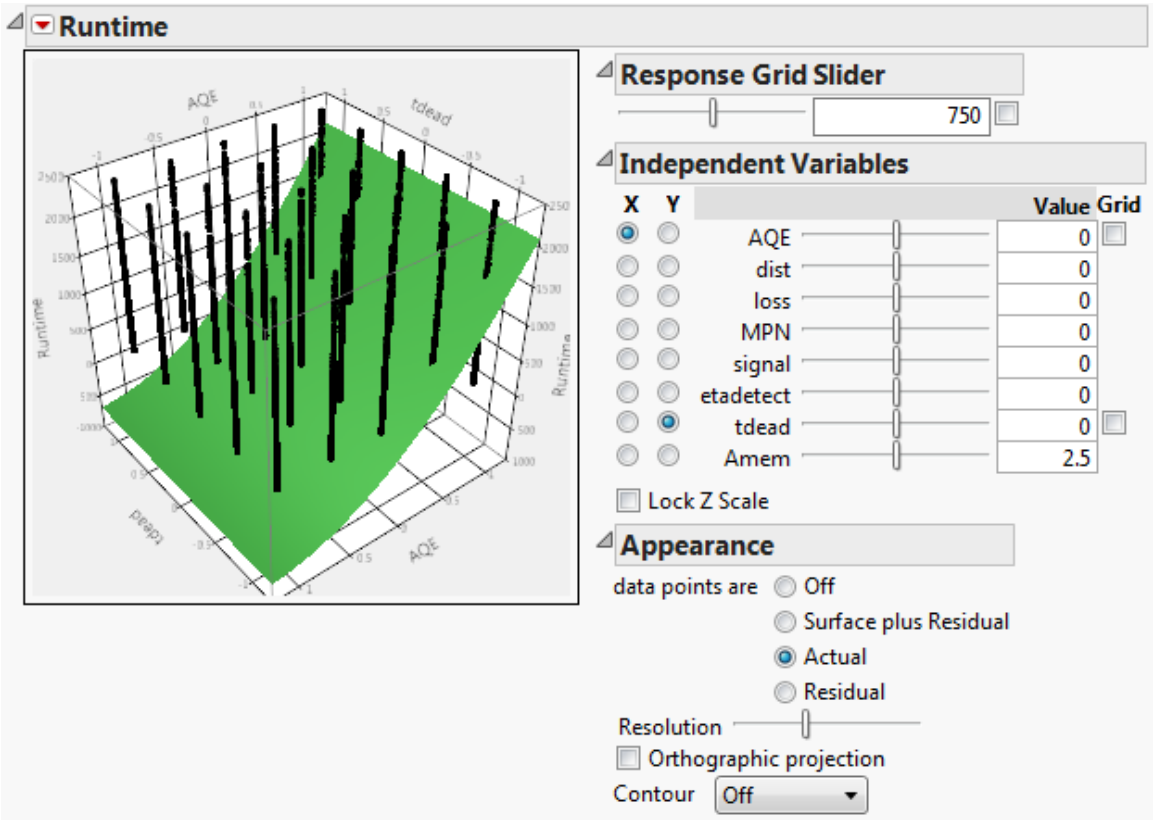


Figure 60. Surface profile of AQE vs. tdead with actual

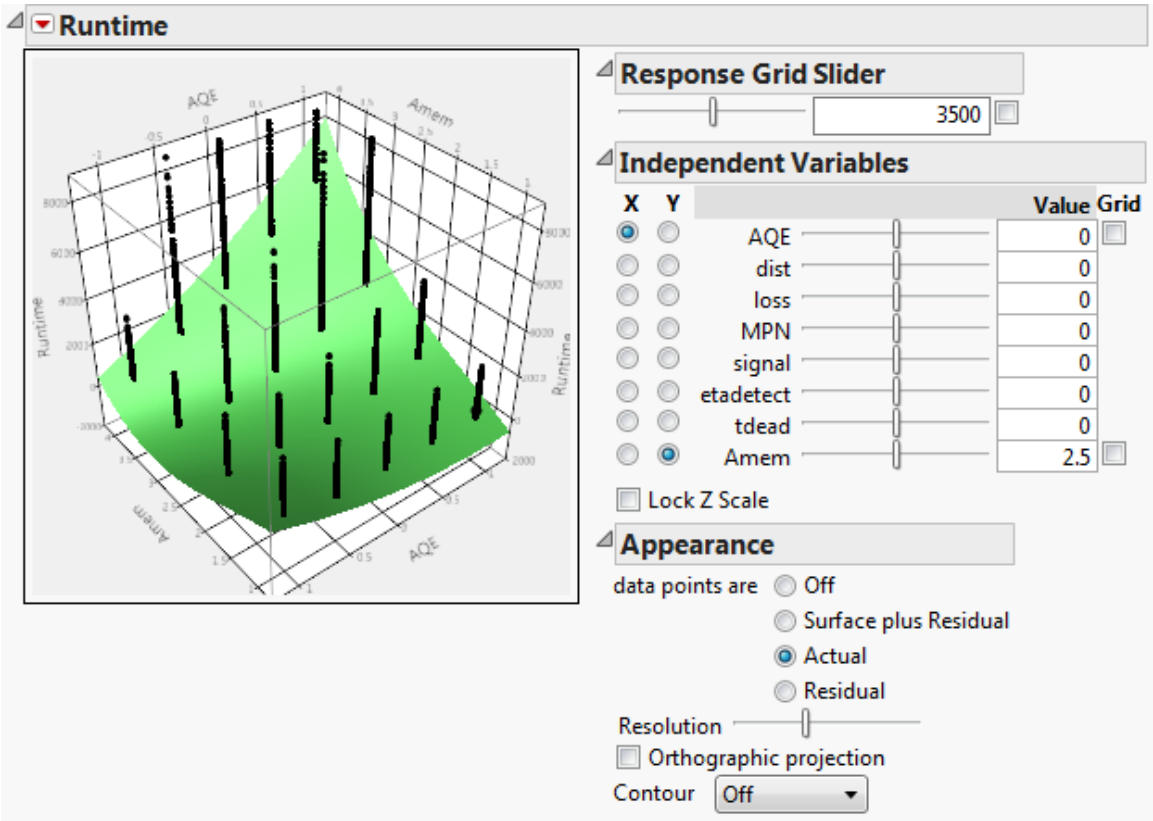


Figure 61. Surface profile of AQE vs. Amem with actual

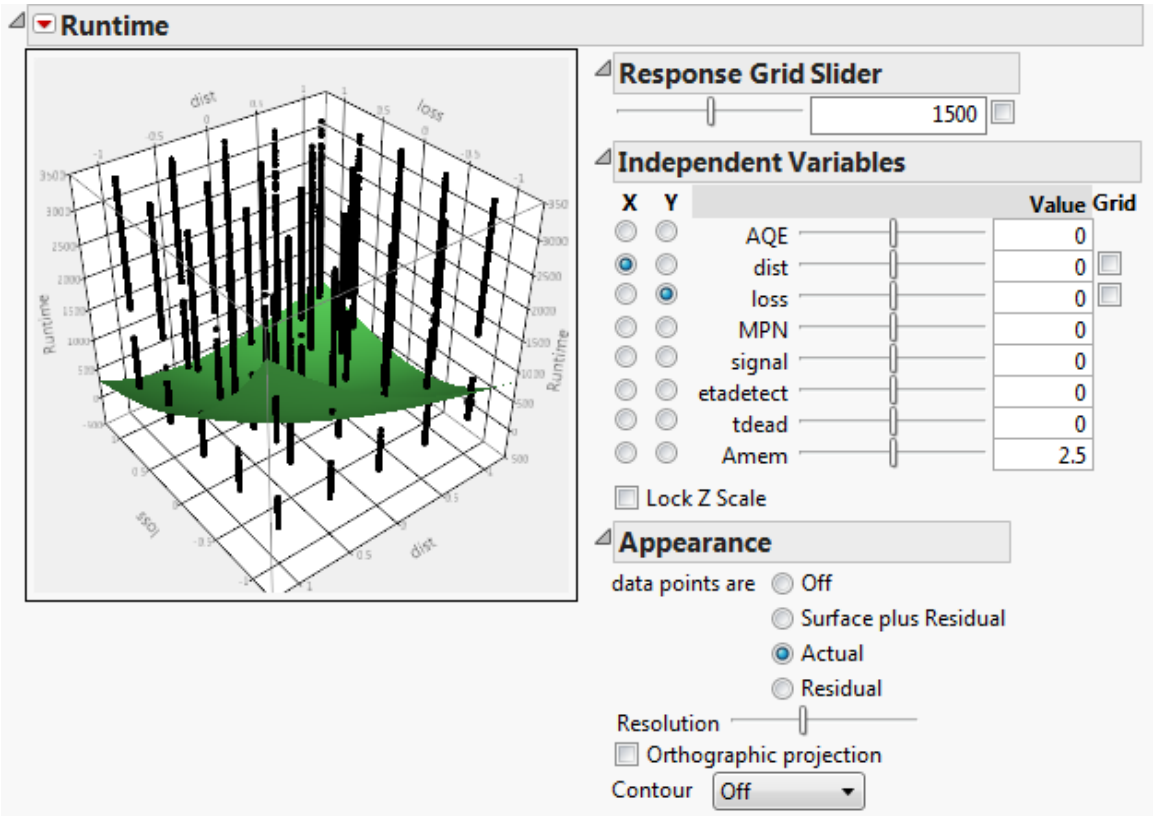


Figure 62. Surface profile of dist vs. loss with actual

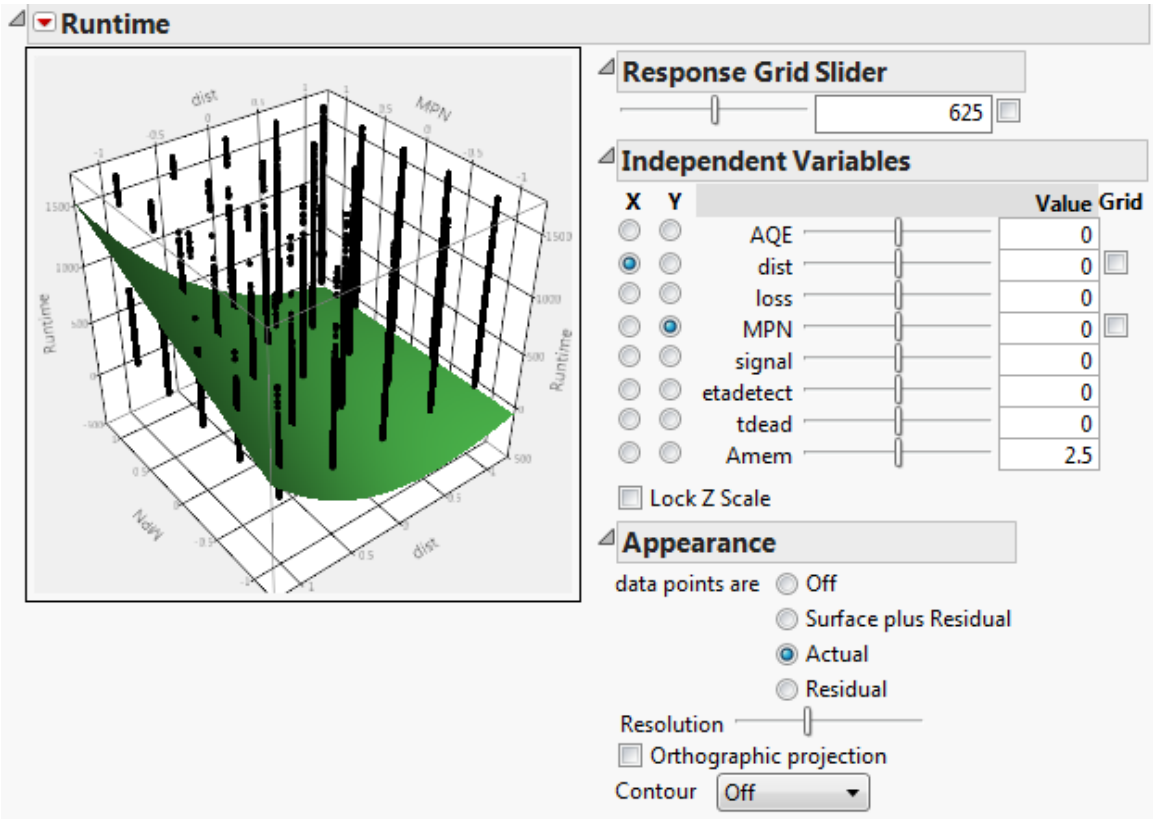


Figure 63. Surface profile of dist vs. MPN with actual

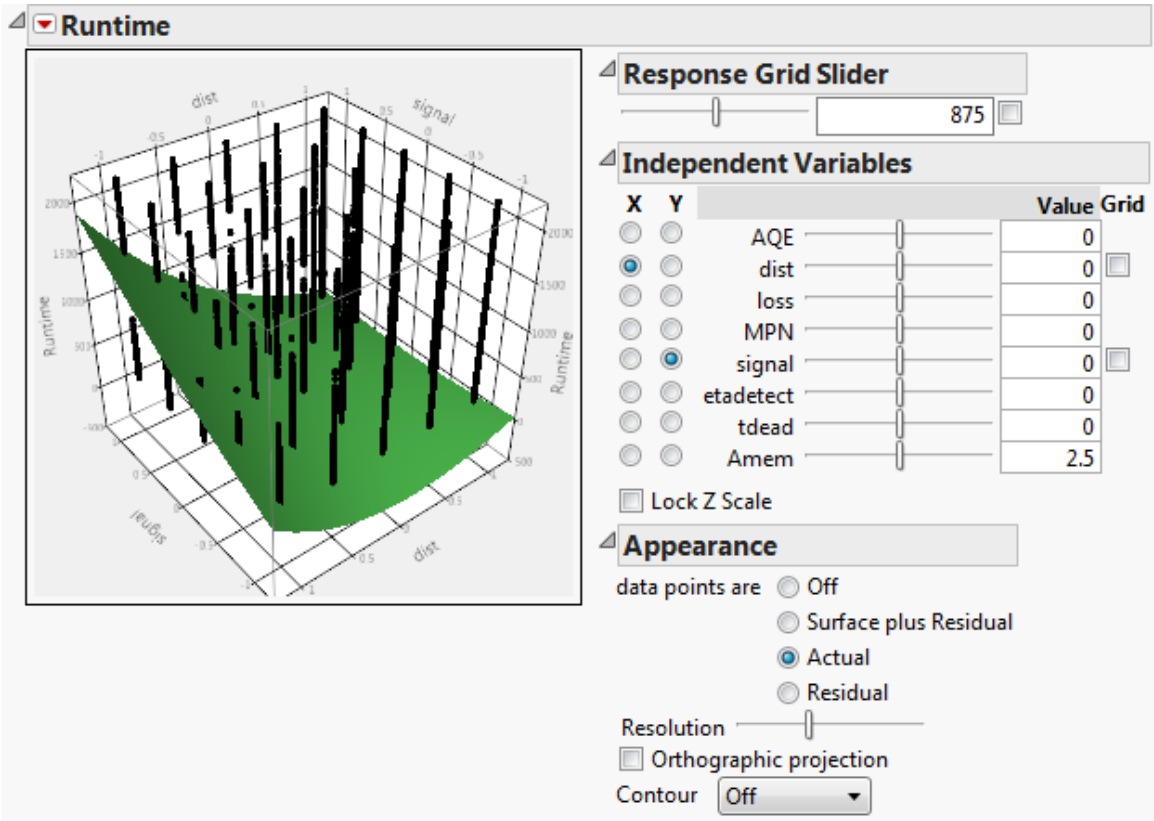


Figure 64. Surface profile of dist vs. signal with actual

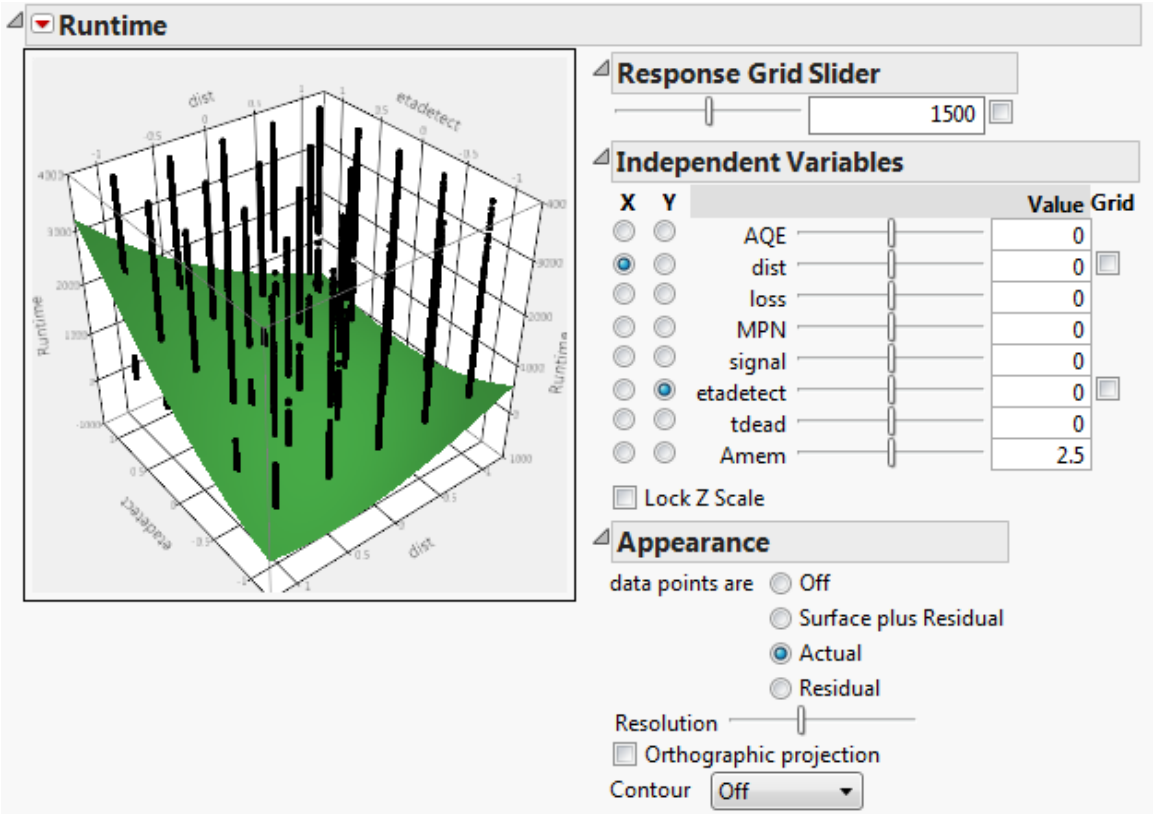


Figure 65. Surface profile of dist vs. etadetect with actual

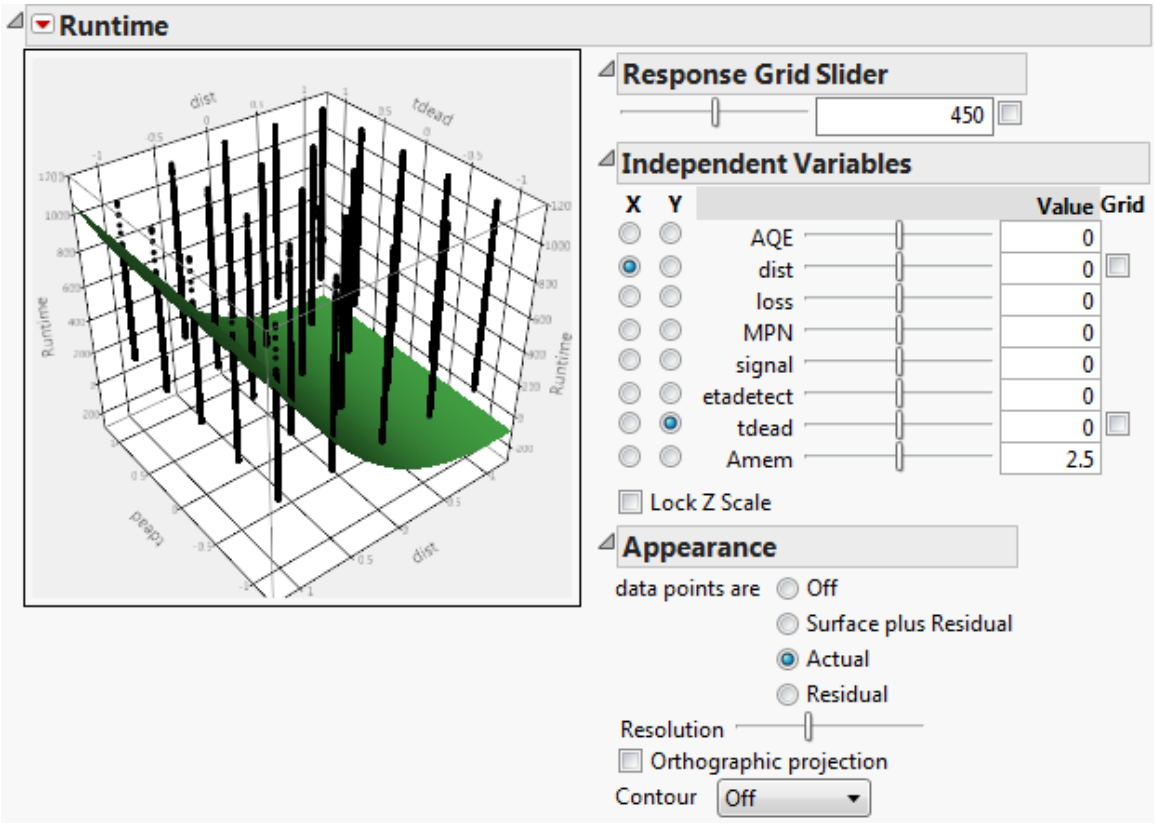


Figure 66. Surface profile of dist vs. tdead with actual

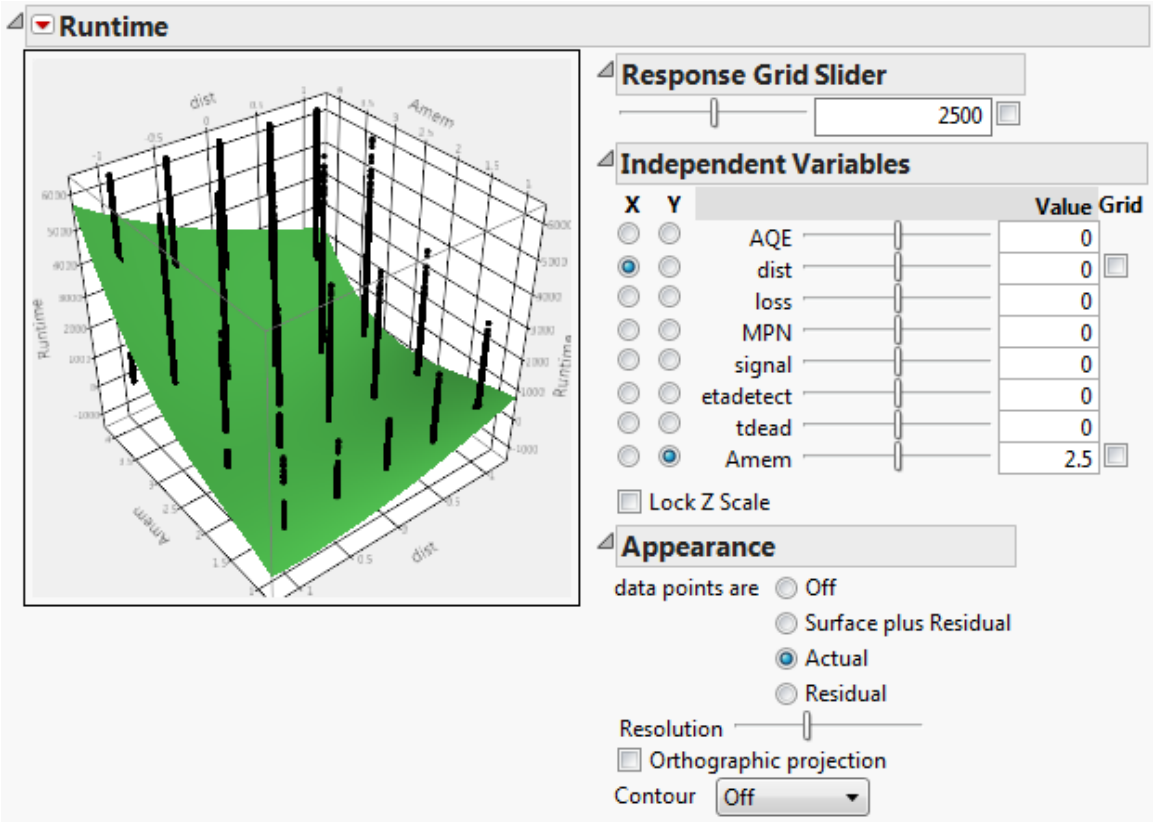


Figure 67. Surface profile of dist vs. Amem with actual

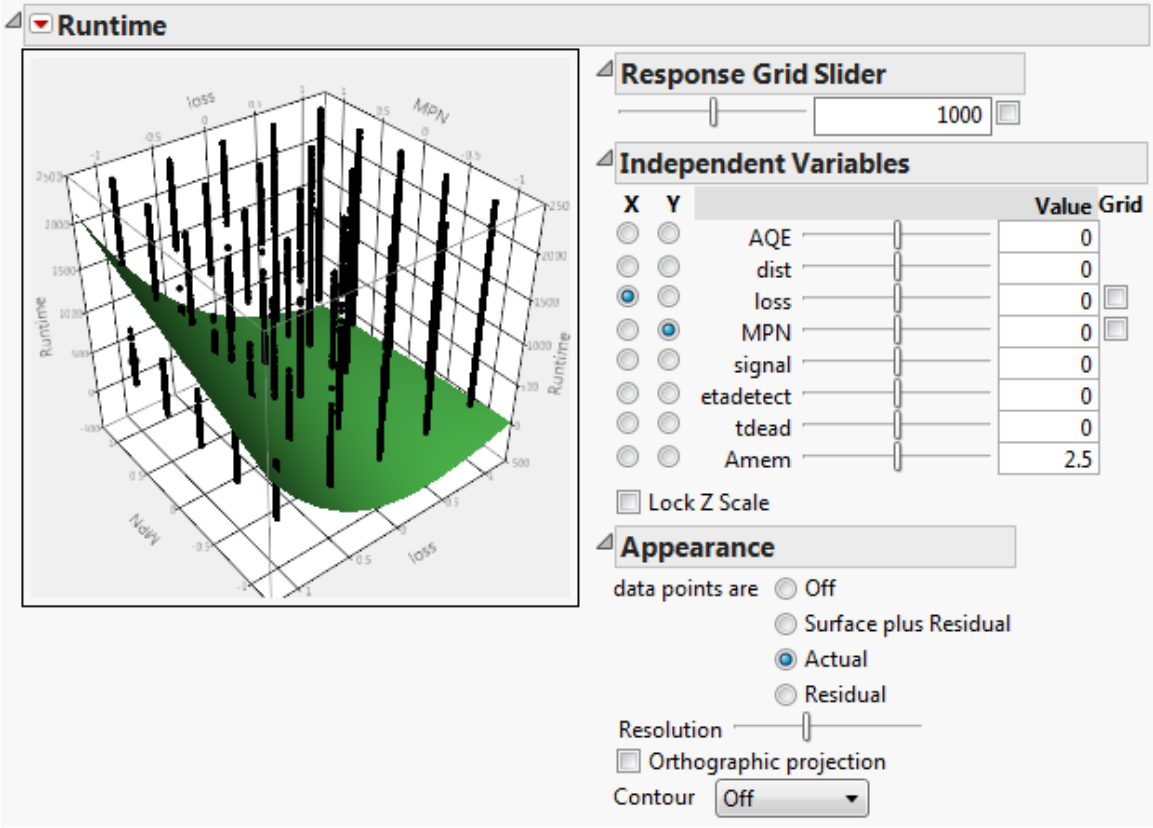


Figure 68. Surface profile loss vs. MPN with actual

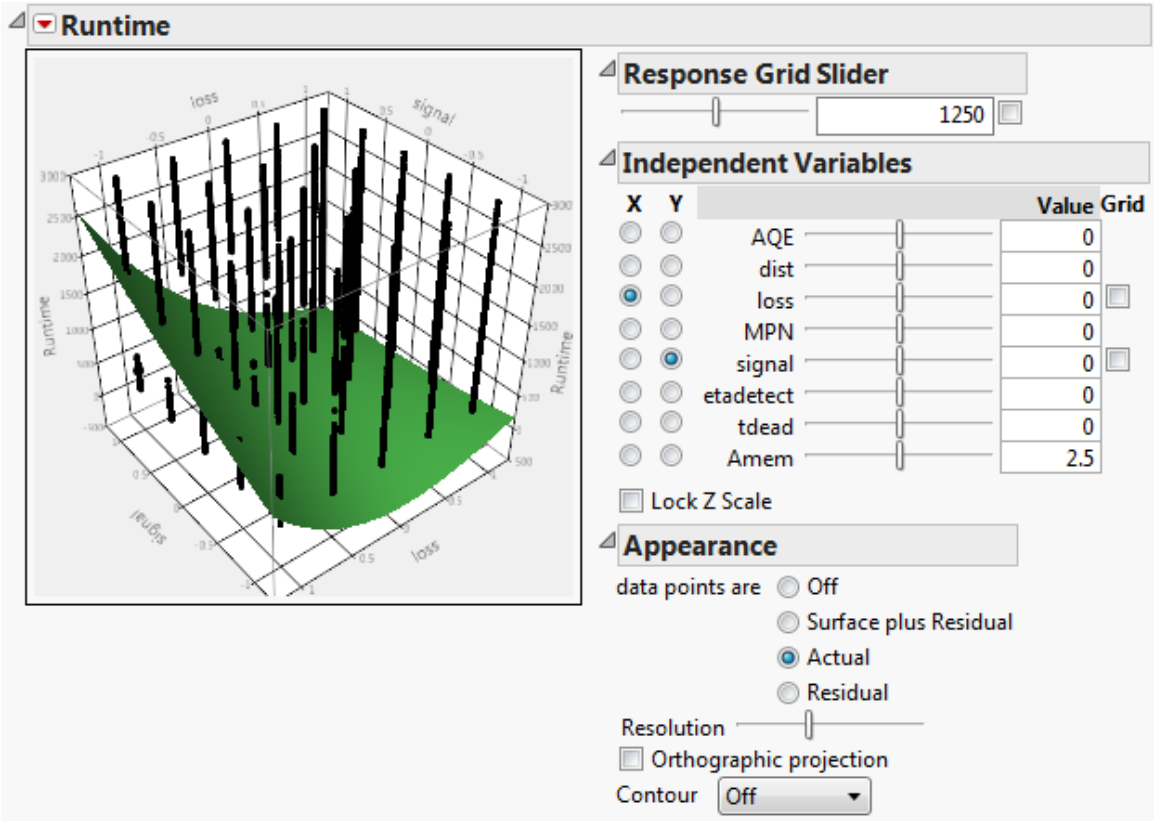


Figure 69. Surface profile of loss vs. signal with actual

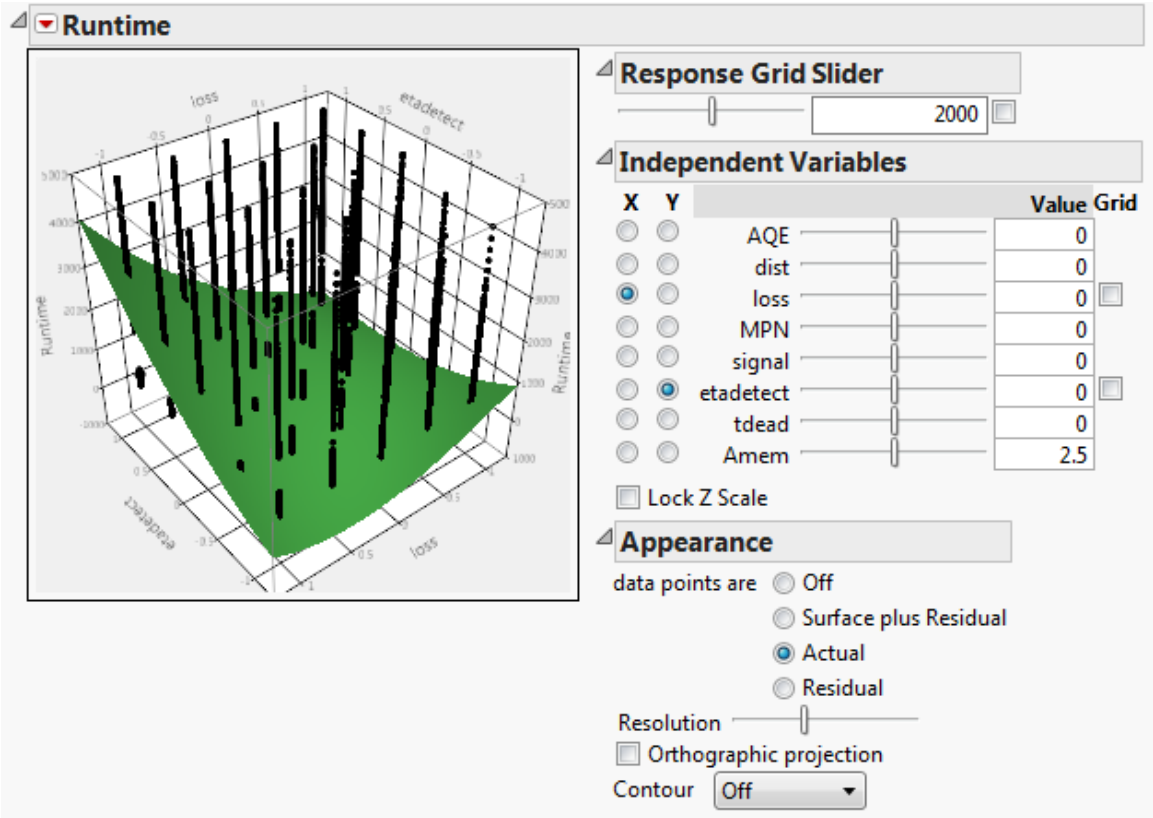


Figure 70. Surface profile of loss vs. etadetect with actual

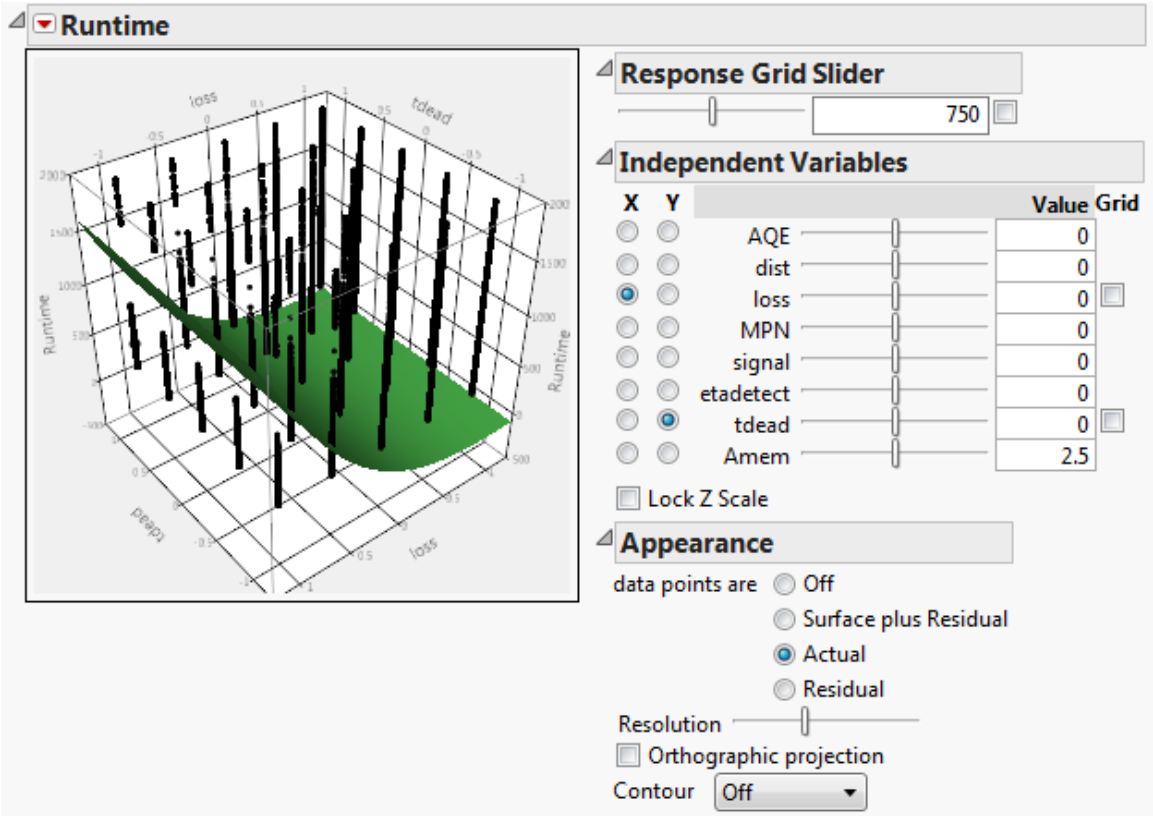


Figure 71. Surface profile of loss vs. tdead with actual

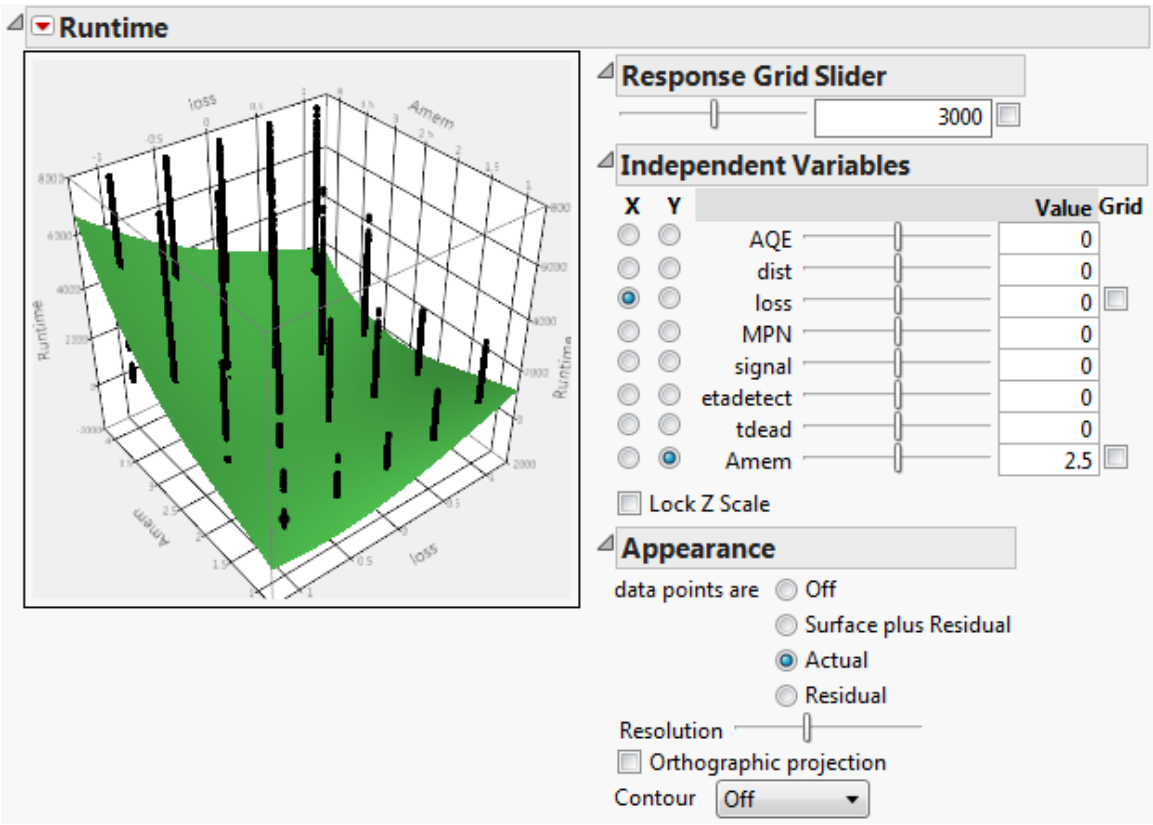


Figure 72. Surface profile of loss vs. Amem with actual

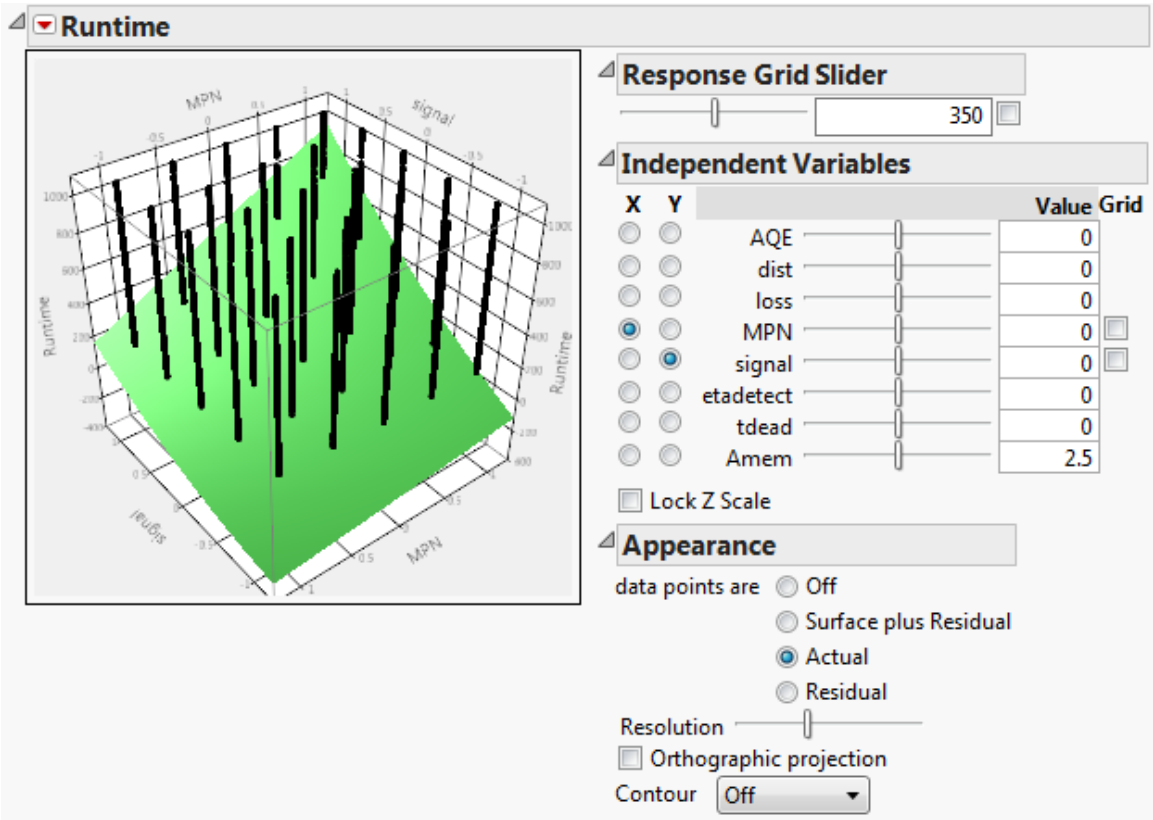


Figure 73. Surface profile of MPN vs. signal with actual

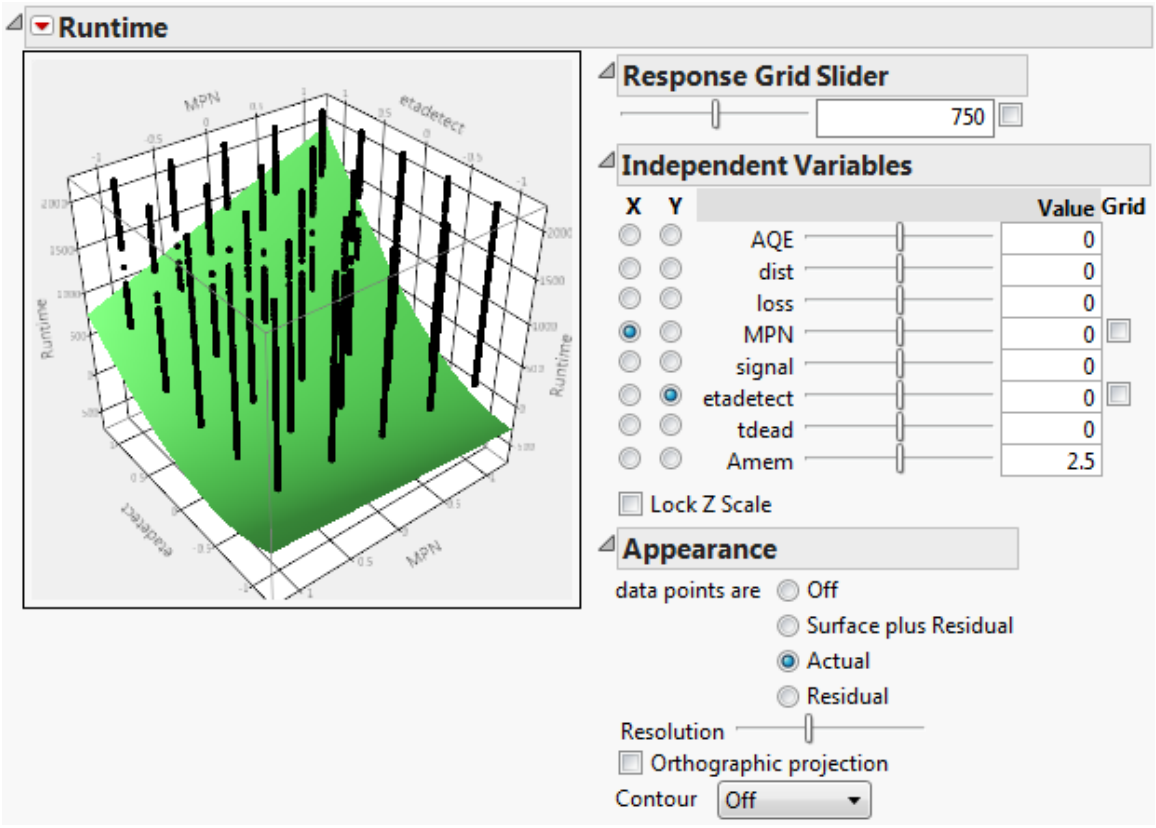


Figure 74. Surface profile of MPN vs. etadetect with actual

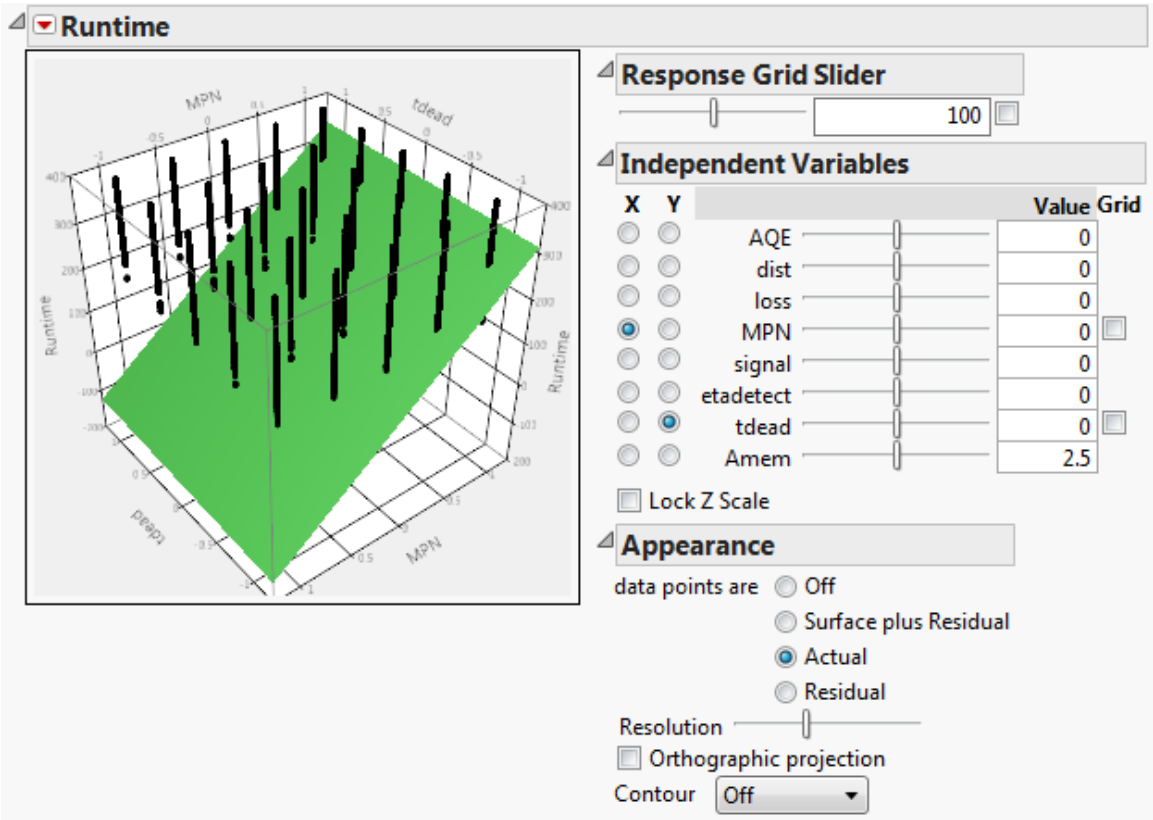


Figure 75. Surface profile of MPN vs. tdead with actual

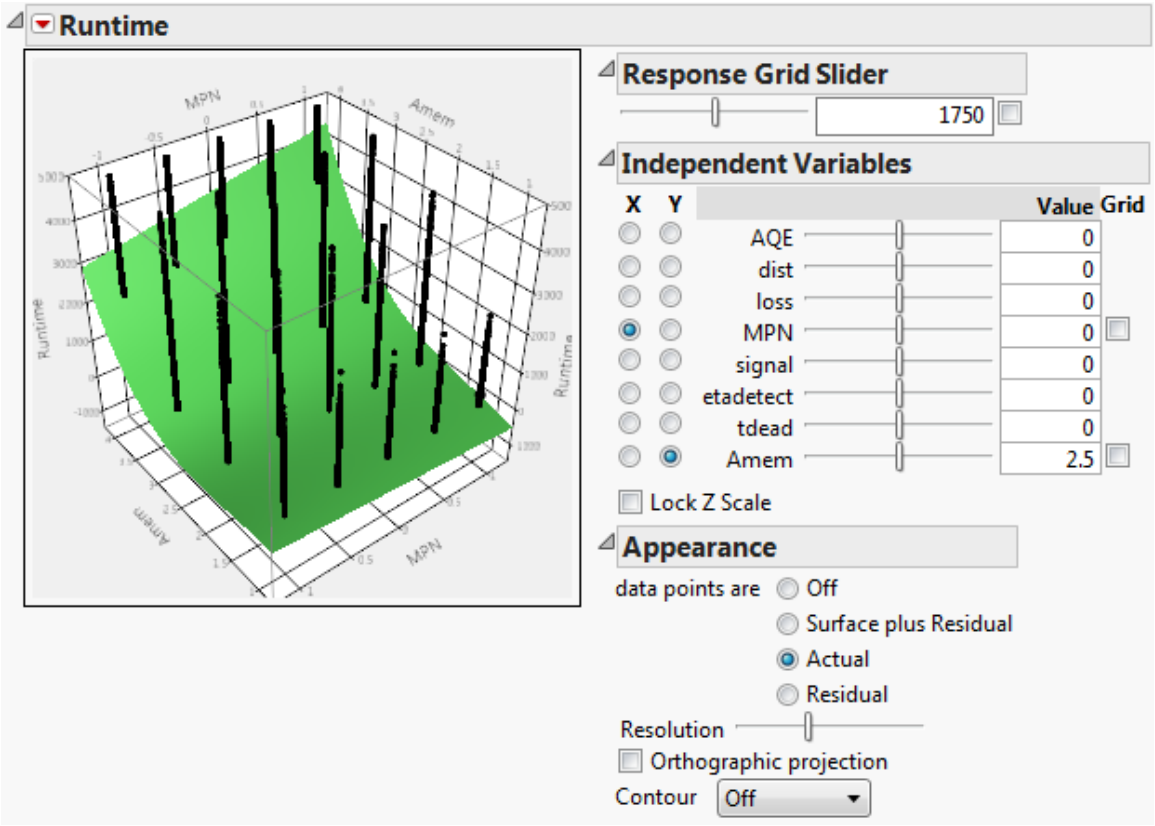


Figure 76. Surface profile of MPN vs. Amem with actual

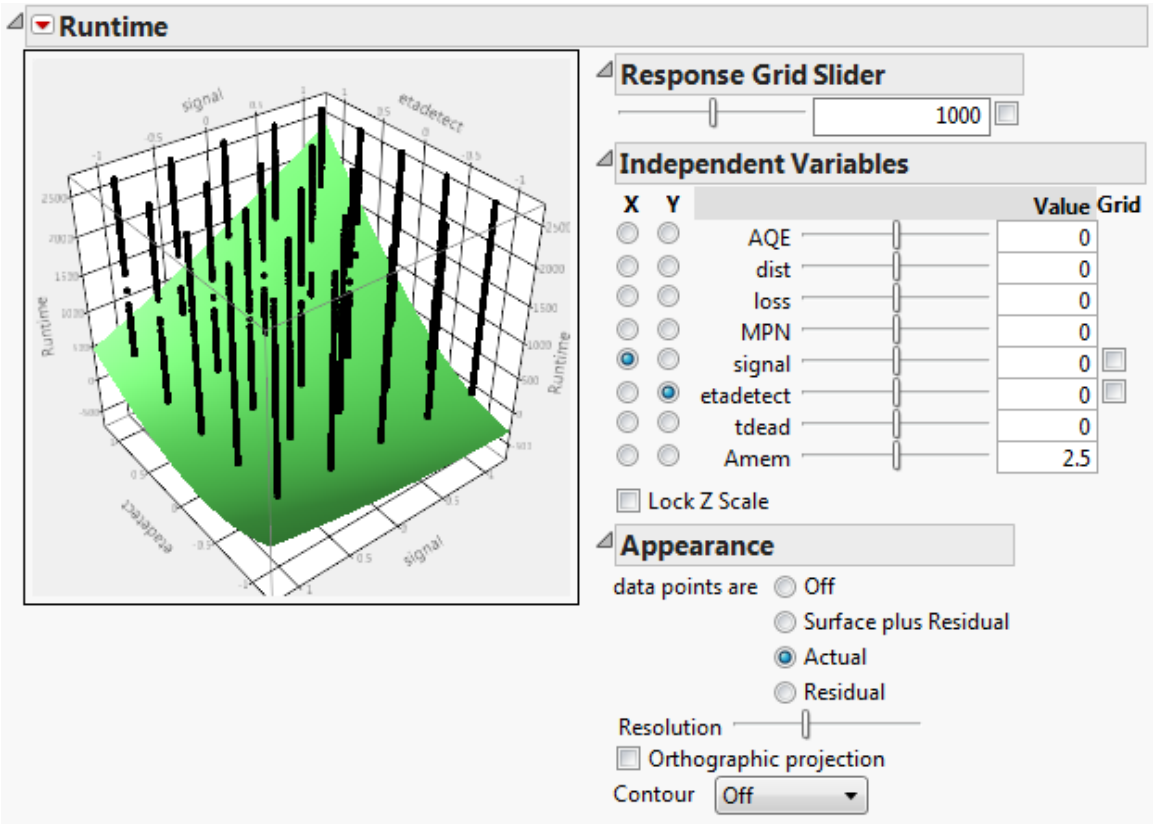


Figure 77. Surface profile of signal vs. etadetect with actual

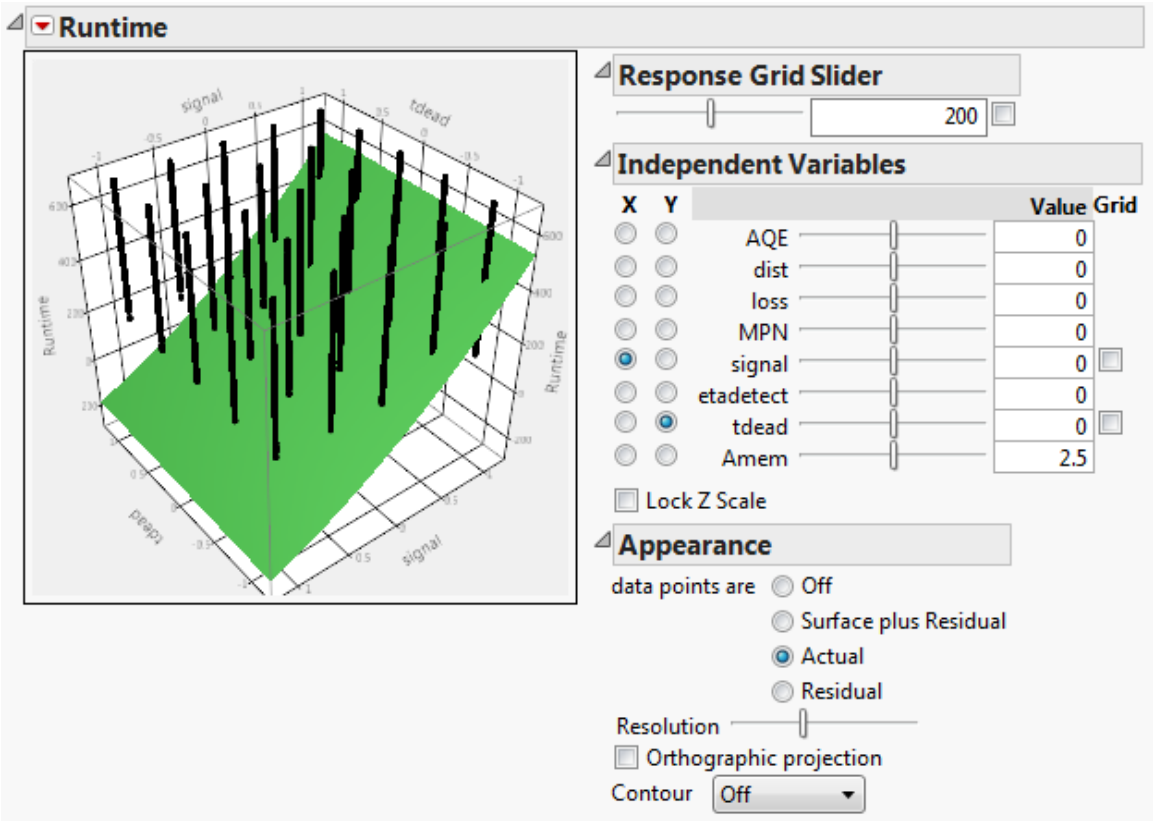


Figure 78. Surface profile of signal vs. tdead with actual

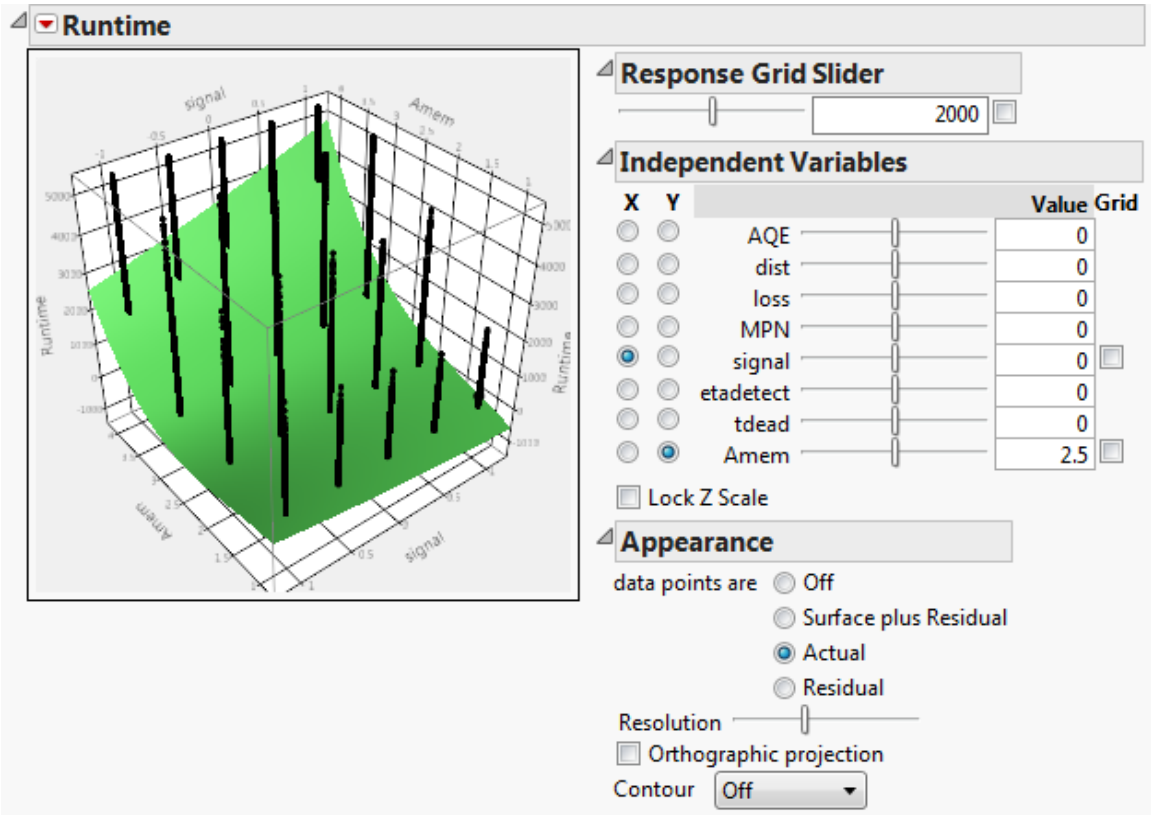


Figure 79. Surface profile of signal vs. Amem with actual

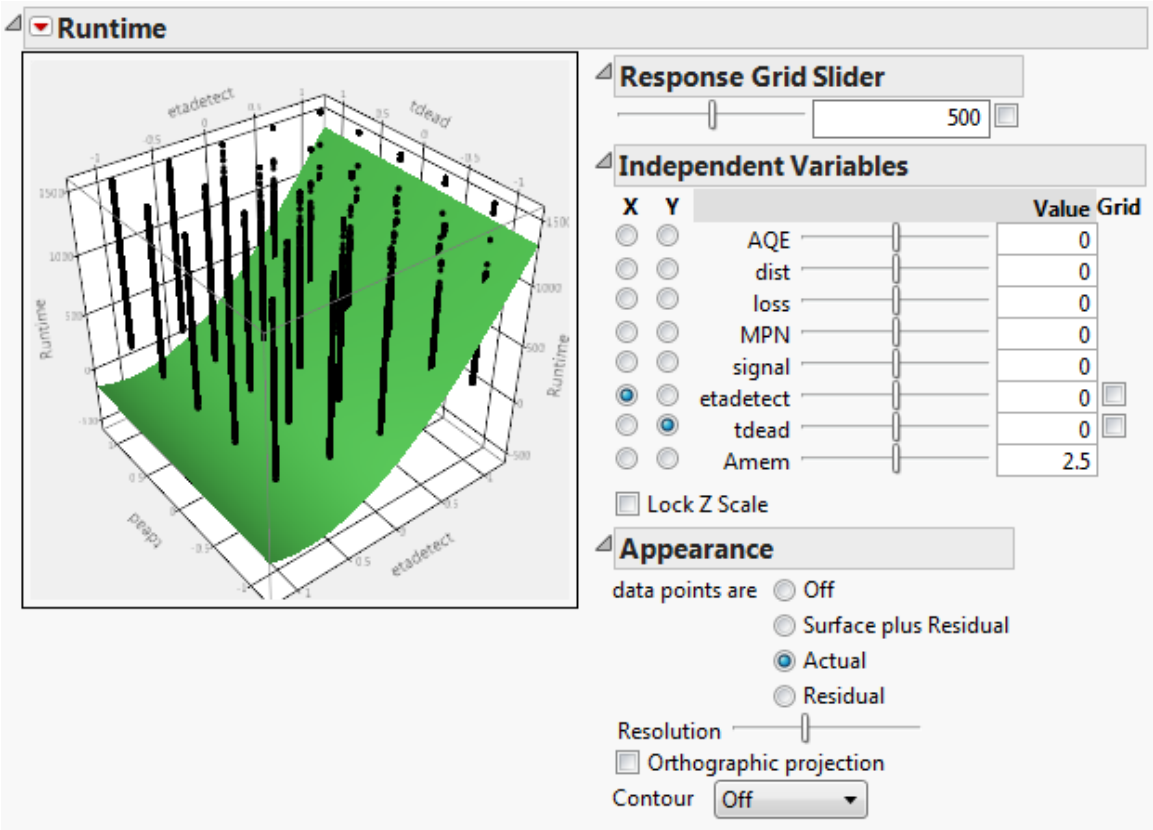


Figure 80. Surface profile of etadetect vs. tdead with actual

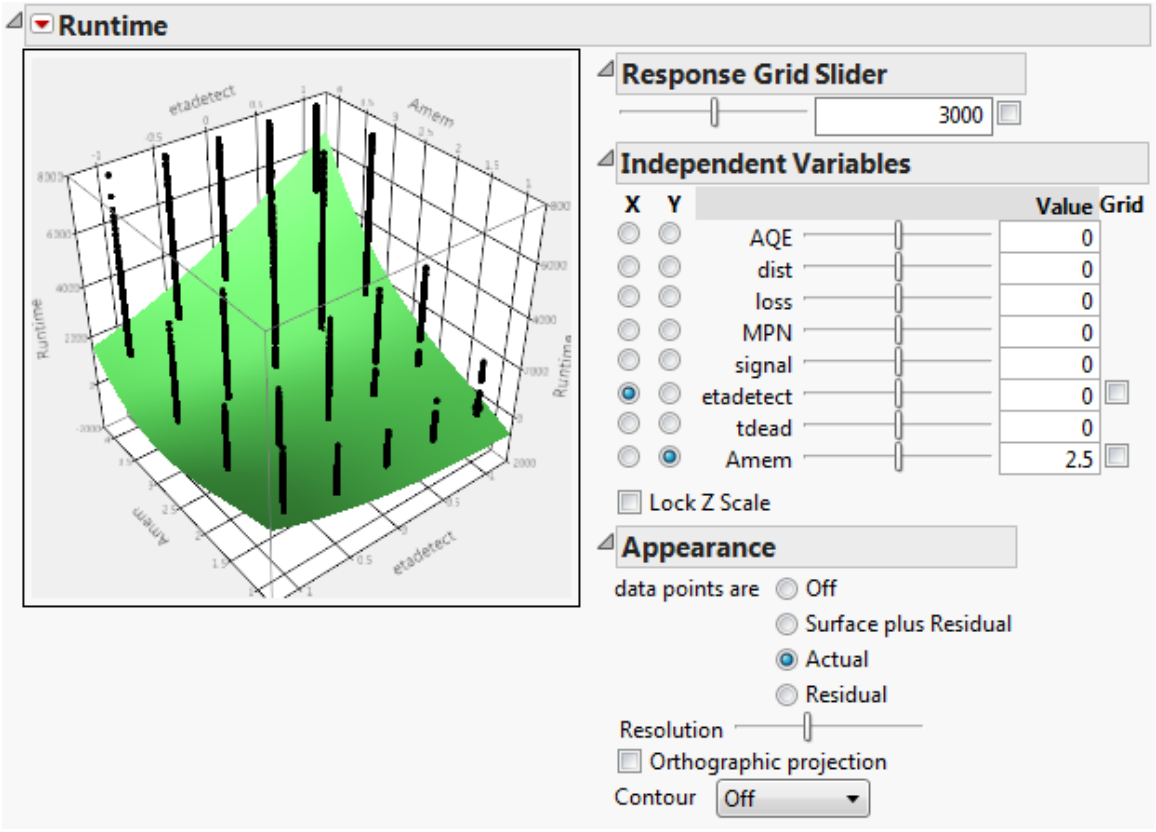


Figure 81. Surface profile of etadetect vs. Amem with actual

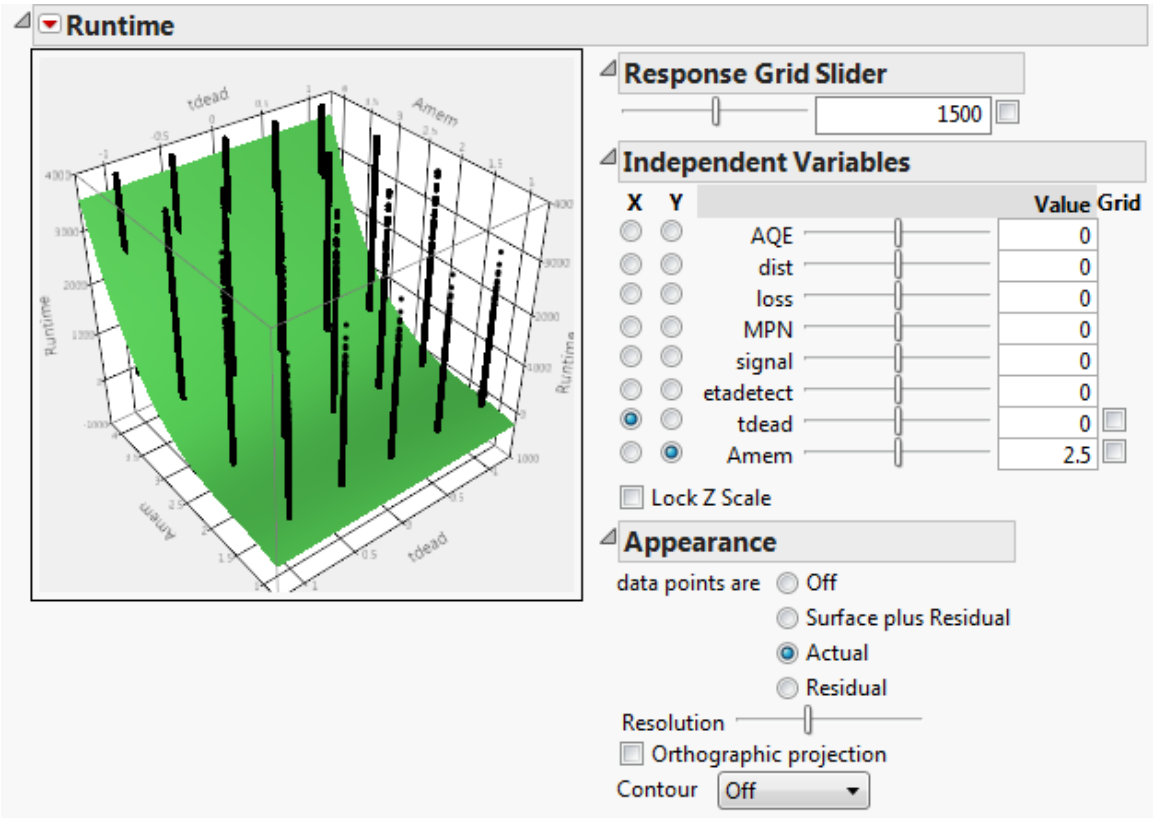


Figure 82. Surface profile of tdead vs. Amem with actual

Bibliography

1. Bennett, C.H., & Brassard, G. 1984. Quantum Cryptology: Public Key Distribu–tion and Coin Tossing. *International Conference on Computers*, **560**, 7–11.

2. Cernera, R. C. 2015. *A system-level throughput model for quantum key distribution.* Master's Thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH.

3. Chen, T., Liang, H., Liu, Y., Chai, W., Ju, L., Liu, W., Wang, J., Yin, HK., Chen, K., Chen, Z., Peng, C., & Pan, J. 2009. Field test of a practical secure communication network with decoy-state quantum cryptography. *Opt. Express*, **17**, 6540.

4. Economystique. *Top 10 The Imitation Game Movie Quotes Alan Turing Benedict Cumberbatch.* `http://economisty.com/top-10-the-imitation-game-movie-quotes-alan-turing-benedict-cumberbatch/`. Accessed: 2015-12-30.

5. Grimaila, M.R., Morris, J., & Hodson, D. 2012. Quantum Key Distribution: A Revolutionary Security Technology. *ISSA Developing and Connecting Cybersecurity Leaders Globally*, 20–27.

6. Kahn, David. 1974. *The codebreakers.* Weidenfeld and Nicolson.

7. Rivest, R.L. 1990. "Cryptology" Chapter 13 of Handbook of Theoretical Computer Science, (ed. J. Van Leeuwen) vol. 1 (Elsevier, 1990). 717–755.

8. Slutsky, B., Rao, R., Sun, P., Tancevski, L., & Fainman, S. 1998. Defense Frontier Analysis of Quantum Cryptographic Systems. *Applied Optics*, 2869.

9. Townsend, John S. 2000. *A modern approach to quantum mechanics.* University Science Books.

10. Wiesner, S. 1983. Conjugate Coding. *ACM SIGACT News, Winter-Spring 1983*, **15**, 78–88.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
24–03–2016	Master's Thesis	OCT 2014 — MAR 2016

4. TITLE AND SUBTITLE

A Response Surface Validation of a Quantum Key Distribution Model

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Ehrlich, Jacob M. Ehrlich., Second Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENS)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENS-MS-16-M-104

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Secretary of Defense
ATTN: Dr. Catherine Warner
1700 Defense Pentagon
Washington D.C. 20301
(703) 697-7247

10. SPONSOR/MONITOR'S ACRONYM(S)

OSD

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Statement A.
Approved for Public Release; distribution unlimited.

13. SUPPLEMENTARY NOTES

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

The need for secure communication in the presence of an adversary introduced the field of cryptology – the practice and study of techniques for secure communication. A common method to secure communication is to distribute a secret key among authorized parties so they can encrypt and decrypt messages between each other. By doing so, ideally, any messages intercepted by a third party are meaningless. An innovative technique to distribute a shared key is Quantum Key Distribution (QKD). QKD uses laws of quantum mechanics to generate and distribute such keys. The purpose of this thesis is to validate an existing mathematical model that is abstract enough to model the essential characteristics of a wide range of QKD system designs. The current model is based on a set of coupled equations. Equation coupling is high as many output variables for a specific phase are inputs for other equations. Because of this, the model output response function is complex, motivating the use of experimentation and response surface modeling to characterize and understand the relationship between inputs and outputs. The mathematical model was designed to capture the essential details associated with a wide variety of system configurations (i.e., designs). Surfaces representing the relationships between inputs and outputs are plotted and used with subject matter experts (SME's) to validate model behavior. After validation, a genetic algorithm is used to optimize the estimated surface. Our findings confirm the complexity of the model and indicate the presence of extreme outliers.

15. SUBJECT TERMS

model validation, design of experiments, response surface methodology, quantum key distribution, genetic algorithm.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Ray Hill, AFIT/ENS
U	U	U	U	103	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x7469 raymond.hill@afit.edu