

**SANAL BİR DENEY LABORATUARININ  
TASARIMI ve UYARLANMASINDA İLK ADIMLAR<sup>1</sup>****Mustafa Y.ATA<sup>2</sup>****ÖZET**

Matematiksel modeli verilen rassal bir olayın benzetimini ve benzetim verilerini kullanarak dizge çıktısının dağılımına ilişkin bir parametrenin Monte Carlo tahminini, verilen bir yakınsanma kistasına göre gerçekleştirecek sanal bir deney laboratuvarının tasarımı ve genel amaçlı bir programlama dili ortamında oluşturulması adım adım sergilenmiştir.

**Anahtar Sözcükler:** Monte Carlo, benzetim, benzetim dili, sanal deney.

**THE FIRST STEPS IN THE DESIGN AND IMPLEMENTATION OF  
A VIRTUAL LABORATORY****ABSTRACT**

The design of a virtual laboratory which will realize the simulation of a random phenomenon with a given mathematical model and the Monte Carlo estimation of a parameter related to the distribution of the system output using the simulated data according to a given convergence criterion, and its implementation in a general purpose programming language is exhibited step by step.

**Keywords:** Monte Carlo, simulation, simulation language, virtual experiment.

**SUNUŞ**

Tek başına suya itildi bir gün. Onu suya iten “Yüzebilirsin.” demişti ve eline bir demet çiçek vermişti, karşı kıyılardan derlenmiş. Böyle yüzmek hoşuna gitti ve şimdye kadar içinde yüzdüğü bu suda batmadı. Ama içinde kulaç attığı suyun engin bir deniz olduğunu biliyor ve bu denizin ulaşamadığı yerlerini çok merak ediyor. Bu denizde dolaşan gemiler olduğunu biliyor, ancak hiç birine güvenmiyor, çünkü hepsi önceden belirlenmiş yolları izliyor. Artık kendi gemisini yapmak istiyor, ancak bunu tek başına yapamayacağını biliyor. Aslında bu gemiyi kendisi için de istemiyor. Ama bir gün, kendisi ile aynı dili konuşan birilerinin kendi gemilerini yaptıklarını, ve bu gemilerle hiçbir sınır tanımadan o engin denizde dolaştıklarını duyarsa, çok sevinecek. Ve bekliyor, çok yakınlarında ve sabırla, o denizde, uzaktan geçen gemilere hep kıyıda bakan, ya da belli bir limanda binip bir başkasında indikten sonra, “benim de kendi gemim olsa” diye düş kuran, ve kendisi ile aynı dili konuşan, genç ve yiğit kişilere, kurdukları düşü gerçek yapmanın ne kadar kolay olduğunu anlatmak, ve onları suya itivermek için...

<sup>1</sup>Bu makalenin geniş bir özeti, 4-6 Temmuz 2005 tarihleri arasında düzenlenen, XV. Ulusal Yöneyem ve Endüstri Mühendisliği Kongresi'ne bildiri olarak sunulmuştur.

<sup>2</sup>Gazi Üniversitesi, Fen-Edebiyat Fakültesi, İstatistik Bölümü, Beşevler/Ankara, yavuzata@gazi.edu.tr

*SDL*, yüzerek ulaşılabilen yakın ve küçük yerlerden toplanmış bir demet kır çiçeği. O, yıllar önce beni bu sulara iten, Sayın Hocam Prof. Dr. İmdat Kara ve ileride *SDL*'den daha güzel çiçek demetleri sunacaklar için...

## 1. GİRİŞ

Girdileri olasılık dağılımlarından rasgele örnekleme ile elde edilen rassal değişken değerleri olan bir dizgenin çıktıları niteliğindeki, olasılık dağılımları bilinmeyen rassal değişkenlerin beklenen değerlerini, doğrudan Büyük Sayılar Yasası(BSY) mantığını kullanarak tahmin etmek amacı ile tasarılan sanal bir deneyin benzetimini bilgisayar ortamında gerçekleştirecek yazılım,

- söz konusu dizgenin özelliklerini göz önünde bulunduran *özel amaçlı* bir programlama dili, ya da
- *genel amaçlı* bir programlama dili

kullanılarak oluşturulabilir.

Sürekli olarak tek düze benzer özellikli sanal deneylerin gerçekleştirildiği ortamlarda birinci seçenek yeğlenebilir, ancak savunma dizgesi gibi geniş kapsamlı ve özgün bir dizgenin benzetimini gerektiren sanal deneylerde genel amaçlı bir programlama dilini kullanmak kaçınılmaz olur. Bu nedenle belirli yapıdaki sanal deneyleri tek başına gerçekleştirme durumunda olan bir benzetimci, genel amaçlı bir programlama dilindeki bilgisini derinleştirmek yerine genellikle belirli bir benzetim diline yönelir. Hemen her bilgisayarda hazır olarak bulunabilen genel amaçlı programlama dillerine göre çok daha çeşitli, pahalı ve erişimleri zor olduğundan, bu tür benzetim dilleri yaygın da değildir. Geniş kapsamlı sanal deney projelerinde genel amaçlı programlama dillerinden birinin seçilmesi, bu dillerin esnekliği yanında yaygın kullanımlarından dolayıdır. Eğer yeterli düzeyde genel amaçlı bir programlama dili bilgisi verilmeden ve yalnızca, şöyle ya da böyle erişilebilen, bir tek benzetim dili ortamında eğitilen bir benzetimci, gerçek iş ortamında, çalıştığı kuruluşun kıt kaynakları ile bedeli ödenerek satın alınan bu benzetim dilinin günün birinde gereksinimi karşılayamaması durumunda, hem kendisi ve hem de çalıştığı kuruluş açısından ciddi sıkıntılara yol açabilir. Böyle bir olumsuzlukta, bu benzetimciye iş veren kadar onu eğiten kurumun da payının olacağı açıktır.

Yeterli düzeyde genel amaçlı bir programlama dili bilgisi ön koşul olmak üzere,

- sanal deney tekniklerinin kuramsal temellerine ve sanal deney tasarım becerilerinin geliştirilmesine ağırlık veren,
- sanal deney uygulamalarının, genel yapısı ve ilkeleri ile birlikte genel amaçlı bir programlama dili ortamındaki çekirdeği verildikten sonra, öğrencinin adım adım kendisinin geliştireceği bir yazılımla gerçekleştirildiği

bir sanal deney eğitim ve öğretiminin, yukarıda sözü edilen olumsuzluktan sorumlu tutulamayacağı açıktır.

Burada, böyle bir yazılımın tasarımı ve uyarlanmasıdaki ilk adımlar, basit bir sanal deney çerçevesinde sergilenerek, bir Sanal Deney Laboratuvarı(*SDL*)'nın çekirdeği oluşturulacaktır.

*SDL*'nin,

- tasarımında, yapısal programlama ilkelerine uyulacak;
- kapsamı ise, yalnızca geliştirildiği programla dili ile sınırlı olacaktır.

*SDL*, bu özellikleri ile her türlü sanal deneyi gerçekleştirmeye kapıları açık bir yazılım olacağından, yeteri ölçüde ve çeşitlilikteki deneylerle zenginleştikten sonra, bir benzetim dilinin temelini de oluşturabilir.

Özel amaçlı bir programlama dili olan *benzetim dillerinin* başlangıç noktası da aslında, geniş kapsamlı ve özgün bir dizgenin benzetimi için genel amaçlı bir programlama dili ile oluşturulan yazılımlardır. Bu geniş kapsamlı benzetim yazılımları, yazılım tekniği gereği, yapısal programlama mantığının tekniklerini kullanarak kendi iç işlevlerini ve standart işlemlerini ayrı birer alt-yordam biçiminde oluşturur ve bir *program kütüphanesi* biçiminde programlama ortamına eklerler. Eğer böyle bir program kütüphanesinde yer alan işlevler ve alt yordamlara ad verilirken belirli bir kurala uyulmuş ise, söz konusu yazılım bir benzetim dili olmaya adaydır. Sanal deney yazılımının program kütüphanesi ve bu program kütüphanesinin içerdiği alt-yordamlara ad verme kuralı, böyle bir benzetim dilinin *temel sözcüklerini* ve *gramer kurallarının* başlangıcını oluşturur. Sözlüğünün çekirdeği, söz konusu sanal deney yazılımının program kütüphanesi olan böyle bir benzetim dili, özgün sanal deneyin yapısı ile kapsamı sınırlı sanal deneylerin

- tasarım boyutları, model ve Monte Carlo(MC) yakınsama kuralı parametreleri gibi *deneysel ortamı* tanımlayan *girdilerinin*,
- her denemedeki sanal gözlemleri mantıksal, matematiksel ve döngüsel işlemlerle girdilerden üreten süreç *modelinin*,
- deney sonundaki MC tahmin ve diğer istatistiklerden oluşan *çıkıntılarının*,

genel yapılarını tanımlamada kolaylık sağlayacak yeni gramer kuralları ve yeni sözcükler üretme çizgisinde ilerleyebilir.

Geniş kapsamlı bir dizgenin benzetimini başlangıç noktası olarak alan benzetim dillerinin gelişim çizgisi tümdengelim niteliklidir. Basit bir sanal deneyle yaşama geçecek olan *SDL* ise, deney çeşitliliği konusundaki deneyimini adım adım arttırarak ve yeterli bir deneyime kavuştuktan sonra, bir benzetim dili olma yolundaki gelişimini tümevarımsal bir çizgide sürdürebilir.

2. Bölümde, *SDL*'nin tasarımı ve kurulmasındaki ilk adımı atmak üzere seçilen rassal olayın matematiksel modeli, böyle bir modelin çıktısına ilişkin bir parametrenin sanal gözlem değerlerinden Monte Carlo tahminini belirli bir yakınsama kıstasına göre elde edecek bir yazılımın genel yapısı ile temel bileşenleri tanıtılacak ve ilk sanal deneyi gerçekleştirmek üzere *SDL* hazır duruma getirilecektir. 3.Bölüm'de ilk güvenilirlik sınaması gerçekleştirilecek olan *SDL*, genel yapısı ve önceki deneylere ilişkin bilgiler korunarak yeni bir deneyi gerçekleştirebilir konuma getirilecektir. Son olarak 4.Bölümde de, *SDL*'nin bir benzetim diline nasıl dönüşebileceği konusunda önerilerde bulunacaktır.

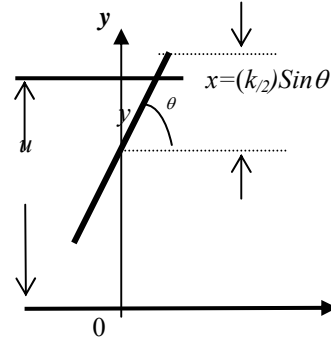
## **2. SANAL DENEY LABORATUARI'NIN TASARIMI VE İLK DENEYE HAZIRLANMASI**

İster sanal ister gerçek olsun, bir laboratuvarın kurulması öncesi düşünsel tasarımındaki ilk adım, söz konusu laboratuvarında gerçekleştirilecek deneyin tasarımıdır. Sanal bir deneyin tasarımına, söz konusu rassal olayın girdi ve çıktılarının dizgesel ilişkilerinin tanımlandığı

matematiksel modeli kurularak başlanır. Matematiksel ya da daha geniş anlamda simgesel bir modeli kurulamayan rassal olayların deney tasarımından söz edilemez. Dolayısı ile *SDL*'nin tasarımındaki ilk adım da ancak, *SDL*'de gerçekleştirilecek ilk sanal deneyin matematiksel modeli kurularak atılabilir.

## 2.1. *SDL*'nda Gerçekleştirilecek İlk Sanal Deneyin Matematiksel Modeli

Tasarımını yapacağımız *SDL*'nda gerçekleştireceğimiz ilk sanal deney olarak, Buffon<sup>1</sup>'un ünlü iğne deneyini seçtik. Şekil:1,  $k$  uzunluğundaki bir doğru parçasının, aralarında  $u$  kadar bir uzaklık bulunan paralel doğrulardan oluşan bir düzlem üzerine rasgele bir atış sonucunu göstermektedir. Doğru parçasının merkezi, bu düzlemdeki paralel doğrulardan bir birine komşu olan herhangi ikisinin arasına düşeceğinden, doğrunun konumu, yalnızca bu iki paralel doğruya göre tanımlanabilir. Doğru parçasının merkezinin daha uzak olduğu paralel doğru absis ve doğrunun merkezinden geçen absise dik eksen ordinat olmak üzere, doğru parçasının merkezinin konumu  $(0, y_i)$  dir ve  $y_i$ ,  $(0, u)$  aralığında tekdüze dağılımlı  $Y$  rassal değişkeninin her denemede gözlenen rassal bir değeridir. Doğru parçasının paralel doğruya yakın ucunun, doğru parçasının merkezinden geçen ve absise paralel bir doğruya uzaklığı olan



Şekil :1 Buffon 'un İğne Deneyi

$x_i = \frac{k}{2} \text{Sin}\theta_i$ ,  $0 \leq x_i \leq \frac{u}{2}$

doğru parçasının yatayla yaptığı  $\theta_i$  açısının bir işlevi, ve  $\theta_i$  de,  $(0, \pi)$  aralığında tekdüze  $\Theta$  rassal değişkeninin her denemede gözlenen rassal bir değeridir. Dolayısı ile, doğru parçasının her deneme sonundaki konumu rassal olarak  $y_i$  ve  $\theta_i$  değerleri ile tanımlanabilir. Doğru parçasının paralel doğrulardan yukarıdakini kesme durumunun,  $x_i > u - y_i$  ve aşağıdakini kesme durumunun da  $x_i > y_i$  koşulunun sağlanmasına bağlı olduğu görülürse, her hangi bir  $i^{\text{inci}}$  deneme sonucu,

<sup>1</sup>Bu deneyde, aralarında  $u$  kadar uzaklık bulunan paralel doğruların yer aldığı bir düzlemin üzerine  $u > k$  uzunluğunda bir doğru parçası rasgele atılır ve paralel doğrulardan birini kesip kesmediği gözlenir. Deneyin amacı, doğru parçasının paralel doğrulardan birini kesme olasılığının Monte Carlo tahminini elde etmek ve olayın kuramsal olasılık tanımından yararlanarak 'bir çemberin çevresinin çapına oranı olan'  $\pi$  parametresinin Monte Carlo tahminine dolaylı olarak ulaşmaktır (Bkz. Gnedenko, 1976:36-37). Bu olayın Monte Carlo benzetimi,  $\pi$  parametresi sanal deney modelinin girdilerinden biri olduğundan, ancak kesme olasılığının deneysel tahmini amacı ile yapılabilir. Bu nedenle Buffon'un iğne deneyine ilişkin olarak aşağıda verilen sanal deney modeli,  $\pi$  parametresinin Monte Carlo tahminine değil, kesme olasılığının deneysel tahminine yönelik olarak kurulmuştur.

$$w_i = \begin{cases} 1, & x_i > u - y_i \vee x_i > y_i \\ 0, & u - y_i \geq x_i \wedge x_i \leq y_i \end{cases} \quad (2)$$

biçiminde bir ikil-değişkenle ifade edilebilir. Bu olayın gerçek deneyinde,  $W$  rassal değişkeninin değeri doğrudan gözlenerek elde edilir. Gözlenen bu değer,  $y_i$  ve  $\theta_i$  değerleri doğrudan ölçülerek  $u$  ve  $k$  parametrelerinin değerleri ile birlikte, (1) ve (2) nolu ilişkilerde yerine konarak elde edildiği düşünülebilir. Sanal deneyde ise, gerçek deneyde doğrudan ölçülen değişkenlerin olasılık dağılımlarından rasgele örnekleme ile elde edilen  $y_i$  ve  $\theta_i$  değerleri  $u$  ve  $k$  parametrelerinin değerleri ile birlikte, (1) ve (2) nolu ilişkilerde yerine konarak sanal deneme sonucu belirlenir. Dolayısı ile sanal deneyin matematiksel modeli,  $Y$  ve  $\Theta$  rassal değişkenlerinin olasılık dağılımlarından rasgele örnekleme ile elde edilecek  $y_i$  ve  $\theta_i$  değerlerinin tanımlanmasını gerektirir. Her iki değişken de tekdüze dağılımlı olduğuna ve  $(a,b)$  aralığında tanımlı tekdüze dağılımlı  $X$  gibi rassal bir değişkenin,  $(0,1)$  aralığındaki birim tekdüze dağılımlı  $R$  gibi rassal bir değişkenden,

$$X=(b-a)R+a \quad (3)$$

dönüşümü ile elde edilebileceğine göre, Buffon'un sanal iğne deneyinin matematiksel modelinin tamamlanması için geriye, birim tekdüze dağılımdan rasgele örnekleme ile  $R$  rassal değişken değerlerinin nasıl seçileceğinin tanımlanması kalır. Alt-düzey programlama dillerinin hepsi de bu işlevi içerir. Ancak hangi algoritma ile elde edildikleri kullanıcı tarafından genellikle bilinmez. Tasarladığımız sanal deney laboratuvarında, her şeyin deneycinin denetiminde olabilmesi için, bir Monte Carlo benzetim dizgesinin en temel girdisi olan bu birim tekdüze rassal değişken değerleri üreticisine ilişkin algoritmanın da kullanıcıya açık olmasını gerektirir. Sanal deney laboratuvarının vazgeçilmez en temel aracı olan bu üretic, şimdilik

$$r_i = \left[ 23 \cdot 10^6 \cdot r_{i-1} \text{Mod}(10^6 + 1) \right] / 10^6 \quad (4)$$

biçiminde bir *doğrusal çarpımsal benzerlik* algoritması olabilir (Ackoff,1962:353). (4)'deki algoritmanın, farklı  $r_0$  başlangıç değerleri verildiğinde birbirinden bağımsız birim tekdüze dağılımlı değişken değerleri üreteceği varsayılabilir. Bu varsayımı daha güçlü biçimde sağlayan üreticiler geliştirilmiş ve geliştirilmeye devam etmektedir. Bu üreticilerin değerlendirilmesi ve en iyisinin seçimi benzetimcinin karar vereceği bir konudur. (4)'deki üretic, şimdiki amaç açısından, yeterli bir örnektir. Bu tanımların ışığında, Buffon'un sanal iğnesinin rasgele atışlarından oluşan sanal bir deneyde  $i^{\text{inci}}$  deneme sonucunu,  $W$  rassal değişkeninin sanal gözlem değeri olarak veren matematiksel modeli,

$$\begin{aligned}
r_i^{\theta} &= \left[ 23 \cdot 10^6 \cdot r_{i-1}^{\theta} \cdot \text{Mod}(10^6 + 1) \right] / 10^6 \\
r_i^y &= \left[ 23 \cdot 10^6 \cdot r_{i-1}^y \cdot \text{Mod}(10^6 + 1) \right] / 10^6 \\
\theta_i &= 3.141592654 r_i^{\theta} \\
y_i &= u \cdot r_i^y \\
x_i &= \frac{k}{2} \text{Sin} \theta_i \\
w_i &= \begin{cases} 1, & (x_i \geq u - y_i) \vee (x_i \geq y_i) \\ 0, & (y_i < u - x_i) \wedge (x_i < y_i) \end{cases}
\end{aligned} \tag{5}$$

biçiminde yazılabilir. Bu deneyin amacı,  $Ol(W=1)$  değerini tahmin etmek olduğuna göre, Monte Carlo tahmin,  $n^{inci}$  deneme sonucunu da içeren,

$$\bar{w}_n = \frac{1}{n} \sum_{i=1}^n w_i \tag{6}$$

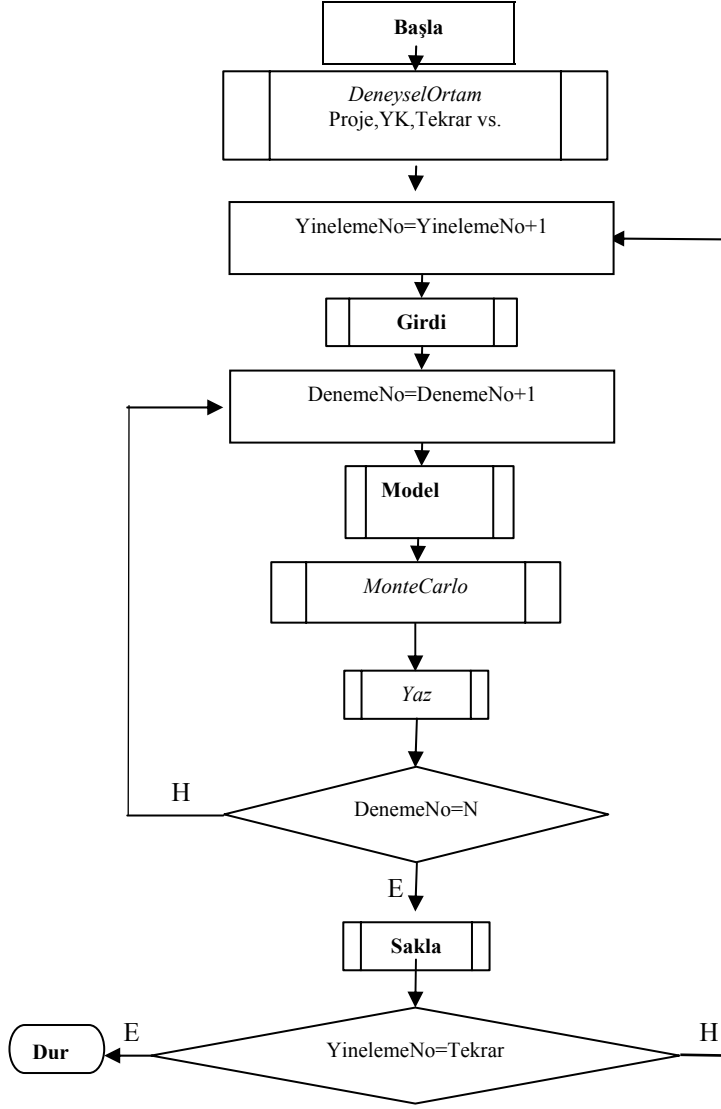
birikimli örnek istatistiğinin,  $N$  gibi yeteri kadar büyük bir sayıdaki deneme sonunda, Büyük Sayılar Yasası gereği

$$\underset{n \rightarrow N}{\text{erey}}(\bar{w}_n) \rightarrow \hat{Ol}(W=1)_{MC} \tag{7}$$

biçiminde yakınsayacağı bir değerdir. Buffon'un iğne deneyi örneğine ilişkin olarak (5), deney çıktısı sanal gözlemleri; (6), tahmin edilecek parametreye ilişkin Monte Carlo istatistiği; ve (7), Monte Carlo tahmini tanımlamaktadır.

## 2.2. SDL'nın Genel Yapısı

Matematikselsel tanımı yapılmış sanal deney çıktısından ardışık olarak hesaplanan birikimli örnek istatistiği deney süresince izlenerek, bu istatistiğin ilişkin olduğu parametrenin Monte Carlo tahmini elde edilebileceğine göre, bu işi başaracak yapısal bir bilgisayar programı, deney boyunca sabit kalacak girdi değerlerinin tanımının ardından gelen bir ana döngüden oluşmalıdır. Bu döngü içinde, yakınsama kıstası sağlanıncaya kadar sanal deney modelinden sanal gözlemler üretilerek birikimli örnek istatistiği hesaplanır ve gerekli bilgiler görüntülenir ya da daha sonra incelenmek üzere saklanır. Böyle bir yazılımın genel yapısı Şekil:2'deki gibi verilebilir.



Şekil:2 Sanal DeneY Laboratuvarı ana programı SDL'nın genel yapısı.

Sanal bir deney laboratuvarı olan *SDL* ana programı, Şekil:2'deki genel yapısı doğrultusunda, çalışmaya başladıktan sonra,

1. *DeneySELOrtam* adlı bir alt-yordamla, Proje Adı olarak, sanal deneyin her denemedeki sanal gözlem değerlerini üreten matematiksel modelinin tanımlandığı alt-yordam adını , sanal deneyin kaç kez yineleneyeceğini, ve yakınsama kıstası parametrelerini ister;
2. *Girdi* adlı bir alt-yordamdan, *SDL genel değişkenlerinin* başlangıç değerlerini, ve söz konusu sanal deneye özgün girdilerin tanımlandığı bir başka alt-yordama ulaşarak alır;
3. *Model* adlı bir alt-yordamdan söz konusu sanal deneye özgün modelin tanımlandığı bir başka alt-yordama ulaşarak her denemedeki sanal gözlem değerlerini üretir;
4. *MonteCarlo* adlı alt-yordamda yakınsama kıstası istatistiklerini hesaplar ve yakınsama gerçekleşmişse *Sakla* alt-yordamını çağırarak bir sonraki yinelemeye geçer;
5. *Yaz* adlı alt-yordamdan, söz konusu sanal deneye özgün çıktıların tanımlandığı bir başka alt-yordama ulaşarak her deneme sonundaki sanal deney çıktılarını görüntüler ve yakınsama gerçekleştiğinde de,
6. *Sakla* adlı alt-yordamla, deney sonuçlarının söz konusu sanal deneye özgün biçimde tanımlandığı bir başka alt-yordama ulaşarak görüntüler ve bir sonraki yinelemeye geçer ve

deneyi belirlenen sayıda yineledikten sonra durur.

Farklı yapıdaki sanal deneyleri ortak bir çerçeve içinde gerçekleştirmek üzere tasarlanan *SDL*'nin genel yapısı içindeki *Girdi*, *Model*, *Yaz* ve *Sakla* alt-yordamları, her sanal deneyin kendisine özgün girdilerinin, modelinin ve çıktılarının tanımlandığı alt-yordamları çağırarak görevlidir. *MonteCarlo* alt-yordamı ise, *SDL*'nin kapsama alanındaki tüm sanal deneyler için geçerli ortak işlemlerin tanımlandığı bir ana alt-yordamdır. *SDL*'nin bu genel yapısı, ancak bir benzetim dilinin kendine özgü olabilecek gramer kurallarından sayabileceğimiz iki ilkenin baştan konma zorunluluğunu gerektirir:

- Her sanal deneyin kendine özgü girdilerinin, modelinin ve çıktılarının tanımlanacağı alt-yordamların adları, proje adı ile ulaşılabilir olacak adlar olmalı,
- ana yordam düzeyinde ortak paylaşımlı olarak tanımlanabilmeleri için, ana yordam ve alt-yordamlar arasında değerleri zorunlu olarak taşınacak değişkenlerin adları aynı olmalıdır.

Bu ilkelerin belirgin gramer kurallarına dönüşmesi, *SDL*'nin uyarlanacağı programlama ortamı seçildikten sonra olabilir. *SDL*'nin bir BASIC programlama ortamındaki uyarlanmasına göre bu iki kuralı şimdilik benimseyeceğiz ve çok daha uygun bir programlama ortamına karar vermeden *SDL*'nin kendisine özgü bu gramer kurallarını arttırmayacağız.

### 2.3. *SDL*'nin Temel Bileşenleri

*SDL*'nin temel bileşenleri, Şekil:2'deki genel yapı içinde yer alan,

- *DeneySELOrtam* alt-yordamı,
- *Girdi* alt-yordamı,
- ana program düzeyindeki döngü,



- her denemede *SDL* ana programı tarafından çağrılan *Model* alt-yordamı,
  - *MonteCarlo* alt-yordamı,
  - *Yaz* ve *Sakla* alt-yordamları ile
  - sanal deneyin matematiksel modelindeki *doğrudan gözlenme durumundaki* rassal değişkenlerin sanal değerlerini, olasılık dağılımlarından rasgele örnekleme ile üretecek olan *SDL işlevleri* niteliğindeki alt-yordamlardan oluşur.
- SDL*'nin bu temel bileşenlerini, BASIC programlama ortamında ve 2. Bölümde tanımlanan sanal deney çerçevesinde adım adım oluşturacağız.

### 2.3.1. *SDL İşlevleri*

(5)'in ilk iki satırı, (4)'deki ortak algoritmaya göre birbirinden bağımsız  $R^Y$  ve  $R^\Theta$  rassal değişkenlerinin sanal değerlerini; 3. ve 4. satırları ise  $Y$  ve  $\Theta$  rassal değişkenlerinin sanal değerlerini (3)'teki ortak algoritmaya göre üretecektir. Dolayısı ile, (3) ve (4) ortak algoritmalarını, birer alt-yordam olarak sanal deney laboratuvarının temel araçları arasında tasarlamak uygun olur. Bu algoritmalar, *SDL*'nin en temel araç ve gereçlerinin başta gelen *belli olasılık dağılımlarından rasgele örnekleme işlevlerinden*, yalnızca ilk iki tanesidir. *SDL*'nda gerçekleştirecek tüm sanal deneylerin matematiksel modellerinde yer alacak temel girdi niteliğindeki rassal değişken değerlerinin rassal değişkenin olasılık dağılımından rasgele örnekleme ile elde ederek sanal değerlerini üretecek bu işlevler, üzerinde çalışılan sanal deney projelerinin gereksinimi doğrultusunda zamanla zenginleşecektir.

*SDL*'nin sahip olması gereken ilk *işlev*, (4)'e göre birim tekdüze dağılımlı  $R$  rassal değişkeninin bir önceki değerinden bir sonraki değerini üretmektir. Bu işlev, *SDL* içinde bir *RAS* adı verilen bir alt-yordamda bir kez tanımlanırsa, *SDL*'nin içinde (4)'deki işlem gerektiği yerde her zaman bir tek sözcükle anlatılabilir konuma gelir. Bu açıdan bakıldığında, *SDL* bir dil; *RAS* da bu dilin içinde özel anlamı olan bir sözcük gibidir. (4)'deki üreteç BASIC yazım kurallarına uygun olarak, şimdilik (8)'deki gibi yazılabilir.

```
FUNCTION RAS (R#)
RAS = (23 * R# * 10 ^ 6 MOD (10 ^ 6 + 1)) / 10 ^ 6
END FUNCTION
```

(8)

Yapısal programlama yaklaşımı, bir yandan (3) ve (4)'deki gibi standartlaştırılabilen işlemleri ayrı birer alt-yordam olarak ele almayı ve değişken değerlerinin ana-program ve alt-yordamlar arasındaki paylaşımında ileride sorunlarla karşılaşmamak için, değişken adlarının verilmesinde de olabildiğince standart tanımların yapılmasını gerektirir. Örneğin, (4)'deki ortak algoritmada birim tekdüze dağılımlı rassal değişken adı olarak  $R$  kullanılırsa, sanal deney modelindeki birbirinden bağımsız birim tekdüze değişkenlere  $R1$ ,  $R2$ , . . . adları verilebilir. Böylece, *RAS* sözcüğü ile, birbirinden bağımsız birim tekdüze dağılımlı  $R1$ ,  $R2$  rassal değişkenlerinin sanal değerleri, örneğin olabildiğince matematik diline yakın

```
R1 = RAS (R1#)
R2 = RAS (R2#)
```

(9)

biçiminde, *SDL*'nin gramerine uygun, 'sözcüklerle belli bir eylem tanımlayan standart tümceler ve deyimler' kullanılarak kolayca üretilebilirler. Benzer biçimde, (3)'deki işlem de *SDL* içinde, *STD* sözcüğü ile, BASIC ortamında

```

FUNCTION STD (A, B, R#)
STD = (B - A) * R# + A
END FUNCTION

```

(10)

biçiminde *SDL*'ye özgü bir işlev olarak tanımlanabilir. Eğer  $X$ ,  $(A1, B1)$  aralığında; ve  $Y$ ,  $(A2, B2)$  aralığında tanımlı birbirlerinden bağımsız tek düze dağılımlı iki rassal değişken ise,  $X1$  ve  $X2$  rassal değişkenlerinin sanal gözlem değeri, *SDL*'nin *STD* işlevi kullanılarak (11)'deki gibi üretilirler.

```
X1 = STD(A1, B1, R1#) : X2 = RAS(A2, B2, R2#)
```

(11)

### 2.3.2. Model Alt-Yordamı

*Model* alt-yordamının, her sanal deneyin belirlenen bir proje adı ile aynı adlı ve sanal deneyin matematiksel modelinin tanımlandığı bir başka alt-yordamı çağırarak görevli bir alt-yordam olduğu söylenmişti. Şimdilik, tek bir proje bulunduğunu ve adının SanalDeney\$ adlı bir değişkene "Buffon" olarak yüklendiğini varsayarak *Model* alt-yordamı

```

SUB Model STATIC
SELECT CASE SanalDeney$
CASE "Buffon"
CALL Buffon
CASE ELSE
PRINT SanalDeney$;" adlı bir proje yok."
STOP
END SELECT
END SUB

```

(12)

Biçiminde oluşturulabilir. Yeni eklenen *RAS* ve *STD* sözcükleri ve BASIC dilinin grameri kullanılarak (5)'deki sanal deney modelinin tanımlandığı *Buffon* alt-yordamı ise,

```

SUB Buffon
R1#=RAS(R1#) : TETA=STD(0, PI, R1#) : R2#=RAS(R2#) : Y=STD(0,
A, R2#)
X = (B / 2) * SIN(TETA)
IF X > A - Y OR X > Y THEN
Y = 1
ELSE
Y = 0
END IF
END SUB

```

(13)

*Buffon* sanal deneyine özgü matematiksel modeli *Buffon* adlı alt-yordamda tanımlarken, model parametrelerine karşılık gelen değişkenler için A, B; bir dairenin çevresinin çapına olan oranı biçimindeki evrensel sabit için PI; her denemedeki sanal gözleme karşılık gelen değişken için de Y adları kullanıldı. *SDL*'de gerçekleştirilecek diğer sanal deneylerin matematiksel modellerinde de bu adların karşılıkları aynı kalacak biçimde kullanılırsa, bu

değişken adları ana program düzeyinde ortak paylaşımli olarak tanımlanabilir. Bu kural, daha öncede değinildiği üzere, alt-yordamlar arasında sürekli paylaşılma durumunda olan değişkenlerle ilgili düzenlemeleri, her sanal deneye özel olarak yapmak yerine, *SDL*'de gerçekleştirilecek tüm sanal deneylere genelleştiren *SDL*'ye özgü zorunlu gramer kurallarından ilki olacaktır. Bu kurala, şimdilik Buffon sanal deneyindeki biçimi ile uyacağız. Buna göre ana-yordam düzeyinde yapılacak gerekli düzenleme ile ilgili açıklama, *SDL*'nin temel bileşenleri oluşturulduktan sonra ana-yordamı verilirken yapılacaktır.

### 2.3.3. Girdi Alt-Yordamı

*Girdi* alt-yordamı da, *Model* alt-yordamı gibi, her sanal deneyin kendisine özel girdilerinin tanımlandığı bir başka alt-yordamı çağırarak görevli bir alt-yordamdır. *Girdi* alt-yordamının çağıracağı her sanal deneye özel girdilerin tanımlandığı bu alt-yordamlara da proje adı ile ilintili belli bir kurala göre ad vermek gerekir. Her sanal deneyin matematiksel modelindeki parametre değerlerini modelinin yer aldığı alt-yordam yerine ayrı bir alt-yordamda verilmesinin nedeni, model parametrelerinin değerleri sanal deney tasarım noktalarına göre değişebilmesi ve kimi sanal deneylerde model parametrelerinin de rassal değişken olabilmesidir. Sanal deneye özgü girdilerin tanımlandığı alt-yordam adını, proje adının sonuna "G" ekleyerek türeteceğiz.

*Yaz* ve *Sakla* alt-yordamlarına da uyarlanacak bu kuralla, ileride *SDL*'nin deneyimi arttıkça sayıları da artacak bu tür alt-yordamları zamanı geldiğinde dosyalara dönüştürmek ve bu dosya adlarına, proje adını anahtar sözcük olarak kullanarak bu amaç için düzenlenmiş bir dizin dosyasından ulaşabilmek için program bütününde yapılacak düzenlemeyi kolaylaştırmış olacağız. *SDL*'nin deneyim düzeyi, tüm deneyler için belirli veri yapılarını genelleyecek yeterliğe ulaşmaya kadar, sanal deneyin girdi ve çıktılarının yan bellek yerine ana bellek üzerinden erişimi daha uygundur. Bu nedenle, her sanal deneye özel olan girdi ve çıktı alt-yordamları, şimdilik veri dosyalarına başvurmadan, klavye, ekran ya da yazıcı gibi doğrudan çevre birimlerine yönelik olarak oluşturulacaktır. Bu açıklamaların ışığında, *Girdi* alt-yordamı, ve *Girdi* alt-yordamının çağıracağı *BuffonG* alt-yordamı (14) ve (15)'deki biçiminde oluşturulabilir.

```
SUB Girdi
N = 1000000:ToplamY# = 0 :ToplamKY# = 0:StHataMCist# = 0
R1# = .1636143 * YinelemeNO:R2# = .3012415 * YinelemeNO
SELECT CASE SanalDeney$
  CASE "Buffon"
    CALL BuffonG (14)
  CASE ELSE
    PRINT SanalDeney$;" adlı projenin girdileri tanımlı
değil."
  STOP
END SELECT
END SUB
```

```
SUB BuffonG (15)
A = 10: B = 5
END SUB
```

### 2.3.4. MonteCarlo Alt-Yordamı

*MonteCarlo* alt-yordamı, *Model* alt-yordamında sanal gözlem değerlerinden hesaplanan birikimli bir örnek ortalaması olan Monte Carlo istatistiğini deney boyunca izleyerek, belli bir yakınsama kıstasına göre Monte Carlo tahmine ulaşıldığında deneye son verecektir. Akla ilk gelebilecek böyle bir kıstas, Monte Carlo istatistiğinin ardışık değerlerinin, rassal bir sürecin  $n$  deneme boyunca gerçekleşmiş bir biçimi olduğu gerçeğine ve dolayısı ile Merkezi Limit Teoremine doğrudan dayandırılabilir.  $\varepsilon$  gibi bir Monte Carlo tahmin hata değeri ile  $\alpha$  gibi, tahmin edilecek parametrenin bu hata miktarına göre sınırları belirlenecek bir tahmin aralığında olmama olasılığı ilişkin değer, benzetimci tarafından sanal deney öncesinde belirlenecek böyle bir kıstasın, iki parametresi olacaktır. Böyle bir kıstasa göre, Monte Carlo istatistiğinin ardışık her değerlerine göre süreç düzeyi için ardışık güven aralığının alt ve üst sınırlarının hesaplanması gerekir. Süreç değişkesi bilinmediğine göre, böyle süreç biçiminde bir güven aralığı oluşturabilmek için, tahmin edilmeye çalışılan parametre yanında, süreç değişkesinin de Monte Carlo tahminine gerek vardır. Dolayısı ile bu, birikimli örneğin standart sapmasının da örnek ortalaması ile birlikte adım adım hesaplanmasını gerektirir.

Kabul edilebilir güven aralığı yarı genişliği, KGAYG; her deneme için hesaplanan güven aralığı genişliği, GAYG; standart normal değişken için  $\%100(1-\alpha)$ 'lık güven aralığının üst sınırı ZALFA değişkenlerinin değerleri *MonteCarlo* alt-yordamına, söz konusu sanal deney girdilerinin tanımlandığı alt-yordamdan aktarılabilir ya da kullanıcı tarafından proje adı ile birlikte girilebilir. Ana program düzeyinde hepsinin de ortak paylaşımlı olarak tanımlanacağı, MC örnek çapına karşılık olarak ana yordamdan gelen deneme sıra numarası I; *Buffon* alt-yordamından gelen sanal deney gözlem değeri Y; kullanıcı tarafından ana-yordam düzeyinde proje adı ile birlikte girilen yakınsama kıstası parametreleri KGAYG, GAYG, ZALFA; başlangıç değerleri *Girdi* alt-yordamında verilen birikimli toplam değişkenleri ToplamY#, ToplamKY#, ve MC tahmin MCist# olmak üzere, *MonteCarlo* alt-yordamı (16)'daki biçiminde oluşturulabilir. Buna göre Monte Carlo istatistik değerlerinden hesaplanacak güven aralığı yarı genişliği olan GAYG, öngörülen Monte Carlo tahmin hatasına ilişkin *Kabul edilebilir Güven Aralığı Yarı Genişliği* olan KGAYG değerine indiğinde, yakınsama sağlanmış olacak ve *MonteCarlo* alt-yordamı program denetimini *Sakla* adlı alt-yordama bırakacaktır. Monte Carlo tahmin değeri olarak en son denemedeki birikimli örnek ortalamasının değeri ve yakınsamanın gerçekleştiği deneme sayısı, MCist# ve N1 değişkenleri ile iki alt-yordam arasında paylaşılırak *MonteCarlo* alt-yordamından *Sakla* alt-yordamına aktarılır. Yakınsama gerçekleşinceye kadar her deneme sonundaki ekranda görülmeye değer bulunan istatistikler ise, *Yaz* alt-yordamı ile görüntülenir.

```

SUB MonteCarlo STATIC
ToplamY# = ToplamY# + Y:ToplamKY# = ToplamKY# + Y ^ 2
OrtalamaY# = ToplamY# / I:OrtalamaKY# = ToplamKY# / I
DegiskeY# = OrtalamaKY# - OrtalamaY# ^ 2:MCist# =
OrtalamaY#
StHataMCist# = SQR(DegiskeY# / I)
GAYG = ZALFA * StHataMCist#
IF I > 100 AND KGAYG >= GAYG THEN

```

(16)

```

N1 = I
CALL Sakla
I = N
END IF
END SUB

```

### 2.3.5. Yaz ve Sakla Alt-Yordamları

*Girdi* alt-yordamında olduğu gibi, *Yaz* ve *Sakla* alt-yordamları da, her sanal deneyin kendisine özel çıktılarının tanımlandığı bir başka alt-yordamı çağırarak görevli birer alt-yordamdır. *Yaz* ve *Sakla* alt-yordamlarının çağıracağı her sanal deneye özel çıktıların tanımlandığı bu alt-yordamlara verilecek adları da, *Girdi* alt-yordamı oluşturulurken belirlenen kurala göre, proje adının sonuna “Y” ve “S” ekleyerek türeteceğiz. *BuffonY* alt-yordamını çağırarak olan *Yaz* alt-yordamı, ve *BuffonY* alt-yordamı

```

SUB Yaz
SELECT CASE SanalDeney$
  CASE "Buffon"
    CALL BuffonY
END SELECT
END SUB

```

(17)

```

SUB BuffonY
LOCATE 4, 1:PRINT YinelemeNO; ". Yineleme "
LOCATE 4, 18:PRINT USING "#####"; I;
PRINT ". Denemede MC Güven Aralığı :";
PRINT USING "####.#####"; MCist#;
PRINT " "; CHR$(241); : PRINT USING "####.#####";
GAYG
END SUB

```

(18)

biçimlerinde oluşturulabilir. *Yaz* alt-yordamının görevi deney süresince, her deneme sonucuna ilişkin bilgilerden benzetimcinin istediklerini görüntülemektir. Bunların başında, deneme sıra numarası ve söz konusu deneme sonunda hesaplanan Monte Carlo istatistiğinin değeri ile güven aralığı yarı genişliği gelebilir. Deney boyunca bu değerler ekranda aynı konuma yazdırılırsa, giderek değişmez olan hanelerden görsel olarak yakınsama süreci izlenebilir. *BuffonY* alt-yordamının (18)'deki uyarlaması, *SDL* genel değişkenlerinden deneyin YinelemeNO I'nın değerini ana-yordamdan; MCist# ve GAYG değişkenlerinin değerlerini de MonteCarlo alt-yordamından alarak, her deneme sonunda ekranda aynı konuma yazdırır.

Benzer biçimde, (19)'daki gibi *Sakla* alt-yordamı, (20)'deki gibi de *BuffonS* alt-yordamı oluşturularak, deneyin her yinelemesi sonunda ekranda görüntülenecek bilgiler biçimlendirilebilir. *BuffonS* alt-yordamı, ana-yordam düzeyinde tanımlı YinelemeNO ve *MonteCarlo* alt-yordamında tanımlı yakınsamanın sağlandığı MC örnek çapına ilişkin N1 ve MC tahmine ilişkin MCist# adlı *SDL* genel değişkenlerinin her yineleme sonunda elde edilen değerlerini, ekranda belirlenen konumdan başlayarak alt alta yazar.

```

SUB Sakla
SELECT CASE SanalDeney$
  CASE "Buffon"
    CALL Buffons
END SELECT
END SUB

```

(19)

```

SUB Buffons
IF SUTUN = 0 THEN SUTUN = 1
LOCATE 6 + SATIR, SUTUN:PRINT USING "###"; YinelemeNO;
PRINT USING "#####"; N1;:PRINT USING "####.#####";
MCist#
IF YinelemeNO = 15 * INT(YinelemeNO / 15) THEN
  SATIR = 0: SUTUN = SUTUN + 20
ELSE
  SATIR = SATIR + 1
END IF
IF YinelemeNO = 60 * INT(YinelemeNO / 60) THEN
  SATIR = 0: SUTUN = 0
END IF
END SUB

```

(20)

Buffon sanal deneyini gerçekleştirmek üzere temel bileşenleri oluşturulan *SDL*'nin Şekil.2'de verilen genel yapısı izlenerek, evrensel sabit  $\pi$ 'nın ve *SDL* genel değişkenlerinin ortak paylaşımlı tanımları ile başlayan ana-yordamı (21)'de olduğu gibi yazılabilir ve ilk deneyi gerçekleştirmeye artık hazır olan *SDL*, derlendikten sonra ilk sınamadan geçirilebilir.

```

'SDL Ana Yordamı
COMMON SHARED A, B, I, N,Y,N1, R1#, R2#,GAYG,KGAYG,ZALFA,
YinelemeNO
COMMON SHARED SanalDeney$,MCist#, StHataMCist#, ToplamY#,
ToplamKY#
CONST PI = 3.1415192546#
CALL DeneyselOrtam
FOR YinelemeNO = 1 TO Tekrar
  CALL Girdi
FOR I = 1 TO N
  CALL Model
  CALL MonteCarlo
  CALL Yaz
NEXT I
  CALL Sakla
NEXT YinelemeNO
END

```

(21)

### 3. SDL’NİN GÜVENİLİRLİĞİ VE YENİ BİR DENEY İÇİN HAZIRLANMASI

*SDL*’nin ne kadar güvenilir olduğunu anlamanın en kestirme yolu, kesin olarak doğru sonucun bilindiği bir durumda onu sınamaktır. (15)’deki *BuffonG* alt-yordamında paralel doğrular arasındaki uzaklık  $A = 10$  ve doğru parçasının boyu  $B = 5$  olarak verilmişti. Eğer *SDL* doğru iş yapıyorsa, ilk deneyin yinelenen sonuçları, tanımlanan deneysel ortamda doğru parçasının paralel doğrulardan birini kesme olasılığı olan

$$Ol(W = 1) = \frac{2B}{\pi A} = 0.318309886$$

değerine çok yakın değerler vermeli, ya da deneysel olarak oluşturulan güven aralıklarının bu değeri kapsama oranı güven düzeyine yakın olmalıdır.

SanalDeney\$ = “Buffon”, YinelemeNO = 60, KGAYG = 0.01 ve ZALFA = 1.96 olmak üzere çalıştırılan *SDL*, deney sonunda Şekil.3’de yer alan değerleri ekranda görüntülemiştir. Buffon sanal deneyinin 60 kez yinelenmesi sonunda, 5 birim uzunluğundaki bir doğru parçasının, birbirinden 10 birim uzaklıktaki paralel doğrulardan birini kesme olasılığına ilişkin MC tahmin değeri 0.3171191 olarak bulunmuş ve 60 yinelemenin her birinde elde edilen MC tahmin değerlerinin tanımladığı %95’lik güven aralıklarının %96.7’si gerçek olasılık değerini kapsamıştır. Gerçekleşen güven düzeyi ile öngörülen güven düzeyinin hemen hemen aynı çıkması, yakınsama kıstasının başarısı yanında *SDL*’nin de ilk güvenilirlik sınavından başarı ile geçtiğinin kanıtıdır.

Deney :Buffon		Kabul edilebilir GAYG :0.01									
Yineleme Sayısı :60		St.Normal GA üst sınırı :1.96									
1	8357	0.319732	16	8285	0.314665	31	8269	0.313581	46	8303	0.315910
2	8363	0.320220	17	8198	0.308612	32	8300	0.315663	47	8354	0.319607
3	8351	0.319363	18	8244	0.311742	33	8324	0.317395	48	8306	0.316157
4	8350	0.319281	19	8305	0.316075	34	8366	0.320464	49	8268	0.313498
5	8267	0.313415	20	8349	0.319200	35	8349	0.319200	50	8274	0.313875
6	8250	0.312242	21	8395	0.322573	36	8339	0.318503	51	8352	0.319444
7	8412	0.323823	22	8379	0.321399	37	8311	0.316448	52	8357	0.319732
8	8316	0.316859	23	8288	0.314913	38	8195	0.308481	53	8238	0.311362
9	8226	0.310601	24	8347	0.319037	39	8238	0.311362	54	8371	0.320750
10	8322	0.317231	25	8373	0.320912	40	8207	0.309248	55	8302	0.315828
11	8358	0.319813	26	8405	0.323260	41	8547	0.334035	56	8307	0.316239
12	8312	0.316530	27	8246	0.311909	42	8386	0.321846	57	8291	0.315040
13	8311	0.316448	28	8336	0.318258	43	8358	0.319813	58	8372	0.320831
14	8253	0.312371	29	8380	0.321480	44	8439	0.325749	59	8321	0.317149
15	8354	0.319607	30	8250	0.312242	45	8268	0.313498	60	8256	0.312621
Monte Carlo Tahmin : 0.3171191											
Güven Aralıklarının Beklenen Değeri Kapsama Oranı : 0.96666666											

#### Şekil.3 *SDL*’nin güvenilirliğini sınamak için gerçekleştirilen bir deneyin sonuçları.

İlk güvenilirlik sınavından başarı geçen *SDL*’nin, yeni bir deneye hazırlanmasında mantıksal olarak ilk iş, yeni deneyin matematiksel modelini uygun bir proje adı verilerek

sanal deneye özel model alt-yordamına yerleştirmek ve bunun zorunlu sonucu olarak söz konusu sanal deneye özel *SDL*'nin diğer girdi ve çıktı bileşenlerini de birlikte oluşturmaktır. Eğer yeni deneyin çıktı alt-yordamları, ilk deneyin çıktı alt-yordamlarının birer kopyası olacaksa, ilk deneye ilişkin (18) ve (20)'deki *BuffonY* ve *BuffonS* alt-yordamlarının adlarını örneğin *Standart1Y* ve *Standart2S* olarak değiştirerek, (17) ve (19)'daki *Yaz* ve *Sakla* alt-yordamlarını, yeni sanal deneye proje adı olarak "SD02" verildiğini varsayarak,(22) ve (23)'deki biçime dönüştürürüz.

```

SUB Yaz
SELECT CASE SanalDeney$
  CASE "Buffon", "SD02"
    CALL Standart1Y
  CASE
    PRINT SanalDeney$;" adlı projenin çıktıları tanımlı
değil."
  STOP
END SELECT
END SUB

```

(22)

```

SUB Sakla
SELECT CASE SanalDeney$
  CASE "Buffon", "SD02"
    CALL Standart1S
  CASE ELSE
    PRINT SanalDeney$;" adlı projenin çıktıları tanımlı
değil."
  STOP
END SELECT
END SUB

```

(23)

Yeni deney, olasılık dağılımlarından rasgele örnekleme işlevleri tanımlanmamış yeni rassal değişkenler içerebilir. *SDL*'ye bu işlevlerin de eklenmesi ile *SDL* sözlüğü giderek zenginleşir. Kuşkusuz, yeni sanal deney modeli, *SDL genel değişkenleri* arasına yeni evrensel sabitlerin ve parametrelerin alınmasını gerektirebilir. Bu değişken adlarının, *SDL* ana-yordamında ortak paylaşımlı değişkenlerin tanımlandığı satırlara belirli bir disiplin içinde eklenmesinden sonra *SDL*, yeni deneyi de gerçekleştirecek duruma gelir.

EK'de verilen *SDL*'nin son biçiminde, *Standart1Y* ve *Standart1S* alt-yordamlarındaki çıktı yapısını Buffon sanal deneyi ile ortak paylaşan bir ikinci sanal deney de kapsamıştır. SD02 proje adı ile *SDL* kapsamına giren bu sanal deneyle ilgili kısa açıklayıcı bilgiler ilgili alt-yordamlarda verilmiştir. SD02 deneyi ile, *SDL sözlüğüne* NRM ve STNRM işlevleri; *SDL genel değişkenleri* arasına C, D ve BeklenenDeger eklenmiş; (20)'de tanımlananlara ek olarak *Standart1S* alt-yordamı, deneyin yinelenmesi tamamlandığında MC tahmininin tüm deney için genel ortalamasını ve her yinelenme sonunda hesaplanan %100(1-ZALFA)'lık güven aralıklarının sanal deneye özel girdi alt-yordamında tanımlanan BeklenenDeger'i kapsama oranını da verecek biçimde düzenlemiştir. Her yeni deney modeli için ve her yeni



eklentiden sonra 3. Bölümde sözü edilen türde bir güvenilirlik sınavasından doğal olarak yeniden geçirilmesi gerekecektir.

#### 4. SDL’NİN GELECEĞİ

Burada ana çizgileri ve yalnızca bir başlangıç biçimi ile verilmeye çalışılan *SDL*, Giriş Bölümü’nde belirtildiği üzere, sanal deney tekniklerinin eğitim ve öğretiminde benimsenen belirli bir yaklaşımın gereği oluştu. Şimdiki durumu ile *SDL*, kendi bölümünde yaklaşık yirmi yıldır sanal deney tekniklerinin eğitim ve öğretimini üstlenmiş, hemen her alanda bu teknikleri kullanan çok geniş bir uzmanlar kümesinin deneyimine ve bir bilişimciye göre, yazılım bilgi ve becerisi sınırlı bir istatistikçinin girişimidir. Günümüzde, tek bir kişinin bilgi ve becerisi ile sınırlı *SDL* benzeri düşünce ürünlerinin, zaman içinde gelişerek yaşama şansları yoktur. *SDL*’nin bu şansı yakalayabilmesi, onun oluşumuna yol açan yaklaşımı benimseyerek böyle bir düşünce ürününün aynı zamanda doğal üretim ve tüketim ortakları olacak uzmanların işbirliğine bağlıdır. Böyle bir proje ortaklığı, bilişim teknolojisinin sağladığı olanaklarla, gönüllülük ve burada olduğu gibi *açık yazılım* temeli üzerinde oluşturulabildiğinde, *SDL* çok kısa bir sürede gelişerek, erişimi ve eğitimi kolay, yaygın kullanımlı bir benzetim diline dönüşecektir. *SDL* girişiminin önü tıkanmadan başarı ile sürdürülebilmesi için, piyasadaki yüzlerce benzetim dillerinden bir veya bir kaçının kuyruğuna takılmadan, çok çeşitli ve farklı deney yapılarında deneyim kazanılncaya kadar,

- sanal deneylerin matematiksel modelini, girdilerini ve çıktılarını tanımlamada kolaylık sağlayacak alt-yordamlar ve *SDL işlevleri*, *SDL*’nin geliştirildiği genel amaçlı programlama diline bir *program kütüphanesi* olarak eklenmelidir.
- özellikle matematiksel modelin, doğrudan matematiksel ifadelerle ve bir metin düzenleyici ile yazılıp-okunmasını sağlayacak bir ara-yüz yazılımının tasarımına asla çaba harcanmamalıdır. Böyle bir çaba, *SDL*’nin kendine özgü bir gramerinin olmasını gerektirir ve erken bir aşamada gramer kuralları devreye sokulursa, *SDL* tıkanabilir. Bu iş, en son karar verilerek gerçekleştirilmesi gereken işlerdendir.
- sanal deneylere özel model, girdi ve çıktı alt-yordamlarının da program kütüphanesine eklenmesi yoluna gidilerek, bu alt-yordamların ayrı dosyalarda saklanması, erişilmesi, ve içerdikleri bilgilerin *SDL* içinde yorumlanması düşünülmemelidir.
- model tanımlarken sıkça karşılaşılan örneğin (13)’teki gibi koşullu bir *Y* değişkenini, *SDL*’ işlevleri arasına,

```
FUNCTION EgerVeya (A, B, C, D)
IF X > A OR X > B THEN
  Y = C
ELSE
  Y = D
END IF
EgerVeya = Y
END FUNCTION
```

işlevini ekleyerek,

```
A = A - Y : Y = B
Y = EgerAveyaB (A, B, 1, 0)
```

biçimine indirgeyen bir öneriyi, *SDL*'nin gramer kurallarını tartışırken bir seçenek olarak göz önüne alınmak üzere, arşivde tutmak da gerekir. Bu tür öneriler arasından hangilerinin, *SDL* gramerine ileride temel oluşturabileceğine kullanıcıların ve üreticilerin oyuyla karar verilebilir.

*SDL* program kütüphanesi, alt kütüphanelere bölünmeyi gerektirecek boyuta eriştiğinde,

- benzer amaçlı alt-yordamlar arasından en uygunları ayıklanarak, sanal deney havuzundaki tüm deneylerin gerçekleştirilebildiği bir alt-yordam uzayı oluşturulur ve
- bu sözlükteki sözcüklerin oluşum kuralları ile başlayan ve *SDL*'nin geliştirildiği genel amaçlı programlama dilinin sözlüğü ve grameri kullanılarak ifade edilebilen her tür işlemi, bu sözcüklerle *kendine özgü yapılarda ifade etme* ve *yorumlama* kuralları adım adım oluşturulmaya başlanabilir.

*SDL* grameri tamamlandıktan sonra, son aşama olarak metin düzenleyici ile okuma-yazma ve grafik ara-yüzü tasarımları ve uyarlamaları başlayabilir.

#### KAYNAKLAR

Gnedenko, B.V. (1976), *The Theory of Probability*, Mir Publishers, Moscow.  
Ackoff, Russell L. *Scientific Method*, John Wiley and Sons, Inc., New York, 1962.

EK: SDL

#### Ana-yordam

```

DECLARE SUB DeneyselOrtam ()
DECLARE SUB Girdi ()
DECLARE SUB Model ()
DECLARE SUB MonteCarlo ()
DECLARE SUB Yaz ()
DECLARE SUB Standart1Y ()
DECLARE SUB Sakla ()
DECLARE SUB Standart1S ()
DECLARE FUNCTION RAS! (R#)
DECLARE FUNCTION STD! (a!, B!, R#)
DECLARE FUNCTION NRM! (Ortalama!, Degiske!)
DECLARE FUNCTION STNRM! (R11#, R22#)
DECLARE SUB BuffonG ()
DECLARE SUB Buffon ()
DECLARE SUB SD02 ()
DECLARE SUB SD02G ()
COMMON SHARED
A, B, C, D, I, N, Y, N1, R1#, R2#, Tekrar, GAYG, KGAYG, ZALFA YinelemeNO
COMMON SHARED
BeklenenDeger, MCist#, ToplamY#, ToplamKY#, StHataMCist#
COMMON SHARED SanalDeney$
CONST PI = 3.1415192546#
CALL DeneyselOrtam

```

74

```
FOR YinelemeNO = 1 TO Tekrar
CALL Girdi
FOR I = 1 TO N
    CALL Model
    CALL MonteCarlo
    CALL Yaz
NEXT I
    CALL Sakla
NEXT YinelemeNO
END
```

### Alt-yordamlar

```
SUB DeneyselOrtam
COLOR 5, 0: LOCATE 1, 1: PRINT "Deney          ";
COLOR 7, 0: INPUT "", SanalDeney$
COLOR 5, 0: LOCATE 2, 1: PRINT "Yineleme Sayısı :";
COLOR 7, 0: INPUT "", Tekrar
COLOR 5, 0: LOCATE 1, 40: PRINT " Kabul edilebilir GAYG  ";
COLOR 7, 0: : INPUT "", KGAYG
COLOR 5, 0: LOCATE 2, 40: PRINT " St.Normal GA üst sınırı :";
COLOR 7, 0: INPUT "", ZALFA
END SUB
```

```
SUB Girdi
ToplamY# = 0: ToplamKY# = 0: StHataMCist# = 0: N = 1000000
R1# = .1636143 * YinelemeNO / Tekrar: R2# = .30124 *
YinelemeNO / Tekrar
SELECT CASE SanalDeney$
CASE "Buffon"
CALL BuffonG
CASE "SD02"
CALL SD02G
END SELECT
END SUB
```

```
FUNCTION RAS (R#)
RAS = (23 * R# * 10 ^ 6 MOD (10 ^ 6 + 1)) / 10 ^ 6
END FUNCTION
FUNCTION STD (A, B, R#)
STD = (B - A) * R# + a
END FUNCTION
```

```
FUNCTION NRM (Ortalama, Degiske)
R1# = RAS(R1#): R2# = RAS(R2#)
NRM = Ortalama + STNRM(R1#, R2#) * SQR(Degiske)
END FUNCTION
```

```

FUNCTION STNRM (R11#, R22#)
STNRM = SIN(2 * PI * R11#) * SQR(-2 * LOG(R22#))
END FUNCTION

```

```

SUB Model STATIC
SELECT CASE SanalDeney$
CASE "Buffon"
CALL Buffon
CASE "SD02"
CALL SD02
END SELECT
END SUB

```

```

SUB MonteCarlo STATIC
ToplamY# = ToplamY# + Y
ToplamKY# = ToplamKY# + Y ^ 2
OrtalamaY# = ToplamY# / I
OrtalamaKY# = ToplamKY# / I
DegiskeY# = OrtalamaKY# - OrtalamaY# ^ 2
IF DegiskeY# <= 0 THEN DegiskeY# = .000001
MCist# = OrtalamaY#
StHataMCist# = SQR(DegiskeY# / I)
GAYG = ZALFA * StHataMCist#
IF I > 100 AND KGAYG >= GAYG THEN
  N1 = I
  I = N
END IF
END SUB

```

```

SUB Buffon
R1# = RAS(R1#): TETA = STD(0, PI, R1#): R2# = RAS(R2#): Y =
STD(0, A, R2#)
X = (B / 2) * SIN(TETA)
IF X > A - Y OR X > Y THEN
  Y = 1
ELSE
  Y = 0
END IF
END SUB

```

```

SUB BuffonG
A = 10: B = 5: BeklenenDeger = (2 * B) / (PI * A)
END SUB

```

```

SUB Sakla STATIC
SELECT CASE SanalDeney$
CASE "Buffon", "SD02"
CALL Standart1S

```

```
END SELECT
END SUB
```

```
SUB Yaz
SELECT CASE SanalDeney$
CASE "Buffon", "SD02"
CALL Standart1Y
END SELECT
END SUB
```

```
SUB SD02
'SD02G alt-yordamından,A,B,C,ve D girdilerini alır, N(A,B)
yığınından ras- 'gele C çaplı örneklerin ortalamasını ve
standart sapmasını hesaplar, örnek
'ortalamasının standart değerinin D değerinden küçük
olmasının başarı 'olduğu bir Bernoulli değişkenin sanal
gözlem değerini üretir.
FOR J = 1 TO C
Y = NRM(A, B):TopY = TopY + Y:TopY2 = TopY2 + Y ^ 2
NEXT J
ORTY = TopY / C
StSapma = SQR(TopY2 / (C - 1) - ORTY ^ 2)
IF (SQR(C) * (ORTY - A) / StSapma) < D THEN
  Y = 1
ELSE
  Y = 0
END IF
END SUB
```

```
SUB SD02G
'Amaç: Z, N(0,1) olmak üzere, Olasılık(Z<D) değerinin MC
tahminini, proje adı ve deneyin toplam yinelenme sayısı ile
birlikte verilen güven aralığı
'yakınsama kıstası(GAYK)nın KGAYG ve ZALFA parametre
değerlerine göre elde
'etmek ve tüm yinelemelerin sonunda gerçekleşen güven
aralıklarının
'BeklenenDegeri kapsama oranını bulmak.
'Çıktı Tanımları: Standart1Y ve Standart1S alt-yordamlarında.
A = 0: B = 1: C = 30: D = 1.645: BeklenenDeger = .95
END SUB
```

```
SUB Standart1Y
LOCATE 4, 1: PRINT USING "####"; YinelemeNO;
COLOR 3, 0: PRINT ". Yineleme ": LOCATE 4, 18
IF I = N THEN
COLOR 7, 0: PRINT USING "#####"; N1;
ELSE
```

```

COLOR 7, 0: PRINT USING "#####"; I;
END IF
COLOR 3, 0: PRINT ". Denemede ";
COLOR 6, 0: PRINT "MC Güven Aralığı :";
COLOR 7, 0: PRINT USING "#####.#####"; MCist#;
PRINT " "; CHR$(241);
PRINT USING "#####.#####"; GAYG
END SUB

```

```

SUB Standart1S STATIC
'İlk 5 satırı, SDL Proje Adı, Yineleme Sayısı, Kabul
edilebilir GAYG ve
'%100(1-ALFA)'lık standart normal güven aralığının üst sınırı
ZALFA girdi
'değerleri ile yakınsama gerçekleştikten sonra deneyin
yinelenme sıra
'numarasını, her yineleme sonunda ulaşılan toplam deneme
sayısı N1'i, ve
'MC örnek ortalamasının değerini,yerel olarak MC örnek
ortalamasının tüm
'yinelemeleri içeren genel ortalamasını ve her yineleme
sonunda gerçekleşen
'güven aralıklarının BeklenenDeger'i kapsama oranını
hesaplayarak yazar.
IF SUTUN = 0 THEN SUTUN = 1 : LOCATE 6 + SATIR, SUTUN
IF RENK = 0 THEN RENK = 7
IF YinelemeNO = 60 * INT(YinelemeNO / 60) THEN
  IF RENK = 7 THEN
    RENK = 15
  ELSEIF RENK = 15 THEN
    RENK = 7
  END IF
COLOR RENK, 0
ELSE
RENK = RENK: COLOR RENK, 0
END IF
PRINT USING "###"; YinelemeNO;
PRINT USING "#####"; N1;
PRINT USING "#####.#####"; MCist#
IF YinelemeNO = 15 * INT(YinelemeNO / 15) THEN
  SATIR = 0 : SUTUN = SUTUN + 20
  ELSE
  SATIR = SATIR + 1
END IF
IF YinelemeNO = 60 * INT(YinelemeNO / 60) THEN
  SATIR = 0: SUTUN = 0
END IF

```

```
'Yinelenen deneyin, MC örnek ortalamalarının birikimli
toplamını
've tüm yinelenmeleri kapsayan genel ortalamasını bulur.
  GenelToplam# = GenelToplam# + MCist#
  GenelOrtalama# = GenelToplam# / YinelemeNO

'Deney sonunda, her yineleme için ayrı ayrı hesaplanan güven
aralıklarının
'Beklenen Değeri kapsama oranını hesaplar.
IF (MCist# + KGAYG)>BeklenenDeger AND BeklenenDeger>(MCist# -
KGAYG) THEN
    W = 1
    ELSE
    W = 0
END IF
    TW = TW + W: KapsamaO = TW / YinelemeNO
COLOR 6, 0: LOCATE 22, 1: PRINT "Monte Carlo Tahmin:";
COLOR 7, 0: PRINT USING "###.#####"; GenelOrtalama#;
COLOR 3, 0: LOCATE 23, 1: PRINT "GA'lıklarının Beklenen
Değeri Kapsama Oranı:";
COLOR 7, 0: PRINT USING "###.#####"; KapsamaO
END SUB
```