ANEXO 3

En el siguiente documento se presenta el anexo 4 de la monografía “Estudio de la herramienta Ptolemy II para el Modelamiento y Diseño de Sistemas Telemáticos y Generación de una Metodología para la Construcción de Actores”, el cual describe los actores que se encuentran en las diferentes librerías de la herramienta Ptolemy II versión 1.0.1.

El presente documento se compone de cada una de las librerías de Ptolemy II en donde se nombran y describen los actores correspondientes a dichas librerías.

A.4.1. ACTORES FUENTE (SOURCES)

Los actores fuentes son utilizados para producir los diferentes tipos de señales que se manejan en Ptolemy II. A continuación se presentan todos los actores que conforman la librería de actores fuente, indicando su función, puertos, tipos de datos que utiliza y los parámetros que maneja.

- Bernoulli. Produce una secuencia aleatoria de boléanos.
  - Puertos. trigger(multipuerto), output(booleano).
  - Parámetros. trueProbability (real), seed(long).

- Clock. Produce una señal periódica constante.
  - Puertos. trigger(multipuerto de entrada, general), output(tipo de elementos de values).
  - Parámetros. offsets(arreglo de reales), period(real), values(arreglo), stopTime(real).

- Const. Produce una salida constante con valor dado por el parámetro value.
  - Puertos. trigger(multipuerto de entrada, general), output(tipo de valor).
  - Parámetros. value(general).

- CurrentTime. Produce una señal de salida con valor igual al tiempo actual.
  - Puertos. trigger(multipuerto de entrada, general), output(real).
  - Parámetros. stopTime(real).

- DiscreteRandomSource. Produce señales con la función de probabilidad pmf dada.
    - Puertos. trigger(multipuerto de entrada, general), output(real).
    - Parámetros. pmf(arreglo de dobles), values(arreglo), seed(long).

- Gaussian. Produce una secuencia aleatoria donde cada salida es el resultado de una variable aleatoria Gausiana.
    - Puertos. trigger(multipuerto de entrada, general), output(real).
    - Parámetros. mean(real), standardDeviation(real), seed(long).

- Interpolator. Produce una secuencia de salida interpolando un grupo especifico de valores. Este puede ser usado para generar formas de ondas complejas.
    - Puertos. trigger(multipuerto de entrada, general), output(tipo de elementos de values).
    - Parámetros. indexes(arreglo de enteros), order(entero), period(entero), values(arreglo de reales).

- PoissonClock. Produce una señal donde las transiciones ocurren de acuerdo al proceso de Poisson.
    - Puertos. trigger(multipuerto de entrada, general), output(tipo de elementos de values).
    - Parámetros. meanTime(real), values(arreglo), stopTime(real).

- Pulse. Produce una secuencia de valores en índices de iteración especificados. La secuencia se repite cuando el parámetro repeat se pone en true.
    - Puertos. trigger(multipuerto de entrada, general), output(tipo de elementos de values).
    - Parámetros. indexes(arreglo de enteros), values(arreglo), firingCountLimit(entero), repeat(booleano).

- Ramp. Produce una secuencia que empieza con el valor dado por init y es incrementado por step después de cada iteración. Los tipos de init y step deben soportar la adición.
    - Puertos. trigger(multipuerto de entrada, general), output(lub(init, step)).
    - Parámetros. init(general), step(general), firingCountLimit(entero).

- SequentialClock. Produce una secuencia de valores en los tiempos dados por el parámetro offsets.
    - Puertos. output(tipo de elementos de values).
    - Parámetros. offsets(arreglo de reales), period(real), values(arreglo).

- Sinewave. Produce una onda seno.
    - Puertos. output(real).
    - Parámetros. samplingFrecuency(real), frecuency(real), phase(real).

- SketchedSource. Produce una señal que ha sido dibujada por el usuario en la pantalla.
    - Puertos. trigger(multipuerto de entrada, general), output(real).
    - Parámetros. dataset(entero), length(entero), period(entero).

- Uniform. Produce una secuencia aleatoria con una distribución uniforme.
    - Puertos. trigger(multipuerto de entrada, general), output(real).
    - Parámetros. lowerBound(real), upperBound(real), seed(long).

- VariableClock. Una extensión de Clock con una entrada para controlar el periodo dinámicamente.
    - Puertos. trigger(multipuerto de entrada, general), output(tipo de elementos de values), periodControl(input, real).

- Parámetros. offsets(arreglo de reales), period(real), values(arreglo), stopTime(real).

A.4.2. ACTORES DE DESPLIEGUE DE SEÑALES (SINKS)

Los actores de despliegue de señales son el último destino de las señales. Estos no tienen salidas, y se encuentran actores que despliegan datos en un plano, en forma de texto o tablas. A continuación se presentan los actores que componen la librería de actores sinks, con sus principales características.

- BarGraph. Realiza graficas de barras, dados los arreglos de reales como entradas.
    - Puertos. input(multipuerto, arreglo de reales).
    - Parámetros. fillOnWrapup(booleano), legend(cadena), startingDataset(entero), iterationsPerUpdate(entero).

- Discard. Consume y descarta señales de entrada.
    - Puertos. input(multipuerto, general).

- Display. Despliega señales de entrada en un área de texto en la pantalla.
    - Puertos. input(multipuerto, general).
    - Parámetros. rowsDisplayed(entero), columnsDisplayed(entero), title(cadena).

- HistogramPlotter. Despliega un histograma de los datos en cada canal de entrada.
    - Puertos. input(multipuerto, real).
    - Parámetros. binOffset(real), binWidth(real), fillOnWrapup(booleano), legend(cadena).

- MatrixViewer. Despliega las entradas de una matriz en una tabla.
    - Puertos. input(matriz).
    - Parámetros. heigth(entero), width(entero).

- Recorder. Registra las entradas para introducirlas en una cola después.
    - Puertos. input(multipuerto, general).
    - Parámetros. capacity(entero).

- SequencePlotter. Dibuja las señales de entrada vs. su índice.
    - Puertos. input(multipuerto, real).
    - Parámetros. fillOnWrapup(booleano), legend(cadena), startingDataset(entero), xInit(real), xUnit(real).

- SequenceScope. Dibuja secuencias que son potencialmente infinitamente largas en el estilo de un osciloscopio.
    - Puertos. input(multipuerto, real).
    - Parámetros. fillOnWrapup(booleano), legend(cadena), startingDataset(entero), persistence(entero), width(entero), xInit(real), xUnit(real).

- Test. Verifica los flujos de entrada contra un valor de parámetro y lanza una excepción si ellos no corresponden.
    - Puertos. input(multipuerto, real).
    - Parámetros. correctValues(arreglo), tolerance(real).

- TimedPlotter. Dibuja entradas como una función de tiempo.
    - Puertos. input(multipuerto, real).
    - Parámetros. fillOnWrapup(booleano), legend(cadena), startingDataset(entero).

- TimedScope. Dibuja entradas como una función del tiempo en el estilo de un osciloscopio.
    - Puertos. input(multipuerto, real).
    - Parámetros. fillOnWrapup(booleano), legend(cadena), startingDataset(entero), persistence(entero), width(entero).

- XYPlotter. Despliega un plano de los datos en cada canal inputX vs. los datos en el correspondiente canal inputY.
    - Puertos. inputX(multipuerto, real), inputY(multipuerto, real).
    - Parámetros. fillOnWrapup(booleano), legend(cadena), startingDataset(entero).

- XYScope. Despliega un plano de los datos en cada canal inputX vs. los datos en el correspondiente canal inputY con persistencia finita.
    - Puertos. inputX(multipuerto, real), inputY(multipuerto, real):
    - Parámetros. fillOnWrapup(booleano), legend(cadena), startingDataset(entero), persistence(entero).

A.4.3. ACTORES MATEMATICOS (MATH)

La librería matemática consiste de actores que calculan alguna función matemática. Alguno de estos actores operan en tipo escalar, esto significa que abarcan todos los tipos de datos numéricos (complejo, real, punto fijo, entero, entero largo). A continuación se presentan los actores que componen esta librería con sus características.

- AbsoluteValue. Produce una salida en cada activación con un valor que es igual al valor absoluto de la entrada.
    - Puertos. input(escalar), output(escalar).

- AddSubtract. Suma señales en los canales de entrada plus y resta señales en los canales de entrada minus.
    - Puertos. plus(multipuerto, general), minus(multipuerto, general), output(lub(plus, minus)).

- Average. Saca el promedio de las entradas desde la última vez que una señal true fue recibida en el puerto reset. La entrada reset podría ser dejada desconectada en la mayoría de los dominios.
    - Puertos. input(general), reset(booleano), output(tipo de la entrada).
    - Parámetros. function(cadena).

- DotProduct. Saca el producto punto de dos arreglos de entrada.
    - Puertos. input1(arreglo), input2(arreglo), output(arreglo).

- Expresión. En cada activación, evalúa el parámetro expresión, cuyo valor es puesto por una expresión que podría incluir referencias a cualquier

puerto de entrada que ha sido adicionado al actor. Esta expresión debe ser soportada por la sintaxis utilizada en Ptolemy II.

- Puertos. output(general).
- Parámetros. expression(cadena).

- MathFunction. Produce una señal de salida con el valor que es una función de las entradas. La función es especificada por el atributo function, donde las funciones validas son exp, log, modulo, sign, square, y sqrt.
  - Puertos. firstOperand(real), output(real).
  - Parámetros. function(cadena).

- Maximum. Produce una señal de salida en cada activación en maximunValue con un valor que es el máximo de los valores de los canales de entrada. El índice de este máximo es sacado en channelNumber.
  - Puertos. input(multipuerto, escalar), maximunValue(multipuerto, escalar), channelNumber(multipuerto, entero).

- Minimum. Produce una señal de salida en cada activación en minimunValue con un valor que es el mínimo de los valores de los canales de entrada. El índice de este mínimo es sacado en channelNumber.
  - Puertos. input(multipuerto, escalar), minimunValue(multipuerto, escalar), channelNumber(multipuerto, entero).

- MultiplyDivide. Multiplica señales en la entrada multiply, y divide señales en la entrada divide.
  - Puertos. multiply(multipuerto, general), divide(multipuerto, general), output(lub(multiply, divide)).

- Quantizer. Produce una señal de salida con el valor en niveles que es el más cercano al nivel de entrada.
  - Puertos. input(real), output(real).

- Parámetros. level(arreglo de reales).

- Remainder. Produce una señal de salida con el valor que es el residuo después de dividir la señal en el puerto de entrada por el divisor.
  - Puerto. Input(real), output(real).
  - Parámetros. divisor(real).

- Scale. Produce una salida que es el producto de la entrada y el factor.
  - Puertos. input(general), output(lub(input, factor)).
  - Parámetros. factor(general).

- TrigFunction. Produce una señal de salida con un valor que es una función de la entrada. La función es especificada por el atributo function, donde las funciones validas son acos, asin, atan,, cos, sin, y tan.
  - Puertos. input(real), output(real).
  - Parámetros. function(cadena).

A.4.4. ACTORES DE CONTROL DE FLUJO (FLOW CONTROL)

Los actores de control de flujo afectan el enrutamiento de las señales de una topología. A continuación se describen los actores que componen la librería de actores de control de flujo.

- Chop. Corta una secuencia de entrada y de ella construye una secuencia nueva. Este actor puede ser usado, por ejemplo, para relleno de ceros, traslape de ventanas, líneas de retraso.
    - Puertos. input(multipuerto, general), output(tipo de la entrada).
    - Parámetros. numberToRead(entero), numberToWrite(entero), offset(entero), usePastInputs(booleano).

- Commutator. Intercala los datos en los canales de entrada en una sola secuencia en la salida.
    - Puertos. input(multipuerto, general), output(tipo de entrada).

- Distributor. Distribuye los datos en la secuencia de entrada en múltiples secuencias en los canales de salida.
    - Puertos. input(general), output(multipuerto, tipo de entrada).

- Multiplexor. Produce como salida la señal en el canal de entrada especificado por la entrada select (una señal es leída de cada canal de entrada en cada activación).
    - Puertos. input(multipuerto, general), select(entero), output(tipo de entrada).

- RecordAssembler. Produce una señal de salida que resulta de combinar una señal de cada uno de los puertos de entrada (los cuales pueden ser

adicionados por el usuario). Para adicionar puertos de entrada al actor en Vergil, haga clic derecho en su ícono y seleccione “Configure Ports”, y luego seleccione “Add”. El nombre de cada campo en el registro es el nombre del puerto que proporciona el campo.

  - Puerto. Output(registro).

- RecordDisassembler. Produce señales de salida en los puertos de salida (los cuales pueden ser adicionados por el usuario) que resultan de separar el registro en el puerto de entrada. El nombre de cada campo extraído del registro es el nombre del puerto de salida al cual el valor del campo es enviado.
  - Puertos. input(registro).

- RecordUpdater. Produce una señal de salida que resulta de reemplazar campos en el registro leído del puerto de entrada con valores proporcionados por otros puertos. El usuario debe crear otros puertos de entrada para usar esto. Los nombres de esos puertos determinan cual campo es reemplazado.
  - Puertos. input(registro), output(registro).

- Repeat. Repite en la salida un bloque de señales leídas de la entrada el número especifico de veces.
  - Puertos. input(multipuerto, general), output(tipo de entrada).
  - Parámetros. numberOfTimes(entero), blockSize(entero).

- SampleDelay. Produce en la salida las señales leídas de la entrada, retrasadas por algún número de activaciones. La cantidad de retraso es la longitud del arreglo initialOutput. El arreglo también proporciona las salidas iniciales para las primeras activaciones.
  - Puertos. input(multipuerto, general), output(tipo de entrada).
  - Parámetros. initialOutputs(arreglo).

- Select. Produce como salida la señal en el canal de entrada especificado por la entrada control(únicamente esta señal es leída).
    - Puertos. input(multipuerto, general), control(entero), output(tipo de entrada).

- Sequencer. Pone señales en orden de acuerdo a sus números en una secuencia.
    - Puertos. input(general), sequenceNumber(entero), output(tipo de entero).
    - Parámetros. startingSequenceNumber(entero).

- Sleep. Produce como salida las señales recibidas en la entrada después de un tiempo de retraso especificado por delay.
    - Puertos. input(multipuerto, general), output(multipuerto, general).
    - Parámetros. delay(long).

- Switch. Produce la señal leída del puerto de entrada en el canal de salida especificado por la entrada control.
    - Puertos. input(general), control(entero), output(multipuerto, tipo de entrada).

- Synchronizer. Espera hasta que al menos una señal exista en cada canal de entrada, y luego produce estas señales en los correspondientes canales de salida.
    - Puertos. input(multipuerto, general), output(multipuerto, tipo de entrada).

A.4.5. ACTORES LÓGICOS (LOGIC)

Los actores lógicos en Ptolemy II proporciona operaciones de comparación, y operaciones con números binarios. A continuación se describen los actores que conforman la librería de actores lógicos.

- Comparator. Produce una señal de salida con un valor que es una comparación de la entrada. La comparación es especificada por el atributo comparación, donde las comparaciones validas son >, >=, <, <=, y ==.
    - Puertos. left(real), right(real), output(real).
    - Parámetros. comparision(cadena), tolerance(real).

- Equals. Consume una señal de cada canal de entrada, y produce una señal de salida con un valor true si estas señales son iguales en valor, y false en otro caso.
    - Puertos. input(multipuerto, general), output(booleano).

- LogicalNot. Produce una señal de salida la cual es la negación lógica de la señal de entrada.
    - Puertos. input(booleano), output(booleano).

- LogicFunction. Produce una señal de salida con un valor que es una función lógica de las señales en los canales de entrada. La función es especificada por el atributo function, donde las funciones validas son and, or, xor, nand, nor, y xnor.
    - Puertos. input(multipuerto, booleano), output(booleano).
    - Parámetros. function(cadena).

(uniformizar los saltos de página entre apartados, trabajarlos como aquí)

A.4.6. ACTORES DE CONVERSIÓN (CONVERSIONS)

Ptolemy II proporciona una librería de actores para realizar la conversión de los tipos de señales manejados por la herramienta, cuando es necesario. A continuación se describen los actores de esta librería.

- BitsToInt. Convierte 32 entradas binarias sucesivas en un entero complemento a dos.
    - Puertos. input(booleano), output(entero).

- CartesianToComplex. Convierte dos señales representando la parte real e imaginaria de un número complejo en su representación compleja.
    - Puertos. real(real), imag(real), output(complejo).

- CartesianToPolar. Convierte un par cartesiano (una señal en x y una señal en y) a dos señales representando su forma polar (la cual es una salida en ángulo y magnitud).
    - Puertos. x(real), y(real), angle(real), magnitud(real).

- ComplexToCartesian. Convierte una señal representando un número complejo en sus componentes cartesianos (los cuales son parte real e imaginaria).
    - Puertos. input(complejo), real(real), imag(real).

- ComplexToPolar. Convierte una señal que representa un número complejo en dos señales que representan su forma polar.
    - Puertos. input(complejo), angle(real), magnitude(real).

- RealToFix. Convierte un real en un número de punto fijo con una precisión específica, usando una estrategia de cuantificación especifica.

- Puertos. input(real), output(fijo).
- Parámetros. precision(matriz de enteros), quantization(cadena).

- FixToReal. Convierte un número de punto fijo en un real, primero colocando la precisión del punto fijo a la precisión proporcionada, usando una estrategia de cuantificación especifica.
  - Puertos. input(punto fijo), output(real).
  - Parámetros. precision(matriz de enteros), cuantificación(cadena).

- FixToFix. Convierte un punto fijo en otro punto fijo con una diferencia de precisión, usando una estrategia especifica de cuantificación y sobreflujo.
  - Puertos. input(punto fijo), output(punto fijo).
  - Parámetros. overflow(cadena), precision(matriz de enteros), quantization(cadena).

- IntToBits. Convierte una entrada entera en 32 salidas binarias sucesivas.
  - Puertos. input(entero), output(enteros).

- PolarToCartesian. Convierte dos señales que representan una coordenada polar (ángulo y magnitud) en dos señales que representan su forma cartesiana (una salida en x y una en y).
  - Puertos. angle(real), magnitude(real), x(real), y(real).

- PolarToComplex. Convierte dos señales que representan coordenadas polares en una señal que representa su forma compleja.
  - Puertos. angle(real), magnitude(real), output(complejo).

- Round. Produce una señal de salida con un valor que es una versión redondeada de la entrada. El método de redondeo es especificado por el atributo function, donde las funciones validas son ceil, floor, round, y truncate.

- Puertos. input(real), output(entero).
- Parámetros. function(cadena).

A.4.7. ACTORES DE ARREGLOS (ARRAY)

Estos actores soportan la manipulación de arreglos, los cuales son una colección de señales de determinado tipo. A continuación se describen los actores que componen la librería de actores de arreglos.

- ArrayAppend. Adiciona arreglos en los canales de salida para producir un solo arreglo de salida.
    - Puertos. input(multipuerto, arreglo), output(arreglo).

- ArrayElement. Extrae un elemento de un arreglo y lo produce en el puerto de salida.
    - Puertos. input(arreglo), output(arreglo).
    - Parámetros. index(entero).

- ArrayExtract. Extrae un sub-arreglo de un arreglo y lo produce en el puerto de salida.
    - Puertos. input(arreglo), output(arreglo).
    - Parámetros. sourcePosition(entero), extractLength(entero), destinationPosition(entero), outputArrayLength(entero).

- ArrayLength. Saca la longitud de un arreglo de entrada.
    - Puertos. input(arreglo), output(entero).

- ArrayToSequence. Extrae todos los elementos de un arreglo de entrada y los produce secuencialmente en el puerto de salida.
    - Puertos. input(arreglo), output(tipo de elemento).
    - Parámetros. arrayLength(entero), enforceArrayLength(booleano).

- SequenceToArray. Colecciona una secuencia de entradas en un arreglo y produce el arreglo en el puerto de salida.
    - Puertos. input(general), output(arreglo).
    - Parámetros. arrayLength(entero).

A.4.8. ACTORES DE PROCESAMIENTO DE SEÑALES (SIGNAL PROCESSING)

La librería de actores de procesamiento de señales se divide en diferentes sub-librerías, las cuales se presentan a continuación con sus respectivos actores.

A.4.8.1. AUDIO

Los actores de la librería audio puede leer y escribir archivos de audio. Pueden capturar datos de una entrada de audio, como un CD o un micrófono, y ejecutar datos de audio a través de los parlantes del computador. Estos actores utilizan la librería javasound, la cual hace parte de la distribución 1.3 de Java.

- AudioCapture. Captura audio de un puerto de entrada de audio del computador, o de su micrófono, y produce las muestras en la salida.
    - Puertos. trigger(multipuerto, general), output(multipuerto, real).
    - Parámetros. sampleRate(entero), bitsPerSample(entero), channels(entero).

- AudioReader. Lee audio de una URL, y produce las muestras en la salida.
    - Puertos. trigger(multipuerto, general), output(multipuerto, real).
    - Parámetros. sourceURL(cadena).

- AudioPlayer. Ejecuta muestra de audio en el puerto de salida de audio del computador, o en sus parlantes.
    - Puertos. input(multipuerto, real).
    - Parámetros. sampleRate(entero), bitsPerSample(entero), channels(entero).

- AudioWriter. Escribe datos de audio en un archivo.
    - Puertos. input(multipuerto, real).
    - Parámetros. pathName(cadena), sampleRate(entero), bitsPerSample(entero), channels(entero).

A.4.8.2. COMUNICACIONES

La librería de actores de comunicaciones contiene tres actores que soportan el modelamiento y diseño de sistemas de comunicación digital.

- LineCoder. Lee una secuencia de boléanos (de longitud wordLength) y los interpreta como un índice binario en una taba, de la cual una señal es extraída y enviada a la salida.
    - Puertos. input(booleano), output(tipo de elemento de la tabla).
    - Parámetros. table(arreglo), wordLength(entero).

- LMSAdaptive. Filtra la entrada con un filtro adaptivo, y actualiza los coeficientes del filtro usando la señal de error de entrada de acuerdo al algoritmo LMS (el menor cuadrado significativo).
    - Puertos. input(real), error(entrada, real), output(real), tapValues(salida, arreglo de reales).
    - Parámetros. decimation(entero), decimationPhase(entero), stepSize(real), errorDelay(entero), initialTaps(arreglo de reales).

- RaisedCosine. Un filtro FIR (Respuesta a Impulso Finito) con una respuesta en frecuencia de coseno. Éste es usado típicamente en sistemas de comunicaciones como un moldeador de pulso o un filtro acoplado.
    - Puertos. input(real), output(real).

- Parámetros. decimation(entero), decimationPhase(entero), interpolation(entero), length(entero), excessBW(real), root(booleano), symbolInterval(entero).

A.4.8.3. FILTRADO

Esta librería proporciona los actores necesarios para realizar procesos de filtrado en las señales de entrada.

- DelayLine. En cada activación saca las n más recientes señales de entrada coleccionadas en un arreglo, donde n es la longitud de initialValues. Al comienzo, antes que estén las n señales más recientes, usa las señales de initialValues.
  - Puertos. input(general), output(arreglo).
  - Parámetros. initialValues(arreglo).

- DownSample. Lee las entradas factor y produce únicamente una de ellas en la salida.
  - Puertos. input(general), output(tipo de entrada).
  - Parámetros. factor(entero), phase(entero).

- FIR. Produce una señal de salida con un valor que es la entrada filtrada por un filtro FIR con coeficientes dados por taps.
  - Puertos. input(general), output(general).
  - Parámetros. decimation(entero), decimationPhase(entero), interpolation(entero), taps(arreglo).

- IIR. Produce una señal de salida con un valor que es la entrada filtrada por un filtro IIR (Respuesta a Impulso Infinito) usando una forma directa de implementación.

- Puertos. input(real), output(real).
- Parámetros. numerator(arreglo de reales), denominator(arreglo de reales).

- Lattice. Produce una señal de salida con un valor que es la entrada filtrada por un filtro entramado FIR con coeficientes dados por reflectionCoefficients.
  - Puertos. input(real), output(real).
  - Parámetros. reflectionCoefficients(arreglo).

- LMSAdaptive. Filtra la entrada con un filtro adaptivo, y actualiza los coeficientes del filtro usando la señal de entrada de error de acuerdo al algoritmo LMS.
  - Puertos. input(real), error(entrada, real), output(real), tapValues(output, arreglo de reales).
  - Parámetros. decimation(entero), decimationPhase(entero), stepSize(real), errorDelay(entero), initialTaps(arreglo de reales).

- RecursiveLattice. Produce una señal de salida con un valor que es la entrada filtrada por un filtro entramado recursivo con coeficientes dados por reflectionCoefficients.
  - Puertos. input(real), output(real).
  - Parámetros. reflectionCoefficients(arreglo).

- UpSample. Lee una señal de entrada y produce salidas factor, con todas menos una de las salidas siendo un cero del mismo tipo de la entrada.
  - Puertos. input(general), output(tipo de la entrada).
  - Parámetros. factor(entero), phase(entero).

- VariableFIR. Filtra la secuencia de entrada con un filtro FIR con coeficientes dados en el puerto de entrada newTaps. El parámetro blockSize especifica el número de entradas sucesivas que son procesadas por cada grupo de taps que son proporcionados por newTaps.
    - Puertos. input(general), newTaps(entrada, arreglo), output(general).
    - Parámetros. decimation(entero), decimationPhase(entero), interpolation(entero), blockSize(entero).

- VariableLattice. Filtra la secuencia de entrada con un filtro entramado FIR con coeficientes daos en el puerto de entrada newCoefficients. El parámetro blockSize especifica el número de entradas sucesivas que son procesadas para grupo de taps proporcionados en newCoefficients.
    - Puertos. input(real), newTaps(input, arreglo de reales), output(real).
    - Parámetros. blockSize(entero).

- VariableRecursiveLattice. Filtra la secuencia de entrada con un filtro entramado recursivo con coeficientes dados en el puerto de entrada newCoefficients. El parámetro blockSize especifica el número de entradas sucesivas que son procesadas para cada grupo de taps proporcionados en newCoefficients.
    - Puertos. input(real), newTaps(input, arreglo de reales), output(real).
    - Parámetros. blockSize(entero).

A.4.8.4. PROCESAMIENTO DE IMAGENES

Esta librería se encarga del procesamiento de imágenes, y se divide en dos sub-librerías, las cuales se presentan a continuación.

A.4.8.4.1. ESPECTRO

- DB. Produce una señal que es el valor en decibeles ($k*\log_{10}(z)$) de la señal recibida, donde k es 10 si inputIsPower es true, de lo contrario es 20. La salida nunca es menor que min.
    - Puertos. input(real), output(real).
    - Parámetros. inputIsPower(booleano), min(real).

- FFT. Una transformada de Fourier de segundo orden
    - Puertos. input(complejo), output(complejo).
    - Parámetros. orden(entero).

- IFFT. Una transformada inversa de Fourier de segundo orden.
    - Puertos. input(complejo), output(complejo).
    - Parámetros. order(entero).

- LevinsonDurbin. Calcula los coeficientes de predicción lineal (para los filtros FIR y entramado) para la entrada de auto correlación especificada.
    - Puertos. autocorrelation(entrada, arreglo de reales), errorPower(salida, arreglo de reales), linearPredictor(salida, arreglo de reales), reflectionCoefficients(salida, arreglo de reales).

- MaximumEntropySpectrum. Un estimador de espectro selectivo que usa el algoritmo de LevinsonDurbin para calcular los coeficientes de predicción lineal, y luego usa esos como un modelo parametrico para el proceso aleatorio.
    - Puertos. input(real), output(real).
    - Parámetros. order(entero), numberOfInputs(entero), log2resolution(entero).

- SmoothedSpectrum. Un estimador de espectro llamado el algoritmo Blackman-Tukey, el cual estima una función de autocorrelación

promediando productos de las muestras de entrada, y luego calcula el FFT de ese estimado.

- Puertos. input(real), output(real).
- Parámetros. order(entero), numberOfInputs(entero), log2resolution(entero).

- Spectrum. Un estimador de espectro simple que calcula el FFT de la entrada. Para un proceso aleatorio, este es llamado el periodograma espectral estimado.
  - Puertos. input(real), output(real).
  - Parámetros. order(entero), numberOfInputs(entero), log2resolution(entero).

- PhaseUnwrap. Una simple fase desenvuelta.
  - Puertos. input(real), output(real).

A.4.8.4.2. ESTADÍSTICA

- Autocorrelation. Estima la autocorrelación promediando productos de las muestras de entrada.
  - Puertos. input(general), output(arreglo del tipo de la entrada).
  - Parámetros. numberOfInputs(entero), numberOfLags(entero), biased(booleano), symmetricOutput(booleano).

- PowerEstimate. Estima la potencia de la señal de entrada.
  - Puertos. input(real), output(real).
  - Parámetros. forgettingFactor(real).

A.4.9. ACTORES DEL DOMINIO TIEMPO CONTINUO (CONTINUOUS TIME)

La librería de actores de tiempo continuo, proporciona un grupo de actores diseñados específicamente para el uso en el dominio CT. A continuación se describen los actores que componen la librería de este dominio.

- Integrator. Integra la señal de entrada con respecto al tiempo para producir la señal de salida. Esto es, la entrada es la derivada de la salida con respecto al tiempo. Este actor puede ser usado para cerrar ciclos realimentados en CT para definir sistemas de ecuaciones diferenciales.
    - Puertos. input(real), output(real).
    - Parámetros. initialState(real).

- ContinuousTransferFunction. Filtra la entrada con la función de transferencia de Laplace.
    - Puertos. input(real), output(real).
    - Parámetros. denominator(arreglo de reales), numerator(arreglo de reales), C.

- LinearStateSpace. Filtra la entrada con un sistema lineal.
    - Puertos. input(multipuerto, real), output(multipuerto, real).
    - Parámetros. A(matriz de reales), B(matriz de reales), C(matriz de reales), D(matriz de reales), initialStates(matriz fila de reales).

- DifferentialSystem. Filtra la entrada con el sistema especificado, el cual puede ser no lineal, y es especificado usando el lenguaje de expresión de Ptolemy.
    - Parámetros. stateVariableNames(arreglo de cadenas), initialStates(arreglo de reales), C.

- CTPeriodicSampler. Hace un muestreo de la señal de entrada con la tasa especifica, produciendo eventos de salida discretos.
    - Puertos. input(multipuerto, real), output(multipuerto, real).
    - Parámetros. samplePeriod(real).

- CTTriggeredSampler. Hace un muestreo de la señal de entrada en tiempos donde la entrada trigger tiene unos eventos de entrada discretos.
    - Puertos. input(multipuerto, real), trigger(entrada, general), output(multipuerto, real).

- RateLimiter. Limita la primera derivada de la señal de entrada, y produce el resultado como una secuencia de salida.
    - Puertos. input(real), output(real).
    - Parámetros. risingSlewRate(real), fallingSlewRate(real).

- ThresholdMonitor. Saca true si los valores de entrada están en el intervalo [a, b], el cual es centrado en thresholdCenter y tiene ancho thresholdWidth. Este actor controla el tamaño de paso de integración así que la entrada no cruza el umbral sin producir al menos una salida true.
    - Puertos. input(real), output(real).
    - Parámetros. thresholdWidth(real), thresholdCenter(real).

- ZeroCrossingDetector. Cuando trigger es cero (dentro del errorTolerance especificado), entonces saca el valor del puerto de entrada como un evento discreto. Este actor controla el tamaño de paso de integración para resolver exactamente el tiempo en el que el cruce por cero ocurre.
    - Puertos. input(real), trigger(entrada, real), output(real).
    - Parámetros. errorTolerance(real).

- ZeroOrderHold. Convierte eventos discretos en la entrada a señales de tiempo continuo en la salida manteniendo el valor del evento discreto hasta que el siguiente evento discreto llega.
    - Puertos. input(real), output(real).

A.4.10. ACTORES DEL DOMINIO EVENTOS DISCRETOS (DISCRETE EVENT)

Esta librería de actores es proporcionada para soportar modelos de eventos discretos. En los modelos de eventos discretos, las señales son eventos colocados en determinados tiempos. A continuación se describen los actores que componen la librería de este dominio.

- TimedDelay. Retarda los eventos de entrada una cantidad especifica.
  - Puertos. input(general), output(tipo de la entrada).
  - Parámetros. delay(real).

- Merge. Fusiona eventos de entrada en una sola señal.
  - Puertos. input(multipuerto, general), output(tipo de la entrada).

- Sampler. En cada entrada trigger, produce en la salida la más reciente entrada vista.
  - Puertos. input(multipuerto, general), trigger(input, general), output(tipo de la entrada).

- Server. Retarda los eventos de entrada hasta que ellos han sido “servidos” por la cantidad de tiempo especificada.
  - Puertos. input(general), newServiceTime(entrada, real), output(tipo de la entrada).
  - Parámetros. serviceTime(real).

- SingleEvent. Produce un solo evento con el tiempo y valor especificados .
  - Puertos. output(tipo del valor).
  - Parámetros. time(real), value(general).

- TimeGap. Produce en la salida la cantidad de tiempo entre los eventos de entrada.

    - Puertos. input(general), output(real).

- Timer. Dado un valor de tiempo de entrada, produce un valor en la salida que es la cantidad de tiempo en el futuro.
    - Puertos. input(real), output(tipo de la entrada).
    - Parámetros. value(general).

- VariableDelay. Retarda eventos de entrada por la cantidad especificada.
    - Puertos. input(general), delay(entrada, real), output(tipo de entrada).
    - Parámetros. defaultDelay(real).

- WaitingTime. Mide la cantidad de tiempo que un evento tiene que esperar para que un evento llegue. Hay un evento de salida para cada evento que llega de esperar, donde el valor de esa salida es el tiempo gastado esperando.
    - Puertos. waiter(entrada, general), waitee(entrada, general), output(real).