SignalSim JSON format specification

Version 1.0

1. JSON Format Description

The JSON format gives better flexibility, expandability and more compact form to organize configuration data than XML format. So JSON is used to replace old XML configuration file. The XML is still been supported, but future update and extension will be adopted in JSON as higher priority.

The configuration parameters to control the behavior of SignalSim are organized within a file that complies with JSON standard. All configuration data are put in a JSON object, and each parameter set is a key/value combination. Besides the definitions in the following chapters, the following combinations are accepted to help better understand the contents. The keys that not yet defined are reserved and will be ignored on processing.

- ✓ “version”:number
- ✓ “description”:string
- ✓ “comment”:string

The name of key is a word start with lower case character. If the name is a word combination, then all following words start with upper case character. The JSON standard does not support comments, so the “comment” key is used to insert comment sentences.

2. Start Time

The key to define start time is “time”, and the value is an object that contains following key/value combinations:

- ✓ “type”: ”[GPS|GLONASS|BDS|Galileo|UTC]”
- ✓ “week”: number
- ✓ “second”: number
- ✓ “leapYear”: number
- ✓ “day”: number
- ✓ “year”: number
- ✓ “month”: number
- ✓ “hour”: number
- ✓ “minute”: number

The key “type” defines the type of the start time. If the value is “GPS”, “BDS” or “Galileo”, the start time uses the corresponding system time and must have value “week” and “second” keys. For example:

```
“time”: {
    “type”: ”BDS”,
    “week”: 749,
    “second”: 594707.3
}
```

If the value is “GLONASS”, then keys of “leapYear”, “day” and “second” must be included. For example:

```
“time”: {
    “type”: ”GLONASS”,
    “leapYear”: 7,
    “day”: 138,
    “second”: 703.3
}
```

If the value is “UTC”, then the complete date should be included. For example:

```
“time”: {
    “type”: ”UTC”,
    “year”: 2020,
    “month”: 5,
    “day”: 16,
    “hour”: 21,
    “minute”: 11,
    “second”: 43.3
}
```

3. Receiver Trajectory

The key to define receiver trajectory is “trajectory”, the value is an object with following key/value combination:

✓ “name”: string
✓ “initPosition”: object
✓ “initVelocity”: object
✓ “trajectoryList”: array

If name key is missing, it is an anonymous trajectory. There can be multiple trajectory with at most one anonymous trajectory.

The key “initPosition”, “initVelocity” and “trajectoryList” are used to define initial position, initial velocity and trajectory list respectively. The trajectory list must be put behind the initial position and velocity.

Initial position contains the following key/value combinations:

- ✓ “type”: ”[LLA|ECEF]”
- ✓ “format”: “[d|dm|dms|rad]”
- ✓ “longitude”: number
- ✓ “latitude”: number
- ✓ “altitude”: number
- ✓ “x”: number
- ✓ “y”: number
- ✓ “z”: number

The key “type” defines the position is longitude/latitude/altitude coordinate or ECEF coordinate. The type “LLA” needs “longitude”, “latitude” and “altitude” to be defined. If no altitude parameter, the default value is 0. The key “format” defines the format of longitude and latitude to be degree, 0 to 3 digits degree plus minute with 2 digit integer part or a radian angle value. Positive longitude means east and positive latitude means north. The following formats are all valid:

```
“format”: “d”, “longitude”: -121.06542
“format”: “dm”, “longitude”: -12103.9252
“format”: “dms”, “longitude”: -1210355.512
```

The type “ECEF” needs “x”, “y” and “z” to be defined with unit of meter.

Initial velocity contains the following key/value combinations:

- ✓ “type”: “[SCU|ENU|ECEF]”
- ✓ “speedUnit”: “[mps|kph|knot|mph]”
- ✓ “angleUnit”: “[degree|rad]”
- ✓ “speed”: number
- ✓ “course”: number
- ✓ “east”: number
- ✓ “north”: number
- ✓ “up”: number
- ✓ “x”: number
- ✓ “y”: number
- ✓ “z”: number

The key “type” defines the format of initial velocity. If the value is “SCU”, then the key “speed”, “course” and “up” are needed to define horizontal speed, direction of speed (clockwise angle from north) and vertical speed. If the value is “ENU”, then the key “east”, “north” and “up” are needed to define the speed of three directions. If the value is “ECEF”, then the key “x”, “y” and “z” are needed to define the speed at x/y/z coordinates.

The unit of speed is m/s, km/h, knots or miles per hour with default setting as m/s.

The unit of angle is degree or radian with default setting as angle.

If the “up” key is missing, the default vertical speed is 0.

The key “trajectoryList” defines the trajectory of the receiver. The value is an object array with order. Each element defines one trajectory segment. The format is:

```
 “trajectoryList”: [
    trajectory 1,
    trajectory 2,
    …
    trajectory n
]
```

Each object in the array has a key “type” to define the type of the segment and other key/value combination varies depend on the type. The format of key “type” is:

```
“type”: “Const|ConstAcc|VerticalAcc|Jerk|HorizontalTurn]”
```

3.1 Uniform Linear Motion

The type “Const” indicates the vehicle moves at constant speed. The segment has only one parameter to define the duration time with unit of seconds.

- ✓ “time”: number

3.2 Uniformly accelerated linear motion

The type “ConstAcc” indicates the vehicle moves with constant acceleration at the direction of velocity, so the track is a straight line. The segment has two of the following three parameters:

- ✓ “time”: number
- ✓ “acceleration”: number
- ✓ “speed”: number

The key “time” is duration time with unit of m/s. The key “acceleration” is acceleration with unit of m/s^2. Positive value means acceleration and negative value means deceleration. The key “speed” is the speed at the end of segment. If acceleration and end speed are defined, then the sign of the acceleration is ignored.

3.3 Vertical Acceleration

The type “VerticalAcc” indicates the trajectory that the vehicle has vertical acceleration while the horizontal speed unchanged. The segment has two of the following three parameters:

- ✓ “time”: number
- ✓ “acceleration”: number
- ✓ “speed”: number

The definitions of the keys are similar to the uniformly accelerated motion.

3.4 Variable Acceleration Linear Motion

The type “Jerk” indicates the vehicle moves with linear changing acceleration at the direction of velocity, so the track is a straight line. The segment has two of the following three parameters:

- ✓ “time”: number
- ✓ “rate”: number
- ✓ “acceleration”: number

The key “time” is duration time with unit of m/s. The key “rate” is acceleration with unit of m/s^3. Positive value means acceleration and negative value means deceleration. The key “acceleration” is the acceleration at the end of segment. If rate and acceleration are defined, then the sign of the rate is ignored.

3.5 水平转弯

The type “HorizontalTurn” indicates the vehicle has uniform circular motion. Because the vehicle may have vertical speed, so the actual track may be a spiral curve. The keys used to define parameters are:

- ✓ “time”: number
- ✓ “angle”: number
- ✓ “acceleration”: number
- ✓ “rate”: number
- ✓ “radius”: number

In above keys, “time” defines the duration time in unit of seconds. Key “angle” defines the angle to turn, positive value means increase on direction angle and negative value means decrease on direction angle. Key “acceleration” defines the horizontal acceleration, positive value means increase on direction angle and negative value means decrease on direction angle. Key “rate” defines the rate of direction angle change, positive value means increase on direction angle and negative value means decrease on direction angle. Key “radius” defines the radius of the circle formed by the trajectory projected onto the ground, positive value means increase on direction angle and negative value means decrease on direction angle.

The trajectory only needs two parameters of above types. The valid combinations are: the first two keys, or one from first two and one from the latter three. If key “angle” is selected combine with one the latter three, the sign of the value from the latter three is ignored.

4. Ephemeris

The key “ephemeris” is used to define the file to load ephemerides with following format:

- ✓ “ephemeris”: object
- ✓ ”ephemeris”: array

The object or each object in the array defines one ephemeris file. If multiple ephemeris files are defined, the ephemerides in the files are combined together. Each object has following keys:

- ✓ ”type”: “RINEX”
- ✓ ”name”: string

The only supported file type at this time is RINEX, so this key can be ignored. The key “name” defines the name of the file, absolute directory and relative directly are both accepted.

5. Almanac

The key “almanac” is used to define the file to load almanac with following format:

- ✓ “almanac”: object
- ✓ ”almanac”: array

The object or each object in the array defines one almanac file. If multiple almanac files are defined, the almanacs in the files are combined together. Each object has following keys:

- ✓ ”system”: ”[GPS|GLONASS|BDS|Galileo”
- ✓ ”name”: string

Because different satellite system uses different almanac format, so the format for each satellite system are described as following:

GPS uses the YUMA format. The following link can be used to download historical almanacs: https://www.navcen.uscg.gov/archives
Galileo uses the XML format. The following link can be used to download historical almanacs: https://www.gsc-europa.eu/gsc-products/almanac
BDS has no historical almanacs and no official almanac format. Latest almanac can be obtained from following link: https://www.csno-tarc.cn/en/system/almanac. The

user needs to copy the contents of the table and save it as text format. The order of the parameters is the same as the table, and one row represents one satellite. Here is the example of one line:

01 7.8234937973E-004 345600 0.0958150411 -5.0144945881E-010 6493.494226 -3.0206159672E+000 -0.986917360 3.1595536040E-001 -7.2207988706E-004 3.4531488779E-011 782

GLONASS official website does not provide historical almanac download anymore and the .agl format file is obsolete. The way to download latest almanac is similar to BDS by accessing the following website: https://glonass-iac.ru/en/ephemeris/. Also copy the contents of the table to a text file and one line for one satellite with following order:

Parameter	Description	Note	Sample
NS	Slot number	range 1~24	01
Date	Reference date	dd.mm.yy	14.02.24
TΩ	Equator time	in unit of second	34254.47
To6	Period	in unit of second	40544.656
e	Eccentricity		0.00035
i	Orbital inclination	in unit of degree	64.55989
LΩ	Longitude of ascending node	in unit of degree	-141.29602
ω	Argument of perigee	in unit of degree	-34.45862
δt2	Correction to board time scale	in unit of second	-7.6293945E-5
nl	Frequency slug	range -7~6	1
ΔT	Rate of draconic period variation	in unit of second	-0.0015258789

The almanac parameters are only used in synthesizing the navigation messages. If the almanacs are not specified for a certain satellite system, they can be automatically generated based on the ephemeris. In the navigation messages, the corresponding pages will be set as invalid pages or set as unavailable satellites if the ephemeris or almanac is missing.

6. Output Control

The key to control output is “output”, the value is an object containing following combinations:

✓ “type”: ”[position|observation|IFdata]”

✓ “format”: ”[ECEF|LLA|NMEA|KML|RINEX|IQ8|IQ4]”

✓ “name”: string

- ✓ “interval”: number
- ✓ “sampleFreq”: number
- ✓ “centerFreq”: number
- ✓ “config”: object
- ✓ “systemSelect”: array

The “systemSelect” key will indicate which constellation and frequencies are used by SignalSim both on satellite parameter calculation and result output. Each object in the array has following key/value combinations:

- ✓ ”system”: ”[GPS|GLONASS|BDS|Galileo]”
- ✓ ”signal”: string
- ✓ ”enable”: true|false

If there is no “systemSelect” key, only GPS L1C/A is selected.

In the above settings, valid “system” and “signal” combinations are listed below:

- ✓ GPS：L1CA、L1C、L2C、L2P、L5
- ✓ BDS：B1C、B1I、B2I、B3I、B2a、B2b
- ✓ Galileo：E1、E5a、E5b、E5、E6
- ✓ GLONASS：G1、G2

If “system” and “signal” does not match, this object will be ignored. If key “enable” is missing, the default value is true. If key “signal” is not specified for a system, master signal (L1CA/B1C/E1/G1) is used as default.

If the value of “type” is “position”, the output is the receiver reference position, the format of the position is list of ECEF or LLA coordinates NMEA0183 message or KML file. If the value of “type” is “observation”, the output is RINEX3 format observation file. If the value of “type” is “IFdata”, the output is IF data file with IQ8 or IQ4 format. The IQ8 format means each sample has 8bit I value and 8bit Q value with the format of 2’s complement. The first byte is I data and second byte is Q data and then I data of next sample etc. The IQ4 format means each sample occupies one byte with sign mag format. The placement from MSB to LSB is $IS/IM_2/IM_1/IM_0/QS/QM_2/QM_1/QM_0$. The magnitude bits 000 to 111 means amplitude of 1, 3, …, 15 respectively.

The key “interval” defines the output interval for position and observation with the unit of second. The interval will be force to round to multiple of 1 millisecond. The default interval is 1 second if it is not defined.

The key “sampleFreq” and “centerFreq” defines the sampling frequency and equivalent LO frequency for IF data. The unit is MHz and must be multiple of kHz. Based on Nyquist sampling method, signals that beyond the bandwidth will not be output.

The key “name” defines the output file name. The name can include absolute or relative directory.

The key “config” defines the object of output configuration with following combination:

- ✓ ”elevationMask”
- ✓ “maskOut”: array

The key “elevationMask” defines the cutoff elevation with unit of degree. The key “maskOut” defines the array of satellites that need to be mask out from the output. Each object in the array has following two keys to specify corresponding satellite.

- ✓ ”system”: ”[GPS|GLONASS|BDS|Galileo]”
- ✓ ”svid”: number|number array

Please be noted that the output control defines multiple formats and the JSON interpretation function in SignalSim will recognize all of the formats. But the user program may only support part of the format.

7. Signal Power Configuration

The key used to control signal power is “power”, the value is an object with following key/value combination:

- ✓ “noiseFloor”: number
- ✓ “initPower”:
- ✓ “elevationAdjust”: true|false
- ✓ “signalPower”: object|array

The key “noiseFloor” defines the noise floor with the unit of dBm/Hz. The basic thermal noise floor from sky is -174dBm/Hz. The value can be adjusted according to the insertion noise of RF front-end. If the insertion noise is 2dB, the noise floor can be set to -172dBm/Hz. This parameter is used to convert between dBm and dBHz.

The key “initPower” defines the default signal power with following two keys:

- ✓ ”unit”: [dBHz|dBm|dBW]”
- ✓ ”value”: number

The default unit is dBHz.

The key “elevationAdjust” determines whether the signal power changes with the elevation. If the value is true, the fading of extra $25 \times (1 - \sqrt{\sin(\mathrm{El})})\mathrm{dB}$ is applied to the satellite power.

The key “signalPower” defines the signal power change for each satellite. The value is an object or an array of objects with following key/value combinations:

- ✓ “system”: ”[GPS|GLONASS|BDS|Galileo]”
- ✓ “svid”: number|array
- ✓ “powerValue”: object|array

The key “system” and “svid” define the specific satellite or satellites. If the key “svid” is missing, all satellites in specified constellation are included.

The key “powerValue” gives an object or an array to indicate power change at the specified time with following keys:

- ✓ “time”: number
- ✓ ”unit”: [dBHz|dBm|dBW]”
- ✓ “value”: number

The key “time” defines the epoch to change power with unit of seconds. The value is time elapse from the start time.

The key “unit” defines the unit of signal power.

The key “value” defines the corresponding signal power. If the unit is dBHz and no “value” key is specified, the signal power will return to initial value.

8. Delay Configuration

TBD