Applying all the different SINDy methods to the spring mass damper w/ sin term system (the ‘synthetic data’)

w/out noise

|  |  |  |
| --- | --- | --- |
| System of interest |  | Chart, line chart, histogram  Description automatically generatedChart, histogram  Description automatically generated |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SINDy (original)** | | | |
| **Description** | **Output** | | **Comment** |
| SINDy (following pysindy example 3 – Lorenz system)  Adds noise levels (0.0001 🡪 1)  Uses polynomial library (w/ poly order 3)  STLSQ optimiser | model with noise level = 0.0001  (x0)' = 1.000 x1  (x1)' = 3.333 1 + -0.667 x0 + -1.333 x1  model with noise level = 0.001  (x0)' = 1.000 x1  (x1)' = 3.333 1 + -0.667 x0 + -1.333 x1  model with noise level = 0.01  (x0)' = 1.000 x1  (x1)' = 3.333 1 + -0.667 x0 + -1.333 x1  model with noise level = 0.1  (x0)' = 1.000 x1  (x1)' = 3.332 1 + -0.666 x0 + -1.332 x1  model with noise level = 1.0  (x0)' = 1.506 x1 + -0.111 x0 x1 + -3.283 x1^2 + 0.668 x0 x1^2 + 5.181 x1^3 + -0.918 x0 x1^3 + -2.869 x1^4 + 0.352 x0 x1^4 + 0.480 x1^5  (x1)' = 3.079 1 + 2.481 x0 + -0.531 x1 + -1.227 x0^2 + -6.961 x0 x1 + -0.856 x1^2 + 0.121 x0^3 + 2.663 x0^2 x1 + 0.949 x0 x1^2 + -0.260 x0^3 x1 + -0.128 x0^2 x1^2 + 3.044 x0 x1^3 + 1.180 x1^4 + -0.650 x0^2 x1^3 + -0.837 x0 x1^4 + -0.760 x1^5 | | The noise is added during the model.fit  Doesn’t look very noisy ? |
| SINDy  No noise  Fourier Library (default)  *['sin(1 x0)', 'cos(1 x0)', 'sin(1 x1)', 'cos(1 x1)']*  STLSQ optimiser | Model  (x0)' = 0.378 sin(1 x0) + -0.089 cos(1 x0) + 1.043 sin(1 x1) + 0.387 cos(1 x1)  (x1)' = 1.679 sin(1 x0) + -0.486 cos(1 x0) + -1.149 sin(1 x1) + 1.756 cos(1 x1) | |  |
| SINDy  No noise  Concatenated library   * Polynomial, degree 2 * Fourier, default   STLSQ optimiser | Model  (x0)' = 1.000 x1  (x1)' = 3.333 1 + -0.667 x0 + -1.333 x1 | | The terms in the equations are correct, it’s just missing the sin(t) term (the fourier library has sin(X) terms)  Maybe add higher order polynomial terms to express sin as an expansion |
| SINDy  No noise  Library of just sin(X) | Library:  ['sin(x0)', 'sin(x1)']  Model (no noise)  (x0)' = 1.173 sin(x1)  (x1)' = -0.551 sin(x1) | | Its sin(t) terms we need in the library |
| SINDy  No noise  Custom library with just known terms (1, X, sin(X) | Custom Library:  [1, 1, 'x0', 'x1', 'sin(x0)', 'sin(x1)']  Model (no noise)  (x0)' = 1.000 x1  (x1)' = 1.667 1 + 1.667 1 + -0.667 x0 + -1.333 x1 | | Not sure how to have only 1 constant (1) term  This is the wrong way to include sin (it should be time not state dependant) |
| SINDy  No noise  Custom library with known terms (1, X, sin(X)) concat. w/ polynomial library | Custom Library:  [1, 1, 'x0', 'x1', 'sin(x0)', 'sin(x1)', '1', 'x0', 'x1', 'x0^2', 'x0 x1', 'x1^2']  Model (no noise)  (x0)' = 0.500 x1 + 0.500 x1  (x1)' = 1.111 1 + 1.111 1 + -0.333 x0 + -0.667 x1 + 1.111 1 + -0.333 x0 + -0.667 x1 | | Its correct if you tidy it up (ignoring that there should be a sin term) |
| SINDy  Noise  Polynomial library | 1% rmse | Model (w/ noise)  (x0)' = 1.000 x1  (x1)' = 3.333 1 + -0.667 x0 + -1.333 x1 |  |
| 10% rmse | Model (w/ noise)  (x0)' = 1.000 x1  (x1)' = 3.333 1 + -0.667 x0 + -1.333 x1 |
| 50% rmse | Model (w/ noise)  (x0)' = 1.000 x1  (x1)' = 3.333 1 + -0.667 x0 + -1.333 x1 |
|  |  | |  |

|  |  |  |
| --- | --- | --- |
| **SINDy-PI with different libraries** | | |
| **Description** | **Output** | **Comment** |
| Following example 9 – high order ode  SINDy-PI  PDE library  Sindypi optimiser | Very long equations – no point in copying in, just know they were long/busy  x0\_t = -0.0009 x0 + 0.9910 x1 + 0.0002 x0x0 + 0.0003 x1x1 + 0.0055 x0x0\_t + 0.0007 x0x1\_t + -0.0007 x1x0\_t + 0.0001 x1x1\_t + 0.0009 x0x0x0\_t + -0.0024 x0x0x1\_t + -0.0002 x1x1x0\_t + 0.0034 x1x1x1\_t  x1\_t = 2.6280 1 + -1.0326 x0 + -1.5092 x1 + 0.1011 x0x0 + -0.4609 x0\_t + 0.5995 x0x0\_t + 0.1206 x0x1\_t + -0.0411 x1x0\_t + 0.0072 x1x1\_t + 0.4045 | Model 14  The equations copied in are the ones for x0 dot and x1 dot, note that all the other equations very busy |
| SINDy-PI  PDE library  Sindypi optimiser, w/ threshold 1e-4 🡪10 | Still very long busy equations  x0\_t = 0.0007 1 + 0.9901 x1 + 0.0008 x1x1 + 0.0002 x1\_t + 0.0029 x0x0\_t + -0.0002 x1x0\_t + 0.0020 x1x1\_t + 0.0001 x0x0x1\_t + 0.0013 x1x1x1\_t  x1\_t = 1.5846 1 + -0.6814 x0 + -0.7654 x1 + 0.0720 x0x0 + 0.0605 x1x1 + 0.2972 x0x0\_t + 0.2607 x0x1\_t + -0.0282 x1x0\_t + -0.3370 x1x1\_t + 0.7311 | Increasing threshold a lot didn’t really make the equations much sparser |
| SINDy-PI  Fourier library  Sindypi optimiser, w/ threshold 1e-4 | Long equations | The fourier library doesn’t have any of the correct terms is it so it would never look wuite right |
| SINDy-PI  Concatenated PDE and Fourier library  Sindypi optimiser, w/ threshold 1e-4 | x0\_t = 0.0031 x0 + 0.9579 x1 + -0.0006 x0x0 + 0.0101 x0x0\_t + -0.0010 x1x0\_t + 0.0097 x1x1\_t + 0.0021 x0x0x0\_t + -0.0005 x1x1x0\_t + 0.0103 x1x1x1\_t + 0.0002 sin(1 x0) + -0.0030 cos(1 x0) + -0.0026 sin(2 x0) + 0.0006 cos(2 x0) + 0.0083 sin(2 x1) + 0.0002 cos(3 x0) + -0.0004 cos(3 x1)  x1\_t = 1.0766 1 + -0.0553 x0x0 + -1.5226 x0\_t + 0.6174 x0x0\_t + -0.0639 x1x0\_t + -0.0140 x1x1\_t + 0.3629 x0x0x0\_t + -0.0223 x0x0x1\_t + -0.0328 x1x1x0\_t + -0.0460 x1x1x1\_t + 0.0994 sin(1 x0) + 0.6500 cos(1 x0) + 0.3243 cos(1 x1) + 0.0703 sin(2 x0) + 0.0441 cos(2 x0) + 0.0122 sin(3 x0) + 0.0044 cos(3 x0) + 0.0124 sin(3 x1) + -0.0358 cos(3 x1) |  |
| SINDy-PI  No noise  PDE library with just sin(t) in library features | x0\_t = 0.1755 1 + -0.1263 sin(t) + -0.1263 sin(t) + -0.1666 x1\_t + 0.5147 sin(t)x0\_t + 0.5147 sin(t)x1\_t + 0.2557 sin(t)x0\_t + 0.2557 sin(t)x1\_t  x1\_t = 0.1010 1 + -0.0710 sin(t) + -0.0710 sin(t) + -0.4281 x0\_t + 0.2742 sin(t)x0\_t + 0.2742 sin(t)x1\_t + 0.4748 sin(t)x0\_t + 0.4748 sin(t)x1\_t |  |
| SINDy-PI  No noise  PDE library with x and sin(t) | x0\_t = 0.0877 x1 + -0.0001 sin(t) + -0.0001 sin(t) + 0.0002 x1\_t + 0.1823 x0x0\_t + 0.3651 x0x1\_t + 0.0456 x1x0\_t + 0.0456 x1x1\_t + -0.0001 sin(t)x0\_t + 0.2728 sin(t)x1\_t + -0.0001 sin(t)x0\_t + -0.0001 sin(t)x1\_t  x1\_t = 3.3335 1 + -0.6667 x0 + -0.1666 sin(t) + -0.1666 sin(t) + -0.7872 x0\_t + -0.1092 x0x0\_t + -0.2184 x0x1\_t + -0.0273 x1x0\_t + -0.0273 x1x1\_t + -0.0001 sin(t)x0\_t + -0.1638 sin(t)x1\_t |  |
| SINDy-PI  No noise  PDE library  ['1', 'x0', 'x1', 'sin(t)', 'sin(t)', 'x0\_t', 'x1\_t'] | x0\_t = 0.9997 x1 + -0.0002 x1\_t  x1\_t = 3.3318 1 + -0.6664 x0 + -0.4933 x1 + -0.1665 sin(t) + -0.1665 sin(t) + -0.8392 x0\_t | Equations are pretty close, theres a derivative term in each one that i wasn’t expecting  Theres 2 sin terms  Expecting ~0.333sin(t) (if you tidy the equation you roughly get this) |
| SINDy-PI  0.1% random noise  PDE library  ['1', 'x0', 'x1', 'sin(t)', 'sin(t)', 'x0\_t', 'x1\_t'] | x0\_t = 0.9994 x1 + 0.0023 x1\_t  x1\_t = 3.3191 1 + -0.6639 x0 + -1.3154 x1 + -0.1659 sin(t) + -0.1659 sin(t) + -0.0117 x0\_t | With added noise the eqautions are roughly the same |
| SINDy-PI  1% random noise  PDE library  ['1', 'x0', 'x1', 'sin(t)', 'sin(t)', 'x0\_t', 'x1\_t'] | x0\_t = 0.0018 1 + 0.9947 x1 + -0.0003 sin(t) + -0.0003 sin(t) + 0.0181 x1\_t  x1\_t = 3.1763 1 + -0.6358 x0 + -1.2577 x1 + -0.1587 sin(t) + -0.1587 sin(t) + -0.0090 x0\_t | The noise has added a constant and sin terms to the x0’ equation |
| SINDy-PI  10% random noise  PDE library  ['1', 'x0', 'x1', 'sin(t)', 'sin(t)', 'x0\_t', 'x1\_t'] | x0\_t = 1.4713 1 + -0.2897 x0 + 0.3147 x1 + -0.0796 sin(t) + -0.0796 sin(t) + -0.0060 x1\_t  x1\_t = 2.1556 1 + -0.4362 x0 + -0.7601 x1 + -0.1041 sin(t) + -0.1041 sin(t) + -0.0065 x0\_t | Added terms in the equations, but the graph still looks similar |
| SINDy-PI  50% random noise  PDE library  ['1', 'x0', 'x1', 'sin(t)', 'sin(t)', 'x0\_t', 'x1\_t'] | x0\_t = 2.8085 1 + -0.5421 x0 + -0.4275 x1 + -0.1843 sin(t) + -0.1843 sin(t) + -0.0008 x1\_t  x1\_t = -0.2411 1 + 0.0314 x0 + 0.1836 x1 + 0.0118 sin(t) + 0.0118 sin(t) + -0.0009 x0\_t | Not recovering anything, equations further off |

|  |  |  |
| --- | --- | --- |
| **Abrupt-SINDy** from ‘sparsereg’ github | | |
| Description | Output | Comment |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **SINDy-SA** | | |
| Description | Output | Comment |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Ensemble-SINDy** | | |
| Description | Output | Comment |
| Ensemble-SINDy  No noise  Polynomial Library  Default STLSQ optimiser  V1 ensembling and V2 ensembling | Ensemble with replacement (V1)  (x0)' = 0.997 x1 + 0.001 x0 x1 + 0.001 x1^2  (x1)' = 2.757 1 + -0.305 x0 + 0.694 x1 + -0.048 x0^2 + -0.178 x0 x1 + -1.306 x1^2    Ensemble without replacement (V2)  (x0)' = 0.997 x1 + 0.001 x0 x1 + 0.001 x1^2  (x1)' = 2.757 1 + -0.305 x0 + 0.694 x1 + -0.048 x0^2 + -0.178 x0 x1 + -1.306 x1^2 | V1 and V2 are the exact same |
| Ensemble-SINDy  1% rmse noise  Default STLSQ optimiser  V1 ensembling and V2 ensembling | Ensemble with replacement (V1)  (x0)' = 0.484 1 + -0.096 x0 + 0.634 x1 + 0.070 x0 x1  (x1)' = 3.112 1 + -0.619 x0 + -1.044 x1^2    Ensemble without replacement (V2)  (x0)' = 0.999 x1  (x1)' = 3.112 1 + -0.619 x0 + -1.044 x1^2 | V2 more sparse, x0’ equation ~correct, |
| Ensemble-SINDy  1% rmse noise  Default STLSQ optimiser  V1 ensembling and V2 ensembling  Added sin term to PDE library | Library:  ['1', 'x0', 'x1', 'sin(t)', 'sin(t)', 'x0\_t', 'x1\_t', '1', 'x0', 'x1', 'x0^2', 'x0 x1', 'x1^2']    Ensemble with replacement (V1)  (x0)' = 1.000 x0\_t  (x1)' = 1.000 x1\_t    Ensemble without replacement (V2)  (x0)' = 1.000 x0\_t  (x1)' = 1.000 x1\_t | Adding the sin(t) term – the model looks right but the equations aren’t  Actually, those equations aren’t wrong?   * Need to get rid of x0\_t and x1\_t or the equations would always simplify to that? |
| Ensemble-SINDy  No noise  Default STLSQ optimiser  V1 ensembling and V2 ensembling  Custom library  [1, 1, 'x0', 'x1', 'x0^2', 'x1^2', 'sin(t)', 'sin(t)'] | Ensemble with replacement (V1)  (x0)' = 1.000 x1  (x1)' = 1.667 1 + 1.667 1 + -0.667 x0 + -1.333 x1 + -0.167 sin(t) + -0.167 sin(t)    Ensemble without replacement (V2)  (x0)' = 1.000 x1  (x1)' = 1.667 1 + 1.667 1 + -0.667 x0 + -1.333 x1 + -0.167 sin(t) + -0.167 sin(t) | If you sum the terms its correct  How to get rid of second constant and second sin term? |
| Ensemble-SINDy  0.1% rmse noise  Default STLSQ optimiser  V1 ensembling and V2 ensembling  Custom library  [1, 1, 'x0', 'x1', 'x0^2', 'x1^2', 'sin(t)', 'sin(t)' | Ensemble with replacement (V1)  (x0)' = 0.013 1 + 0.013 1 + -0.007 x0 + 0.996 x1 + -0.005 x1^2 + -0.001 sin(t) + -0.001 sin(t)  (x1)' = 1.664 1 + 1.664 1 + -0.665 x0 + -1.331 x1 + -0.001 x1^2 + -0.166 sin(t) + -0.166 sin(t)    Ensemble without replacement (V2)  (x0)' = 0.013 1 + 0.013 1 + -0.007 x0 + 0.996 x1 + -0.005 x1^2 + -0.001 sin(t) + -0.001 sin(t)  (x1)' = 1.664 1 + 1.664 1 + -0.665 x0 + -1.331 x1 + -0.001 x1^2 + -0.166 sin(t) + -0.166 sin(t) | Added a constant, an x0, and x1^2, and a sin term to the first equation, added a x1^2 term to the second equation |
| Ensemble-SINDy  1% rmse noise  Default STLSQ optimiser  V1 ensembling and V2 ensembling  Custom library  [1, 1, 'x0', 'x1', 'x0^2', 'x1^2', 'sin(t)', 'sin(t)' | Ensemble with replacement (V1)  (x0)' = 0.148 1 + 0.148 1 + -0.087 x0 + 0.824 x1 + 0.006 x0^2 + 0.063 x1^2 + -0.023 sin(t) + -0.023 sin(t)  (x1)' = 1.619 1 + 1.619 1 + -0.642 x0 + -1.184 x1 + -0.001 x0^2 + -0.101 x1^2 + -0.147 sin(t) + -0.147 sin(t)    Ensemble without replacement (V2)  (x0)' = 0.148 1 + 0.148 1 + -0.087 x0 + 0.824 x1 + 0.006 x0^2 + 0.063 x1^2 + -0.023 sin(t) + -0.023 sin(t)  (x1)' = 1.619 1 + 1.619 1 + -0.642 x0 + -1.184 x1 + -0.001 x0^2 + -0.101 x1^2 + -0.147 sin(t) + -0.147 sin(t) |  |
| Ensemble-SINDy  10% rmse noise  Default STLSQ optimiser  V1 ensembling and V2 ensembling  Custom library  [1, 1, 'x0', 'x1', 'x0^2', 'x1^2', 'sin(t)', 'sin(t)' | Ensemble with replacement (V1)  (x0)' = 1.560 1 + 1.560 1 + -1.246 x0 + -0.544 x1 + 0.123 x0^2 + 0.554 x1^2 + -0.213 sin(t) + -0.213 sin(t)  (x1)' = 0.848 1 + 0.848 1 + -0.649 x0 + 0.044 x1 + 0.061 x0^2 + -0.389 x1^2 + -0.012 sin(t) + -0.012 sin(t)    Ensemble without replacement (V2)  (x0)' = 1.560 1 + 1.560 1 + -1.246 x0 + -0.544 x1 + 0.123 x0^2 + 0.554 x1^2 + -0.213 sin(t) + -0.213 sin(t)  (x1)' = 0.848 1 + 0.848 1 + -0.649 x0 + 0.044 x1 + 0.061 x0^2 + -0.389 x1^2 + -0.012 sin(t) + -0.012 sin(t) | Equations look messy, but looking at plot, it seems to be recovering the dynamics |
| Ensemble-SINDy  50% rmse noise  Default STLSQ optimiser  V1 ensembling and V2 ensembling  Custom library  [1, 1, 'x0', 'x1', 'x0^2', 'x1^2', 'sin(t)', 'sin(t)' | Ensemble with replacement (V1)  (x0)' = 3.231 1 + 3.231 1 + -2.994 x0 + -0.741 x1 + 0.293 x0^2 + 0.385 x1^2 + -0.403 sin(t) + -0.403 sin(t)  (x1)' = -0.090 1 + -0.090 1 + -0.226 x0 + 0.283 x1 + 0.053 x0^2 + -0.163 x1^2 + 0.053 sin(t) + 0.053 sin(t)    Ensemble without replacement (V2)  (x0)' = 3.231 1 + 3.231 1 + -2.994 x0 + -0.741 x1 + 0.293 x0^2 + 0.385 x1^2 + -0.403 sin(t) + -0.403 sin(t)  (x1)' = -0.090 1 + -0.090 1 + -0.226 x0 + 0.283 x1 + 0.053 x0^2 + -0.163 x1^2 + 0.053 sin(t) + 0.053 sin(t) | Even with a lot of noise, the model still looks close to the noiseless true plot, even though the equations are messy |
| Ensemble-SINDy  1% noise  STLQS optimiser  V1, V2, V3, and V4 ensembling  Custom library |  |  |
|  |  |  |
|  |  |  |

28/07 it seems that ive just been plotting x\_test and not the model (at least for spring-damp-sin in ens sindy)

Troubleshooting

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Results |  | comments |
|  | No ensemble | V1 ensembling |  |
| Comparing x\_sim | No ensembling  (x0)' = 0.936 1 + -0.406 x0 + -0.351 x1 + 0.043 x0^2 + 0.260 x0 x1 + 0.437 x1^2  (x1)' = 2.603 1 + -0.262 x0 + 0.847 x1 + -0.050 x0^2 + -0.208 x0 x1 + -1.344 x1^2 | Ensemble with replacement (V1)  (x0)' = 0.936 1 + -0.406 x0 + -0.351 x1 + 0.043 x0^2 + 0.260 x0 x1 + 0.437 x1^2  (x1)' = 2.603 1 + -0.262 x0 + 0.847 x1 + -0.050 x0^2 + -0.208 x0 x1 + -1.344 x1^2 | Nothing different seems to be happening if ensembling or not ? |