Structure and Synthesis of Robot Motion

Making Sense of Sensorimotor Systems

Subramanian Ramamoorthy School of Informatics

26 January, 2012

What do you Need to Know about your Robot?



Recap: What Problem is the Robot Solving?



<u>Problem</u>: How to generate actions, to achieve high-level goals, using limited perception and incomplete knowledge of environment & adversarial actions?

What does Robot Need to Know?

- Given access to raw data channels for various uninterpreted sensors and motors
- Devise a procedure for learning that will tell you what you need for various tasks (as yet unspecified)
 - What types of models?
 - What types of learning methods?

Discussion: How will you do this?

What are you Learning from?

B. 66007 4. 323841 22. 664233 4. 202899 1. 213200 22. 664233 1. 66784 2. 245012 22. 664233 2. 39704 1. 653462 1. 663832 22. 664233 1. 203820 22. 664233 1. 968572 2. 245012 22. 664233 2. 39774 1. 653463 1. 663832 22. 664233 1. 203820 22. 664233 1. 968572 2. 245012 22. 664233 2. 39774 1. 653463 1. 669043 22. 664233 1. 200894 1. 210455 22. 664233 1. 967881 2. 245012 22. 664233 2. 39774 1. 653463 1. 669043 22. 664233 1. 200455 2. 264253 1. 967881 2. 245012 22. 664253 2. 39774 1. 653463 1. 669043 22. 664253 1. 200455 2. 2. 644253 1. 968572 2. 245012 22. 664253 2. 39774 1. 653562 1. 690433 22. 664253 1. 200455 2. 2. 644253 1. 968572 2. 245012 22. 664253 2. 39774 1. 650524 1. 690433 22. 664253 1. 200856 2. 341766 22. 664253 2. 39774 2. 664253 2. 305457 <										
1 1 22.664233 2.264233 <td>1.896007</td> <td>4.323841</td> <td>22.664253</td> <td>4.202899</td> <td>1.213200</td> <td>22.664253</td> <td>1.967881</td> <td>2.245012</td> <td>22.664253</td> <td>2.397054</td>	1.896007	4.323841	22.664253	4.202899	1.213200	22.664253	1.967881	2.245012	22.664253	2.397054
1. 638462 1. 6993363 22. 664253 2. 200025 2. 627862 22. 664253 1. 966572 2. 245012 22. 664253 2. 39774 1. 656573 1. 69943 22. 664253 4. 200399 1. 211126 22. 664253 1. 967861 2. 245012 22. 664253 2. 39774 1. 665573 2. 245012 22. 664253 2. 39774 1. 665573 2. 245012 22. 664253 1. 9775 2. 245012 22. 664253 2. 39774 1. 665573 2. 245012 22. 664253 1. 977597 3. 2. 43338 22. 664253 4. 200395 1. 211126 1. 199597 3. 2. 445133 22. 664253 1. 20045 1. 211126 1. 199597 3. 2. 44513 22. 664253 1. 198577 3. 2. 43338 22. 664253 1. 198575 1. 11826 1. 199577 3. 2. 43338 22. 664253 1. 198575 1. 11826 1. 199577 3. 2. 43338 22. 664253 1. 198575 1. 11826 1. 199577 2. 2.0400 2. 341766 22. 6642 2. 2. 664253 1. 198575 1. 11826 1. 199579 3. 2. 664253 1. 198575 1. 11826 1. 199579 3. 2. 664253 1. 198575 1. 11826 1. 199579 3. 2. 664253 1. 198575 1. 11826 1. 199579 3. 2. 664253 1. 198575 1. 11826 1. 199579 3. 2. 664253 1. 198575 1. 11826 1. 199579 3. 2. 664253 1. 198575 1. 11826 1. 199579 3. 2. 664253 1. 198578 2. 664253 1. 198578 2. 664253 1. 198578 2. 664253 1. 198578 2. 664253 1. 198578 2. 664253 1. 198578 2. 664253 2. 1064253 2. 1064253 2. 1064253 2. 1064253 2. 1064253 2. 2. 664253 2.	1.894624	4.323150	22.664253	4.202899	1.213891	22.664253	1.968572	2.245012	22.664253	2.397745
1.639462 1.699043 22.664233 2.290625 2.290625 2.2627882 2.264253 1.96784 2.245012 2.2664253 2.39774 1.895313 4.123150 22.664253 2.29000 2.600425 2.264253 2.29000 2.600425 2.264253 2.39774 2.265253 2.39774 2.265253 2.29075 1.200426 2.264253 2.29000 2.600425 2.2664253 2.39774 2.265253 2.29075 1.200426 2.2664253 2.2664253 2.2664253 2.2664253 2.29774 2.265253 2.29774 2.265253 2.29774 2.265253 2.29774 2.265253 2.29774 2.265253 2.29774 2.265253 2.29774 2.265253 2.29774 2.265253 2.29774 2.265253 2.29774 2.265253 2.29774 2.265253 2.29774 2.265253 2.29774 2.265253 2.29774 2.265253 2.29774 2.265253 2.29774 2.265253 2.29546 2.265253 2.29546 2.265253 2.29546 2.265253 2.29546 2.265253 2.29546 2.265253 2.29546 2.265253 2.29546 2.265253 2.29546 2.265253 2.29546 2.265253 2.29546 2.265253 2.2654253 2.2	1.635462 1.894624	1.698352 4.323150	22.664253 22.664253	2.293389 4.202899	2.627882 1.211126	22,664253	1,968572	2,245012	22,664253	2,397745
1: 0:0000 1: 0:00000 1: 0:00000 1: 0:00000 1: 0:00000 1: 0:000000 1: 0:	1.635462	1.699043	22.664253	2.290625	2.627882	22 664252	1 067991	2 245012	22 664252	2 207745
1.895131 4.23150 22.664253 4.200334 1.210435 22.664253 1.200354 1.200456 22.664253 1.200553 1.21126 1.630624 1.699043 22.664253 1.2164253 1.21126 1.630624 1.699043 22.664253 1.2164253 1.21126 1.630624 1.699043 22.664253 1.2164253 1.21126 1.630624 1.699043 22.664253 1.2164253 1.2164253 1.2164253 1.210510 1.630624 1.699043 22.664253 1.2164253 1.266425	1.635462	1.699043	22.664253	2.289934	2.630646	22.004235	1.90/001	2.243012	22.004235	2.59/745
1.835315 4.22150 22.664233 2.260423 2.260423 1.21126 2.260423 2.26	1.895315 1.636153	4.323150 1.699043	22.664253 22.664253	4.206354 2.294080	1.210435 2.630646	22.664253	1.968572	2.245012	22.664253	2.397745
1.375757 1.443338 1.22.664253 1.113270 22.664253 2.341766 22.664253 2.341766 22.664253 2.341766 22.664253 2.341766 22.664253 2.341766 22.664253 2.341766 22.664253 2.341766 22.664253 2.341766 22.664253 2.341766 22.664253 2.341766 22.664253 2.341766 22.664253 2.341766 22.664253 2.341766 22.664253 2.341766 22.664253 2.341766 22.664253 2.3664253 2.06013 1.48614 1.60288 2.2664253 1.36613 4.40053 2.311346 22.664253 2.664253 <td>1.895315</td> <td>4.323150</td> <td>22.664253</td> <td>4.200825</td> <td>1.211126</td> <td></td> <td></td> <td> 3630</td> <td>22.664253</td> <td>2.397745</td>	1.895315	4.323150	22.664253	4.200825	1.211126			3630	22.664253	2.397745
16.883917 1.723966 2.293462 2.264233 1.931848 3.931848 <t< td=""><td>4.375673</td><td>2.443358</td><td>22.664253</td><td>22.664253</td><td>1.119210</td><td></td><td></td><td>2080</td><td>2.341766</td><td>22.664253</td></t<>	4.375673	2.443358	22.664253	22.664253	1.119210			2080	2.341766	22.664253
22.664253 2.322415 22.664253 22.664253 2.000363 1.54838 22.664253 2.33106 2.360423 1.168236 2.36923 2.1664233 2.564233 22.664253 2.31106 2.364233 2.0664233 2.6642	1.638917 22.664253	1.725996 22.664253	2.295462 1.250519	22.664253 22.664253	1.953368 22.664253			8499	22.664253	2.130981
12:664233 2:730656 2:730656 2:042311 22:664233 2:064233 <	22.664253	2.322415	22.664253	22.664253	2.000363			64253	2 050813	1 5/8383
27. 694233 1.196133 2.400334 2.310396 2.32624 2.4002 1.480184 1.60289 2.139504 22. 664233 2.664233 22. 664233 2. 2. 664233 2. 664233 2. 664233 2. 664233 2. 664233 2. 664	22.664253	2.323106	2.730856	2.043211	22.664253	A	a a	04255	1. 1050015	1.540505
1.25904 22.664253 1.96517 2.15171 2.2564253 22.664253 22	22.664253 2.359735	1.196613 22.664253	4.400553 2.068782	2.316196 22.664253	2.352824 22.664253	R	n - In St	2802	1.486184	1.602980
12:664253 1:901376 22:664253 1:301376 22:664253 1:30111 22:664253 1:97598 2:664253 2:664253 1:30374 22:664253 1:997598 2:664253 2:664253 1:30374 22:664253 1:997598 2:664253 1:30374 22:664253 1:997598 2:664253 2:25054 22:664253 1:997598 2:664253 2:25054 22:664253 1:997598 2:664253 2:25054 22:664253 1:997598 2:664253 2:25054 22:664253 1:997598 2:664253 2:25054 2:664253 1:99769 2:664253 1:99769 2:664253 2:664253 1:90846 2:10221 2:002365 2:664253 1:617493 2:664253 2:19254 1:90846 2:10321 2:002365 2:064253 2:664253 1:61749 2:664253 2:19254 2:664253 1:002436 2:00206 2:2664253 2:064253 1:69766 2:19254 2:1064253 2:064253 2:064253 2:064253 1:674505 1:65759 1:	1.259504	22.664253	1.965117	2.151714	22.664253			9839	1.538708	22.664253
27.467-366 22.664233 2.25261 3.06498 3.06498 3.06498 3.064233 2.664233	22.664253	1.991378	22.664253	2.374248	22.664253			0373	22.664253	22.664253
2.450960 22.664253 2.664253 2.664253 2.108300 22.664253 2.264253 2.25261 22.664253 2.2	2.46/546 22.664253	22.664253 1.997598	22.664253 2.197326	22.664253	1.321011 2.168300			1388	22.664253	2.250541
2.160960 22.664253 2.6604253 22.664253 1.303734 2.457871 22.664253 1.617493 22.664253 22.664253 1.617493 22.664253 22.664253 1.617493 22.664253 1.617493 22.664253 2.664253 1.617493 22.664253 2.664253 1.617493 22.664253 2.664253 1.617493 22.664253 2.664253 1.617493 22.664253 2.664253 1.69766 2.664253 2.664253 1.69766 2.664253 1.69766 2.664253 1.69766 2.664253 1.69766 2.664253 1.69766 2.664253 1.69766 2.664253 1.69766 2.664253 1.69766 2.664253 1.69766 2.664253 1.69766 2.664253 1.69766 2.664253 1.69766 2.664253 1.69766 2.664253 1.69766 2.664253 1.697736 2.664253 1.69766 <td>2.450960</td> <td>22.664253</td> <td>2.661054</td> <td>22.664253</td> <td>1.303734</td> <td></td> <td></td> <td>1388</td> <td>22 664253</td> <td>2 250541</td>	2.450960	22.664253	2.661054	22.664253	1.303734			1388	22 664253	2 250541
22.064233 1.99610 22.064233 2.00206 22.064233 2.00206 22.064233 2.00206 22.064233 2.00206 22.064233 2.00206 2.00206 2.00206 2.00206 2.00206 2.00206 2.00206 2.00206 2.00206 2.00206 2.00206 2.00206 2.00206 2.00206 2.0064233 2.00206 2.0064233 2.00206 2.0064233 2.00206 2.00206 2.00206 2.00206 2.00206 2.00206 2.00206 2.00206 2.00206 2.00206 2.0064233 2.00206 2.0064233 2.00206 2.001054 1.601493 2.001054 1.601493 2.00206 2.001054 <td>2.450960</td> <td>22.664253</td> <td>2.660363</td> <td>22.664253</td> <td>1.303734</td> <td></td> <td></td> <td>1300</td> <td>22.004255</td> <td>2.250541</td>	2.450960	22.664253	2.660363	22.664253	1.303734			1300	22.004255	2.250541
1. 244990 1. 976174 2.004509 2. 356970 22. 664253 1. 6617493 22. 664253 2. 756426 22. 664253 1. 908446 2. 112321 2. 002436 64253 1. 617493 22. 664253 2. 664253 1. 164823 1. 886331 1. 891860 2. 328635 3818 1. 526268 22. 664253 2. 664253 1. 101242 4. 308637 1. 858687 2. 064253 2. 664253 1. 68769 2. 664253 1. 101242 4. 308637 1. 885687 2. 067441 2. 664253 1. 788195 1. 65343 2. 664253 2. 064253 1. 065304 1. 849012 2. 2. 27735 64253 1. 788195 1. 65343 2. 664253 2. 664253 1. 065304 1. 849012 2. 2. 27735 64253 1. 788195 1. 65343 2. 664253 2. 664253 1. 087420 22. 664253 2. 664253 1. 087420 2. 664253 2. 664253 2. 664253 2. 664253 2. 664253 2. 664253 2. 664253 2. 664253 2. 664253 2. 664253 2. 2664253 2. 664253 2. 664253 2. 2664253 2. 664253 2. 664253	2.457871	22.664253	22.656908	22.664253	1.306498			8300	22.004253	2.252614
1.133158 22.664253 1.908446 2.112321 2.002436 3.316217 2.086059 4.537391 3.118563 22.664253 1.617493 22.664253 22.664253 1.164823 1.886331 1.81860 2.238635 3818 1.526268 22.664253 22.664253 1.101242 4.308637 1.858687 2.067441 2.664253 2.664253 2.664253 2.664253 2.664253 2.664253 2.664253 2.664253 2.664253 2.664253 1.653703 1.69766 22.664253 2.0664253 2.0664253 2.0664253 2.7664253 1.788195 1.65343 22.664253 2.4664253 2.0564253 2.79723 2.2664253 2.7400 2.190415 1.67139 22.664253 2.464253 2.019023 2.664253 2.019023 2.664253 2.464253 2.464253 2.464253 2.464253 2.264253 2.20131 1.67139 2.264253 2.264253 2.20131 1.617493 2.264253 2.20132 1.617493 2.264253 2.20132 1.617493 2.264253 2.20131 1.617493 2.264253 1.6125	1.244990	1.976174	2.004509	2.356970	22.664253			9881	1.555294	22.664253
3.1021/ 22.664253 2.164223 1.684253 1.684253 1.684253 2.264253 2.664253	1.193158	22.664253	1.908446	2.112321	2.002436			64253	1.617493	22.664253
2.111567 2.788908 2.090206 22.664253 22.664253 22.664253 2645 1.585703 1.69766 2.2.664253 2.070855 4.485559 22.664253 2.664253 1.65343 1.69766 2.2.664253 2.0664253 2.050813 4.466898 22.664253 64253 1.788195 1.65343 2.0664253 2.064253 2.064253 2.064253 2.786143 2.019023 22.664253 2.740 2.190415 1.67139 22.664253 2.064253 2.064253 2.064253 2.786143 2.019023 22.664253 2.4299 22.664253 1.74050 22.664253 2.664253 2.664253 2.664253 2.664253 2.664253 2.24224 2.264253 2.2	22.664253	1.164823	1.886331	1.891860	2.328635		10,000	3818	1.526268	22.664253
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	2.411567	2.788908	2.090206	22.664253	22.664253			2645	1.585703	1.697661
22.004233 22.004233 1.003304 1.043012 22.27733 1.05343 1.05343 1.05343 22.006233 2.004233 1.023147 3.835233 2.2664253 2.2664253 2.190415 1.67139 22.664253 22.664253 2.7664253 2.7664253 2.2664253 2.40824 2499 22.664253 1.67139 22.664253 2.2664253 1.087420 22.664253 3.797223 2499 22.664253 1.74050 22.664253 2.2664253 0.993430 1.832425 5410 22.664253 2.264253 2.2013 1.815839 22.664253 2.664253 2.000363 5043 2.344531 2.2013 1.815839 22.664253 2.664253 2.001054 5043 2.344531 2.21736 1.815839 22.664253 2.664253 2.001054 5043 2.344531 2.21451 1.815839 22.664253 2.664253 2.001054 1.815839 3.524238 2.095734 2.345222 2.22151 1.815839 22.664253 2.664253 2.001054 1.815839 3.524238 2.095734 <td>2.382541</td> <td>2.786835</td> <td>2.070855</td> <td>4.485559</td> <td>22.664253</td> <td></td> <td></td> <td>64252</td> <td>1 799105</td> <td>1 652421</td>	2.382541	2.786835	2.070855	4.485559	22.664253			64252	1 799105	1 652421
22.664253 4.712930 1.023147 3.835233 2.20824 2740 2.190415 1.67139 22.664253 22.664253 1.087420 22.664253 3.797223 2499 22.664253 1.74050 22.664253 2.428845 22.664253 1.991378 4.363234 2499 22.664253 2.20133 1.815839 2.2664253 2.001364 5043 2.344531 2.21736 22.664253 22.664253 22.664253 2.064253 2.001054 5043 2.344531 2.21736 22.664253 22.664253 2.664253 2.001054 5043 2.344531 2.222427 1.815839 3.524238 2.095734 2.344531 2.22427 1.815839 22.664253 22.664253 2.664253 2.001054 2.344531 2.22427 2.264253 2.64253 2.64253	2.406730	22.664253	2.050813	4.466898	22.664253	10		64253	1./88195	1.053431
22. 664253 22. 664253 1.087420 22. 664253 1.79723 22. 664253 2.42845 22. 664253 1.991378 4.363234 22. 664253 2.42845 22. 664253 2.2. 22013 1.815839 2.2. 664253 2.344531 2.2. 21736 1.815839 22. 664253 22. 664253 2.001054 5043 2.344531 2.2. 22151 1.815839 3. 524238 2.095734 2.344531 2.2. 22427 1.815839 22. 664253 22. 664253 2.001054 1.815839 3. 524238 2.095734 2.344531 2.222427 1.815839 22. 664253 22. 664253 2.001054 2.2. 664253	22.664253	4.712930	1.023147	3.835233	2.220824).		2740	2.190415	1.671399
22.604233 2.428845 22.604233 1.9913/8 4.39234 22.604253 22.604253 22.604253 22.604253 2.264253 2.24224 22.604253 2.428845 22.604253 22.604253 2.000363 5043 2.34531 2.22013 1.815839 22.604253 22.604253 2.001054 5043 2.344531 2.21736 1.815839 22.604253 22.604253 2.001054 5043 2.344531 2.21736 1.815839 22.604253 22.604253 2.001054 5043 2.344531 2.21736 1.815839 22.604253 22.604253 2.001054 5043 2.344531 2.2221 1.815839 22.604253 22.604253 2.001054 5043 2.344531 2.22427 1.815839 22.604253 22.604253 2.001054 2.344531 2.2222 2.22151 1.815839 22.604253 22.604253 22.604253 2.001054 2.344531 2.2222 2.22082 1.815839 22.604253 22.604253 22.604253 2.001054 2.345222 2.2082 2.2082	22.664253	22.664253	1.087420	22.664253	3.797223		a. E ≤	2499	22.664253	1.740509
22.664253 2.428845 22.664253 2.000363 5043 2.344531 2.22013 1.815839 22.664253 2.664253 2.664253 2.001054 5043 2.344531 2.21736 22.664253 22.664253 2.664253 2.001054 5043 2.344531 2.21736 22.664253 22.664253 22.664253 2.001054 5043 2.34531 2.21736 22.664253 22.664253 2.664253 2.001054 5043 2.34531 2.21736 1.815839 22.664253 2.664253 2.001054 5043 2.34531 2.222151 1.815839 22.664253 2.664253 2.001054 2.34531 2.222427 1.815839 22.664253 22.664253 2.001054 2.34531 2.22427 1.815839 22.664253 22.664253 2.001054 2.34531 2.22427 1.815839 22.664253 2.664253 2.001054 2.34531 2.22427 1.815839 22.664253 2.664253 2.001054 2.345322 2.345322 2.20822 1.815839 2.664253 2.66	22.664253	22.664253	22.664253	0.993430	1.832425			5410	22.664253	2.242248
1.815839 22.664253 22.664253 2.001054 22.664253 22.664253 2.664253 2.001054 22.664253 22.664253 2.664253 2.001054 22.664253 22.664253 2.664253 2.001054 22.664253 22.664253 2.001054 2.344531 2.21736 22.664253 22.664253 2.664253 2.001054 2.345222 2.22151 1.815839 22.664253 22.664253 2.001054 2.344531 2.22427 1.815839 22.664253 22.664253 2.001054 2.344531 2.22427 1.815839 22.664253 22.664253 2.001054 2.344531 2.22427 1.815839 22.664253 22.664253 2.001054 2.345222 2.2082 1.815839 22.664253 22.664253 2.001054 2.345222 2.2082 1.815839 22.664253 22.664253 2.001054 2.345222 2.2082 1.815839 22.664253 2.664253 2.001054 2.345222 2.2082 1.815839 22.664253 2.664253 2.001054 2.2	22.664253 22.664253	2.428845 22.664253	22.664253 4.680449	22.664253 0.993430	2.000363 22.664253		and the second	5043	2.344531	2,220133
121:004213 22:064213 20:064213 20:064213 20:064213 20:041213	1.815839	22.664253	22.664253	22.664253	2.001054			5042	2 244521	2 217268
22.664253 22.664253 4.691306 0.990666 22.664253 1.815839 3.524238 2.095734 2.345222 2.22111 1.815839 22.664253 22.664253 22.664253 2.001054 2.264253 2.001054 2.345222 2.22427 1.815839 22.664253 22.664253 22.664253 2.664253 2.664253 2.34531 2.22427 1.815839 22.664253 22.664253 2.664253 2.664253 2.664253 2.34531 2.22427 1.815839 22.664253 22.664253 2.664253 2.664253 2.001054 2.345222 2.2082 1.815839 22.664253 2.664253 2.001054 2.345222 2.2082 1.815839 22.664253 2.664253 2.001054 2.345222 2.2082 1.815839 22.664253 2.664253 2.001054 2.345222 2.339693 2.21736 1.815839 22.664253 2.664253 2.001745 2.094352 2.339693 2.21736 1.809619 22.664253 2.664253 2.064253 2.064253 2.064253 2.064253 2.664253	1.815839	22.664253	22.664253	22.664253	2.001054			5045	2.344331	2.21/308
22.664253 22.664253 4.680449 0.993430 22.664253 1.815839 3.524238 2.095734 2.344531 2.22427 1.815839 22.664253 22.664253 22.664253 2.001054 2.34522 2.22082 1.815839 22.664253 22.664253 22.664253 2.001054 2.34522 2.2082 1.815839 22.664253 22.664253 22.664253 2.001054 2.34522 2.2082 1.815839 22.664253 22.664253 22.664253 2.001054 2.34522 2.339693 2.21736 1.809619 22.664253 2.664253 2.001745 1.814457 2.156551 2.094352 2.339693 2.21642 1.807546 22.664253 2.664253 2.001745 3.797223 1.822059 2.347986 2.070164 22.6642 1.807546 22.664253 22.664253 22.664253 2.664253 2.664253 2.664253 2.664253 2.347986 2.070164 22.6642 1.807546 22.664253 22.664253 22.664253 22.664253 2.664253 2.34582 2.343148 2.30168 <tr< td=""><td>22.664253 1.815839</td><td>22.664253 22.664253</td><td>4.691506 22.664253</td><td>0.990666 22.664253</td><td>22.664253 2.001054</td><td>1.815839</td><td>3.524238</td><td>2.095734</td><td>2.345222</td><td>2.221515</td></tr<>	22.664253 1.815839	22.664253 22.664253	4.691506 22.664253	0.990666 22.664253	22.664253 2.001054	1.815839	3.524238	2.095734	2.345222	2.221515
22.664253 22.664253 22.664253 22.664253 22.664253 22.664253 22.664253 22.664253 22.664253 22.664253 22.664253 22.664253 2.001054 22.664253 2.001054 22.664253 2.001054 22.664253 2.001054 2.001745 2.001745 2.001745 2.001745 2.001745 2.001745 2.001745 2.001745 2.001745 2.001745 2.001745 2.001745 2.001745 2.001745 2.0010	22.664253	22.664253	4.680449	0.993430	22.664253	1.815839	3.524238	2.095734	2.344531	2.224279
1.610539 22.004233 22.004233 22.004233 22.004233 22.004233 22.004233 22.004233 22.004233 22.004233 22.004233 22.004233 22.004233 22.004233 2.001034 22.004233 2.001034 22.004233 2.001034 22.004233 2.001034 22.004233 2.001034 2.0010	22.664253	22.664253	4.676993	0.991357	22.664253	1.815839	3.524238	2.095734	2.345222	2.220824
1.809619 22.664253 2.470311 22.664253 2.001745 4.424050 22.664253 22.664253 1.10226 22.664253 3.797223 1.822059 2.347986 2.070164 22.66425 1.807546 22.664253 22.664253 22.664253 22.664253 22.664253 2.070164 22.66425 2.054268 4.423244 22.664253 22.664253 22.664253 1.003797 1.808237 1.823441 2.343148 2.30168 2.142729 1.689368 22.664253 22.664253 22.664253 1.003797 1.808237 1.823441 2.343148 2.30168	22.664253	22.664253	4.682522	22.004253 0.995503	22.664253	1.814457	2.156551	2.094352	2.339693	2.217368
1.807546 22.664253 22.6642	1.809619 4.424050	22.664253 22.664253	2.470311 22.664253	22.664253 1.110226	2.001745	3,797223	1.822059	2,347986	2,070164	22,664253
2.094208 4.492344 22.004233 22.004233 22.004233 1.003/9/ 1.80823/ 1.823441 2.343148 2.30188 2.142729 1.689368 22.664253 2.464782 22.664253	1.807546	22.664253	22.664253	22.664253	22.664253	1 002707	1 0000007	1 922441	2.0/0104	2 201602
	2.142729	4.432344 1.689368	22.664253	22.004253	22.664253	1.003/9/	1.808237	1.823441	2.343148	2.301682

Consider the Various Tasks



What does the robot need to know? How to get it?

Some Requirements

- Models of motion
 - Own dynamics
 - Object dynamics
 - Other agents
- Models of environment
 - Space & how I move in space
 - Other navigation considerations
- Models of self
 - What is the connection between my sensors and actuators?
 - What do the sensorimotor channels even mean?
 - How to ground all of the above at this low level?

Linear Time Invariant (LTI) Systems

- Consider the simple spring-mass-damper system:
- The force applied by the spring is $F_s = -kz(t)$
- Correspondingly, for the damper: $F_d = \gamma \dot{z}(t)$
- The combined equation of motion of the mass becomes:

$$x\ddot{z}(t) = -\gamma\dot{z}(t) - kz(t)$$

• One could also express this in state space form:

$$\begin{aligned} x(t) &= [x_1(t), x_2(t)]' = [z(t), \dot{z}(t)]' \\ \dot{x}(t) &= \begin{pmatrix} \dot{z}(t) \\ \ddot{z}(t) \end{pmatrix} = \begin{pmatrix} x_2(t) \\ -\frac{1}{m}(\gamma x_2(t)) + kx_1(t) \end{pmatrix} \\ \dot{x}(t) &= \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & \frac{\gamma}{m} \end{pmatrix} x(t) = Ax(t) \end{aligned}$$



How to Get Models?

- One may not always want to derive analytical models, one often obtains them from experiment
 - System Identification
- By observing inputs and outputs of a system, one tries to fit a suitable model that captures the dynamics
- The model may then be used as a basis for controller synthesis, planning, etc.
- In robotics, you need to do this at *many* different levels
 - Low level control of sensors and actuators
 - "Maps" of the world and objects
 - Models of people and other agents

Low-level System Identification

• In order to design a controller, you **must** have a **model**



- Model system that transforms an input signal into output signal.
- Often, may be represented by differential or difference equations.
- A model is used, either explicitly or implicitly, within control design

Signals & Systems Definitions



- Typically, a system receives an input signal, x(t) (generally a vector) and transforms it into the output signal, y(t). In modelling and control, this is further broken down:
- u(t) is an input signal that can be manipulated (**control signal**)
- w(t) is an input signal that can be measured (measurable disturbance)
- v(t) is an input signal that cannot be measured (unmeasurable disturbance)
- *y*(*t*) is the **output signal**
- Typically, the system may have hidden states x(t)

Example: Solar-Heated House (Ljung)





0

- The sun heats the air in the solar panels
- The air is pumped into a heat storage (box filled with pebbles)
- The stored energy can be later transferred to the house
- We're interested in how solar radiation, w(t), and pump velocity, u(t), affect heat storage temperature, y(t).

Example: Military Aircraft (Ljung)

- Aim is to construct a mathematical model of dynamic behaviour to develop simulators, synthesis of autopilots and analysis of its properties
- Data below used to build a model of pitch channel, i.e. how pitch rate, y(t), is affected by three separate control signals: elevator (aileron combinations at the back of wings), canard (separate set of rudders at front of wings) and leading flap edge



Figure 1.5 The Swedish fighter aircraft JAS-Gripen.



SAAB AB, Sweden. The pitch rate and the elevator leading edge flap, and considered by

Identifying a System

- Models of car behaviour (acceleration/steering) are built from experience/observational data
- Generally, there may exist some prior knowledge (often formed from earlier observational data) that can be used with the existing data to build a model. This can be combined in several ways:
- Use past experience to express the equations (ODEs/difference equations) of the system/sub-systems and use observed data to estimate the parameters
- Use past experience to specify prior distributions over parameters
- The term modelling generally refers to the case when substantial prior knowledge exists, the term system identification refers to the case when the process is largely based on observed inputoutput data.

Auto-Regressive with eXogenous inputs Model

• Consider this ARX model with no disturbances

 $y(t) + y_1 y(t-1) + \dots + y_n y(t-2) = y_1 u(t-1) + \dots + y_m u(t-2)$

• This can be used for prediction using:

 $y(t) = - y(t - 1) - \dots - y_n y(t - 1) + y_1 u(t - 1) + \dots + y_m u(t - 1)$

and introducing the vectors:

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\mu}_1 & \cdots & \boldsymbol{a}_n & \boldsymbol{b}_1 & \cdots & \boldsymbol{b}_m \end{bmatrix}^{-1}$$

$$\mathbf{x}(t) = \underline{-} (t-1) & \cdots & - (t-1) & u(t-1) & \cdots & u(t-1) \end{bmatrix}^{-1}$$

- the model can be written as $y(t) = z'(t)\theta$
- Note, that the prediction is a function of the estimated parameters which is sometimes written as

$$\hat{y}(t \,|\, \boldsymbol{\theta}) = \boldsymbol{\zeta}(t) \boldsymbol{\theta}$$

Least Squares Estimation

- By manipulating the control signal u(t) over a period of time $1 \le t \le N$, we can collect the data set: $Z^{N} = u(1), v(1), \dots, u(N), v(N)$
- Lets assume that the data is generated by: $y(t) = \mathbf{x}'(t)\mathbf{\theta} + \mathbf{x}', \sigma$
- Where θ is the "true" parameter vector and $\aleph(0, \sigma^2)$ generates zero mean, normally distributed **measurement noise** with standard deviation σ .
- We want to find the estimated parameter vector, $\hat{\theta}$, that "best fits" this data.
- Note that because of the random noise, we can't fit the data exactly, but we can minimise the prediction errors squared using $\hat{\theta} = \mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{y}$

Here, **X** is the matrix formed from input vectors (one row per observation, one column per input/parameter) and **y** is the measured output vector (one row per observation

Example: First Order ARX model

• Consider the simple, first order, linear difference equation, ARX model: y(t) + ay(t -) = bu(t -)

where 10 data points $Z^{10} = \{u(1), y(1), ..., u(10), y(10)\}$ are collected.

• This produces the (9*2) input matrix and (9*1) output vector:

$$\mathbf{X} = \begin{bmatrix} y(1) & u(1) \\ y(2) & u(2) \\ \vdots & \vdots \\ y(9) & u(9) \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y(2) \\ y(3) \\ \vdots \\ y(10) \end{bmatrix}$$

• Therefore the parameters $\theta = [a \ b]^T$ can be estimated from

$$\hat{\boldsymbol{\theta}} = \mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{y}$$

In Matlab: thetaHat = inv(X'*X)*X'*y

Model Quality and Experimental Design

The variance/covariance matrix to be inverted

$$\mathbf{X}'\mathbf{X} = \begin{bmatrix} \sum_{t} x_{1}^{2}(t) & \sum_{t} x_{1}(t)x_{2}(t) & \cdots & \sum_{t} x_{1}(t)x_{n}(t) \\ \sum_{t} x_{2}(t)x_{1}(t) & \sum_{t} x_{2}^{2}(t) & \cdots & \sum_{t} x_{2}(t)x_{n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{t} x_{n}(t)x_{1}(t) & \sum_{t} x_{n}(t)x_{2}(t) & \cdots & \sum_{t} x_{n}^{2}(t) \end{bmatrix}$$

- Strongly determines the quality of the parameter estimates. This in turn is determined by the distribution of the measured input.
- Control *signal should be chosen* to make the matrix as wellconditioned as possible (similar eigenvalues)
- Number of training data & sampling time both affect the accuracy and condition of the matrix
- Experimental design: choose experiments to optimally estimate model parameters

System Identification Process

- In building a model, the designer has control over three parts of the process
- 1. Generating the **data set** Z^N
- Selecting a (set of) model structure (ARX for instance)
- 3. Selecting the **criteria** (least squares for instance), used to specify the optimal parameter estimates
- A very popular approach involves (recursive) parameter estimation



Some Questions

Who makes all the decisions? How to deploy that aspect?

- What variables should I be using for the model?
 - Egocentric/allocentric frames, transformations, etc.
- Once we have such a model, we can define control tasks in terms of this where do the specifications come from?
- What happens at the higher levels (maps, objects) and how do we tie this to that?

A Bootstrap Learning Framework

- Learn models of robot and environment with no initial knowledge of what sensors and actuators are doing
- Many learning methods begin this way, e.g., RL, but the goal here is to construct a representation incrementally and continually as well



D. Pierce, B.J. Kuipers, Map learning with un-interpreted sensors and effectors, *Artificial Intelligence* 91:169-227, 1997.

Simple Scenario

- Robot critter has a set of distance sensors (range) – one of which is defective – but it doesn't know that yet
- Other sensors: battery power, digital compass
- It has a track-style motor apparatus – turn by differentially actuating its wheels



What do you learn from?

Randomized actions (hold a randomly chosen action for 10 time steps), repeatedly applied



How does environment appear in the data?

A Simple but Complete Procedure

Problem.

- Given: A robot with an uninterpreted, almost-everywhere approximately linear sensorimotor apparatus in a continuous, static environment.
- Learn: Descriptions of the structure of the robot's sensorimotor apparatus and environment and an abstract interface to the robot suitable for *prediction* and *navigation*.

Solution.

Representation: A hierarchical model. At the bottom of the hierarchy are egocentric models of the robot's sensorimotor apparatus. At the top of the hierarchy is a discrete abstraction of the robot's environment defined by a set of discrete views and actions. *Method*: A sequence of statistical and generate-and-test methods for learning the objects of the hierarchical model.

One Step: Go from Raw Channels to Structure of Sensor Array

- Sensors may come in groupings: ring of distance sensors, array of photoreceptors, video camera, etc.
- We first want to extract groupings based on two criteria:
 - Sensors that have similar values over time
 - Sensors that have a similar frequency domain behaviour
- Two distance metrics:

$$d_{1,ij}(t) = \frac{1}{t+1} \sum_{\tau=0}^{t} |x_i(\tau) - x_j(\tau)|.$$

$$d_{2,ij} = \frac{1}{2} \sum_{l} |(\textit{dist } x_i)_l - (\textit{dist } x_j)_l|,$$

distribution

Example Trace



$$i \approx_k j$$
 if $d_{k,ij} < \min\{\varepsilon_{k,i}, \varepsilon_{k,j}\}.$

$$\varepsilon_{k,i}=2\min_{j}\{d_{k,ij}\}.$$

Extending the Group Notion

We can reason transitively about similarity:

 $i \sim j$ iff $i \approx j \lor \exists k$: $(i \sim k) \land (k \sim j)$.

So, a wandering trace might yield something like this as groups:

 $(0\ 1\ 2\ 22\ 23)\ (0\ 1\ 2\ 3\ 23)\ (0\ 1\ 2\ 3\ 4)\ (1\ 2\ 3\ 4\ 5)\ (2\ 3\ 4\ 5\ 6)\ (3\ 4\ 5\ 6\ 7)\ (4\ 5\ 6\ 7)\ (5\ 6\ 7\ 8\ 9)\ (7\ 8\ 9\ 10)\ (7\ 8\ 9\ 10\ 11)\ (8\ 9\ 10\ 11\ 12)\ (9\ 10\ 11\ 12\ 13)\ (10\ 11\ 12\ 13)\ (10\ 11\ 12\ 13\ 14\ 15)\ (12\ 13\ 14\ 15\ 16)\ (13\ 14\ 15\ 16\ 17)\ (14\ 15\ 16\ 17\ 18\ 19)\ (17\ 18\ 19\ 21)\ (17\ 18\ 19\ 21)\ (20)\ (19\ 21\ 22\ 23)\ (0\ 21\ 22\ 23)\ (0\ 1\ 21\ 22\ 23)\ (24)\ (25)\ (26)\ (27)\ (28).$

After Transitive Closure

(0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 21 22 23)

(20) defective

- (24) battery voltage
- (25) east
- (26) *north*
- (27) west

(28) south



Getting at Structure of Array

- Task is to find an assignment of positions (in space) to elements that captures the structure of the array as reflected in distance metric d₁.
- Distance between positions in image ≈ distance between elements according to d₁.

$$\|(pos y_i) - (pos y_j)\| = d_{1,ij},$$

- This is a constraint satisfaction problem: n sensor elements yield n(n-1)/2 constraints.
- Solve by metric scaling: $E = \frac{1}{2} \sum_{ij} (\|(pos y_i) (pos y_j)\| d_{ij})^2.$

Structural Model of Distance Array



Roving Eye



If the array slowly roves over a much larger image, can you recover the structure of this array?

Roving Eye Robot









digital compass



turn-slip-slide

d₁ and d₂ for Roving Eye (after 5 mins)





Metric Scaling Procedure



Where are we going with this?

