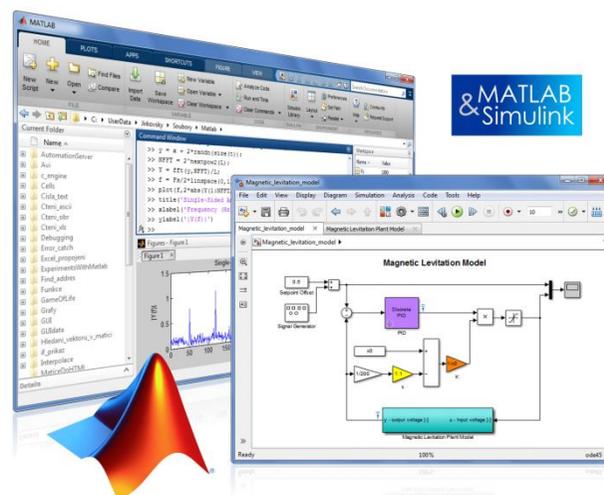


20.11.2019 Technical Computing Prague

Robotic systems development in MATLAB



Michal Blaho

blaho@humusoft.sk

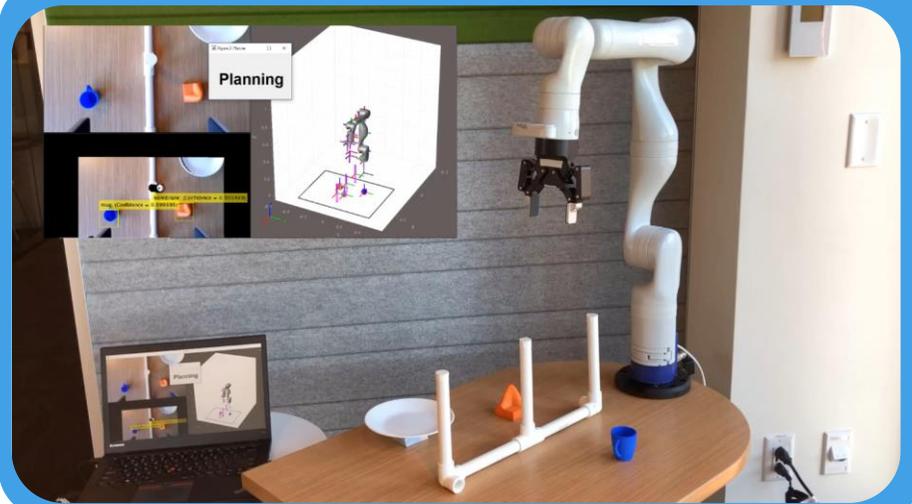
www.humusoft.cz

info@humusoft.cz

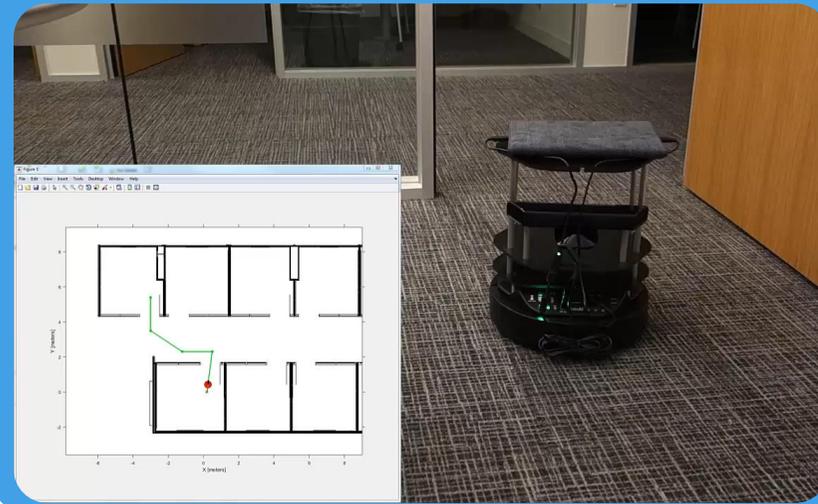
www.mathworks.com

Robot Applications

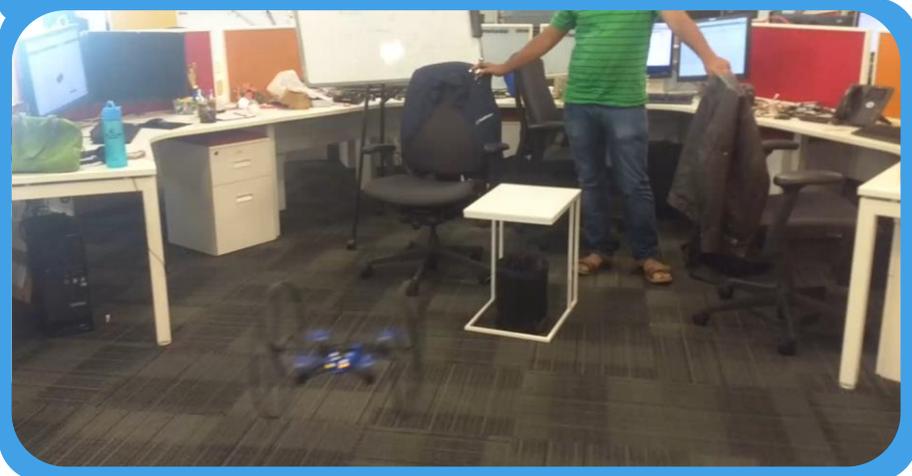
Manipulator Arms



Mobile Robots



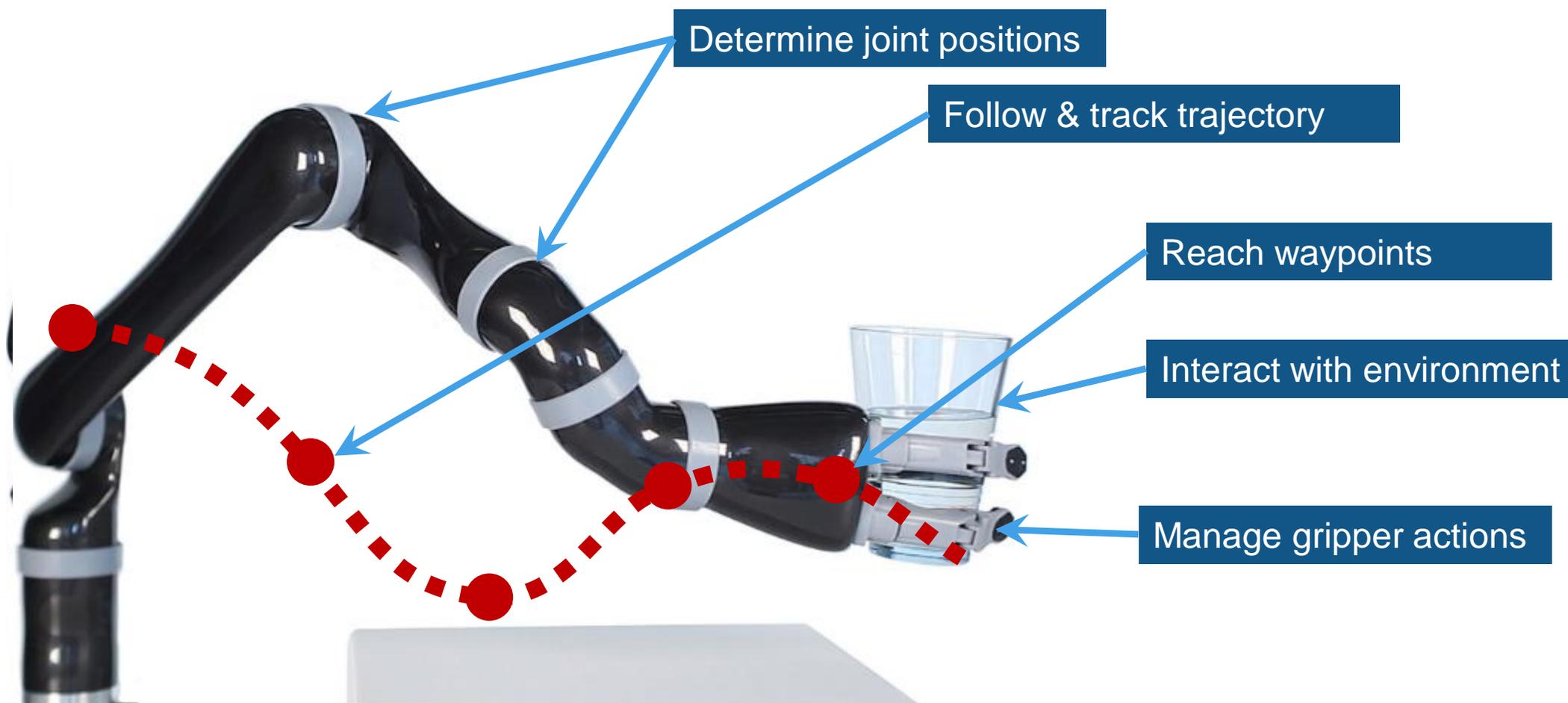
UAVs



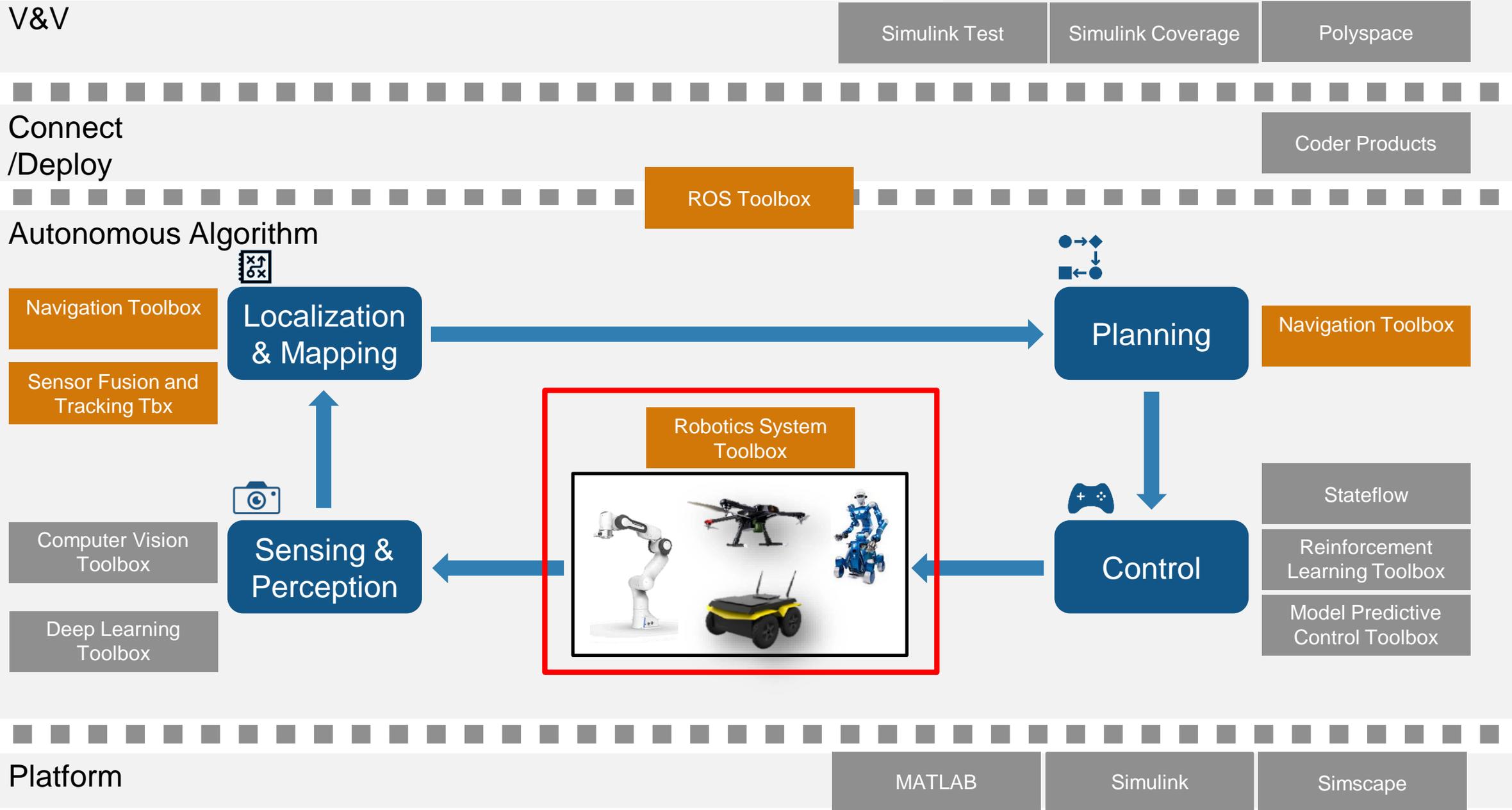
Humanoids



Challenges in Designing Robotics System



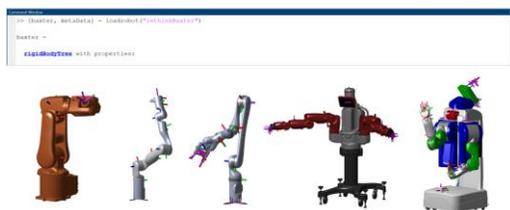
Components of Robotics System Development



Robotics System Toolbox

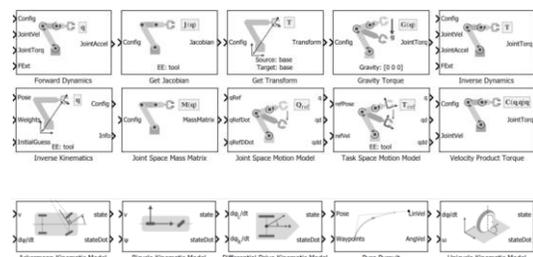
Robot Model

- Rigid body tree (RBT) representation
- Robot model library of commercially available robots
- Import RBT from URDF file, text, Simscape multibody model



Algorithms

- Library of common robot algorithms
 - Manipulator arms
 - Mobile robots
 - UAVs



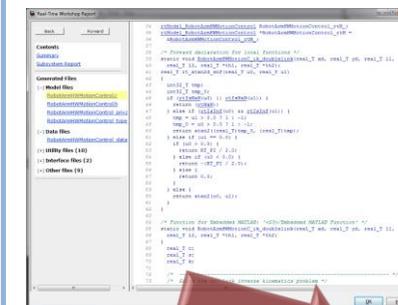
Simulation

- Motion modeling and simulation to prototype algorithms quickly and test behavior of real-world systems.
- Synchronized Gazebo co-simulation from Simulink



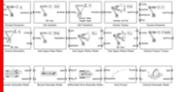
Deployment

- Generate C/C++ code and MEX functions for algorithm acceleration



Robot Model

- Rigid body tree (RBT) representation
- Load a RBT robot model from a library of commonly used robots
- Import a RBT robot mode from URDF file

Robot Model	Algorithms	Simulation	Code Generation
<ul style="list-style-type: none"> • Rigid body tree (RBT) representation • Robot model library of commercially available robots • Import RBT from URDF file, text, Simscape • Multibody model 	Library of common robot algorithms <ul style="list-style-type: none"> o Manipulator arms o Mobile robots o UAVs 	<ul style="list-style-type: none"> • Motion modeling and simulation to prototype algorithms quickly and test behavior of real-world systems. • Synchronized Gazebo co-simulation from Simulink 	<ul style="list-style-type: none"> • Generate C/C++ code and MEX functions for algorithm acceleration 

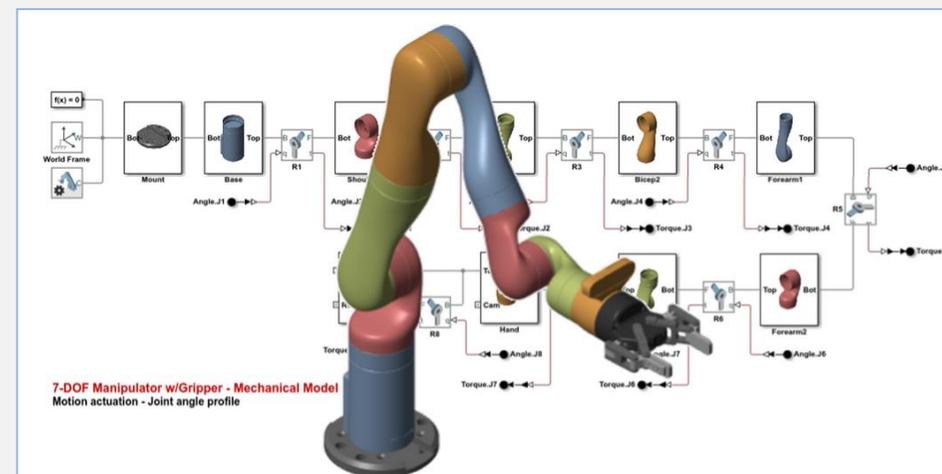
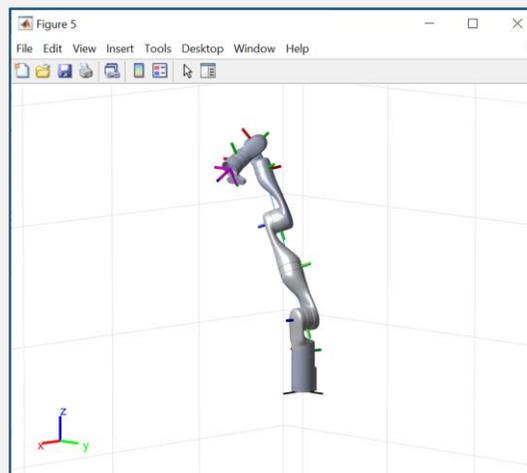
```
>> robot = loadrobot("kinovaGen3")
```

```
robot =
```

```
rigidBodyTree with properties:
```

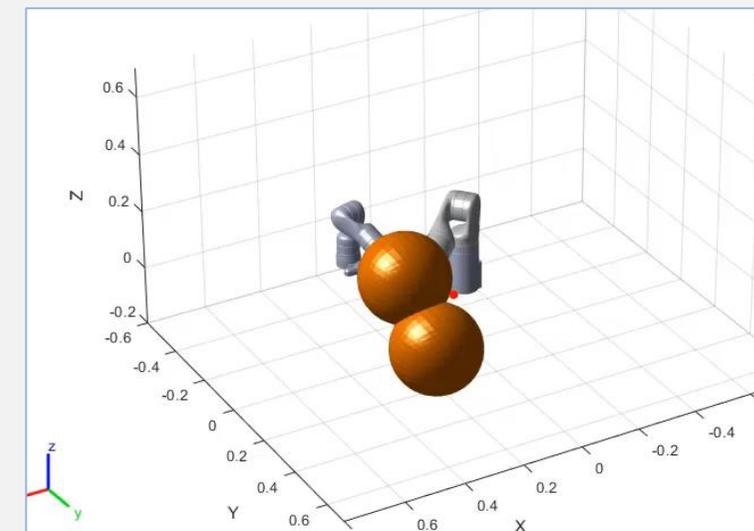
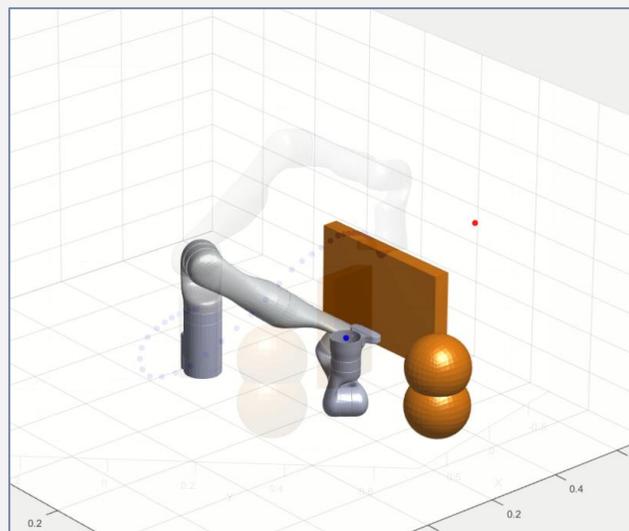
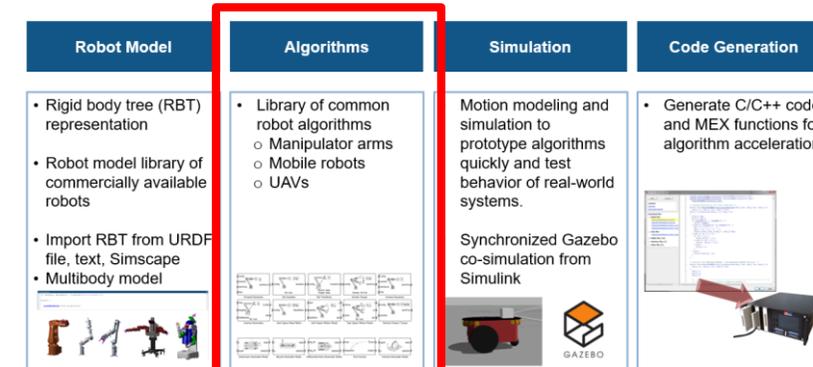
```

NumBodies: 8
  Bodies: {1x8 cell}
    Base: [1x1 rigidBody]
  BodyNames: {1x8 cell}
  BaseName: 'base_link'
  Gravity: [0 0 0]
  DataFormat: 'struct'
```



Algorithms - Manipulation

- Forward and inverse kinematics
- Generalized inverse kinematics & constraints
- Forward and inverse dynamics
- Trajectory generation
- Collision checking



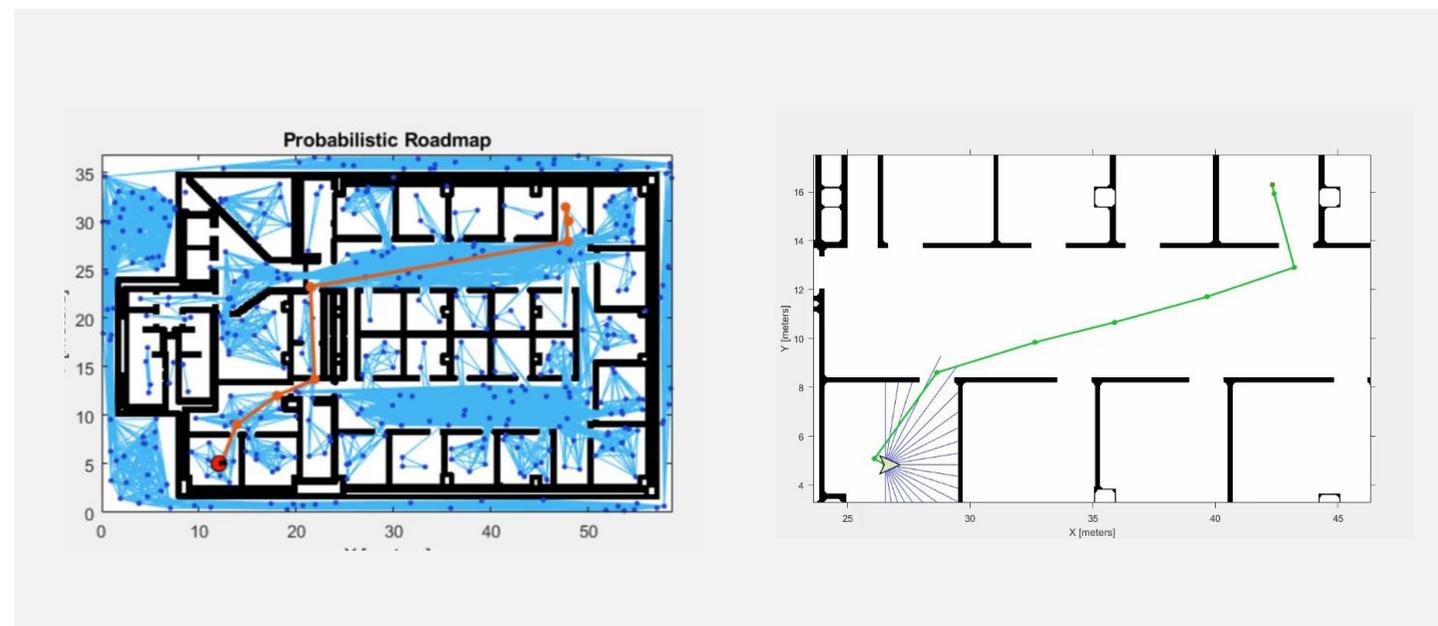
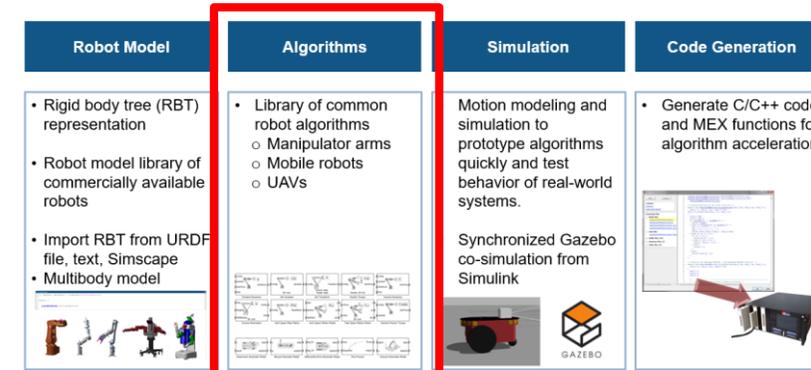
Algorithms – Mobile Robots

- Mapping and map representation
 - Binary occupancy grid

- Localization
 - Odometry
 - stateEstimatorPF

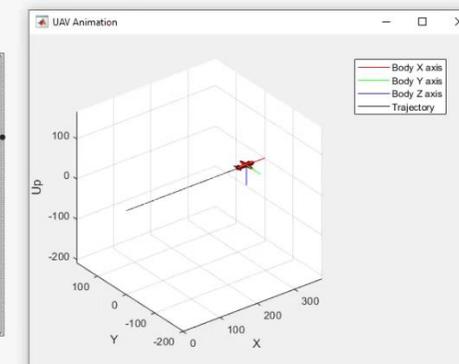
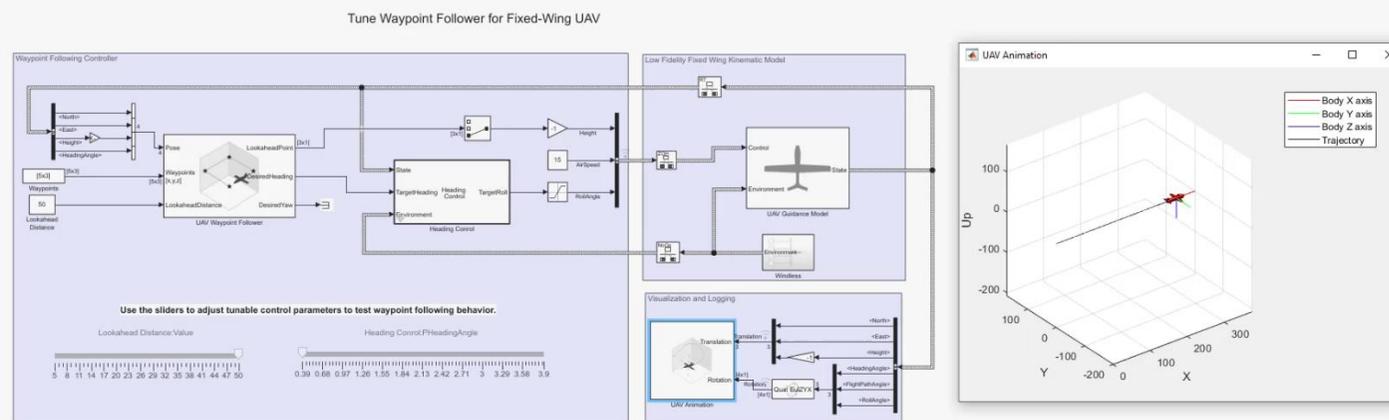
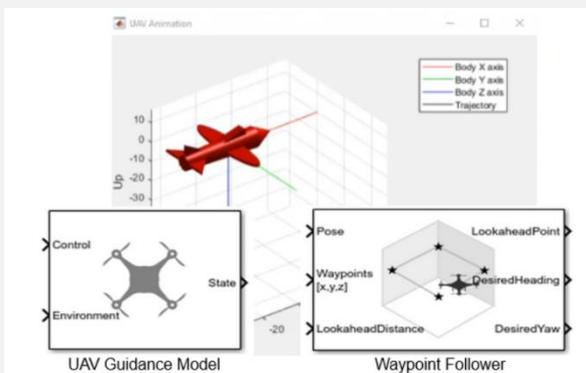
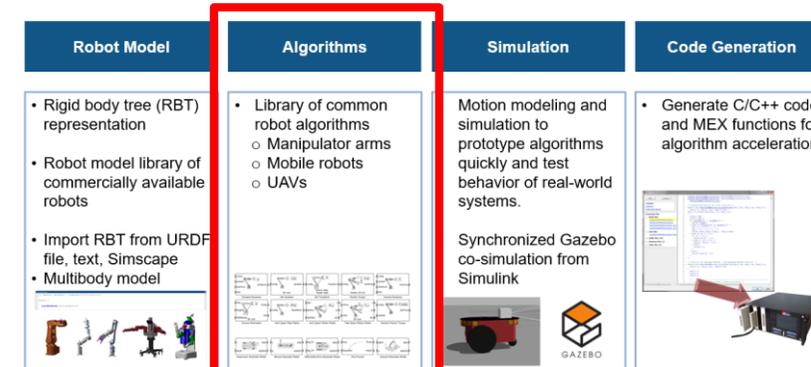
- Path planning
 - Probabilistic roadmap (PRM)

- Path following
 - Pure pursuit



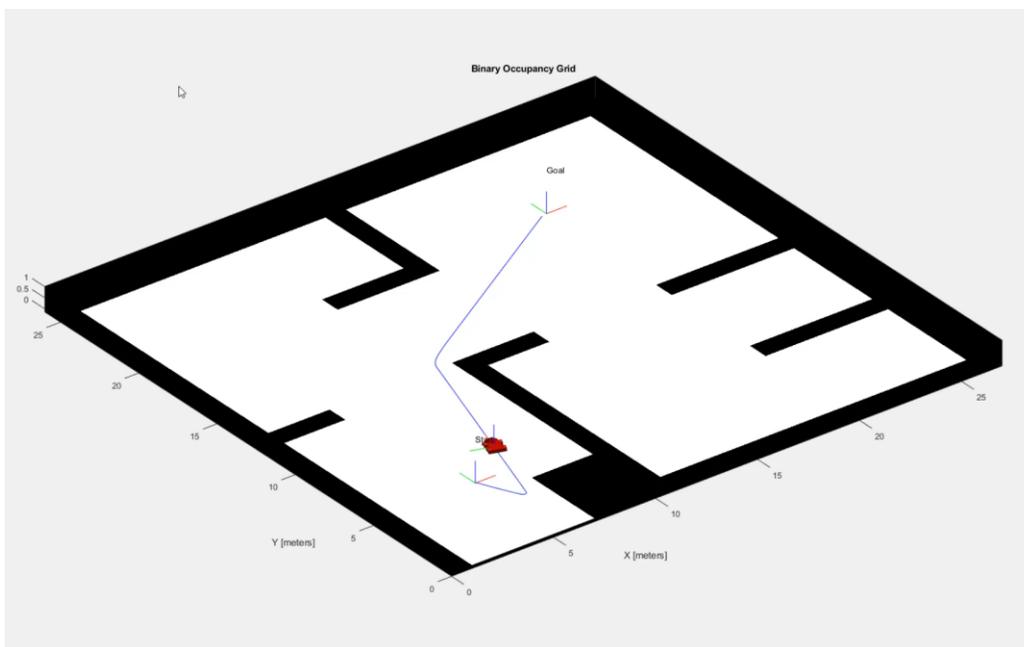
Algorithms – UAVs (Add-On Library)

- Guidance models
 - Reduced-order guidance model for fixed-wing and multi-rotor UAVs
- MAVLink communication
 - Communicate with simulated/physical UAV
 - Import and analyze UAV flight logs
- Waypoint following
 - Execute flight missions based on given waypoints

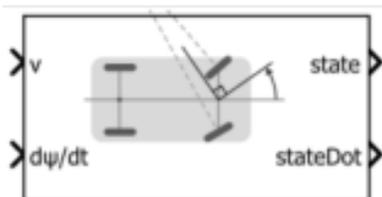
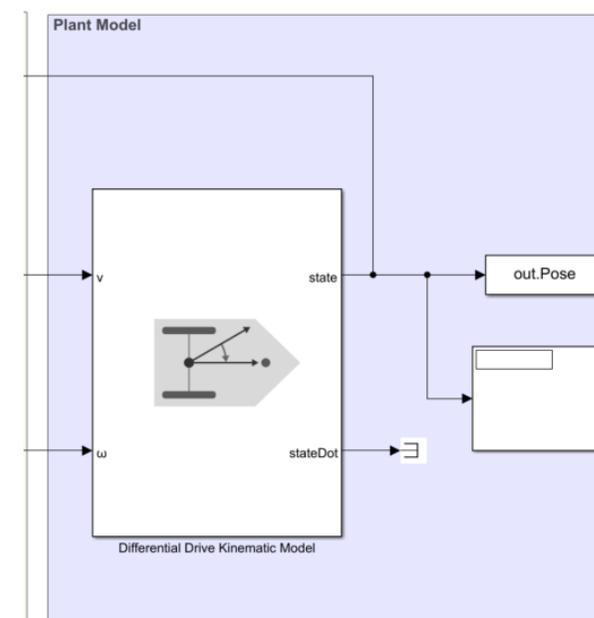


Simulation

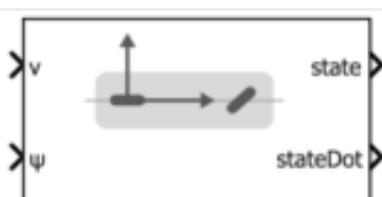
- Low-fidelity simulation



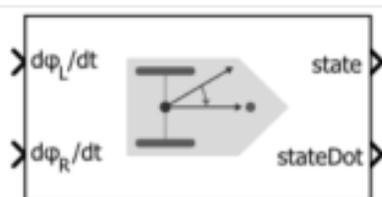
Robot Model	Algorithms	Simulation	Code Generation
<ul style="list-style-type: none"> Rigid body tree (RBT) representation Robot model library of commercially available robots Import RBT from URDF file, text, Simscape Multibody model 	<ul style="list-style-type: none"> Library of common robot algorithms <ul style="list-style-type: none"> Manipulator arms Mobile robots UAVs 	<ul style="list-style-type: none"> Motion modeling and simulation to prototype algorithms quickly and test behavior of real-world systems. Synchronized Gazebo co-simulation from Simulink 	Generate C/C++ code and MEX functions for algorithm acceleration



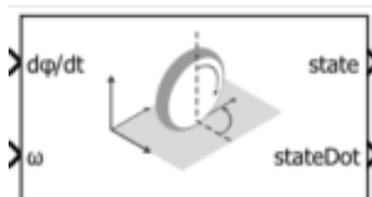
Ackermann Kinematic Model



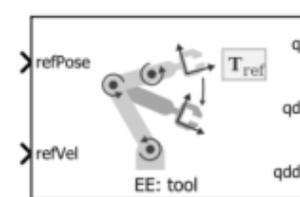
Bicycle Kinematic Model



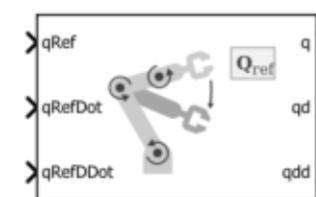
Differential Drive Kinematic Model



Unicycle Kinematic Model



Task Space Motion Model

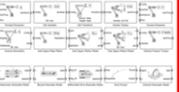


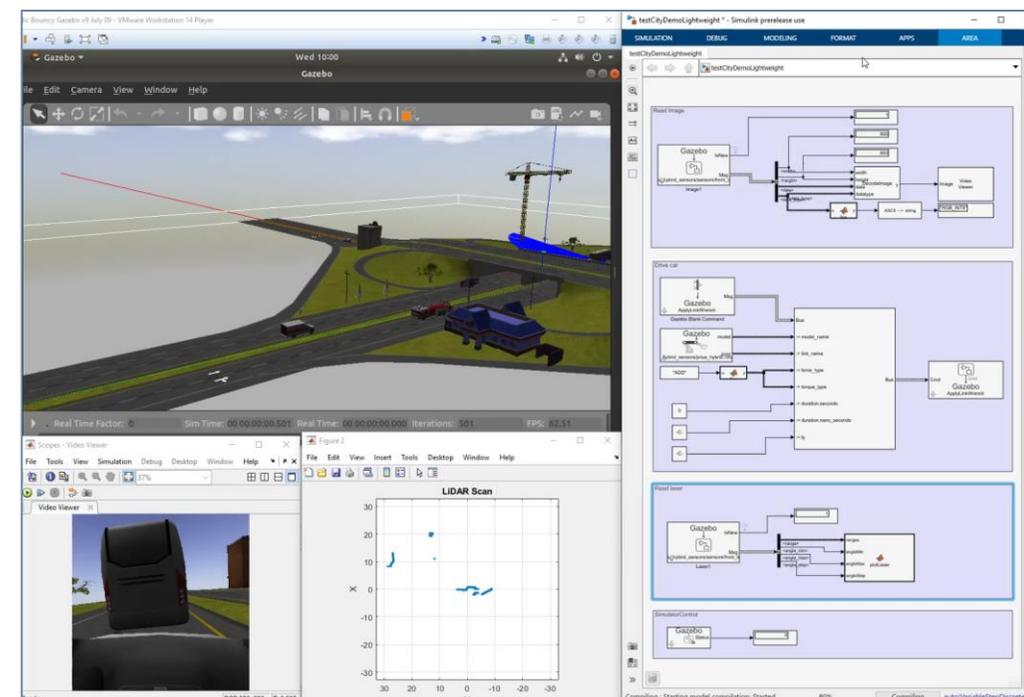
Joint Space Motion Model

Simulation

- Gazebo Co-simulation

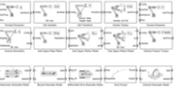
- Provides synchronized stepping between Simulink and Gazebo simulator
- Retrieve sensor data and ground truth pose for models from Gazebo simulator
- Actuate model links and joints in Gazebo simulator

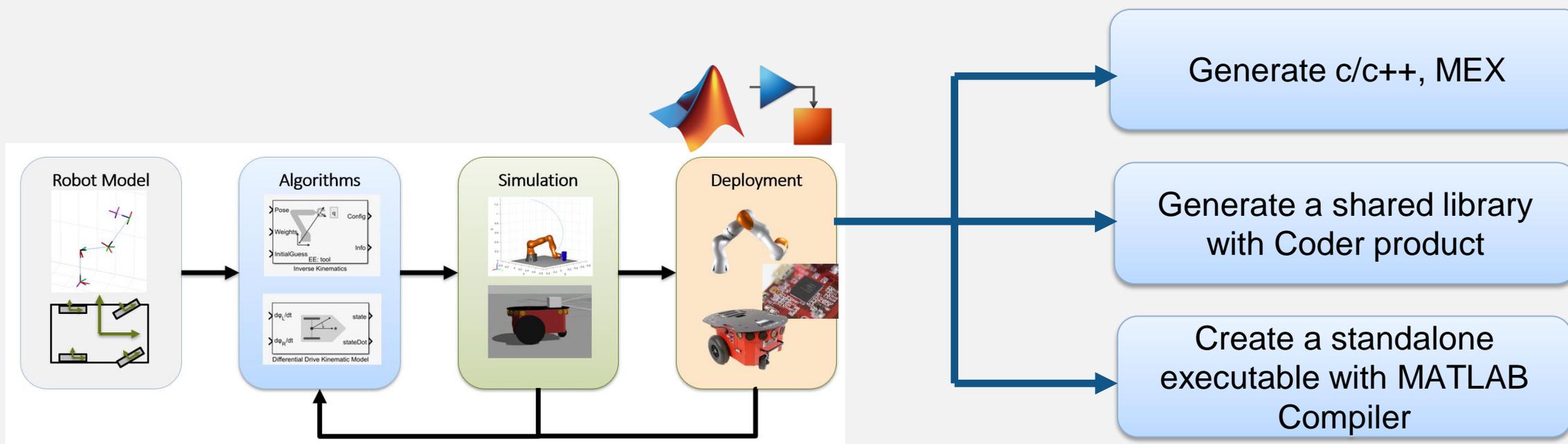
Robot Model	Algorithms	Simulation	Code Generation
<ul style="list-style-type: none"> • Rigid body tree (RBT) representation • Robot model library of commercially available robots • Import RBT from URDF file, text, Simscape • Multibody model 	<ul style="list-style-type: none"> • Library of common robot algorithms <ul style="list-style-type: none"> ○ Manipulator arms ○ Mobile robots ○ UAVs 	<ul style="list-style-type: none"> • Motion modeling and simulation to prototype algorithms quickly and test behavior of real-world systems. • Synchronized Gazebo co-simulation from Simulink 	<ul style="list-style-type: none"> • Generate C/C++ code and MEX functions for algorithm acceleration 



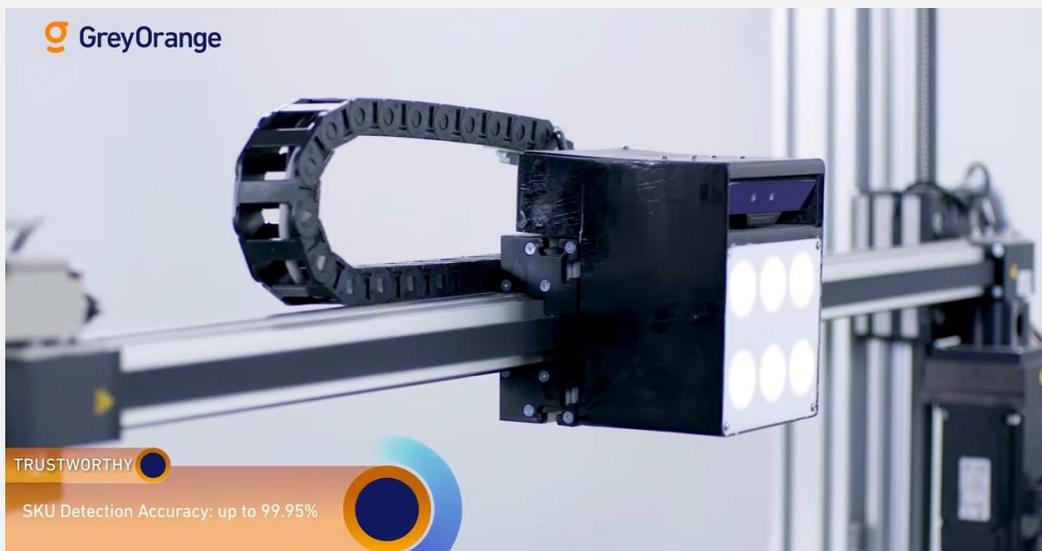
Deployment

- Accelerate robotics algorithms with code generation

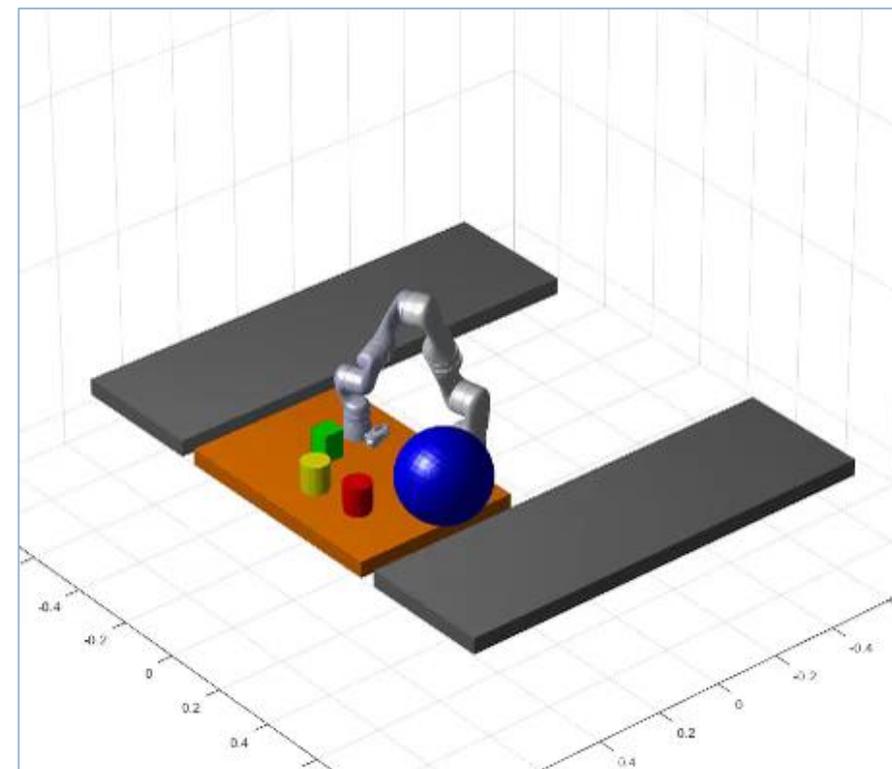
Robot Model	Algorithms	Simulation	Code Generation
<ul style="list-style-type: none"> Rigid body tree (RBT) representation Robot model library of commercially available robots Import RBT from URDF file, text, Simscape Multibody model 	<ul style="list-style-type: none"> Library of common robot algorithms <ul style="list-style-type: none"> Manipulator arms Mobile robots UAVs 	<ul style="list-style-type: none"> Motion modeling and simulation to prototype algorithms quickly and test behavior of real-world systems. Synchronized Gazebo co-simulation from Simulink  	<ul style="list-style-type: none"> Generate C/C++ code and MEX functions for algorithm acceleration 



Example – Pick-and-Place Robot Arm



- Applications in **warehouses**, **manufacturing**, and **medical industries**
- RST: robot model, plan, control, and simulate robot
- MPC: trajectory optimization
- Stateflow: task-level planning and execution



Navigation is critical for autonomous systems

Qualcomm | Invest | Research | Areas of Research | Autonomous Robotics

Autonomous Robotics

Overcoming obstacles in autonomous path planning and navigation.

We have developed **autonomous path planning and navigation systems** for drones and robots, enabling them to move safely through indoor and outdoor environments. For example, prior to flight, the user designates where the drone should go and the bounds of the area it will fly through. The drone's path planning algorithm uses a 3D model of the world (generated through voxel mapping) to build a random graph of unoccupied points in space and safely transitions between them. The graph represents all the collision-free paths the drone could select to reach its goal. The drone may see multiple paths but will pick the shortest path to its destination. Every hundred milliseconds, it updates its 3D voxel map and re-checks the planned path to ensure it is still safe. If at some point the drone encounters a potential hazard or obstacle in its path, it will re-vector to an alternate route based upon its internal decision-making.



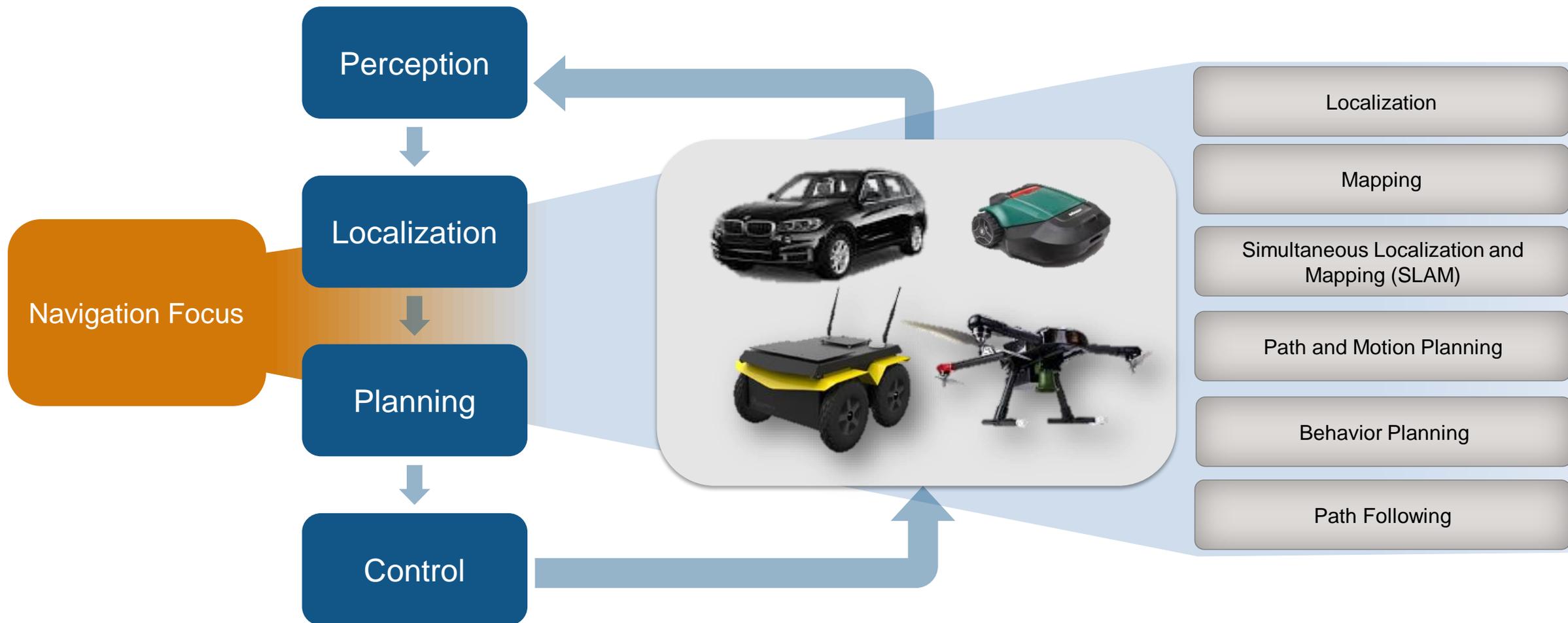
Navigation Tools

- *Where am I going?*
- *What's the best way there?*
- *Where have I been?*
- *Where am I on map?*
- *What if you don't have a map?*

Behavior Planning
Path / Motion Planning
Mapping
Localization
SLAM

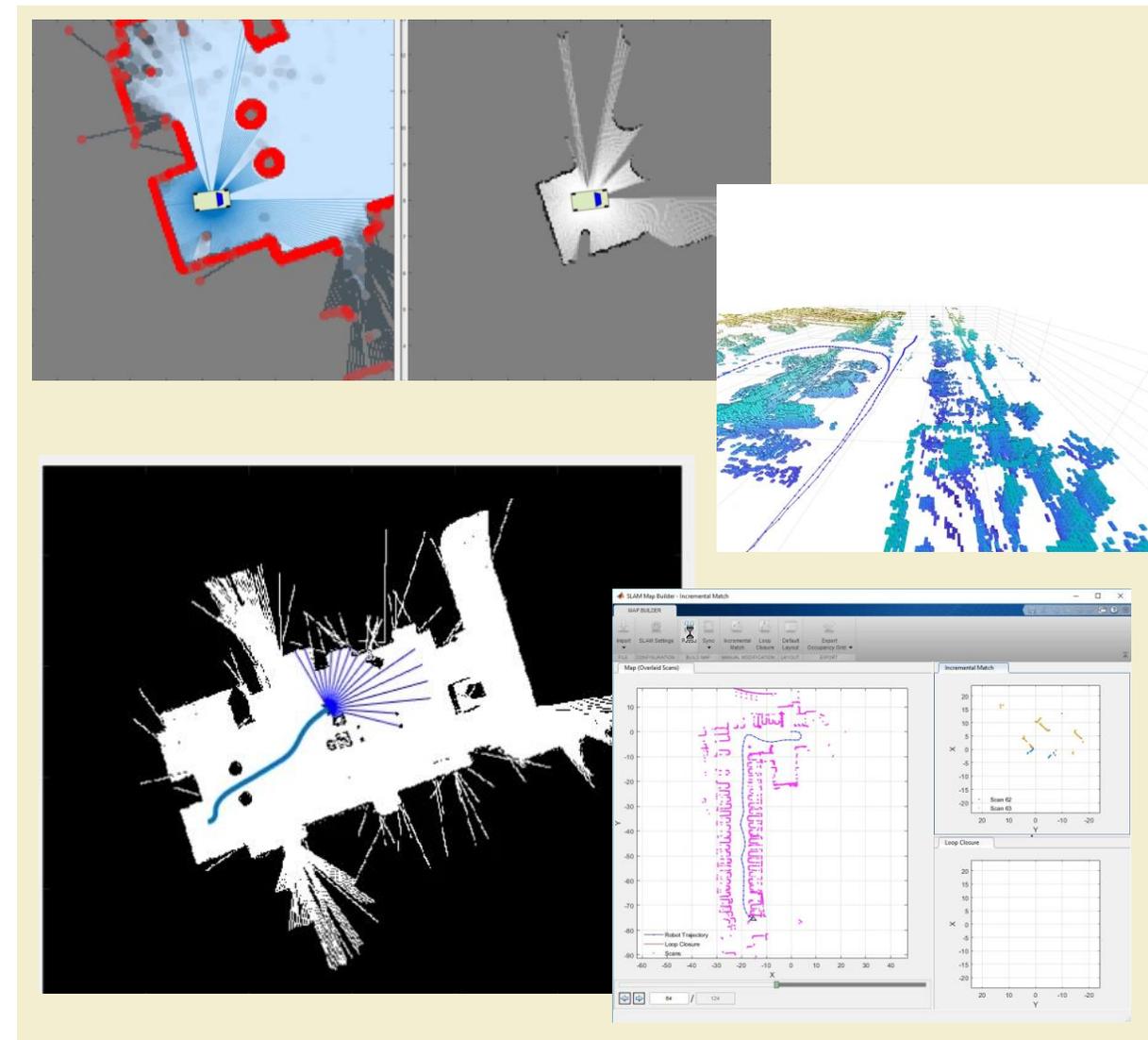


Autonomous Navigation Workflow



Navigation Toolbox

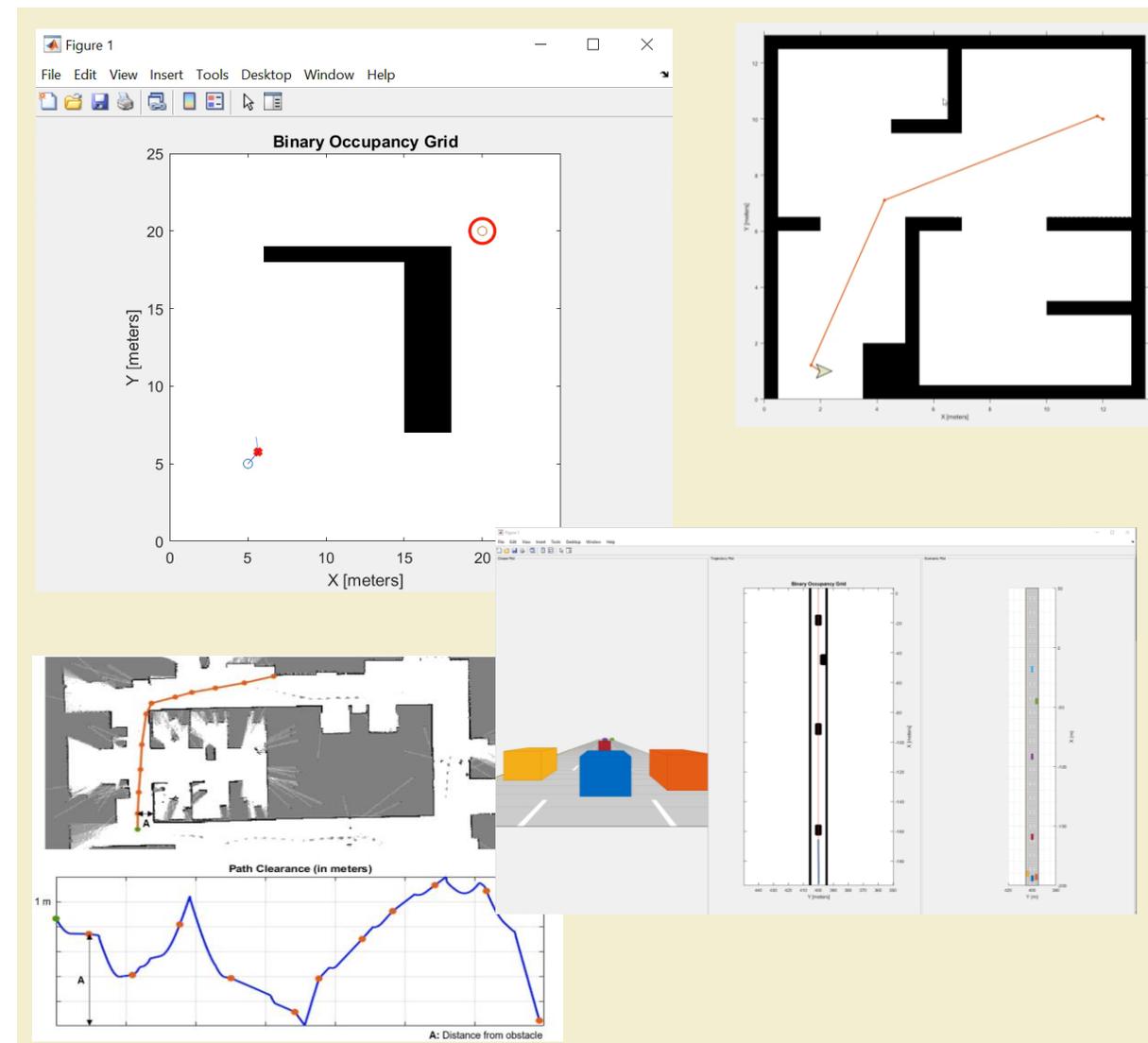
- Mapping and localization
 - 2D and 3D SLAM
 - Egocentric maps
 - SLAM map builder App



Navigation Toolbox

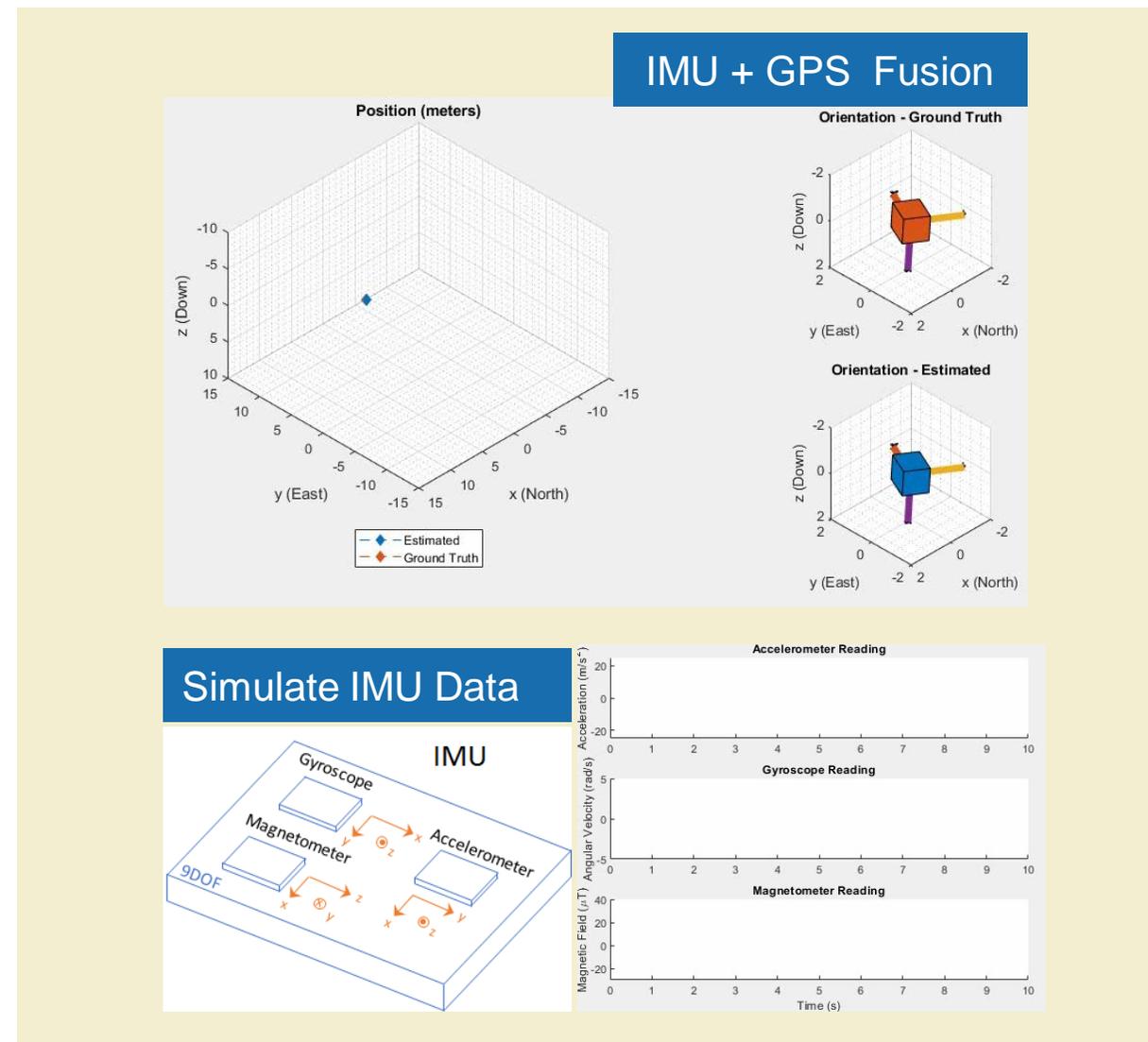
- Mapping and localization
 - 2D and 3D SLAM
 - Egocentric maps
 - SLAM map builder App

- Path planning and Following
 - Algorithms for path planning
 - Planner interface
 - Path metrics
 - Path following and controls



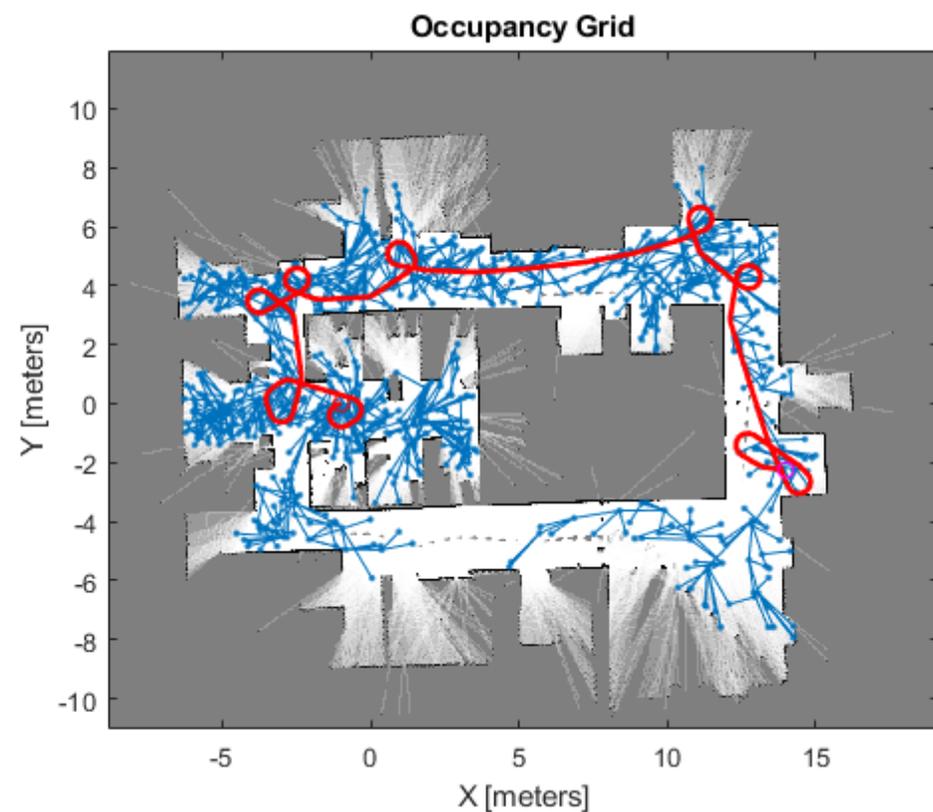
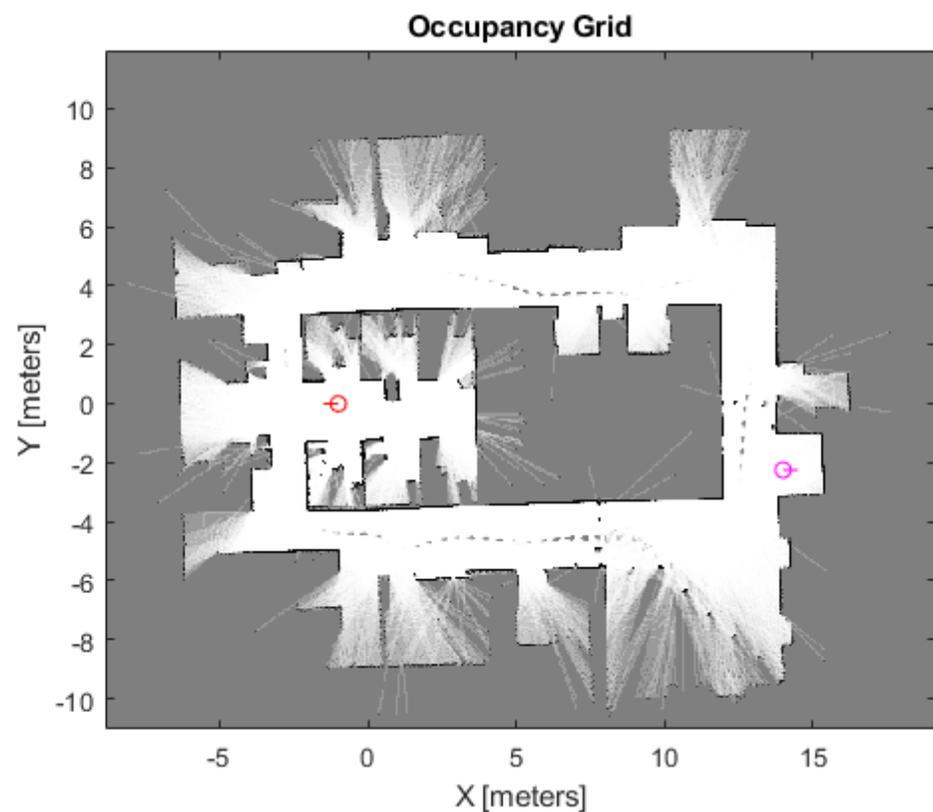
Navigation Toolbox

- Mapping and localization
 - 2D and 3D SLAM
 - Egocentric maps
 - SLAM map builder App
- Path planning and Following
 - Algorithms for path planning
 - Planner interface
 - Path metrics
 - Path following and controls
- **Sensor modeling and simulation**
 - **IMU, GPS, INS sensors**



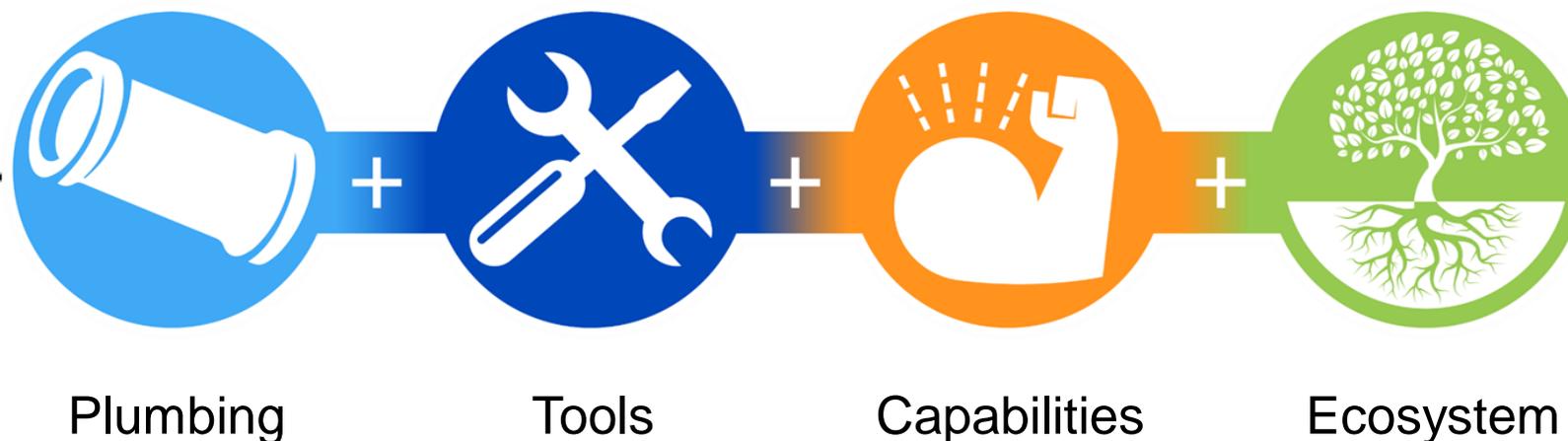
Example – Plan Mobile Robot Paths using RRT

- Load an existing occupancy map of a small office space
- Specify the state space of the vehicle
- Plan a path for a vehicle



ROS – A Distribution in Software for Automation

- Open Source
- Established to prevent re-inventing the wheel
- Maintained by Open Robotics
- Reusable Software Components
- >1,000,000 user downloads/mo



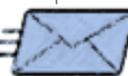
Why ROS? Growth and Adoption of ROS



10 years huge growth

- Plenty of development tools
- Active community (ROS wiki p
- 5,000 packages, 18,000 wiki p

ROS1_Bridge



Completely Re-Written



- Run on range of systems: embedded to workstation
- For use in real-time systems
- For safety- and mission-critical applications and productions

ROS for Windows 10: Microsoft gets back into robotics

By Steve Crowe | October 1, 2018



Amazon releases Robomaker, a platform to test and deploy robotic applications

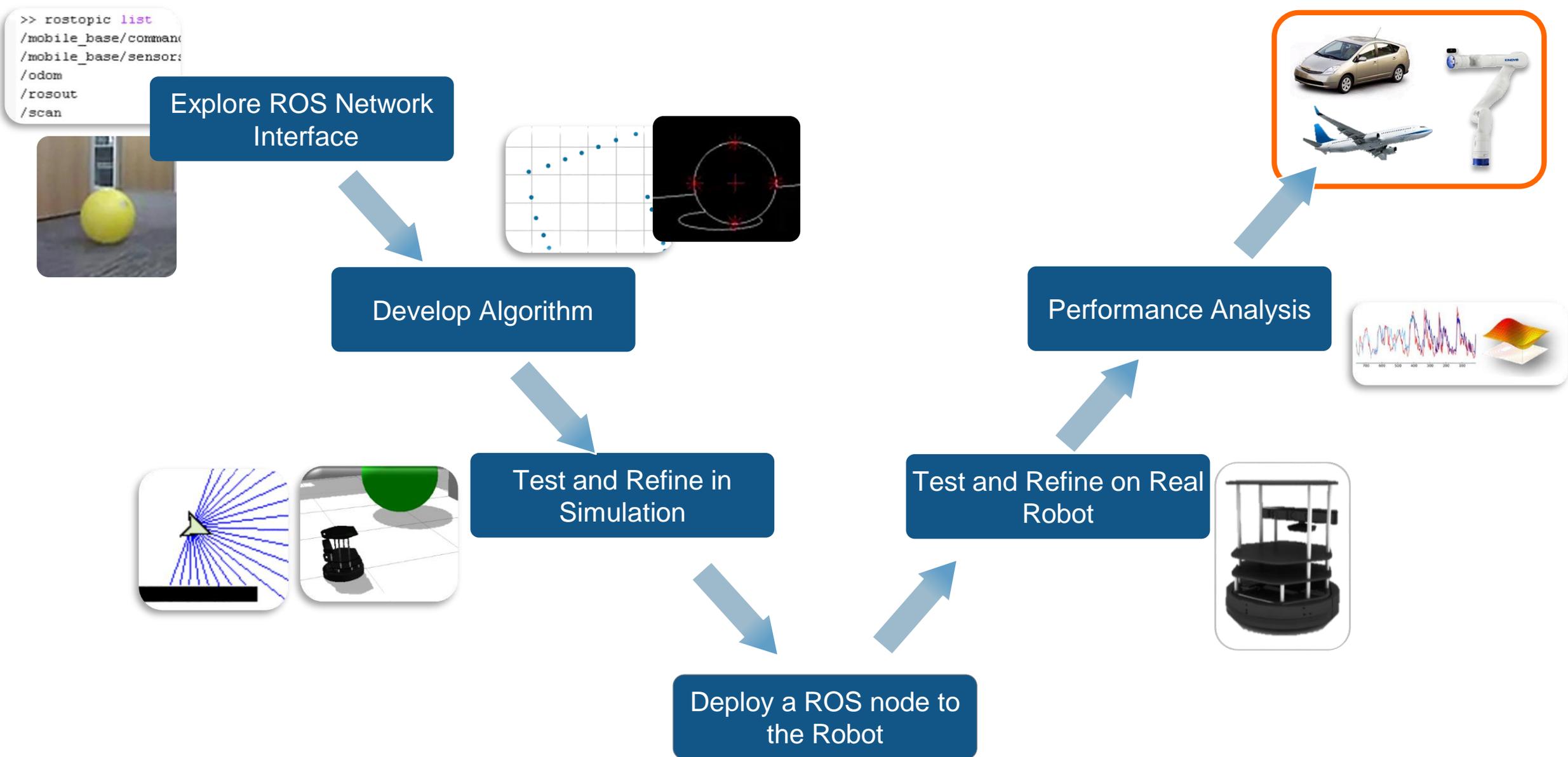
Robot Operating System (ROS) will be integrated into AWS services and given full cloud capabilities.

November 28, 2018
Devin Jones



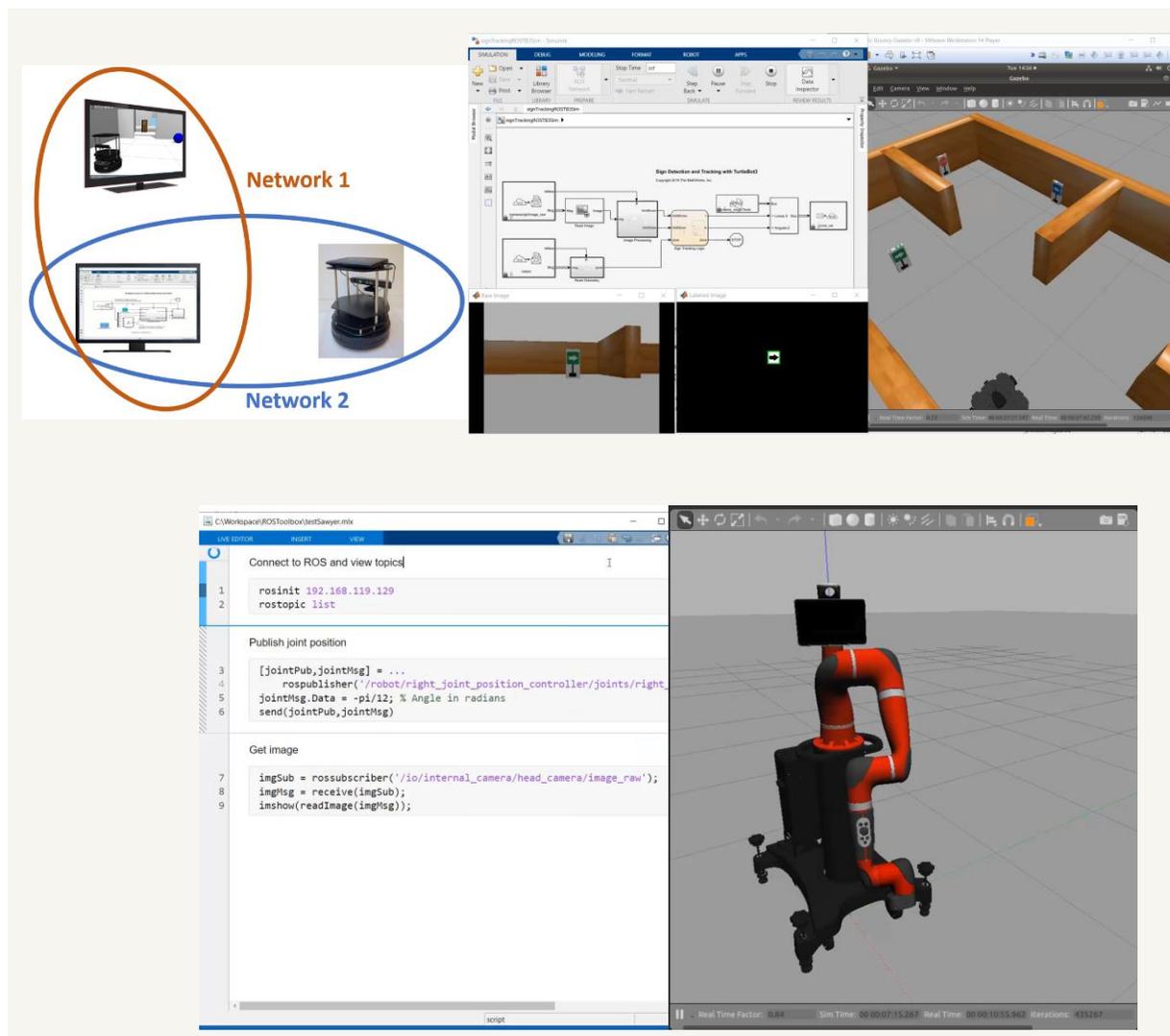
A depiction of the Robomaker logo from Amazon.

Development Workflow for ROS-based Applications



ROS Toolbox

- ROS network and communication
 - Live connectivity from MATLAB and Simulink to ROS and ROS2



The image collage illustrates the ROS Toolbox workflow. On the left, a diagram shows two networks (Network 1 and Network 2) connected to a robot. The top right shows a Simulink block diagram titled "Sign Detection and Tracking with TurtleBot3". The bottom left shows a MATLAB script in a console window:

```

1 Connect to ROS and view topics
2 roscat /
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

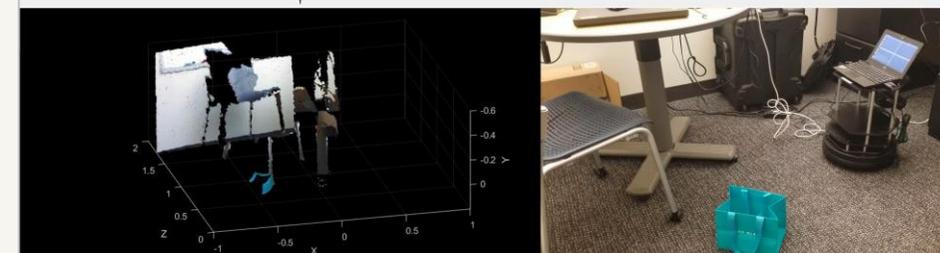
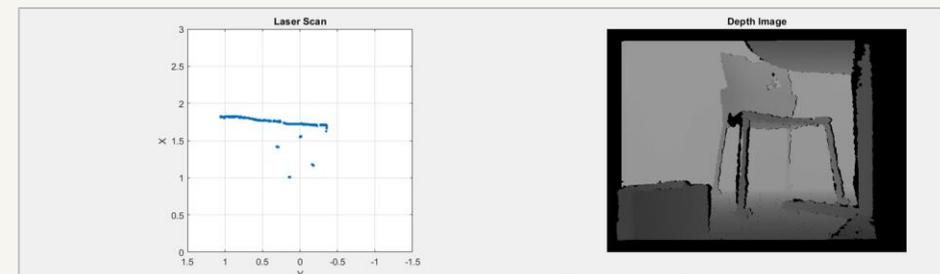
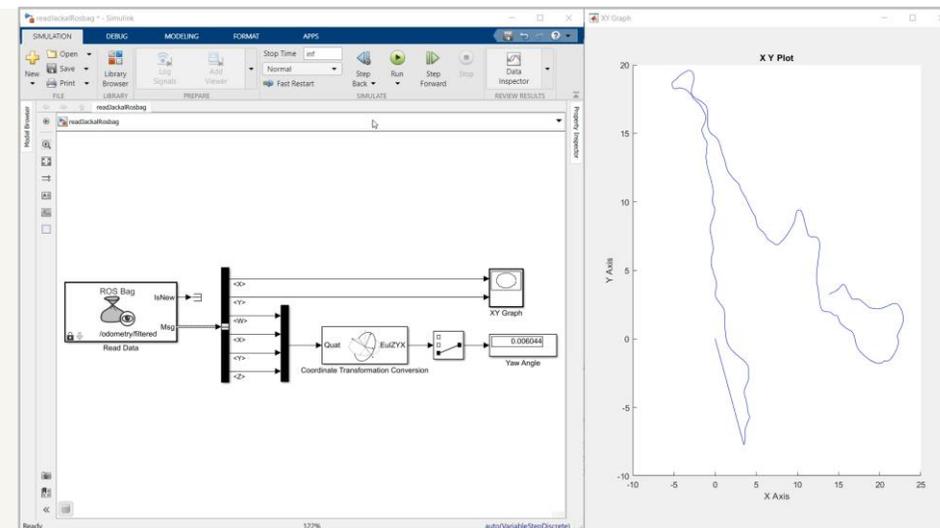
```

The bottom right shows a 3D simulation of a robot in a maze environment.

ROS Toolbox

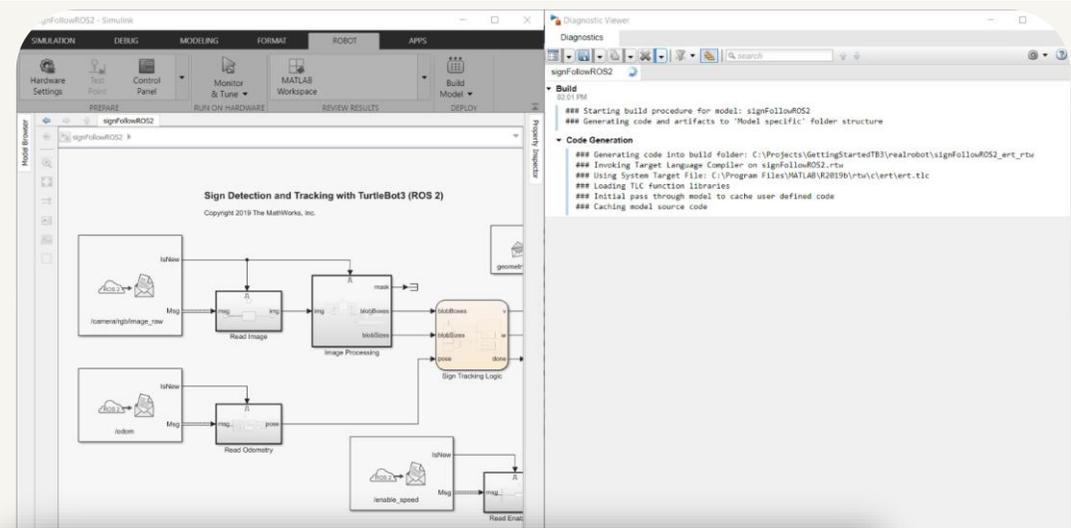
- ROS network and communication
 - Live connectivity from MATLAB and Simulink to ROS and ROS2

- ROS Message
 - rosbag data import and playback
 - Specialized ROS message



ROS Toolbox

- ROS network and communication
 - Live connectivity from MATLAB and Simulink to ROS and ROS2
- ROS Message
 - rosbag data import and playback
 - Specialized ROS message
- ROS node generation
 - Node generation from Simulink for prototyping and deploying autonomous systems



The screenshot shows the ROS Toolbox environment. The main window displays a Simulink model titled "Sign Detection and Tracking with TurtleBot3 (ROS 2)". The model includes blocks for "Read Image", "Image Processing", "Read Odometry", and "Sign Tracking Logic". A "Diagnostics Viewer" window is open on the right, showing build logs for the "signFollowROS2" model. The logs indicate the start of the build procedure, code generation, and caching of model source code.

```

### Starting build procedure for model: signFollowROS2
### Generating code and artifacts to 'Model specific' folder structure
### Code Generation
### Generating code into build folder: C:\Projects\GettingStartedTB3\realrobot\signFollowROS2_ert_rtw
### Invoking Target Language Compiler on signFollowROS2_rtw
### Using System Target File: C:\Program Files\MATLAB\R2019b\rtw\clert\ert.tlc
### Loading TLC function libraries
### Initial pass through model to cache user defined code
### Caching model source code
  
```


Package and send the joint angle trajectory
Interpolate the joint trajectory at 1 ms rate

```

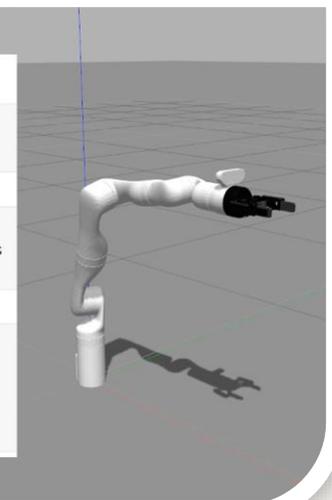
numSamples = waypointTimes(end)*1000 + 1;
[q, qd, qdd] = trajveltraj(jointWaypoints, numSamples, ...
    'EndTime', repmat(diff(waypointTimes), [7 1]));
waypointTimes = linspace(0, waypointTimes(end), numSamples);

Send a joint trajectory action to the robot arm

[trajAct, trajGoal] = rosactionclient( ...
    '/my_gen3/gen3_joint_trajectory_controller/follow_joint_trajectory');
jointNames = {'joint_1', 'joint_2', 'joint_3', 'joint_4', 'joint_5', 'joint_6', 'joint_7'};
packagePrecompJointTrajectory(trajGoal, jointNames, q, qd, qdd, waypointTimes)
sendGoal(trajAct, trajGoal);

Send gripper command actions along the trajectory

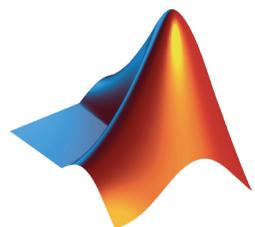
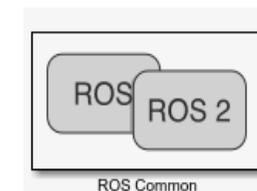
pause(6)
[gripAct, gripGoal] = rosactionclient( ...
    '/my_gen3/robotiq_2f_85_gripper_controller/gripper_cmd');
gripGoal.Command.Position = 0.75;
sendGoal(gripAct, gripGoal);
pause(3)
gripGoal.Command.Position = 0;
sendGoal(gripAct, gripGoal);
  
```



MATLAB/Simulink ROS Functionality

ROS

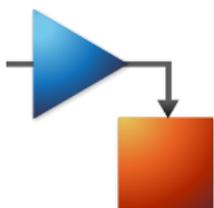
ROS 2



- Topic – Pub / Sub
- Service – Server / Client
- Action – Client
- Parameter Server – Get/Set
- Custom Message
- rosbag read

- Topic – Pub / Sub
- Custom Message

- Read Data
- Read / Write Image
- Read Point Cloud
- Read Occupancy Map



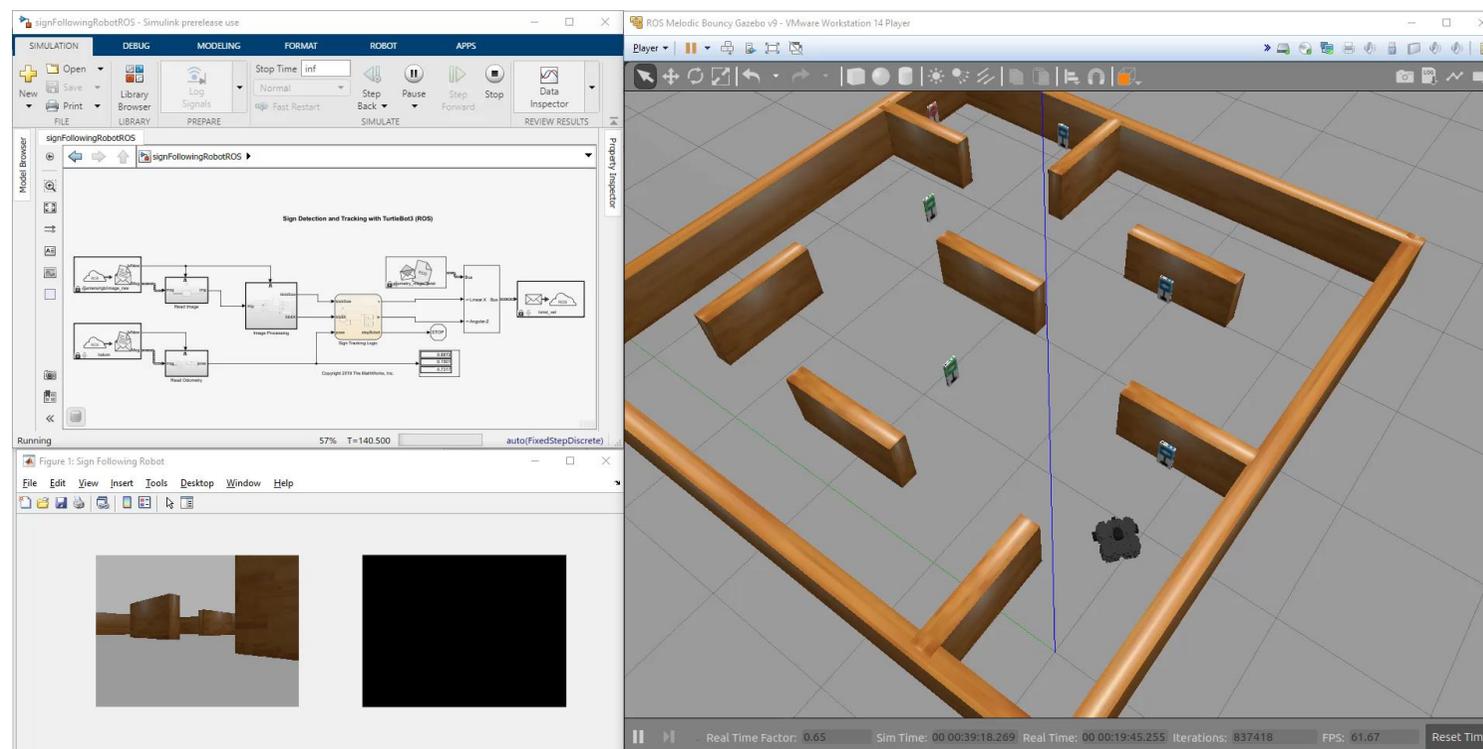
- Topic – Pub / Sub
- Service – Call
- Parameter – Get / Set
- ROS Time
- rosbag playback
- Code Generation

- Topic – Pub / Sub
- Code Generation

- Read Data
- Read Image
- Read Point Cloud

Example – Sign-following Robot

- Detect the color of the sign and send the velocity commands to turn the robot
- Connect with ROS-enabled simulator, i.e., Gazebo
- And connect with hardware



Thank you