TACTICAL UNMANNED GROUND VEHICLE FOR CLOSE-QUARTERS SURVEILLANCE & COMBAT

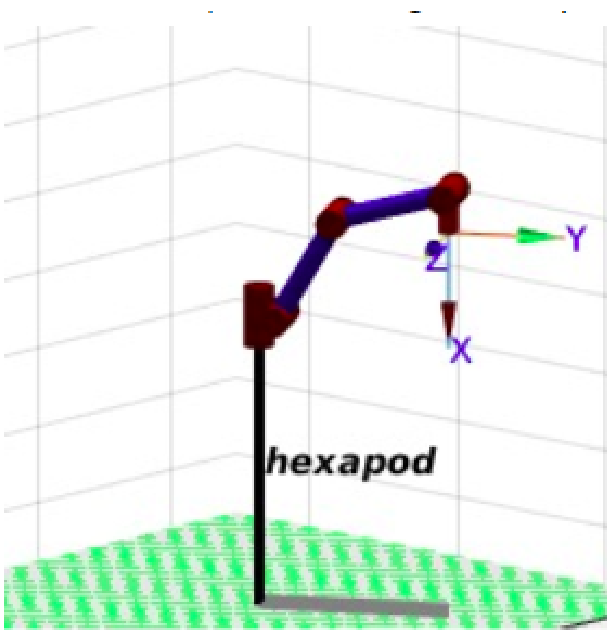
MACHINE DESIGN

Aman Malekar (180100012)
Barath Krishna S. (180100027)
Kritti Sharma (180100060)
Mitalee R. Oza (180100067)
Shubham Agrawal (180041000)
Rishab Khantwal (180100095)
Kratik Bhadoriya(180100059)

Mechatronics

Dynamic Decoupling

For precise manoeuvring of the hexapod over the rocky terrain, we need to have the precise location of the foot at all instants of time. However, the control algorithm that we will establish will only be able to control the joint angles, angular velocities and accelerations. So we need to generate a map from the joint angles to the foot location and vice versa to choose reasonable values for various design parameters such as link lengths, resting configuration, range of each leg and chassis design. To generate this mapping, we used MATLAB. We defined the robot using the DH parameters. For generating the map from the joint angles to the foot location and orientation (forward kinematics), we had to write down the rotations (as specified by the joint angles) in the form of rotation matrices and multiply them to get the foot orientation. The foot location was calculated by taking the individual components of the link lengths along the three principal directions (in the world frame) and adding them up to achieve the result. Next, we move on to inverse kinematics, where the input is the end



effector/foot location and orientation, and the output is the joint angles. In order to keep the foot always perpendicular to the ground, we need to have some relation between the joint angles at A2, A3 and A4, such that the above condition always holds for all permissible foot positions and orientations. We get an implicit relation while solving for the joint angles, and hence we cannot directly predict the joint angles to be given for attaining a specific foot location and orientation. For this reason, we need to use inverse kinematics for finding the individual mapping from end-effector location and orientation to joint angles. We start with writing the transformation matrix from frame 0 to frame 3. Knowing that the axis of the foot frame always remains perpendicular to the ground, we can assume directions of the other two axes as per our convenience. Once this is done, we can calculate theta(1), theta(2) and theta(3) from the transformation matrix. And finally using the coupling relation, [theta(4) = -pi/2 - theta(2) - theta(3)] we can find out theta(4).

This form of relation for the joints helps us take advantage of the available control scheme for hexapods which uses three DoF legs with this system, where we have four joints for each leg. Hence we are able to take the best of legged as well as wheeled robotics and implement it on our system.

Model Specifications Selection Procedure

Based on our script which accounts for the forward kinematics and inverse kinematics, we follow an iterative process in which we first start with an approximate estimate of the lengths of the four links. This estimate is then taken as the input along with the d-h parameters.

Forward Kinematics

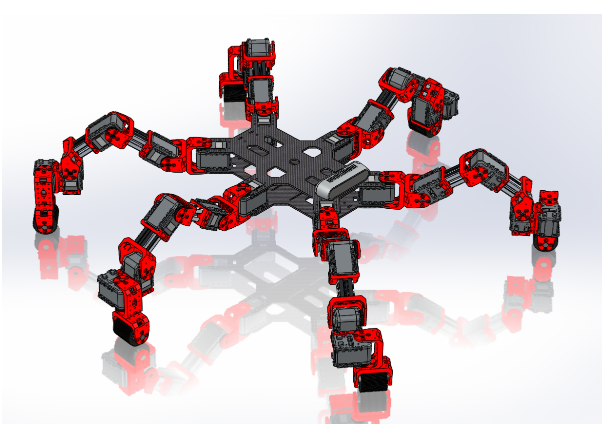
For the given theta positions, we have to check for the end effector positions through the transformation matrix T. We then visualise the end effector position using the **bot.plot(theta')** command. This helps us understand if the configuration is feasible for the actuator specifications and with the dynamics of the bot. For our estimated link length, the configuration appears valid.

Inverse Kinematics

We then decide the end effector positions in **X,Y & Z** and then check for the corresponding theta values through inverse kinematics. The end effector position range was finalized by considering the bot in various terrain scenarios and checking for maximum agility of the bot through the simultaneous movement of the six legs. The resulting inverse kinematic angles provide a estimate of the rotations required at the servo of each joint.

Final Link Lengths

L1 = 0.026 m	L3 = 0.3794 m
L2 = 0.3922 m	L4 = 0.092 m



Actuator Selection

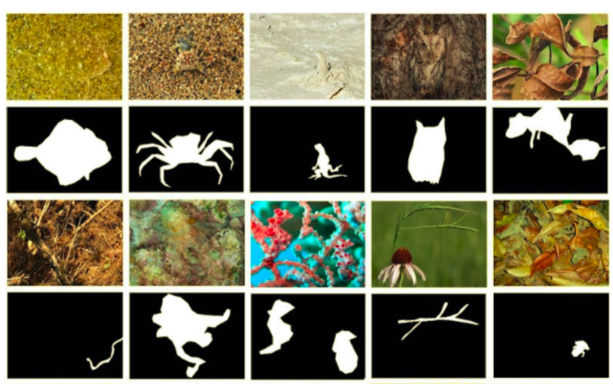
The torque requirements at each joint of the legs is to be found out using the Jacobian, according to the payload requirements. Once the torque at each joint is finalised, we can look for actuators/motors that satisfy these requirements. From our initial analysis, the torque requirement at both the hip and knee joints were approximated to be around 18Nm. Looking at the actuators available in the market, we came to the conclusion that a standalone motor would be too big and heavy to satisfy our requirements. Thus, a motor in combination with a gearset would be required to fulfil the actuation demands.

Camouflage detection

We first started to look for the relevant datasets for our task of camouflage detection and ended up over two standard camouflage segmentation datasets. We will choose the models that work on these dataset and would use them to solve our task.

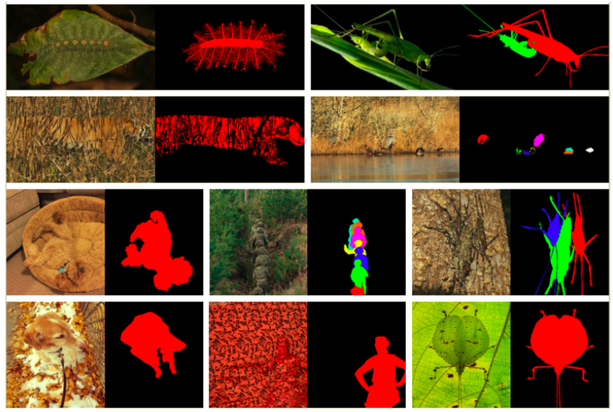
0.1 CAMO dataset

Camouflaged Object (CAMO) dataset specifically designed for the task of camouflaged object segmentation. It focus on two categories, i.e., naturally camouflaged objects and artificially camouflaged objects, which usually correspond to animals and humans in the real world, respectively. Camouflaged object images consists of 1250 images (1000 images for the training set and 250 images for the testing set). Non-camouflaged object images are collected from the MS-COCO dataset (1000 images for the training set and 250 images for the testing set).



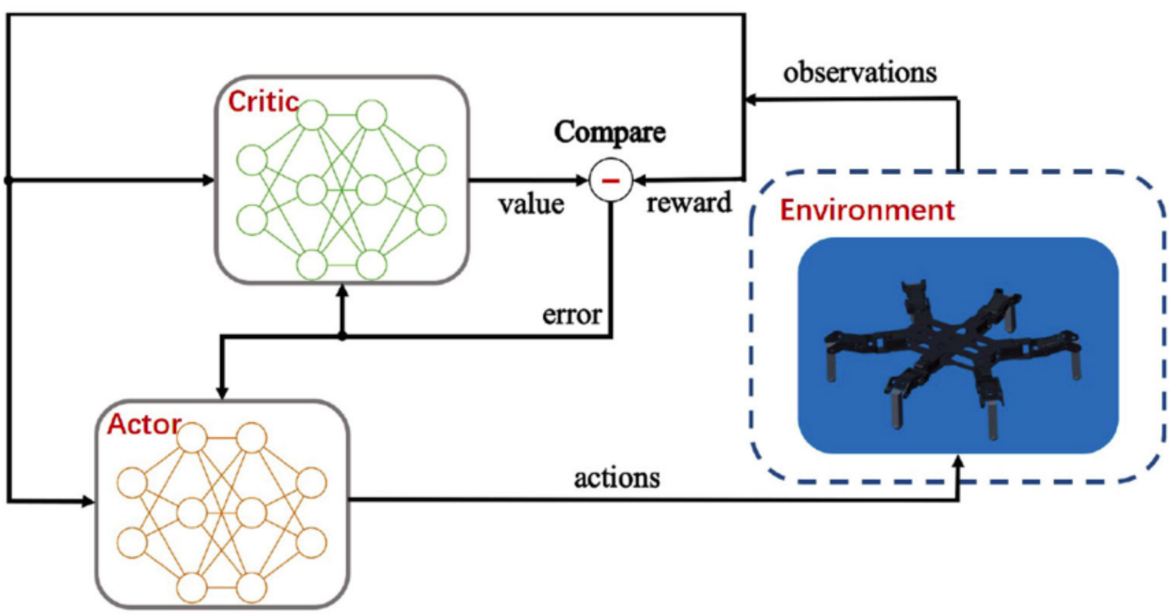
0.2 COD10k dataset

COD10K contains 10,000 images (5,066 camouflaged, 3,000 background, 1,934 noncamouflaged), divided into 10 super-classes, and 78 sub-classes (69 camouflaged, nine non-camouflaged) which are collected from multiple photography websites.



Controls and Dynamics

The problem at hand when determining the locomotion of a hexapod can be very well modelled as a Markov Decision Process with an infinite action space. Hard coding optimal actions for such a continuous action space is complicated and infeasible, thus making the analysis a bit convoluted. Moreover, for the optimal policy, the gradient of cost function turns out to be dependent on the action-value function, which is again unknown! Hence, we have adopted an actor-critic method to tackle this problem with a reinforcement learning strategy, where the actor decides what action to take and the critic informs the actor how good (or bad) was that particular action. This way, the actor learns based on the policy gradient, and the critic evaluates the action, thus estimating the action-value function! Details of the backend theory have been only worded in this submission and mathematical details have been omitted. Stay tuned for the entire theoretical backend discussion in the final submission!



References

- M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal. Fast, robust quadruped-locomotion over challenging terrain. InRobotics and Automation (ICRA), 2010 IEEEIn-ternational Conference on, pages 2665–2670, May 2010. URLhttp://www-clmc.usc.edu/publications/K/kalakrishnan-ICRA2010.pdf. clmc.

- Alexander Winkler, Ioannis Havoutis, Stephane Bazeille, Jesus Ortiz, Michele Focchi,R udiger Dillmann, Darwin Caldwell, and Claudio Semini. Path planning with force-basedfoothold adaptation and virtual model control for torque controlled quadruped robots. In2014 IEEE International Conference on Robotics and Automation (ICRA), pages 6476–6482, 2014. doi: 10.1109/ICRA.2014.6907815.

- Diego Pardo, Michael Neunert, Alexander Winkler, Ruben Grandia, and Jonas Buchli. Hy-brid direct collocation and control in the constraint-consistent subspace for dynamic leggedrobot locomotion. 07 2017. doi: 10.15607/RSS.2017.XIII.042.

- Alexander W. Winkler, C. Dario Bellicoso, Marco Hutter, and Jonas Buchli. Gait and trajec-tory optimization for legged systems through phase-based end-effector parame-terization.IEEE Robotics and Automation Letters, 3:1560–1567, 2018.

- Robert J. Griffin, Georg Wiedebach, Stephen McCrory, Sylvain Bertrand, Inho Lee, andJerry Pratt. Footstep planning for autonomous walking over rough terrain. In2019IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), page 9–16.IEEE Press, 2019. doi: 10.1109/Humanoids43949.2019.9035046. URLhttps://doi.org/10.1109/Humanoids43949.2019.9035046.

- Jiming Ren, Shuo Yang, Zhaoyuan Gu, and Howie Choset. Trajectory optimization for thelegged system, 09 2020.

- Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018.

- Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speedand accuracy of object detection, 2020.

- Alexey Merzlyakov and Steve Macenski. A comparison of modern general-purpose visualslam approaches, 2021.

- D aniel Kiss-Ill es, C. Barrado, and E. Salam ı. Gps-slam: An augmentation of the orb-slamalgorithm.Sensors (Basel, Switzerland), 19, 2019.