DIGITAL SIGNAL PROCESSING LAB

![Logo

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4SfoRXhpZgAATU0AKgAAAAgABgALAAIAAAAmAAAIYgESAAMAAAABAAEAAAExAAIAAAAmAAAIiAEyAAIAAAAUAAAIrodpAAQAAAABAAAIwuocAAcAAAgMAAAAVgAAEUYc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFdpbmRvd3MgUGhvdG8gRWRpdG9yIDEwLjAuMTAwMTEuMTYzODQAV2luZG93cyBQaG90byBFZGl0b3IgMTAuMC4xMDAxMS4xNjM4NAAyMDIxOjA0OjEwIDIwOjAyOjE2AAAGkAMAAgAAABQAABEckAQAAgAAABQAABEwkpEAAgAAAAMwMAAAkpIAAgAAAAMwMAAAoAEAAwAAAAEAAQAA6hwABwAACAwAAAkQAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMTowNDoxMCAyMDowMDowNgAyMDIxOjA0OjEwIDIwOjAwOjA2AAAAAAYBAwADAAAAAQAGAAABGgAFAAAAAQAAEZQBGwAFAAAAAQAAEZwBKAADAAAAAQACAAACAQAEAAAAAQAAEaQCAgAEAAAAAQAAFjsAAAAAAAAAYAAAAAEAAABgAAAAAf/Y/9sAQwAIBgYHBgUIBwcHCQkICgwUDQwLCwwZEhMPFB0aHx4dGhwcICQuJyAiLCMcHCg3KSwwMTQ0NB8nOT04MjwuMzQy/9sAQwEJCQkMCwwYDQ0YMiEcITIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIy/8AAEQgApACsAwEhAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A9/ooAKKACigAooAKKACigAooAKKACigAooAKKACigAooAKKACigAooATI9aMj1pcyCwtFMAooAKKACigAooAKKACigBKKACs2+1uxsPlkl3Sf3E5NYV68KEeabNaNGdWXLEwZfFd5cOY7G0APbILH8hUYi8T3vJeWIe5CfyrxJ4nFYp2pKyPXWHw2GV6rux3/CO63Jy95j6zNR/wjetJyt6Pwmap/s7Gb84/r+FWigNNp4mtPuSyyAf3XDfzp6eJtTsm2X1qD9VKH/CnGvi8I7VNUJ0sLif4bszbsPENhekJv8qQ/wAEnH69K1hXtYfEwrx5oM8mvh50ZcsxaK6TEKKACigAooAKKAEqG6uobSBpp3CIO5qJzUIuUtkOMXKSit2cpd6xf6xN9msEdIzx8v3iPc9hVuw8JxJiS9cyP3ReF/H1rxKdKWPq+0n8C2PXqVI4On7OHxPc6CG2ht0CQxLGo7KMVNXtwpxgrRPJlJyd2FLVkiUyWGOZSkiK6nqGGRUyipKzGm07owr/AMK20+XtT5En93+E/h2rLgv9T0CUQXSNJD0AJz/3yf6V4lahLBVFWp/D1PXo1o4uHsam/RnV2V9BfwCWCQMO47g+hq1XtU6kakVKOzPJnBwk4voFFWSFFABRQAlFICte3sNjbtNKenQDqT6CuVjhvPEl4ZJGMcCn8E9h6mvLx0pVZxw8Ou56ODiqUHXl02Oqs7G3sYBFBGFHc9yferNelTpqnBRj0OCc3OXNLqNd0jUs7BVHUk1mT+IdPh480yH/AKZjNY18VTor3maUcPOq/dRSbxZbg/Lby/iQKF8WwZ+a2kA9mBrg/teF9ju/sypbctw+I9PmOC7RH/bXA/OtSKWOZN0bq6nupzXdQxdOsvdepxVsNUo/Eh9Q3NtDdwtFNGHQ9jW84Ka5ZbGUZOLujlLqxu9Auxc2rs0JPX+jf410mnajFqMHmIcMOGTuprzMG3Qqyw8vVHfikq1NV4+jLo6UtesecFFABRQAVHLKsMbSSMFVRkk1Ldk2xpNuyOTIn8RanjlbdOn+yP8AE11VvBHbQLFEgVF6CvPwMXOUq8uux24ySgo0Y9PzJc4rG1TXo7QmKDEk3fn5V/8Ar+1dOKrqjTv1MMPRdaduhlppmp6w4kupGSPqN4/kv+Natv4csIceYGlb/aPH5VwUME6r9rXOyti1TXs6OxdXTbFBhbWHj/YFK2nWTjDWsP8A3wK9D6rRtblOL6xVvfmZTn8O2EwOxGib1Q/0rIm0e/0tzNZysyjrs6/iO9eficDyP2tHRo7aGMU17OtqmX9N8QLMyw3YCSE4DjoT/Q1vV24TE+2hrutzkxND2U/LoMljSaNo5FDIwwQR2rk7iCbw/qSzQkmBug9R/dP9Kxx8HG1aO8TbBzTvSe0jqrW4jurdJo2yrDNTV3wkpxUkccouLaYUVRIUUAJXO+ILtppUsIckkjdj17D+tceOny0ml10OrCRvVTfTU19OsUsLVYl5Y8sfU1brejTVOmoroYVJuc3J9TA1fVnMv2KzJLsdrMvXPoKm0rQ47ULPcDfPjIB5C/8A164FH6ziW38MTtcvYUFFfFI2aazKilmIAAySTXpnnN2Rxl58TtBtLtoE+03AU4aWFAV/DJ5/Cup0vVLPWLFLyxmEsL9x1B9COxrOFaMnZHLQxlOtNwj0LlBrSx1mNquiR3itLAAs+OR2f61V0jVHgk+xXbNwdqs3VT6GvLnH6tiFNbSPRg/rFBxe8To+1V720jvLV4X6MOD6H1r0akFODi+pwQk4SUl0MHRbh7C/ewn4DN8o9G/+uK6YdK5sBJ+y5HutDoxkf3nMuuotFdpyhRQBFPKsMLyvwqqSa5/RYGu76W8m5Kn/AMeP+ArhxK560IfM66D5KU5/I6SszWb82ltsjOJZAQD6Dua3xE/Z0nJGNCHPUUSDRdN8mMXMwzK4+XP8I/xNbVThKXs6S7lYip7So2Fcj8R7+Wx8ITrExVrh1hyDjAPX9ARW1R2gzgxcuWhJrseIdK9U+EaTCy1Nzn7OZECem7B3fptrhwy99HzmVc31lWPSe1Vn1KxSTynvIFk6bTIAfyr0G0lqfUynGL95ljIPPH1rH1vTRPH9pjX96g+YD+IVz4ql7Sk12OnDVOSomS6Jfm6tvKkbMsQwT/eHY1qU8NP2lJNirw5Kjic/r9qUkjvIuGztYj17GtmyuBc2cUw6sOR6HvWNH3MROPfU1qvnoQl20LFFdxyBRQBl67Lssdg/jbH4d6k0iDyNOj45cbj+NcS97FPyR1PTDrzZf6VzgU6prZJ5iQ9O2B/9ejGauEO7DDe7zT7I6McDHauN8UfECz0C7ayggN3eKMuobaqZ6ZPr7V0Tmqcbs8zFYlYenzsTwv8AEG08QXgsZoDaXTDKAvuV/UA+vtUvxHs3u/B1w6DLQOsxHsDg/oTU8/tKbsc/t1icLKS7M8b0uwfVNUtbCM7TPIqbvQdz+VfQVraWOg6SsMSpb2tuhJJ7AdST+uaxwqteTOLJ6aUZVGeS+K/H97rE72+nSvbaeCQCh2vL7k9h7fnXGYBOSMk9z3rGdRzlc83GYqdaq5X0PW/hZrFzeWN3p9zI0gtSrRsxyQrZ+XPsRx9a9BIzkdQa7qLcoK59JgKjnh4yZzhU6ZrWV4jc/of/AK9dJ2rnwnu89Psz1sV73LPuitqEH2mylj74yPr1rP8AD8pMEkJP3WyPxpz93ExfdBDXDyXZm1RXYcoUUAYevEvLDGPc/wBK2Y1CRKo6AAVyUda036HRU/hQXqRXsnlWcr9wprO0GHbDJKRyxxn6UVNcRFdkOGlCXma56V8+eLLae08V6lHcZ3tMzgnurHIP5U8UvdR89nMW6SfmZtldyWN9bXkQ+eCVZF98HNfQlleWXiLRBNEyy21zGVYemRgg+9RhXvE58nmmpUmeWeHdFl0T4oW+nzLxG0jRMR99Nh2mtf4oa3M89voFpvPmASTKgJL8/KuB9M/lQrxpSXmEVKjhakFvexg6b8NddvoxLN5NmpGQJmy35DpU8vwr11JVWKezkjPV95XH4YqFhp2uYRymrKKdz0Pwj4Wi8Mae8Qk825mYNLLjGcdAPYV0VdtOPLFI+hw9JUqaguhj69DujilxypwfxrSs5PNs4nPUqM1zU1bESXex3z1oRZKehrC0seTqs0XY7h+Rp4j+JTfmFH4JryN+ius5gooAw9UG7Uoh6bf51tjpXLQ/iTOir8EClqx/4l8g9SB+tJpK7dPX3JNH/MT8g/5cfMvVzninwhZeJoFMhMN3GP3c6jkexHcVvOClGzODEUY1qbgzy7Uvh94isJDstRdxg8Pbtn9DzS6B/wAJb4duy1lpl7tc/vIHgYo319D7iuD2dSnK6PnI4avhq3NFHrmmL/asNrqeoaT9kv4twVZMM6Z4OCOx9Kq61No3h2eTXru0d7iQCMyxxl2AA49lHvxXa7cvM15n0L5VT55Lz+ZhL8WNGL4Npehf7xRf8a2dN8eeHtTkEcd8IZScBJxsz+J4/WohiIydtjmo5nRnLleh0ikEZBBz6UtdB6S1WhR1dQ1g3swpdKJOnRg9hj9a5dsR8jo/5cfMu1iQjbrz+5P8qMT8UPUKG0vQ3KK6jnCigDG1MYv4W9h/OtgVzUf4kzep8ESnqi7rCT2wf1o0s/6Co9Cf50W/2j5B/wAufmXKiubmC0ge4uZUihjGWdzgD8a6G+5ztqKuzBi8d+GppjGuqRAju4Kr+ZGKran8Q/D+nxMYrn7ZL2jt+cn69BWTrxtc4p4+gouVyl4L8Sar4n1a+uplWHTokCRxKM4cnP3upOP506w8Ys3jO/8AD+piIxtKVtpAuOwOxvX61KqPlTfUyhi26cJT2kyj4w+HlvcQSX+ixCK5UFmt1GFk+g7GvJ+vY/Q1zV4csro8jMsN7GreOzPQfhx4ouodSj0W7kaW2nB8kscmNgM4z6HnivXK6sPLmgezldZ1aCvutCjqv/HkR6sKk01dthH78/rU74j5Hsf8ufmWqx4Bu1t29Cf5UV94+oUdpehs0V0mAUUAZerJzE/pkVoxOHiVvUZrnp6VpG09acRlwnmQSJ6iqWlP8rxn6iielaLCOtJo0q8x+LdzcBdNtQWFs5d2APDMMYz9MmniG1Tdjy8ybWGlY8w4q7pWk3mtahHY2Ue6VzyT0Re7H0FcEVd2R8pSg6k1BdT3vQtGtvD2jx2UHRBueQ9XbuTXhWuah9r8S31/byEbrkvE4POAeD+ldVf3YqJ7maJUqNOmuh7t4e1Ma1oFnfcbpYxvHow4I/MGvJPGfhG/0zWLm6t7WSaxuJDIjRLnZnkqQOnPSqrJygmjXH03Xw0Zx3L3w98L38uuxanc20kFtbZKmRSpdsEcA84HrXsNXh4uMNTbK6MqdDXqZuqvny4x1zk1fgTy4ET+6AKmnrVkz2J6U4oeehrK05d95LJ9T+Zp1dakEFPSEma1FdBiFFAFS/j8y2b1Xmiwk3WwH904rDat6o23pfMtf1rLYG0v9w4Q8n6Uq+lpdmFLW8TUGCAQeKy9e0Gy8Q6ebS9Q7QdyOvDI3qDWzSkrHNVpqpBwl1OIj+EcPnZl1eUxeixANj65/pXb6LoGneH7XyNPtwgbl3PLOfUmsqdBQfMcOEy+FCXNuzj/AIneIrmwt4dJtHKNcoWmcdQmcYH15/KvJuua5cRJudjxs0quddroj2r4YM7eDo9/QTSBfpn/ABzXZYyK7afwI+hwmtCHoGPTignaCSeBV30OleRmRg3d8ZD91Tn/AArUrChqnLubVtGl2Ibt/Lt2IPJ4FQadHsiLY+8ab1qryBaUmXqK3MQooAawDAg9CKoW2be6aNvuk4rCrpKMjWnrFo0KguoBNGcfeHKmtJx5otEQlyyTIbSbAEL8Htn+VXamjK8dSqitLQKK1MzhfiH4TutcigvtPQSXVuCrRZwXQ88e4P8AOvObLwd4gv7pbddNnhyQGknXaq+5J6/hXDWpSdS66nzuOwNSeI5orRnuGiaVDomj22nwnKwrgserHqT+JzWhmu2KsrHv04ckFFdAzVG8lLnyY+/3iP5VnVlaOnU3pRvIsW0Hkx4/iPWpqqEeWKRM5c0rlC7YzTrEvQfzq6ihECjoBWdPWcpFz0gkPorcyCigAqpdxZxIvUdazqK8WXB2kSwS+Ygz94dalpxfNFMUlZ2K1xbb/nThh196y9Y1S907R7i6t1gaS2QyOJs4ZQM446GspXpu66hUk3SbW6LmiXV1f6Rb3l2sSyToJAsWcAEAgc96o+KtX1DQ9OW9tI7aWMOqOku4HLMAMEfWtHJqNzllOUaPP1Nq2877PH9p8vzsfP5edufbNYGv6vqmmanp9vax2jxXsvkqZd2UbBJJx1HFE5NK6KqzlGHMtzogW2jd97HJHSuQv/EOu6fpJ1Ga0tIka5EMcL7920vtDH+eKJyaV0KtUnCN4mnFf6v/AGpJZ3MMBgMO+OeANy2cFSD3rWt7cRjc3Ln9KzjecrvodFOUlTtLcsVFPL5acfePStpO0bhFXdiG0hIPmt949M1bqaUbRKqO8haK0ICigBKCOKVgKhRoH3L93NWkYOuVNZw0biXLXUWsTxWif8ItqjcAi1k5/wCAmqmk4swqu1OQ3w+ZU8OaaVOR9mjz3/hFZvj2bd4WcMAMXEOf+/i1z8zjCzJqqLwt12OqE0eACecelct4rkRtZ8MkHgX/AP7Ia1nOPKFdWp39DpjcjHyrk+9cr49Z28NqXbaDdwY7fxioqScol4hKNCT8jrYokUbl5J71L2reMUloXe+o2RxGMn8BVZY2nfe/Son7z5TSPurmLg4FFaozCigAooASigBCAwwe9QYaFuOQazkvtIuL6FeTW9NivRZy3ca3BYL5ZPIJ6D8aS8GlajEsN2IJ4/NMQV+QX5+XHr1qk00c7nCScWJZSaVZW0UVm0EUEm5o1jwFOPvEfSqc8/hvXEeSdrS6SBN7GTkKvrz296T5WrEuVPl5OhNaahogsd1rc2620JWM4bAQngA56ZzUOp23hyUtc6jHau0biMtJyVbAIA9Dj0qeWFhyqU5Rs9iz/bekQi3UXUK+auYh3IzgfrUepR6DfN/xMRaTGHcp80A7SACfyBB/Gn7trBKdOcbPYli1DR7CePS4riCGRSFWAHGCeQPx9KkXWtOkvGs4ruJ7kMV8sHnI6j8KbkkhqpDZMsBGlbLHirAGBgDpRBdWbyfQWirJCigAooAKKACkOCOaW4HL33he4utckvVukWGSWKUoQ2RsxxjdtOcdxUsfhtEvxd+e5lF21wQWbbggjbtzjPPXFZcjRyLDNybfcbaeHruN40muoGt4EmSELGQ37w/xEnHHtVW08H3MVpdQzXiFpbQWqlQ5wPX5mP5DFLkbM3hZO13/AFsXH8LK6XUT3Uk8Vw0JYT/OcIckE+hqCbwpObAwJe73F59oR5AwO3btCkqQeB39qfsynhdNBbjw1eS/ZTHcxQyRxhHmRpN5w2f73I/3s9abf+Gry4u7gxXtukEzuxDREsN6qrcg46Lxx3pODsxfVZ9H2LE/hxp9QlulvJV33MU4i/gwm3gj1O3r9PSls/DAtNZS/wDPZx5ksjRMSVDMSQyjOAwBIPrmhU7sv6qk+bzudEOlLW51hRQAUUAFFABRQAlLQAlBAPakw2E2+lGD60rDuHNJg+tGoBg96XaPSiwhelFMBaKYBRQAUUAJ3ooAKKAA0UkAUUwCigAo70AA6UtABRQAUUAFFABRQB//2QD/4THkaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLwA8P3hwYWNrZXQgYmVnaW49J++7vycgaWQ9J1c1TTBNcENlaGlIenJlU3pOVGN6a2M5ZCc/Pg0KPHg6eG1wbWV0YSB4bWxuczp4PSJhZG9iZTpuczptZXRhLyI+PHJkZjpSREYgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdG9yVG9vbD5XaW5kb3dzIFBob3RvIEVkaXRvciAxMC4wLjEwMDExLjE2Mzg0PC94bXA6Q3JlYXRvclRvb2w+PHhtcDpDcmVhdGVEYXRlPjIwMjEtMDQtMTBUMjA6MDA6MDY8L3htcDpDcmVhdGVEYXRlPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDw/eHBhY2tldCBlbmQ9J3cnPz7/2wBDAAMCAgMCAgMDAwMEAwMEBQgFBQQEBQoHBwYIDAoMDAsKCwsNDhIQDQ4RDgsLEBYQERMUFRUVDA8XGBYUGBIUFRT/2wBDAQMEBAUEBQkFBQkUDQsNFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBT/wAARCACkAKwDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD9U6KKKACiiigAopKM0ALRSbh60ZHrQAtFJuHrRuA70ALRTd6jvQJFPRhU8y2uOzHUUmRRuHrVCFopMiigBaKKKACiiigAooooAKKKKACikzRQAnQkikyPpS9TWbr2vWPh3Tpb3ULhLe2j6sx5J7AeprOdSFOLnN2SKhFzkoxV2zR6e1ch4r+KXh7wiTFc3nn3fa2tx5j/AIgdPxryvxB8SPEfxIvzpPhuCe1tWOP3XErr6u38A/zntW/4R/Z7s7fZc69O15OeTbwErGPqerV8ZPOsXmE3SyildLectI/LufUxyrDYGKqZnUs/5I/F8+xh6h+0FrmsXDW3h7RlVicDerTP+S8D8aqrp/xb8THLzXVip5+Z0gH5Dmve9L0Sw0WFYbCzhtIl42xIFH44rQ6Uv7AxeK97HYuT8o6If9tYbD6YPCxXnL3mfPB+DPxAvPmn1wKT13Xsv9BR/wAKR8eWvzRa8pP+zezD+Yr6HFLVf6pYDfmnfvzsP9Zsb/LC3+FHzm/h34seHQDDeXV2i9BHOso/J+ant/jl4s8LyiHxBo6uM4PmRtA59wTlTX0Jmq19p1tqULQ3VvFcxNwUlQMD+dL/AFfxGH1wOLnG3R+8mH9t0cRpi8NCS7r3WcX4T+Mnh3xRIsIuDYXh48i6AUk+zdDXdowIBBznpXlXi74A6TqwebR3/sq6xxGBmE/8B7fUVxeleLvFvwivlsNYgkvNOzhUdywx6xOf/QT+lTHNsdlslDNqfuv7cdvmug3luDzBOeW1Pe/klv8AJ9T6MFLWJ4X8Wad4u01LzTrhZU6Op4dG/usOxrayPWvtKVWnWgqlOV0+p8tUpzpScJqzQtFFFamYUUUUAFFFFADe/vQTt70tVb++h021murmQRQRKWZmOAAKiUlGLk9Ehxi5Oy3M/wAUeKLHwppct9ePgKMJGv3pG7Ko7mvELPTtd+NmvNc3EjWmmQnGRykI7qv95z61addQ+M/i/HzwaVb8g9BEh/m7Y/D8K9w0fSbbQ9PhsrKFYYIhhVA/U+pr4jkqcQ1m5Nxw0X/4G/8AI+u5oZHStFXxEv8AyRf5lTw34U03wpp6WmnW6wqPvN1dz6se5rYzik3bR7V5747+LVr4dZ7PTgl7f9GJb93H7Ejqfavp6+Iw2V0E5e7FaJL9EfPUaOIzCraF5Se7/Vs724uYbOJpZ5liiXktIwAFcbq3xi8M6ZlReNeuOotU3j8+lcBa+B/FfxHmS61i5ltbMncvnrjj/ZjB/nXb6P8ABXw7pwU3CSahIOpmf5f++RivB+vZrjtcHSUId5/5Hs/U8uwf+9VXOXaO33mNN+0JpisRFpd2R23Mi/1psP7Q2nswEul3Sr6pIj/1Fd/b+CfD9su2PR7Ebf8Apiv+FOm8F6FdKUk0izI9oFH9Kf1TPfi+sRv25Q+s5Pt7CVv8RzumfGrwzqLBXnlsW6f6TGVA/EZH612ljqFtqMIltp47iI9HjYMK4zVvgx4d1JWMMElhIejWzkAfgciuB1L4b+JPAdw19ol3LLCv3jbjD4/2k6MPw/Cpljs2wGuKpKpDq47/AHDWEy3GaYao4S7S2+8964rP1rRLHxDYyWd/brcwSDBVh09wexrz/wAD/GCLUpI7LWlW1u2OxJ14jc+hz90/WvTlwBnI+te9hcXhczpc1N8ye6f6o8bEYbEZfVSmrPo/1TPn3XvCutfCHXF1XSJpJtOdsbjkgjP3JR/Jv8n2DwX4ytPGemi4t32TKds1uT80bdfxBrcvrGDUrWa2uYknt5VKPG4yCD1FeE6xpN98H/FkN7Ys8mmy5CKTw694m9x1B/8Ar18xUo1OHqvtaOuHk9V/K31XkfQQqwzul7OrpXWz/m8n5nvy/dp1Z+haxb67pdve2sgkilXIPcHuD7ir9fcQnGpFTg7pnyUouEnGSs0LRRRVkhRRRQA3IzXkfxf8QzalfW/hywy7u6+aEPVyRtX8Op/CvUtU1CPS7G4u5iFihjLsfoM15T8MdJk8QeIr3Xb4F2iYnBH/AC0bnj6Dj8a+XzmpOs6eApPWo9fKPU9/K4wo8+NqbU1p/iex6D4N8Kw+E9FjtIsNMfmmkxgu56mt77vNHauN+JXixvD2keTbNtvroMqN3RR95v8APc1685Uctwt7WjBHmQjVx+Ite8pM574jfEG4kvDoGiFnuJD5cssPLbj/AAJ7+p7fyv8AgD4V22gpHf6kgudSIyqsdyQ/T1Pv+VHwx8Ef2bbJq16m69mXMQbkxoe/1PU16J714mBwM8ZU/tDHLX7MekV/meti8ZHDU/qWDdl9qXWT/wAhF7Ypk88drC0srLHGg3FmIAAxzmpcjHWvBf21vFt34W+BOpx2UrQzanNFp5kVsEI5O8fiqsPxNfTVZqlTc+x8hjcSsHhp4iW0U2ZfiT9uv4ceH9bk0+D+09ZSJtkl5p8CtB77SzAsPcDB7Zr2zwH490P4keHLfXPD98l9p8+cOvDKw6qwPKsPQ1+PhULnB4+ua+4f+Cddvfp4f8ZTvuGkvdQJD/dMwRvMI/4CYwfpXh4LH1q9VRmtGfnWQcT4vMsf9XrRXLK9rdLH2JSOu7IIzSKw29eKx7jxr4ftbsWk+uadDdE7fIkukV8+mCc177aitT9PlUjB+87HLePvhba+I4ZbzT0WDUwOV6JNjs3vjv8ASsP4c+Pp9JuxoWtySAK3lRTT/ejbsjH0xjB/+tXre9WG4EEYyGzXn/xS8Epqlr/alrFm8gXMqoMeavf8R1r5XMMvlhan9oYFWkviXSSPpsDjYYmCwWMd4v4X1i/8j0MEbRWV4m8P2/ibR7ixuB8sg+V8co3Zh9K534X+LG17STaXMm++tAFZj1kT+Fv6Gu4x+Ne5RqUcxwyla8ZI8irTq4Gu4t2lFnjPwx1efwn4kuPDmonZHJIRGpPCSAdvZhz+VezL90d68p+L2gta3VnrlpmOUMI5HUdCDlG/mPyr0LwzrC65oVnejAaVAWUHO1uhH55rxsolLC1KmXTfwax/wv8AyPVzOMcRTp46H2tJf4l/matLSDpS19UfPBRRRQBxPxXvja+G/IBwbiQIf90cn9BVz4daX/ZfhW0+XEkw85+OST0/TFc78WpDcXmnWoPZmx7khRXolnCttZwwoMLGgUD6DFfN4ePts0q1X9hJL56nt1pezy+lTX2m2/yJGIXJJ4HNeRRwnx58QnZzvs4Gzt6rsXgD8Wz+del+Jr06foN9cDhkibH1xx+tcn8JtN8mxu7th88rhA3so/xP6UZjF4rF0cJ9n4n8th4F/V8PVxC3+FfPc75F2qFAwoGBXz18eP2wNC+EOuSaBp+nv4g1+NQ08Sy+XDb5GQHfB+bGDgDgEZxX0I2dpA9K/KH9oPQ9R8P/ABs8YW2phzcS38lwjtn545DujI9RtIH4Edq7sfXnh6S9n10PzHijM8TluFU8NvJ2v2Ptn4Efth6J8YNdTw/f6c3h/XJU3W8bzCWG4wMlVbAw4HO0joO+Ku/treGZvEXwF1aaAbpdNmhvioBOVVsP+Ssx/Cvzp8MeILnwn4k0jXLMZuNOu4ruME9SjBsZ7AgY/E1+r/hjxJoHxo+HiXtnJHfaRq1o0csZPK7lKvGw7MMkEVzYTESxtKVKo9TyMkzSef4Ktg8S17Sz+aaPys8A+Ep/HnjPQ/Dlu5ik1K6jt/NHOxCfmbHsAT+Ffq34f8P+HvhL4HisrOODSND0u3Lu7nARVBLO7HqTySx618NfBn4YXfwt/bI0jwzfRkx2stzLaTOP9dB5EhjkHvjg46EGu+/bw+KF9dajpPw20YTObpFu7+K1BaSfLYhhCryclSxHsvasMHbCUalWS95Ox5eRcmS4HEYqrD94pcqXW/Y80/aC/a+134lalc6X4WvLjRfCsZZA9uxjnvR03O3VVPOFHY/N6D53MaSMWZfMZhyznJY+pJr6E8D/ALEPxF8WWsd3fCx8NxOu5Y9QkLTYPqiD5foTWnqH7AfxDtb2OKz1HRL22b71wZnj2e5XYT+RP4VwSo4us/aTi2fNYzAZ7mU/rNWnJ3/D5Hrf7A/xK1XxL4c13wzqlzJeJo7Ry2s0pLMkUm4GMkk8KynHsfavq9k3BgRuVuuenpXk37OvwDs/gT4XuLRbr+0NYv5BLe3uzaGwDtRRnhVycZ5JJPevW+eM19RhIThRjGpufs+S0MRhsBSpYl3ml/wx4/JCfAvxCDRkx2cx4HQbHOCPwP8AKvYA2VBFeffFrThLa2V4B80bmNiB2YcfqK7Dw5eG/wBBsbhvvSQqT9cc14+XL6tiq+FXw/Evnufb46X1jD0sS99n8iv4w0oa14dvLXGWKFl9mHI/UCuV+DuoF9Ou7JzzFIJFHoGHT8wfzr0GRRtYdfX34rzHwEn9l+NNQtBwreYoH0bI/Q08bH2WYYevHreL/QMLL2mCrUX0tJfqepClpF6UtfSHhhRRRQB5n4+TzvFtiv8AdEf6yV6TH90fSvP/AB1Hs8S2Evqq/o4r0Feg+leFgY2xWI9Ueti5Xw9D0ZzfxDfb4Xul/vlF/NhTfh7CIvDEGBgszn9T/hU3jyLzvDdzgZKlW/JhTfAbD/hHYRnlXcf+PGjl/wCFTmf8n6j5v+E+y/m/Q6KvIfj5+zjoHxz02NrlzpuvWqkWupwqCyjrscfxJnnHY8jvn1361R1zXNP8O6bcalqt5BYWFsu+W5uHCIg9ya9epThUi4z2PnMVh6OJoyp4hJx63PzZ8a/sd/E7wldP5OkJ4itQTtuNKlDEj3jfDA/TNO+Ef/C6vgzrjTaB4S10wzMDc6bc6fK9vNjufRsY+YHP1HFfaun/ALWPwp1K+e1j8X2aOv8Ay0nV4oz9HYBT+dY3jj9sr4aeD7OV7PVf+EkvgPktdKHmZOOMucKo/H8DXgvCYWnL2lOrb5n5u8jyrC1PrWHxfJbs0dx4FiPj6x0TxZ4m8GDw94psxIsUV2Ukntww2sVdeisM8HB9RWN8TdT8CfBnULr4j6zo1xcatcqlm99aWrXEqqoO0eka46nKg8Zrz/8AZh+NvjL46eNvEur38UWneE7O3jgtrCFAwE7NuyZCNzsEHPQDcOKk8J/tJSzfH/xR8NfFqWbWkl40OkXSx7RygYQSg5BJBOG7kY7ivR9tTlTjK9uZ6Nrr3Pqo4/C1MNTmnZzdoya3l0fzM2H/AIKD+BGuNjaJryRZ5kaGLp648zNeheCP2s/hn46uo7W28QLpt652rb6ohtix7AFvlJ/GvKv2j/2NdL1nTbzxH4Ds10/V4g00ukwACC6x18teiP6AcHpgda+FGG44Ktk9VcAHPpjtXmVcZisLNKrZo+Qx2fZxkuIVPFRjKL20tc/aGGRZEDIVYEZBU5FPznmvgr9iv48avpvi618BazdSX+j6iHFg07FntZVUt5YJPKMA3HYgY4NfegxwK9vC4iOJhzxP0DKM0pZthViKSt0a7M5j4iRLL4blzyVkQ/rj+tS+AWLeFbNW6qCv5Mab4/P/ABT7r/ekUfrmrHgqHyfDdoMfey35sTXm8v8Awptr+XX7z7C//Cek/wCb9Dc4rzfS4/J+Jk+O7v8Aqua9JGBmvP8AS1E3xEnkHOGkBx2woFPMo3nQt/MhYCVoVv8ACz0FelLSL0pa9w8kKKKKAOK+IdtlrKcDldyfyP8ASus0+4F1ZwzA5EiBvzFZni2z+16TKQMtEQ4x+v6U3wldibR0ToYm2Eeg7fpXkU4+yx0/7yv9x6VR+0wkH/K7feXtZtft2m3UHXehA+tc78P7jbFc2rHkEOB9Rg/yrrzzjjr7VxM8Z8P+JBKvy275J/3T1/Kli/3VaniOi0foww37ylOj31XyO29xzXxh/wAFEta1KOLwhpCmVNIuGnuJFQkLJKmwKGHQ7QzED39q+zlZXVWUgrjjBri/i18I9A+MnhZ9F16BjGriWC5hIWa3kHR0Yg44JB7EE114qlLEUXCLs2fJ5zg6uPwNTDUXaT/qx+SGF9Dj65rpfh/8Pdb+Jvii08O6Da+feXBy7N9yCPvI5/hUfqeBmvriz/4J12P9obrvxreSWOceXDZIkhXPTeWIz77a+kvhj8IfDHwg0U6d4a01bVZCGnuJDvnnYd3c8n6dB2ArwKGWVZTvV0ifleXcG4upWTxnuwXzbG/Cn4aaV8HPAdpoOnEGO3Qy3F1IAGnlPLyN9f0AA7V+Y/xU8YnxF8XPEviLTLl1Euqvc2lwhwwCP+7YH/gIP5V9d/t2fGbVPCemaf4L0aZrWTV4Hlv7hDiQW+duxT23/Nk+g96+EfvBscevtTzGtFNYeG0S+LcfSjKnl2HVlSt9/Q/Wz4N+Ol+Jvwx8P+IePNvbVTOo/hlXKyL+DBq+Dv2mv2dfEfgbx7q+raZpF1qXhrU7h7qCexiMgty5y0bqAduCTg9CMd6+n/2E5p5fgLaecTsW+uRFn+5uz/PdX0MUDLg9K9aVBYyhDndnY+6q5bT4gyyj9YdpWTuu9j8/v2OPgP4j1D4lWPi3VtLutJ0bSQ0kT3cbRNcylWQBFIyVXJO7pwAK/QL2zg9qRVAwB8vrSSSCFXdjhQMkn0rpw1COGp8sX6nrZTlVLKcP9XpO93dvzOR8fXPmfZbVOWyXYD8h/Oup0u1Fjp9vb/8APNAv6VyVnG3iDxG1ww/cxHcB7D7ortlrjwcXUq1MR0ei9EfW4l+zpwoLpq/VjZMKrZPauH8GQ/aNevboj+83/fTf/WrqPEV0bPS52U4dhsX6ms3wXY/Z7J5iDmZuuOw4/wAadePtcXSivs3YUf3eGqS76HSr0FLSL0pa9c80KKKKAIpo1mV0YZVhg1yuibtH1iW1k/1bnaM+vUH8q66sPxFp/mbbqP76cN9Ox/CvPxVN6Vo7x/I68PNa05bS/M2+vfmszXtKGp2jbQBPH80bf0NSaTfC+t13f61R83+NXvXA5rofJXp7XTMbzo1OzRznh3UjGq2VwdrAfIW4OP7pro+MVj6zoouf31v8kq8sBxn/AOvXF/Ejx3rvg3wHq2saZFYT3OkwSXU0eoCQLLEiFioKdHOBgkYrjp1ZYVezq7dGaYmUPZvER2WrXY9M5pc1yHwu1/VvFngXSdb1mOzhu9Rt47sRWO/YiOgZVJbksAeT0rmf2gPiJ4k+FnhWPX9FttKvrVJ4ree3vzIrlpZFRCrLxgFuQR+NdrqxjDne1rnlSxMIUPrEk+W1/keb/tlfs96x8VLPTfEPhmFbvWdMRoZbLcFa4gPPyk8blbJweoJ5zivkXwv+zX8S/FmtRabH4V1DTtzqJLrU4jDFCO7Fm69c4AJNfqTopvzpVt/ahtzqBQGY2gYRbu+3dzj615b8XviN4v8AAvjDwppuj22i3Fj4ivRp8L3nmiSCTYzlm2nDLhTgDBzXl4rB0Zz9tJvXc+Ozbh7A4qt9erOSva6XU7T4X/D+y+F3gPSPDVixkhsYdjTN1lcnc7n3LEn8a6rduOM1FG0iwqZADJtyzKMLnHb2rwPxd8ZPiJ4P8Dv4pv8ARdFsLeXV10+2sLjzvO8p7gxJK5yACRhtuOh616cqkKMdtEvyPrJ1aWBpRjZ8qXRdEfQO4ZNc34k1B7lhY2xzuOJGXnP+yK5PT/FnjU+MrnQ9VsNPfTZLEXFvqemrJzLv2tCytkAgHIOeRXeaNoq2KiWUBrgjHso9BXJOo8T+6p6Lqz2MLUhy+2kteiehLoulDTLULx5rDLn3rQDcc0vNUdVvxZ2524818hBXYlChTstEiW5Vp3e7MTxFM2pajFZRcqp5I55P+ArpLW3W1t0iQfKgwKx/D+msrG7mBMjfc3enrW6OlcuFpu8q095fkb4iaSjSjsvzHUUlLXonGFFFFADTTZF3KRjPqDTjR6UmlYOpgyW8ml3Hmxf6stn8O4NbVvOlzEHjII/UUskazKUcZBFZojk02bK5ZG6DHH41xqLoPT4X+B039stdzULc4/OvN/2greBvgz41l4SRdHuiG/7ZNW5efFHwtY+IE0K71u1g1ZpI4/sjvhw7gbFPYFsjAPXNN8SR+D/GlnHZa0NP1S2+2NZrDc4dDcAHMeDwTgHj2rWajVg43PJryVWlUpU5K+q3MH4Ptd2vwn8HPE3mRnSbUnHzAful/KuO/a21Tz/gvcrIojK6np+454/4+o69O8MXng7wzpNlZ6FLp+n6bciaa2htcLG4T/WsoHGF744rndW1T4V/FW3uLrUJdF1+2023+0StdYZIouTvO7jbx96uJ0JRpezhMxryhVwP1Vtczja9/I9Ej1O3CKC/zY/u14l+0FewTfED4NMHyo8S/wDtB67vw74v8AR+GzJo+q6dBo2nvHasUl2RwNIQI1Oem4sMZ65rN8caL8Lb+SbVfFFrpF1La3C2ZnvPnaObYGVF7htpBwOcHNaTVWcLNoeKdOth+SE1fRvXTR/8A72TWl24ij3N0+avDv2t5Z5vhNE1xJ5SNrWnbckL/wAvC8816cvxS8E6aulRrrFlCL6IPaIM5kj3bAQMZ68c96p+NrP4c+K5j/wlCaPqb2BliIvwriFkVXkGDwCFdSfQMKmpTlVptSma4ypTxGFnRpNJtWvfudtYWNvDH5kfzOwyX6mrq/dridP8ZeCPCOoW3hCz1LT9NvI2WKHTEfbtZxuVAOgJzkDvnirUHxQ8LX2vSaJaaza3GrpI8ZtFf5y6D516clQDkdsV1xdOnHsbKvCVryV9tzpru6S0Qu3J6Be5rIgs5NUuDNPkR55HqPQfjVhLWW/k3yk7O/GPwFaqKEUBVwBwBWXK60ry+FHepeyVo7scq7VAxj6UtIpyKWuzY5wooopgFFJkUUALRSZFFAB3pJFDLhhkUd6XIqd9GB4j4r+Aup+IPiNda9FrEEGnXN7Y3jwMJt6m32fKFEgjbcUHLIcZ9hV+y+CUNr4kTWjqEzXo1ubVirSS+V5bo6iIRF9gYbx84HYnua9e4prRhuoFc/1eCu4nnRy/DqTm46t3+Z454d+DutWNxaQXur6fNpWnW99BYrBaOs/+ktkmQs5Hyj0AzWL4d/Zv1TT9D1uxvdbhkkvdFXRoXiWZggBz5h8yRsHp8q4H6Y96EI7ZFHluv8XNR7COl0ZvK8K+W6enn3VvyPJrr4Bx3VvrlncatcarZarLp7yrqf79ylvJueNmOMhxwPQkmszUf2fdRbw2+nQ6/wDarhNd/ta3uLxZFbyhD5SQu8bqxKpgbwQTgZzk17Zhu5pNrschsVXsYdi3leGlG3Lbp+p4trPwP1zUP7Ee11W00y8tLZLe41C3ku/PYLMZMAmbDLjPEm7lj24qLxZ8D9d1nXNVez17TbfTNQuJpik1k7zJ58UcUoDCQDhY8rx1PNe3rGx+8c07yVznGTSdBO6XUj+y8NZpp6269jyPVPgrLqnii81iPW7uIz6vaamtlz9mxCIwVZM8s3l53Z4yvHFO8N/AlfD/AI+t/En9oSXS/ab26lspnkaFJJpGZJYk3bUkVXKMcYYEng165tC9qXFNYeF7s0/s/DKXMo63v8wX7owMCnUgpa6z0QooooAKKKKAI2bDCnUUUDfQGo70UUiWDUd6KKEPoA6GhaKKa2EA6UnvRRQMKX+Iiiihkgv3adRRQUFFFFABRRRQAUUUUAFFFFAH/9k=)

**MATLAB CODES OF EXPERIMENTS**

SUBMITTED BY:

SUDIP NAYAK (B220061)

(ETC, Semester-4)

**International Institute of Information Technology**

**Bhubaneswar, Odisha, India**

DSP Lab

Experiment -1

Unit Step Response

Code:

% Generation of a Unit Step Sequence

clf;

% Generate a vector from -10 to 20

n = -10:20;

% Generate the unit step sequence

s = [zeros(1,10) ones(1,21)];

% Plot the unit step sequence

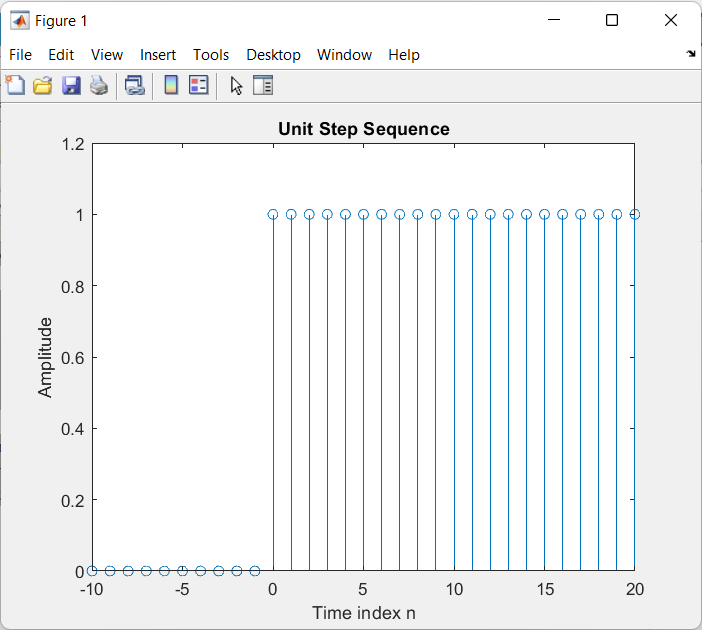
stem(n,s);

xlabel('Time index n');ylabel('Amplitude');

title('Unit Step Sequence');

axis([-10 20 0 1.2]);

Output:



Unit Sample

Code:

% Generation of a Unit Sample Sequence

clf;

% Generate a vector from -10 to 20

n = -10:20;

% Generate the unit sample sequence

u = [zeros(1,10) 1 zeros(1,20)];

% Plot the unit sample sequence

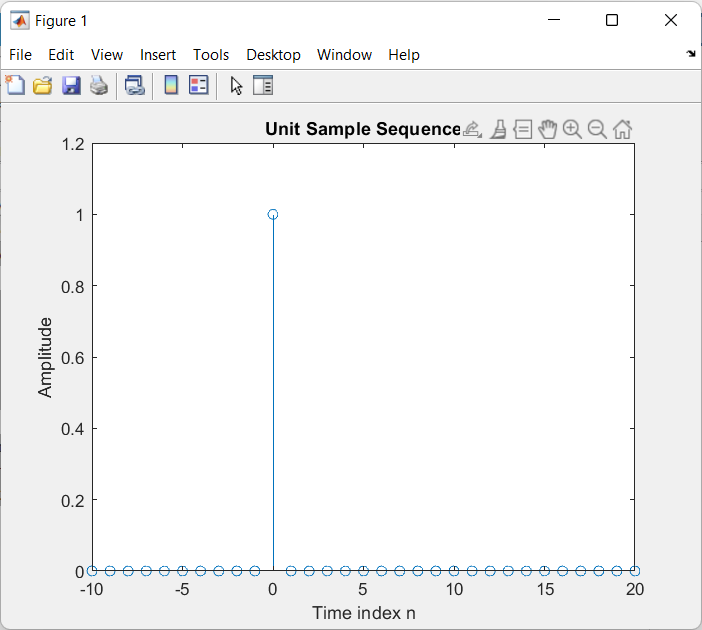
stem(n,u);

xlabel('Time index n');ylabel('Amplitude');

title('Unit Sample Sequence');

axis([-10 20 0 1.2]);

Output



Sinusoidal Signal

Code:  
n = 0:40;

f = 0.1;

phase = 0;

A = 1.5;

arg = 2\*pi\*f\*n - phase;

x = A\*cos(arg);

clf; % Clear old graph

stem(n,x); % Plot the generated sequence

axis([0 40 -2 2]);

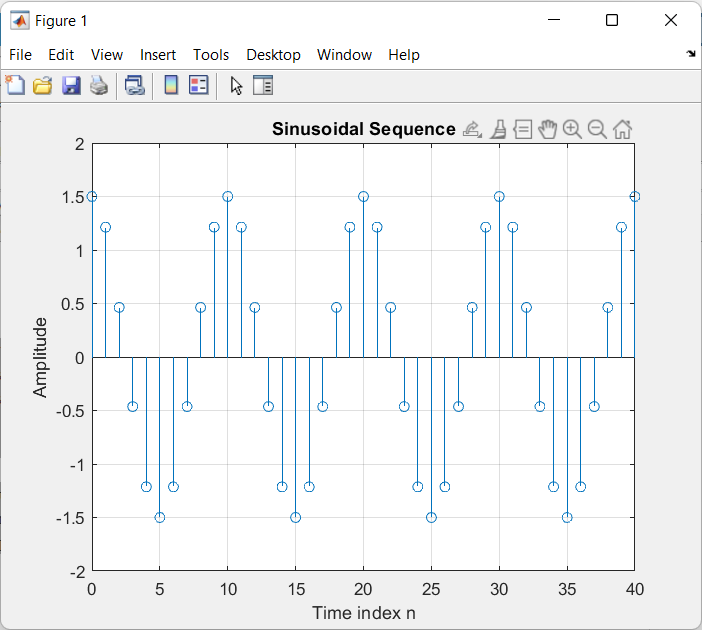
grid;

title('Sinusoidal Sequence');

xlabel('Time index n');

ylabel('Amplitude');

axis;

Output:  


Real Valued Exponential

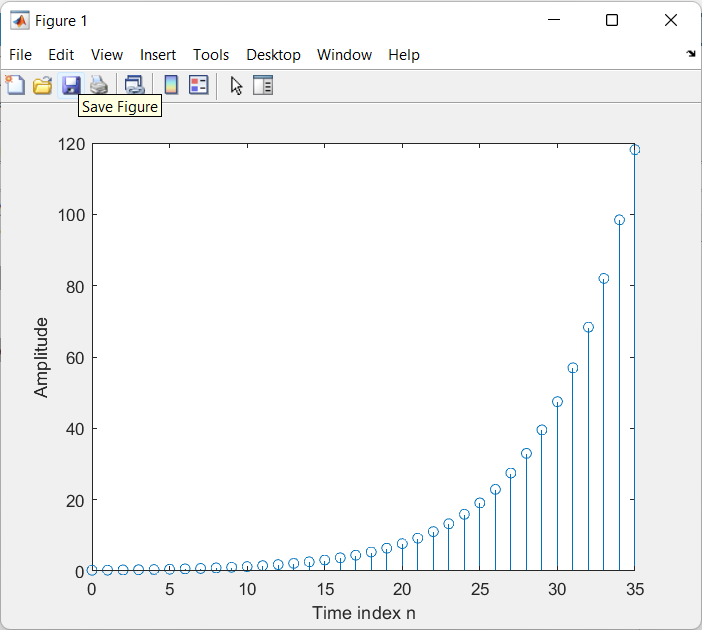
Code:  
clf;

n = 0:35; a = 1.2; K = 0.2;

x = K\*a.^n;

stem(n,x);

xlabel('Time index n');ylabel('Amplitude');

Output:  


Complex Exponential

Code:

clf;

c = -(1/12)+(pi/6)\*1i;

K = 2;

n = 0:40;

x = K\*exp(c\*n);

subplot(2,1,1);

stem(n,real(x));

xlabel('Time index n');ylabel('Amplitude');

title('Real part');

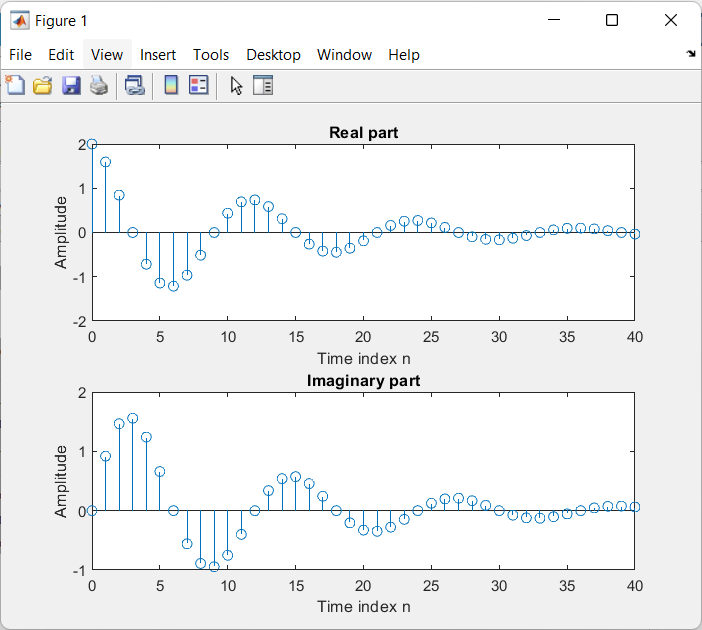
subplot(2,1,2);

stem(n,imag(x));

xlabel('Time index n');ylabel('Amplitude');

title('Imaginary part');

Output:



DSP Lab

Experiment -2

Scaling

Code:  
t = [0 1 2 3 4] ;

x1 = [0 2 -2 1 4] ;

x2 = [2 -3 4 5 0] ;

subplot(321);

stem(t,x1 .\* 0.3);

title('Scaling x1 by 0.3');

xlim([-1 5]);

xlabel('Time');

ylabel('Amplitude');

subplot(322);

stem(t,x1 .\* 0.7);

title('Scaling x1 by 0.7');

xlim([-1 5]);

xlabel('Time');

ylabel('Amplitude');

subplot(323);

stem(t,x1 .\* 1.3);

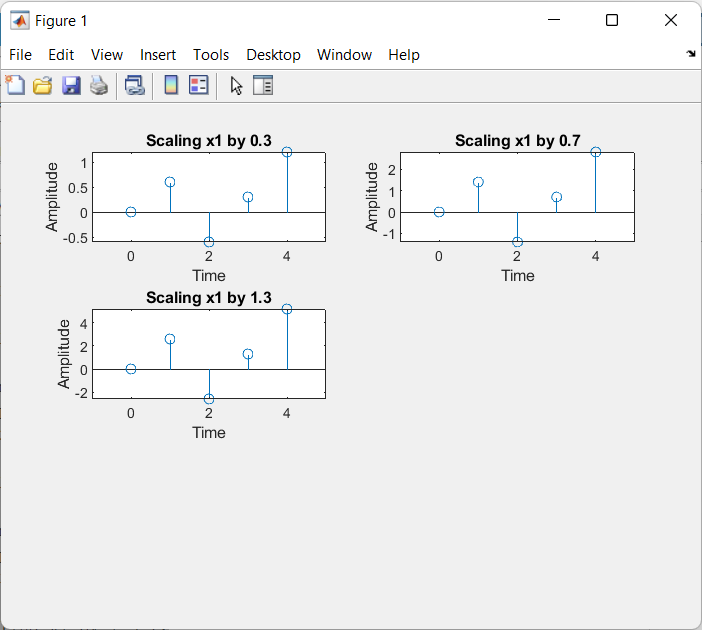
title('Scaling x1 by 1.3');

xlim([-1 5]);

xlabel('Time');

ylabel('Amplitude');

Output:



Time Shifting:  
Code:  
t = [0 1 2 3 4] ;

x1 = [0 2 -2 1 4] ;

x2 = [2 -3 4 5 0] ;

subplot(321);

xlim([-5 10]);

stem(t-3,x1);

xlabel('Time');

ylabel('Amplitude');

title('Shifting x1 to 3 units left');

subplot(322);

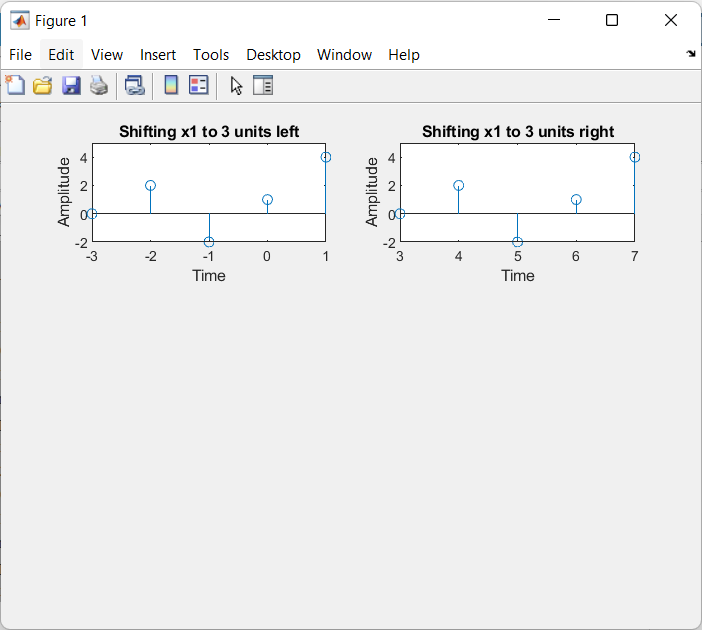
xlim([-5 10]);

stem(t+3,x1);

xlabel('Time');

ylabel('Amplitude');

title('Shifting x1 to 3 units right');

Output:  


Signal Folding

Code:  
t = [0 1 2 3 4] ;

x1 = [0 2 -2 1 4] ;

x2 = fliplr(x1);

t1 = -fliplr(t);

subplot(321);

xlim([-5 10]);

stem(t,x1);

xlabel('Time');

ylabel('Amplitude');

title('x1');

subplot(322);

xlim([-5 10]);

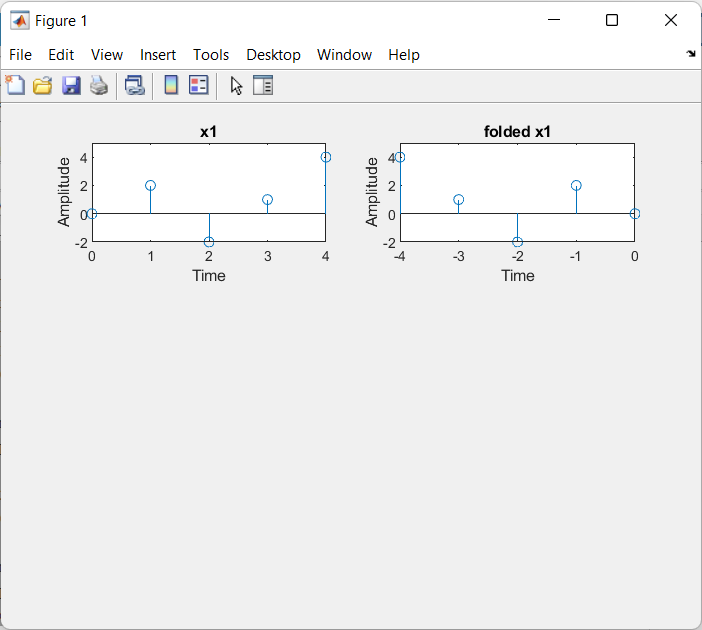
stem(t1,x2);

xlabel('Time');

ylabel('Amplitude');

title('folded x1');

Output:



Odd part and Even part of a signal

Code:  
t = [0 1 2 3 4] ;

x1 = [0 2 -2 1 4] ;

x2 = fliplr(x1);

x3 = (x1+x2)\*(1/2);

x4 = (x1-x2)\*(1/2);

subplot(321);

xlim([-5 5]);

stem(t,x3);

xlabel('Time');

ylabel('Amplitude');

title('Even Component');

subplot(322);

xlim([-5 5]);

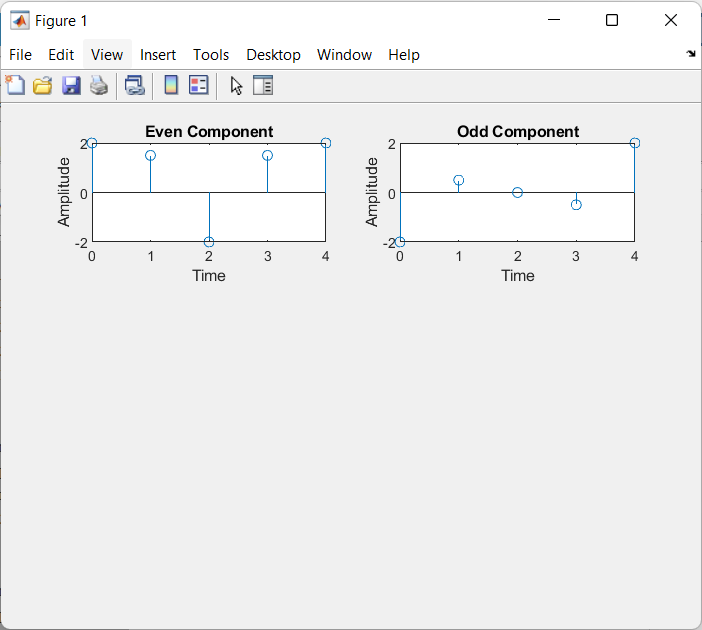
stem(t,x4);

xlabel('Time');

ylabel('Amplitude');

title('Odd Component');

Output:



Product and Sum of 2 signals:

Code:

t = [0 1 2 3 4] ;

x1 = [0 2 -2 1 4] ;

x2 = [2 -3 4 5 0] ;

subplot(321);

stem(t,x1);

title("x1");

xlim([-1 5]);

xlabel('Time');

ylabel('Amplitude');

subplot(322);

stem(t,x2);

title('x2');

xlim([-1 5]);

xlabel('Time');

ylabel('Amplitude');

subplot(323);

stem(t,x1+x2);

title('Sum');

xlim([-1 5]);

xlabel('Time');

ylabel('Amplitude');

subplot(324);

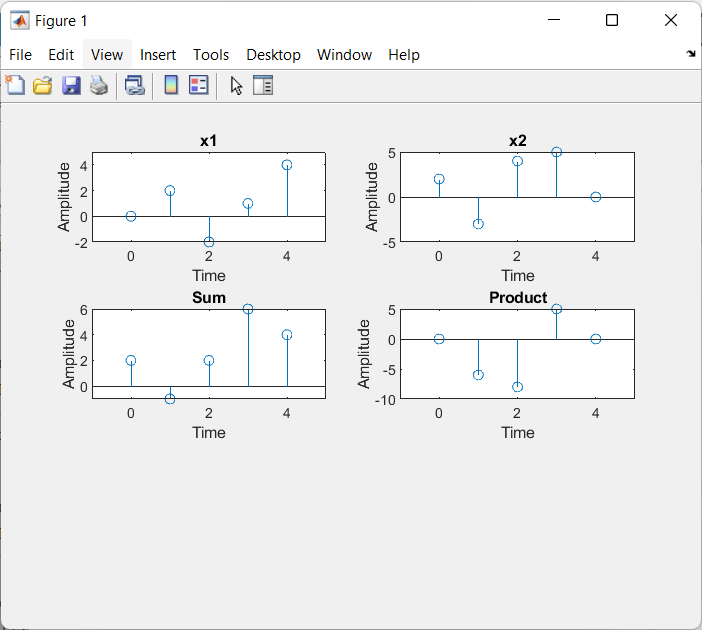
stem(t,x1 .\* x2);

title('Product');

xlim([-1 5]);

xlabel('Time');

ylabel('Amplitude');

Output:  


Energy of A signal

Code:

n = -2:2;

x\_n = n;

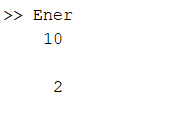
E = sum(abs(x\_n).^2);

P = E/length(n);

disp(E);

disp(P);

Output:



DSP Lab

Experiment -3

Linear Convolution (Without Built In Function)

Code:  
x = [5 6 1 2];

h = [10 6 4 8 9 5];

m=length(x);

n=length(h);

X=[x,zeros(1,n)];

H=[h,zeros(1,m)];

for i=1:n+m-1

Y(i)=0;

for j=1:m

if(i-j+1>0)

Y(i)=Y(i)+X(j)\*H(i-j+1);

else

end

end

end

Y

subplot(223);

stem(Y);

title('linear conv of sequence');

subplot(221);

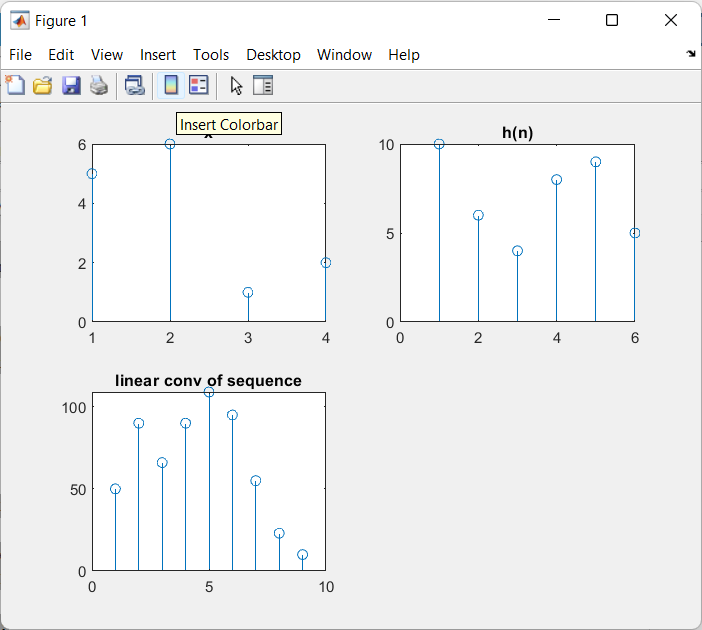
stem(x);

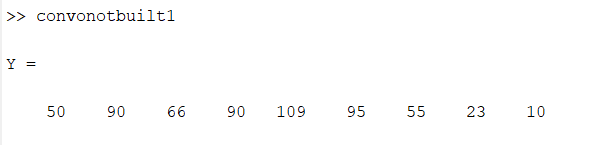
title('x');

subplot(222);

stem(h);

title('h(n)');

Output:  




Linear Convolution (With Built-In Function)

Code:

x = [5 6 1 2];

h = [10 6 4 8 9 5];

f=conv(x,h);

disp(f);

xlabel('time index n');

ylabel('amplitude');

subplot(223);

stem(f);

title('linear conv of sequence');

subplot(221);

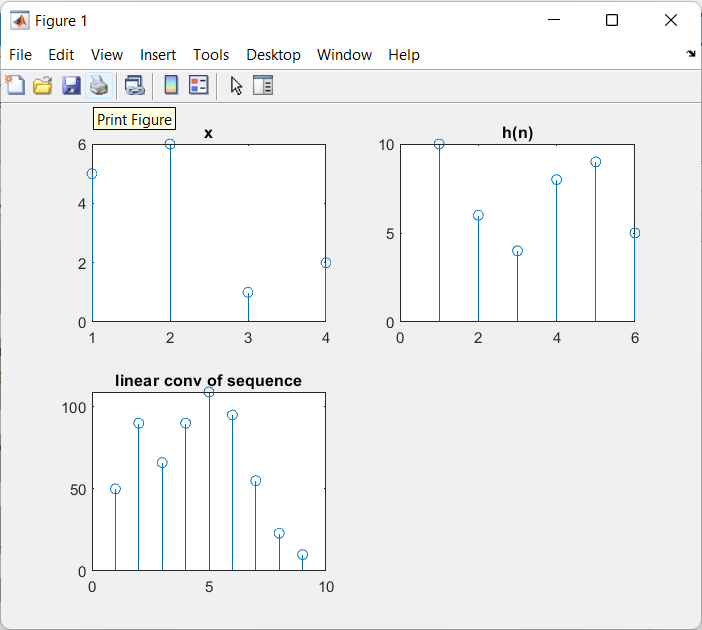
stem(x);

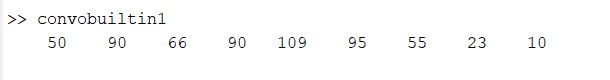
title('x');

subplot(222);

stem(h);

title('h(n)');

Output:  




Circular Convolution

Code:  
x1 = [3 2 1 0];

x2 = [1 2 3 4];

ccirc = ifft(fft(x1).\*fft(x2));

subplot(2,2,3);

stem(ccirc,'filled');

title('Circular Convolution')

subplot(221);

stem(x1);

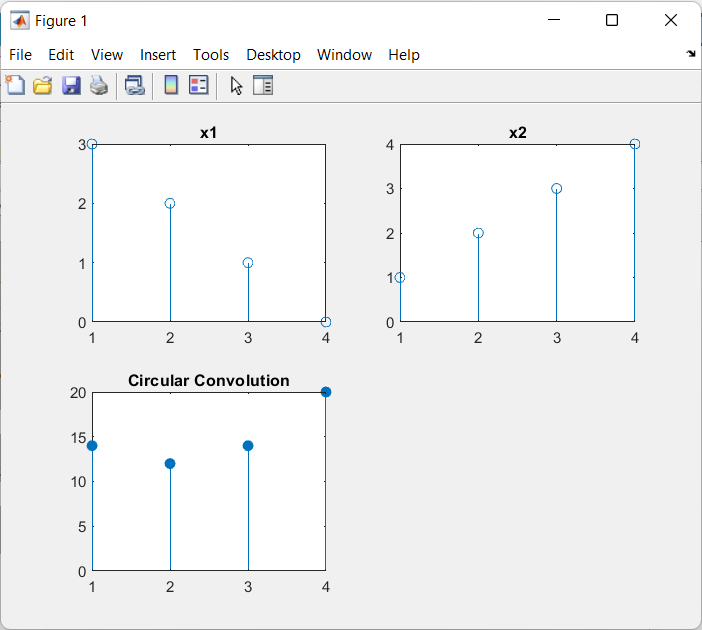
title('x1');

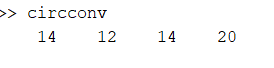
subplot(222);

stem(x2);

title('x2');

disp(ccirc);

Output:  




DSP Lab

Experiment -4

Cross Correlation

Code:  
x = [1 2 3 4];

x2 = [5 6 7 8];

ylim([0 50]);

y = xcorr(x,x2);

subplot(311);

stem(x);

title('Value of X');

subplot(312);

stem(x2);

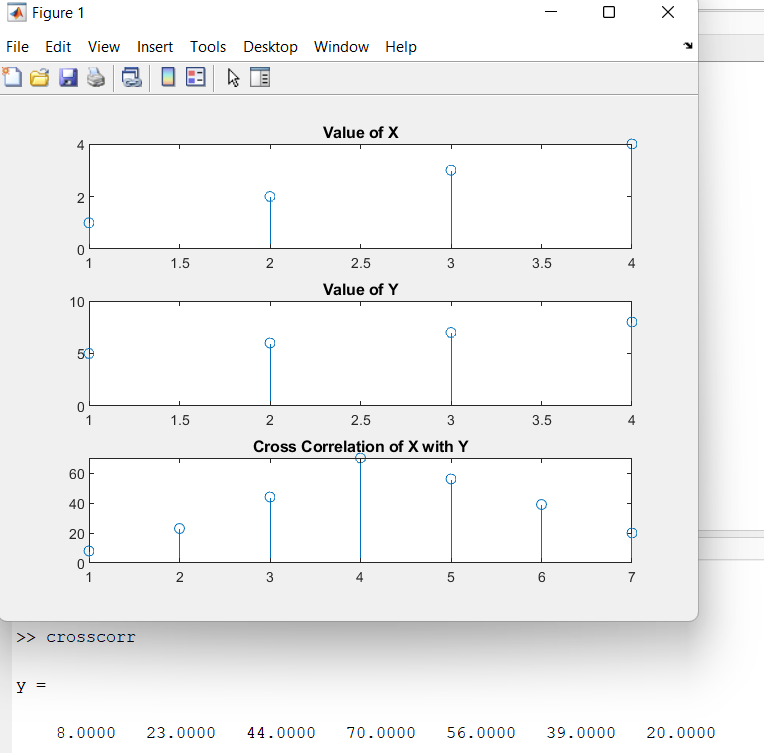
title('Value of Y');

subplot(313);

stem(y);

title('Cross Correlation of X with Y');

display(y);

Output  


Auto-Correlation

Code:  
x = [1 2 3 4];

y = xcorr(x,x);

ylim([0 50]);

subplot(211);

stem(x);

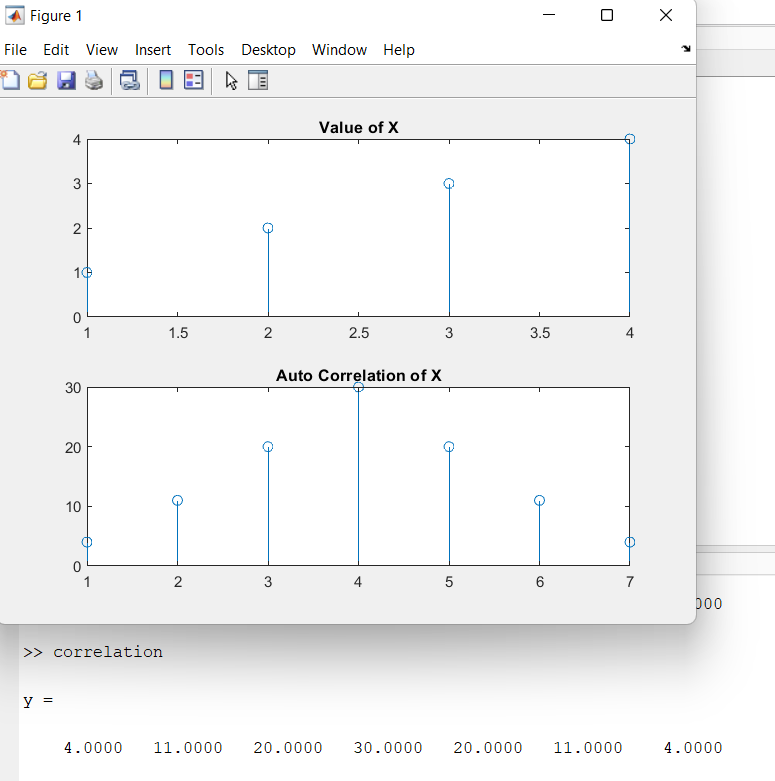
title('Value of X')

subplot(212);

stem(y);

title('Auto Correlation of X');

display(y);



DSP Lab

Experiment -5

Low pass, High pass and Bandpass filters

Code:

%low pass filter

Fpass = 100;

Fstop = 150;

Apass = 0.5;

Astop = 65;

Fs = 1e3;

d = designfilt('lowpassiir',...

'PassbandFrequency',Fpass,'StopbandFrequency',Fstop, ...

'PassbandRipple',Apass,'StopbandAttenuation',Astop, ...

'DesignMethod','butter','SampleRate',Fs);

fvtool(d)

%high pass filter

Fstop = 350;

Fpass = 400;

Astop = 65;

Apass = 0.5;

Fs = 1e3;

d = designfilt('highpassiir','StopbandFrequency',Fstop ,...

'PassbandFrequency',Fpass,'StopbandAttenuation',Astop, ...

'PassbandRipple',Apass,'SampleRate',Fs,'DesignMethod','butter');

fvtool(d)

Fstop1 = 150;

Fpass1 = 200;

Fpass2 = 300;

Fstop2 = 350;

Astop1 = 65;

Apass = 0.5;

Astop2 = 65;

Fs = 1e3;

d = designfilt('bandpassiir', ...

'StopbandFrequency1',Fstop1,'PassbandFrequency1', Fpass1, ...

'PassbandFrequency2',Fpass2,'StopbandFrequency2', Fstop2, ...

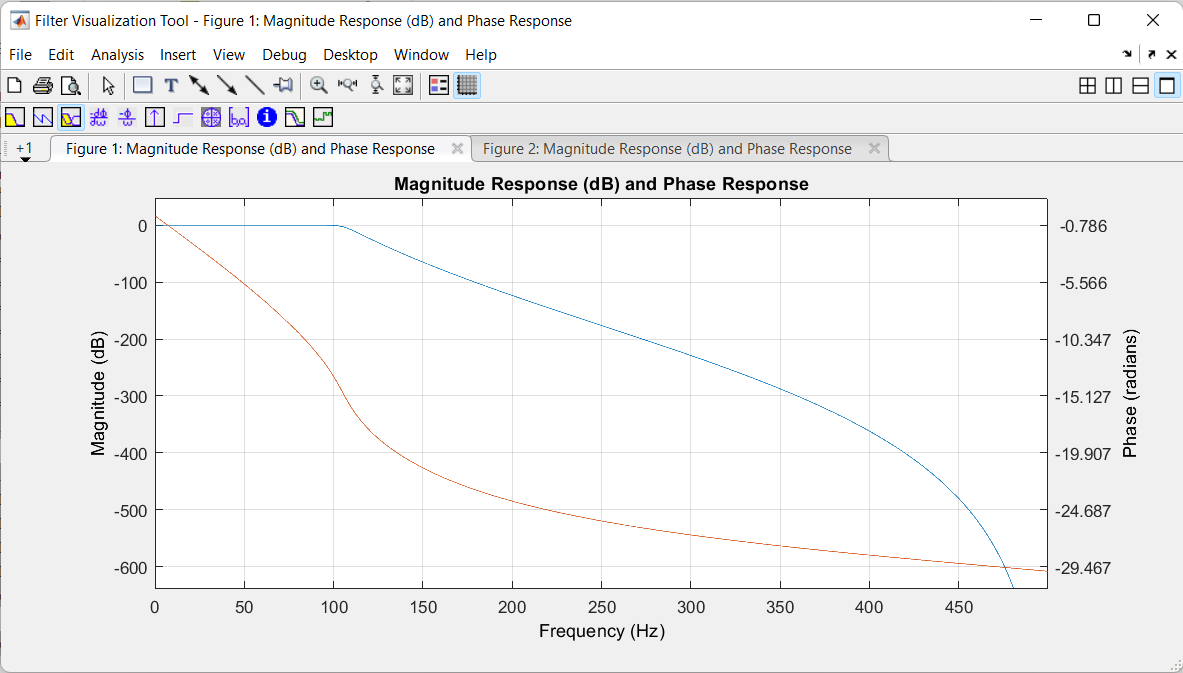
'StopbandAttenuation1',Astop1,'PassbandRipple', Apass, ...

'StopbandAttenuation2',Astop2, ...

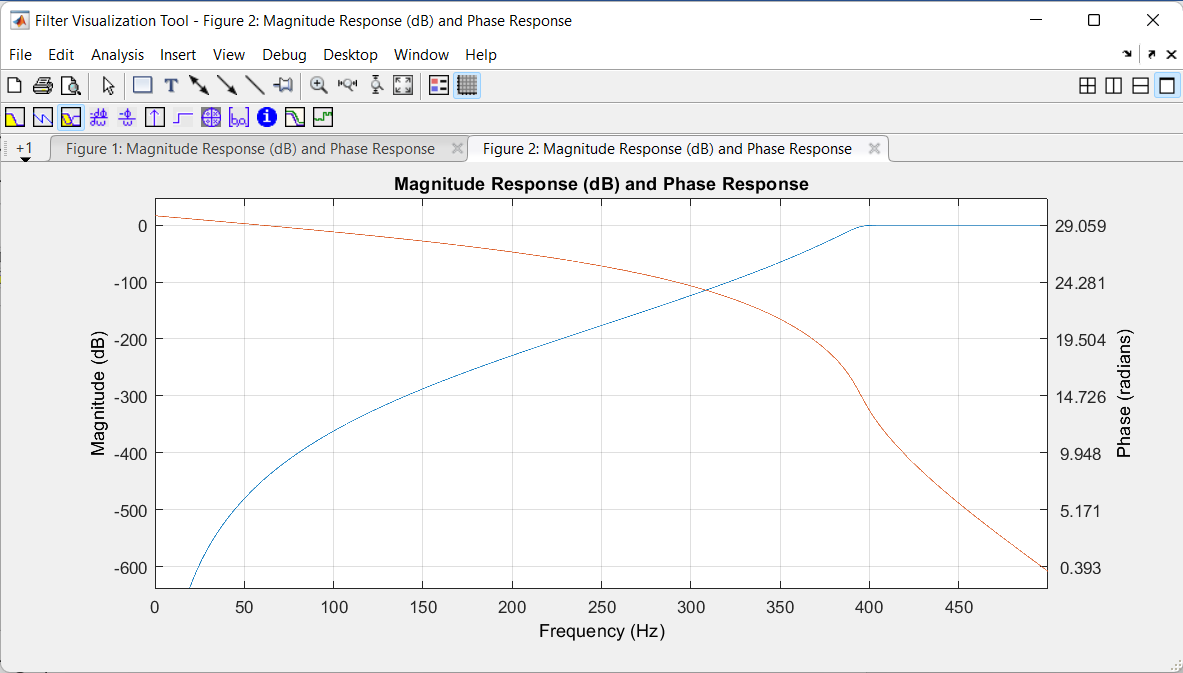
'DesignMethod','butter','SampleRate', Fs);

fvtool(d)

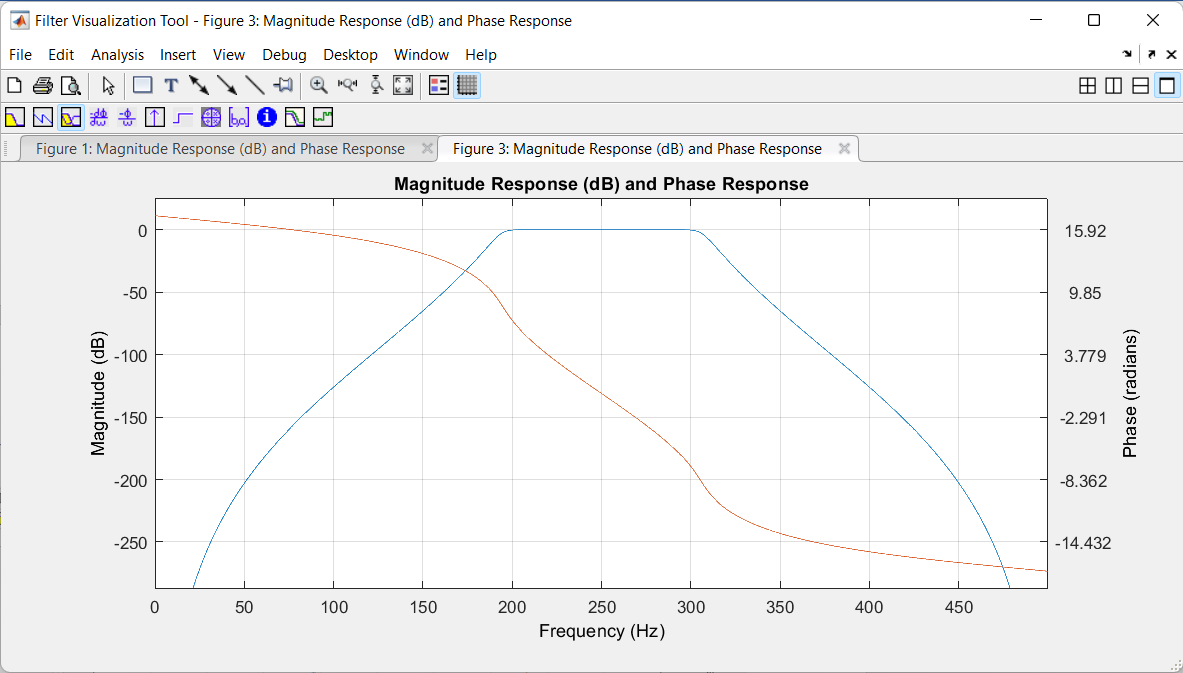
Output:  
Low pass



High pass



Band pass



DSP Lab

Experiment -6

DFT

Code:

%Function eg\_5\_a

clc;

close all;

N= input('Enter the nunber of samples : ');

x= input('Enter the sequence to find DFT : ');

L = length(x);

while(N<L)

x=input('Invalid Sequence given... Enter another sequence to find DFT :');

L=length(x);

end

if(L<N)

xn=[x,zeros(1,N-L)];

else

xn=x;

end

XK=(zeros(1,N));

for k = 0: N - 1

for n=0:N-1

XK(k+1)=XK(k+1)+(xn(n+1)\*exp(-2\*1i\*pi\*n\*k/N));

end

end

t=0:N-1;

subplot(3,2,1);stem(t,xn);title('x(n)');xlabel('n');ylabel('amplitude');

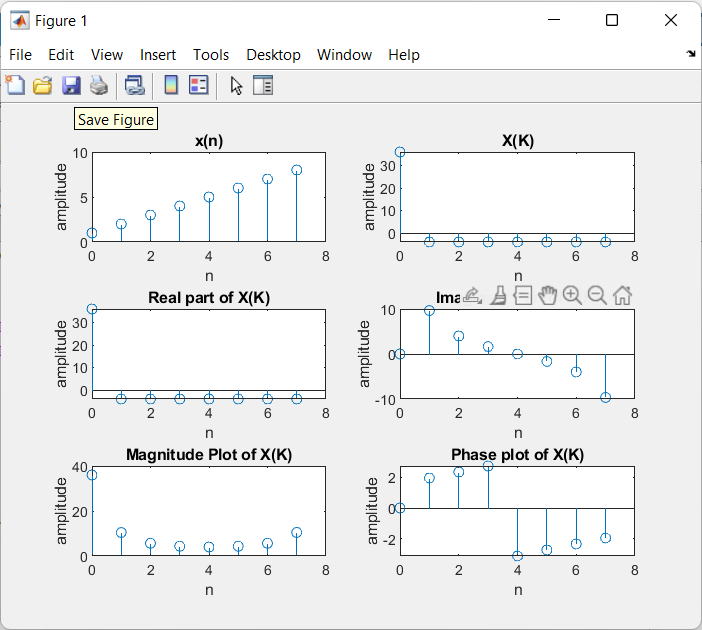
subplot(3,2,2);stem(t,XK);title('X(K)');xlabel('n');ylabel('amplitude');

subplot(3,2,3);stem(t,real(XK));title('Real part of X(K)');xlabel('n');ylabel('amplitude');

subplot(3,2,4);stem(t,imag(XK));title('Imaginary Part of X(K)');xlabel('n');ylabel('amplitude');

subplot(3,2,5);stem(t,abs(XK));title('Magnitude Plot of X(K)');xlabel('n');ylabel('amplitude');

subplot(3,2,6);stem(t,angle(XK));title('Phase plot of X(K)');xlabel('n');ylabel('amplitude');

Output:  


IDFT

Code:  
%Function eg\_5\_a

clc;

close all;

N= input('Enter the nunber of samples : ');

x= input('Enter the sequence to find IDFT : ');

L = length(x);

while(N<L)

x=input('Invalid Sequence given... Enter another sequence to find IDFT :');

L=length(x);

end

if(L<N)

xn=[x,zeros(1,N-L)];

else

xn=x;

end

XK=(zeros(1,N));

for k = 0: N - 1

for n=0:N-1

XK(k+1)=(XK(k+1)+(xn(n+1)\*exp(2\*1i\*pi\*n\*k/N)))/N;

end

end

t=0:N-1;

subplot(3,2,1);stem(t,xn);title('x(n)');xlabel('n');ylabel('amplitude');

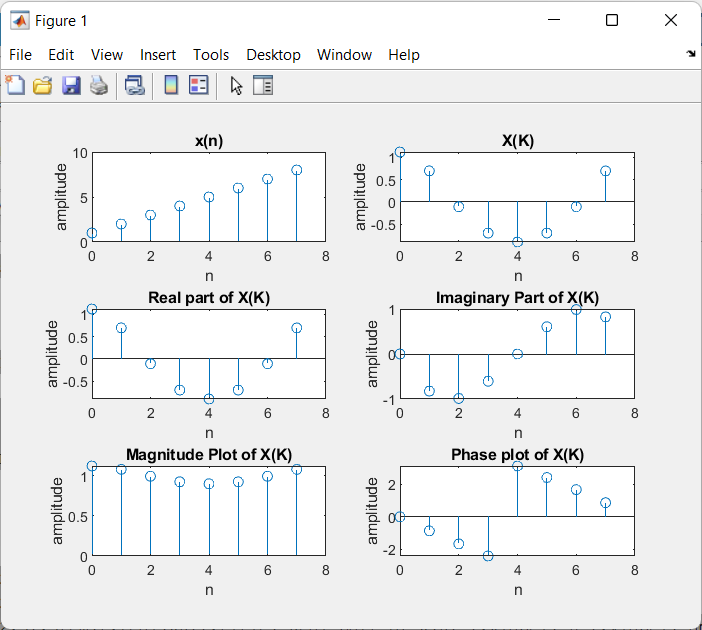
subplot(3,2,2);stem(t,XK);title('X(K)');xlabel('n');ylabel('amplitude');

subplot(3,2,3);stem(t,real(XK));title('Real part of X(K)');xlabel('n');ylabel('amplitude');

subplot(3,2,4);stem(t,imag(XK));title('Imaginary Part of X(K)');xlabel('n');ylabel('amplitude');

subplot(3,2,5);stem(t,abs(XK));title('Magnitude Plot of X(K)');xlabel('n');ylabel('amplitude');

subplot(3,2,6);stem(t,angle(XK));title('Phase plot of X(K)');xlabel('n');ylabel('amplitude');

Output:  


DSP Lab

Experiment -7

DIT-FFT

Code:  
X = [1 2 3 4];

Y = fft(X,4);

subplot(321);

stem(X);

title('Given Signal');

subplot(322);

stem(real(Y));

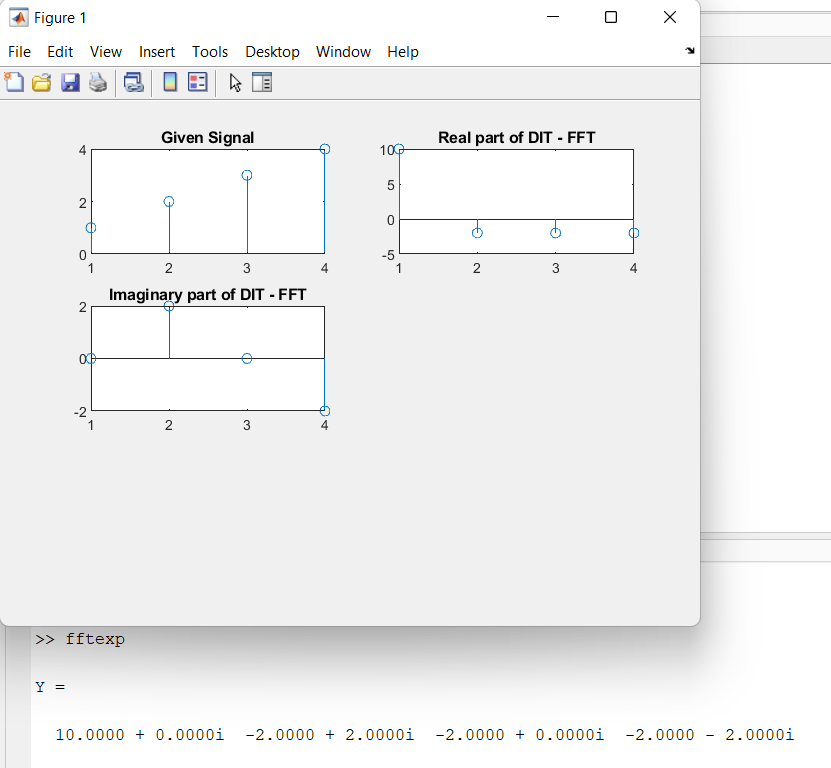
title('Real part of DIT - FFT');

subplot(323);

stem(imag(Y));

title('Imaginary part of DIT - FFT');

display(Y);

Output:  


DIF-FFT

Code:

X = [1 2 3 4];

Y = ifft(X,4);

subplot(321);

stem(X);

title('Given Signal');

subplot(322);

stem(real(Y));

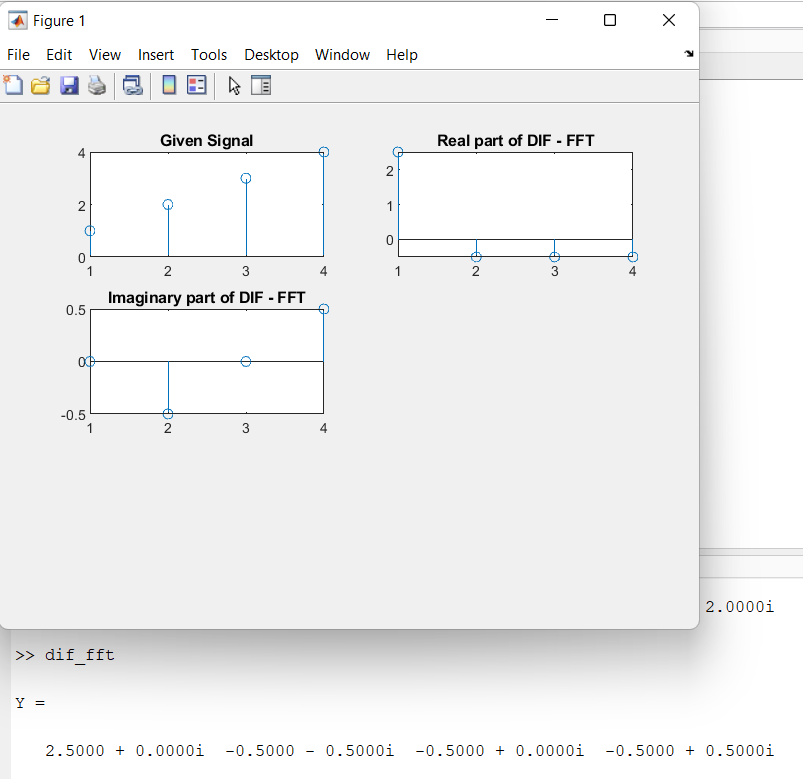
title('Real part of DIF - FFT');

subplot(323);

stem(imag(Y));

title('Imaginary part of DIF - FFT');

display(Y);

Output:  


DSP Lab

Experiment -8

IIR Filters: High pass, Low pass and band pass:

Code:

%To design IIR Butterworth filter:

%for data sampled at 1000Hz, design a LPF with less than 3dB of

%Ripple in the passband. Defined from 0 to 40Hz and at least 60dB

%of attenuation in the stopband. Defined from 150Hz to the Nyquist frequency 500Hz.

%N.B:- Butterworth filter is a signal processing filter designed to have a

% frequency response as flat as possible in passband.

fs = 1000 ;

Wp = 40/500; Ws = 150/500;

figure('Name','Low-Pass');

[n,Wn] = buttord(Wp,Ws,3,60);

[z,p,k] = butter(n,Wn);

SOS = zp2sos(z,p,k);

freqz(SOS,1024,fs); % Plotting the frequency response

figure('Name','High-Pass');

title('High-Pass Filter')

[z,p,k] = butter(n,Wn,'high');

SOS = zp2sos(z,p,k);

freqz(SOS,1024,fs); % Plotting the frequency response

figure('Name','BandPass');

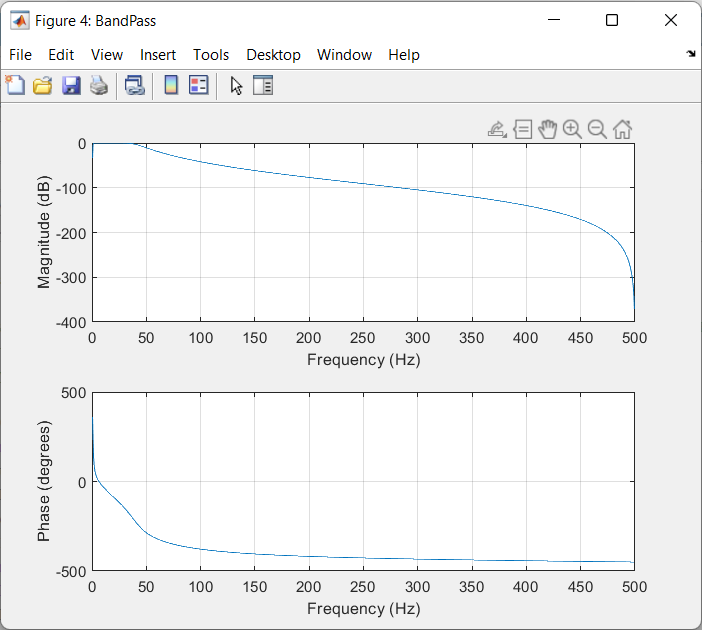
[z,p,k] = butter(n,[1/500 40/500],'bandpass');

SOS = zp2sos(z,p,k);

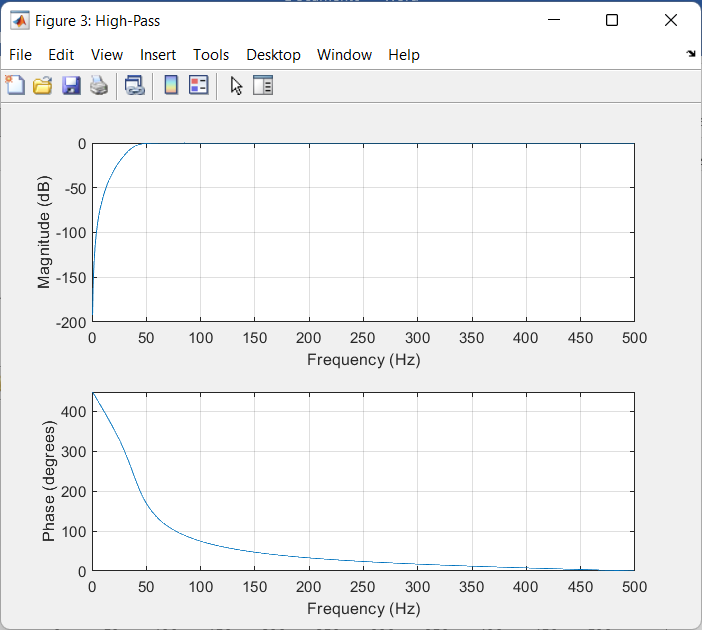
freqz(SOS,1024,fs); % Plotting the frequency response

Output:

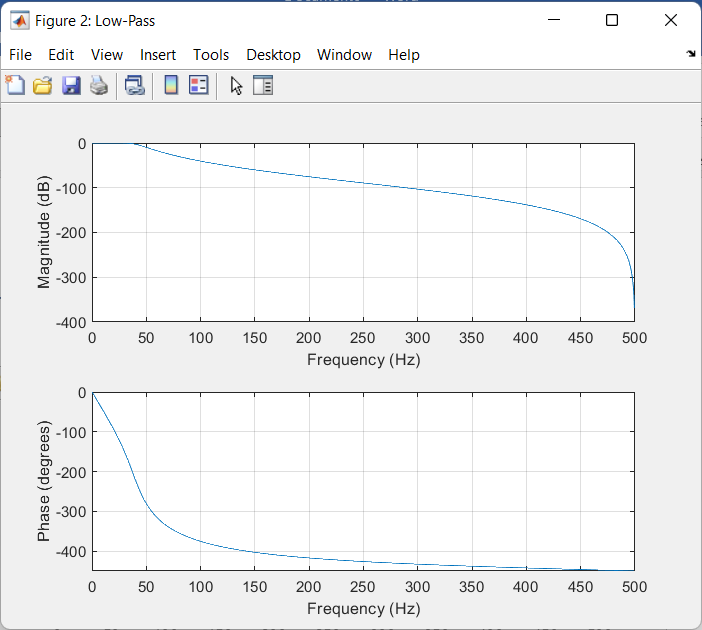
Band pass



High pass



Low pass



DSP Lab

Experiment -9

Overlap Add method for long sequences

Code:

%Q1.Do the Linear convolution of long sequence using overlap add method where

% x(n)=[10 98 12 3 -9 4 -7 5 3 1 ]

% h(n)=[2 3 4]

close All

clear All

clc

x=[10 98 12 3 -9 4 -7 5 3 1 ]; % input sequence

h=[2 3 4];

L=8;

N1=length(x);

M=length(h);

lc=conv(x,h);

x=[x zeros(1,mod(-N1,L))];

N2=length(x);

h=[h zeros(1,L-1)];

H=fft(h,L+M-1);

S=N2/L;

index=1:L;

X=[zeros(M-1)];

for stage=1:S

xm=[x(index) zeros(1,M-1)]; % Selecting sequence to process

X1=fft(xm,L+M-1);

Y=X1.\*H;

Y=ifft(Y);

Z=X((length(X)-M+2):length(X))+Y(1:M-1); %Samples Added in every stage

X=[X(1:(stage-1)\*L) Z Y(M:M+L-1)];

index=stage\*L+1:(stage+1)\*L;

end

subplot(2,1,1)

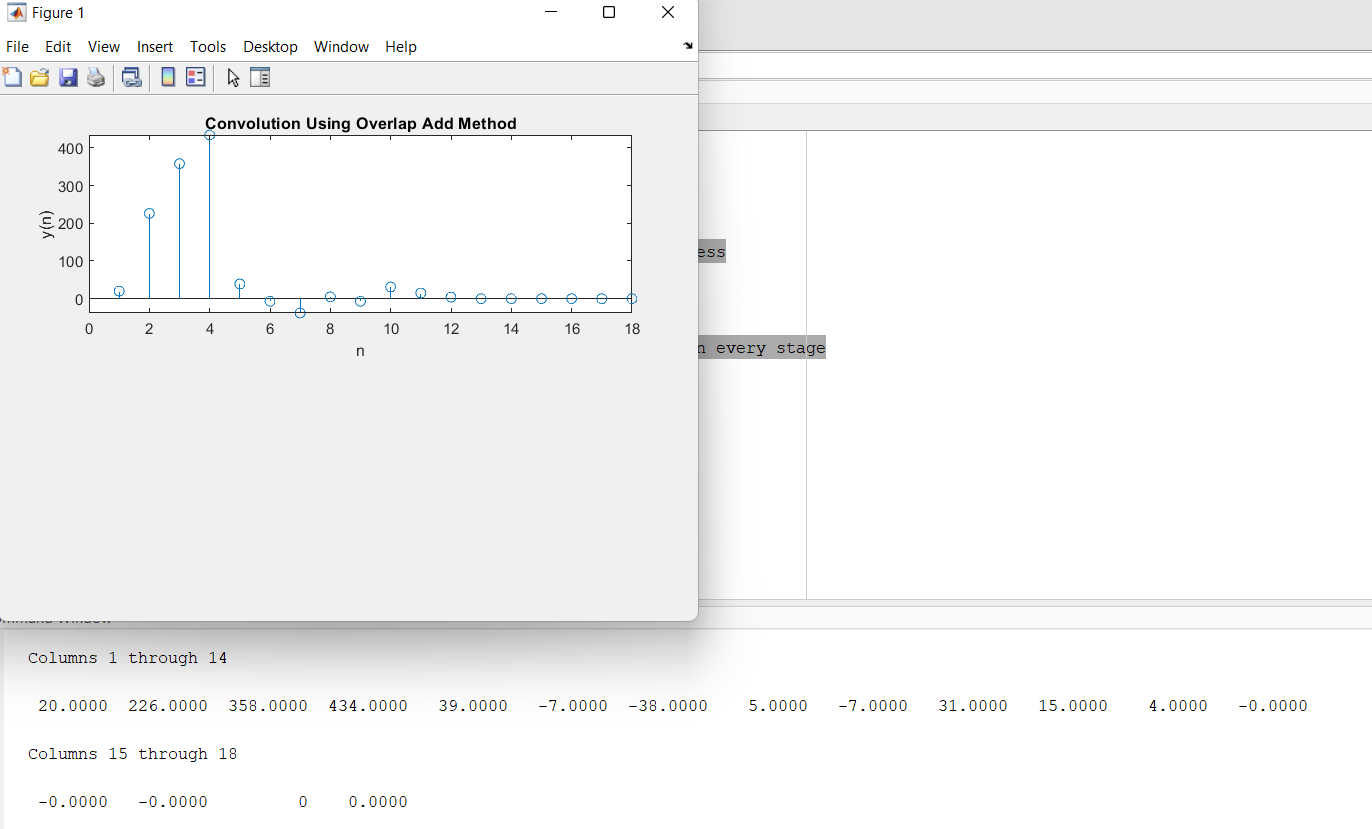
stem(X);

title('Convolution Using Overlap Add Method')

xlabel('n');

ylabel('y(n)');

display(X);

Output:  


DSP Lab

Experiment -9

Overlap Save Method for long sequences

Code:

%Q2.Do the Linear convolution of long sequence using overlap save method where

% x(n)=[1 2 3 4 9 8 4 -9 7 10 12 45]

% h(n)=[1 2 3 4]

close All

clear All

clc

x=[1 2 3 4 9 8 4 -9 7 10 12 45];

h=[1 2 3 4];

L=16;

N1=length(x);

M=length(h);

lc=conv(x,h);

x=[x zeros(1,mod(-N1,L)) zeros(1,L)];

N2=length(x);

h=[h zeros(1,L-1)];

H=fft(h,L+M-1);

S=N2/L;

index=1:L;

xm=x(index); % For first stage Special Case

x1=[zeros(1,M-1) xm];

X=[];

for stage=1:S

X1=fft(x1,L+M-1);

Y=X1.\*H;

Y=ifft(Y);

index2=M:M+L-1;

Y=Y(index2); %Discarding Samples

X=[X Y];

index3=(((stage)\*L)-M+2):((stage+1)\*L); % Selecting Sequence to process

if(index3(L+M-1)<=N2)

x1=x(index3);

end

end;

subplot(1,1,1);

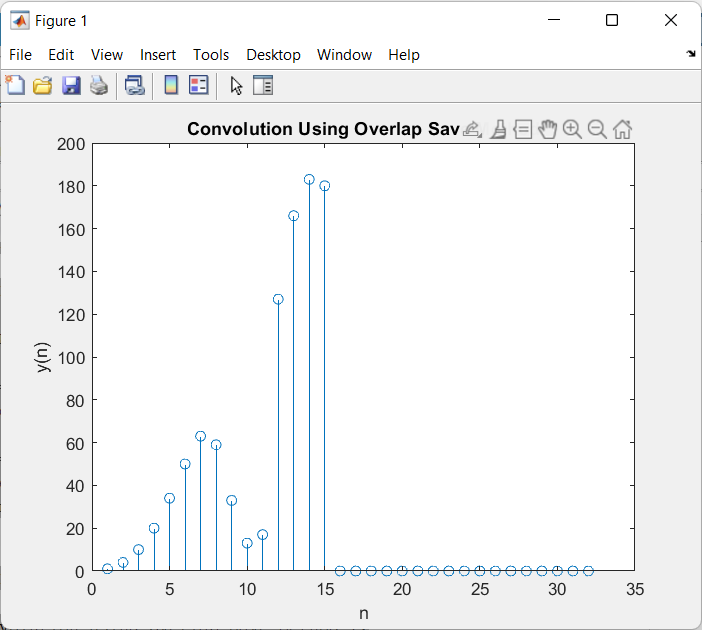
stem(X);

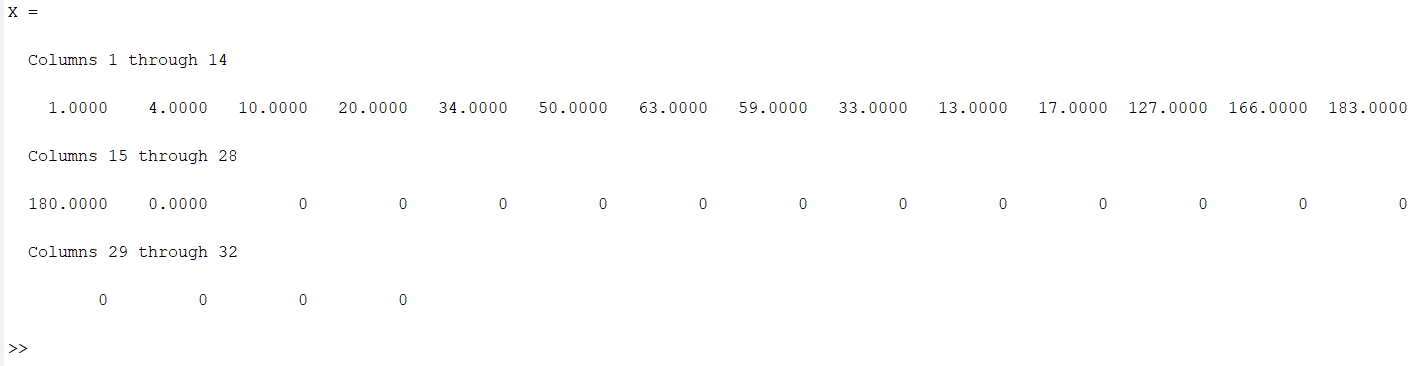
title('Convolution Using Overlap Save Method');

xlabel('n');

ylabel('y(n)');

display(X);

Output:  




DSP Lab

Experiment -10

Moving Average Filter:  
Code:  
close all;

clc;

t=1:1/100:2;

x1=sin(2\*pi\*5\*t);

x2=rand(1,101);

In=x1+x2;

Param='Center';

Len=5;

Out=movavgFilt(In,Len,Param);

subplot(411)

plot(t,x1);title('Original Signal');

subplot(412)

plot(t,In);title('Noisy Signal');

subplot(413)

plot(t,Out);title('Filtered Signal');

fvtool(Out);

subplot(427);

plot(t,abs(fft(Out))); title("Magnitude Response");

subplot(428);

plot(t,angle(fft(Out))); title("Phase Response");

function Out = movavgFilt(In, Len, Param)

Siz = size (In);

Siz\_In = Siz (1, 2);

if (isequal (Param, 'Left'))

Pad = zeros (1, Len - 1);

New\_In = [Pad In];

for i = 1:Siz\_In

temp = 0;

for j = 1:Len

temp = temp + New\_In(i + j - 1);

end

Out(i) = temp / Len;

end

elseif (isequal (Param, 'Center'))

len1 = mod (Len, 2);

if isequal (len1, 0)

error ('Cannot use the Len as an even number for this option. Use Left or Right');

else

Pad\_Len = (Len - 1)/2;

Pad = zeros (1, Pad\_Len);

New\_In = [Pad In Pad];

for i = 1:Siz\_In

temp = 0;

for j = 1:Len

temp = temp + New\_In(i + j - 1);

end

Out(i) = temp / Len;

end

end

elseif (isequal (Param, 'Right'))

Pad = zeros (1, Len - 1);

New\_In = [In Pad];

for i = 1:Siz\_In

temp = 0;

for j = 1:Len

temp = temp + New\_In(i + j - 1);

end

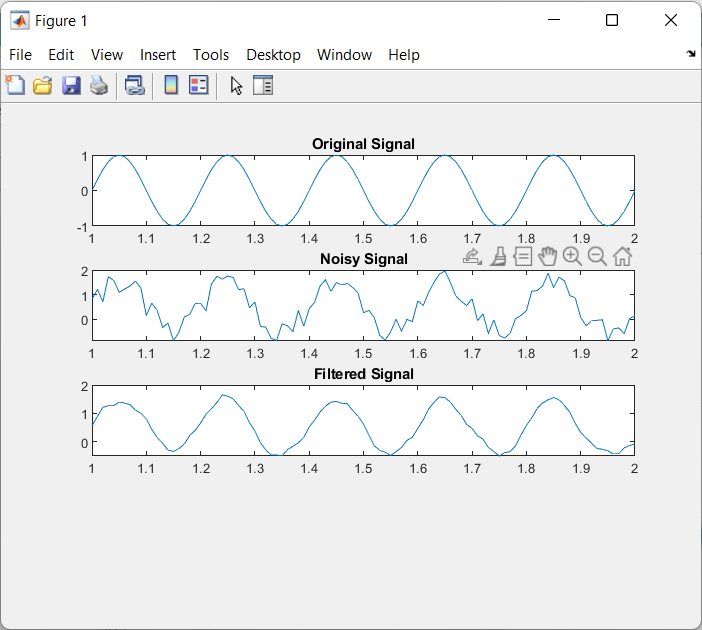
Out(i) = temp / Len;

end

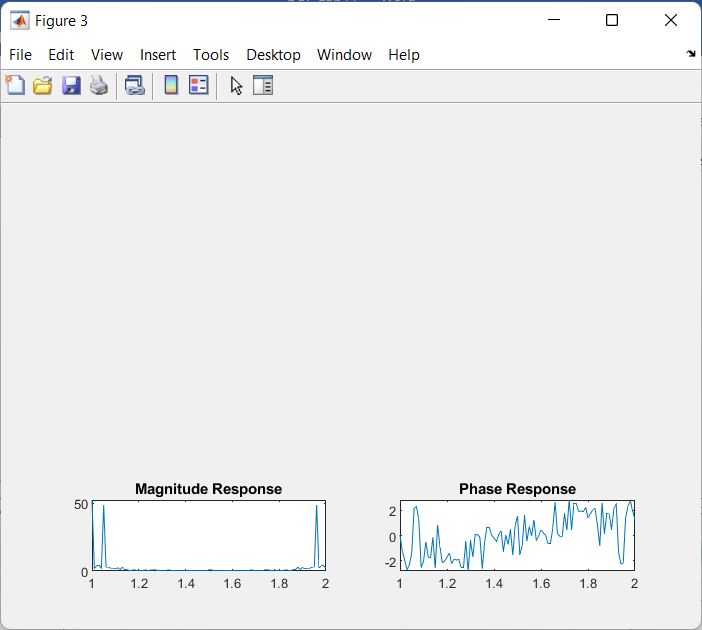
end

end

Output:

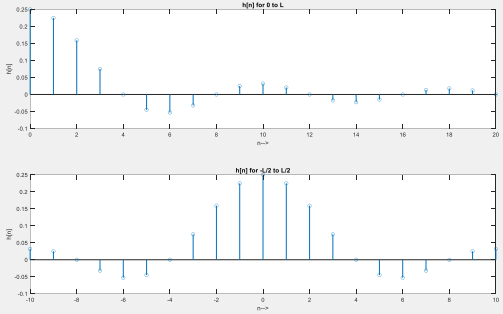




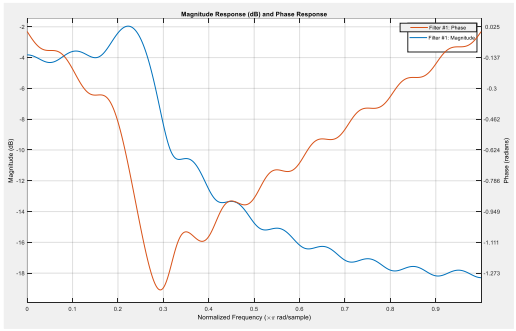


**Q. Implement the low pass filter with the given impulse response: ℎ[𝑛] = sin( 𝑛𝜋 4 ⁄ ) 𝑛𝜋 Determine the magnitude and phase response spectrum of this filter for symmetric and non-symmetric values of n and test the filter both low and high frequency signals.**

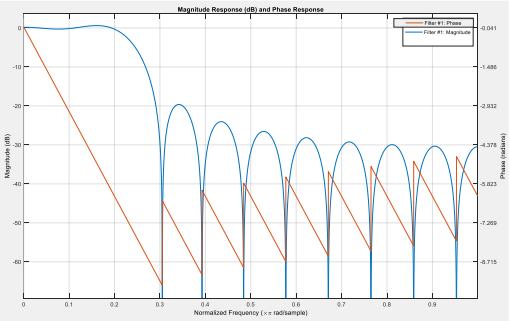
Ans: For 20 samples (i.e., for L = 20)



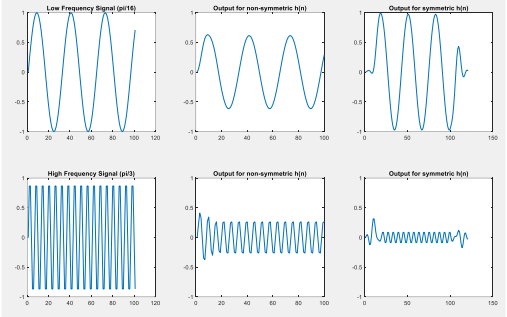
The top figure is for n values from 0 to 20 and is the non-symmetric filter. The bottom figure is for the filt.er with n values from -10 to 10 and is the symmetric filter. As we can see, in the bottom filter, the values a symmetric about n = 0.



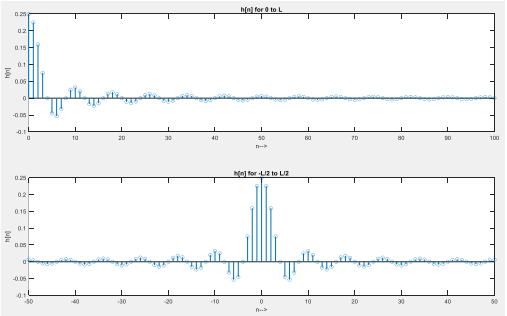
The figure above shows the magnitude and phase response for the non-symmetric filter with 20 samples. As we can see, the cut-off frequency is not precisely defined which is due to the low number of samples used. Also, the maximum attenuation for frquencies above cutoff frequency is -18dB. We can also notice that the phase spectrum is not linear.



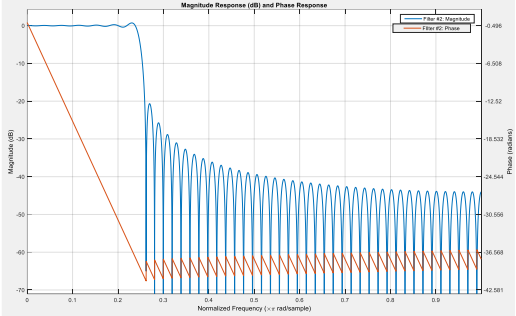
This figure shows the magnitude and phase response for the symmetric filter with 20 samples. As we can clearly see, although the cut-off frequency is still not precisely visible, it is much clearer than the non-symmetric one. Also, the phase spectrum is linear for this filter. In contrast to the non-symmetric filter, here the attenuation is also much better for high frequencies (-30dB as compared to -18dB for non-symmetric filter).



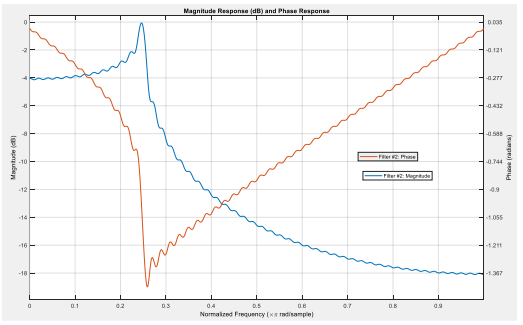
The figure above shows the output when high and low-frequency signals are passed to these filters. For the low-frequency signal, the non-symmetric filter reduces the maximum amplitude of the signal from 1 to roughly 0.6 while the symmetric filter maintains the amplitude at 1. However, the symmetric filter introduces a delay of 5 samples in the output. For 100 samples (i.e., for L = 100) The figure below shows the impulse response of the same filter but with 100 samples instead of 20.



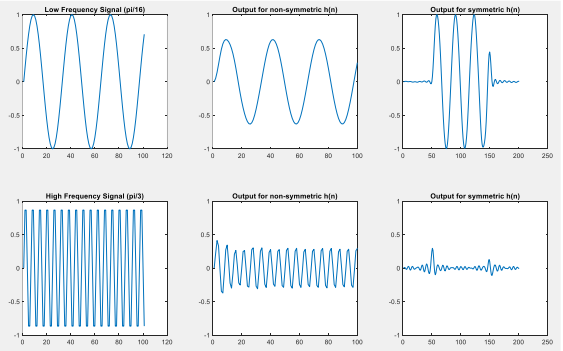
The top figure shows the non-symmetric filter with n ranging from 0 to 100 and the bottom one shows the symmetric filter with n ranging from -50 to 50.



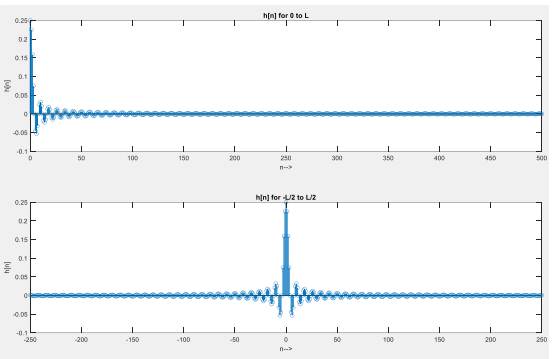
The above figure shows the magnitude and phase response for the symmetric filter. As compared to the symmetric filter with 20 samples, here the cut-off frequency is much more clearly defined. Also, the maximum amplitude attenuation for higher frequencies is roughly - 45dB. The ripple effect near the end of pass-band is also reduced.



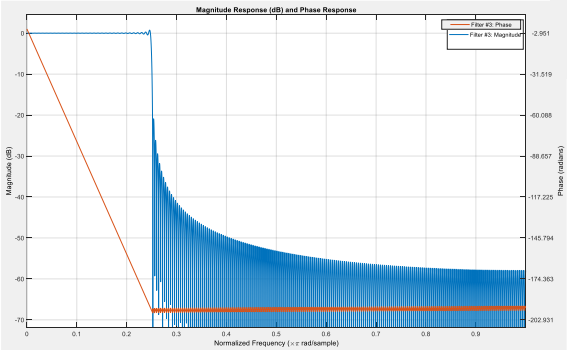
The above figure shows the magnitude and phase response for the non-symmetric filter. As compared to the symmetric filter, the maximum amplitude attenuation for higher frequencies is -18dB which is very un-optimal as compared to the symmetric filter. While the cut-off frequency is easily identifiable from the magnitude response (marked by a sharp peak), there is a high rippling effect near the end of the pass-band.



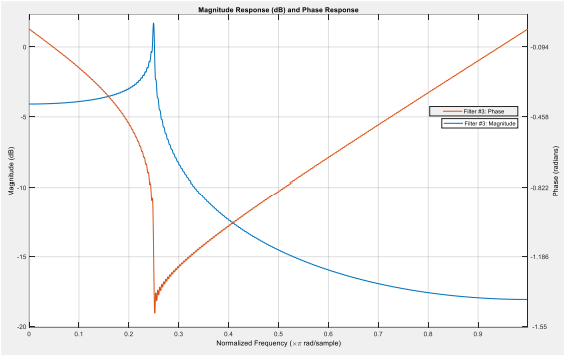
The figure above shows the output when high and low-frequency signals are passed to these filters. For the low-frequency signal, the non-symmetric filter reduces the maximum amplitude of the signal from 1 to 0.5 while the symmetric filter maintains the amplitude at 1. However, the symmetric filter introduces a delay of 50 samples in the output. For the highfrequency signal, the symmetric filter does a much better task of suppressing the signal. From these results, it is evident that the symmetric filter is a much better filter as compared to the non-symmetric one. For 500 samples (i.e., for L = 500) The figure below shows the impulse response of the filter for 100 samples. The top figure is for the non-symmetric filter with n ranging from 0 to 500. The bottom figure is for the symmetric filter with n ranging from -250 to 250.



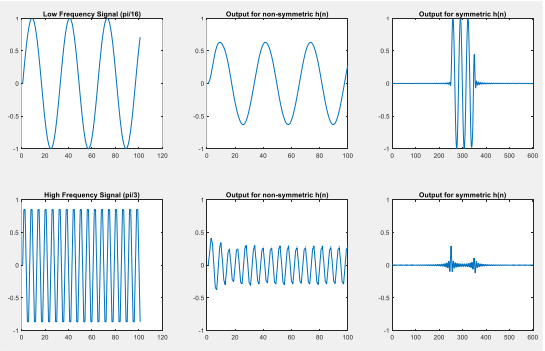
The figure below shows the magnitude and phase response for the symmetric filter with 500 samples. As we can see, increasing the number of samples to 500 makes the rippling effect almost unnoticeable near the end of the passband. Also, the maximum attenuation for the higher frequencies is almost -58dB which almost ideal for most use cases. Also, the phase response is almost perfectly linear.



The figure below shows the magnitude and phase response for the non-symmetric filter. As we can see the magnitude and phase response curves are clearly visible and the cut-off frequency is easily identifiable from the magnitude response as being π/4. However, the maximum attenuation for higher frequencies is roughly -18dB which is much worse as compared to the symmetric filter. Also, the phase response is a curve and not linear in the passband.

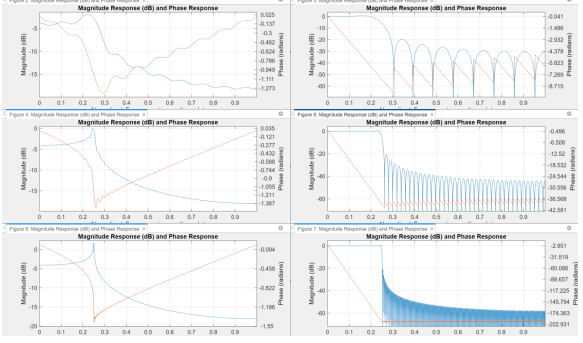


The figure below shows the output of the filter given the input signal.

For the low-frequency signal, the non-symmetric filter compresses amplitude of the signal from 1 to roughly 0.6 while the symmetric filter allows the signal to pass through maintaining the maximum amplitude at 1. However, in the output from the symmetric filter, the output seems to be delayed by 250 (L/2) samples, which is due the implementation of the non-causal filter in MATLAB. For high frequency signal, both filters suppress the signal but the symmetric filter does it much better as the signal is barely noticeable for symmetric filter while the nonsymmetric filter output is not ideal. This output could have been predicted from the magnitude responses of the filters.

Comparision:

The figure above shows the magnitude and phase response of all the filters in one picture to make it easier to compare these. The left side is for non-symmetric filters and the right-side is for symmetric filters. The top row is for 20 sample filters, middle row is for 100 samples and last row is for 500 samples.



The figure above shows the magnitude and phase response of all the filters in one picture to make it easier to compare these. The left side is for non-symmetric filters and the right-side is for symmetric filters. The top row is for 20 sample filters, middle row is for 100 samples and last row is for 500 samples.

**MATLAB CODE:**

%hd = sin(n\*pi/4)/(n\*pi) ;

L = 20 ;

h1 = zeros(1,L+1) ;

n = zeros(1,L+1) ;

for i = 1:L+1

n(i) = i - 1 ;

if n(i) == 0

h1(i) = 1/4 ;

else

h1(i) = sin((i-1)\*pi/4)/((i-1)\*pi);

end

%h1 stores values for n from 0 to L

end

subplot(211);

stem(n,h1); title('h[n] for 0 to L');

xlabel('n-->'); ylabel('h[n]');

fvtool(h1);

h2 = zeros(1,L+1);

n = zeros(1,L+1);

for i = 1:L+1

n(i) = i - floor(L/2) - 1 ;

if n(i) == 0

h2(i) = 1/4 ;

else

h2(i) = sin(n(i)\*pi/4)/(n(i)\*pi);

end

%h2 stores values for n from -L/2 to L / 2

end

subplot(212);

stem(n,h2); title('h[n] for -L/2 to L/2');

xlabel('n-->'); ylabel('h[n]');

fvtool(h2);

figure('Name','Filter with Sinusoidal Input');

n1 = 0 : 1 : 100 ;

subplot(231);

x1 = sin(n1\*pi/16);

plot(x1); title('Low Frequency Signal (pi/16)'); ylim([-1 1]);

subplot(232);

plot(conv(x1,h1)); title('Output for non-symmetric h(n)'); ylim([-1 1]); xlim([0 100]) ;

subplot(233);

plot(conv(x1,h2)); title('Output for symmetric h(n)'); ylim([-1 1]);

subplot(234);

x2 = sin(n1\*pi/3);

plot(x2); title('High Frequency Signal (pi/3)'); ylim([-1 1]);

subplot(235);

plot(conv(x2,h1)); title('Output for non-symmetric h(n)'); ylim([-1 1]);xlim([0 100]) ;

subplot(236);

plot(conv(x2,h2)); title('Output for symmetric h(n)'); ylim([-1 1]);

The above code was run multiple times with changing the value of L for different filters.

**Conclusion:**

From these observations, we concluded that the symmetric low-pass filter is a much better filter as compared to the non-symmetric filter. Also, increasing the number of samples for the filters made the spectrum of the filters much clearer and the output was much clearer for them. While increasing the number of samples made the output look much better, it also increased the time required for simulating the filters. For example, for calculating output of the system with 500 sample filter requires much more computational time and resources as compared to the system with 100 sample filter.

SUBMITTED BY: -

NAME: SUDIP NAYAK

STUDENT ID: B220061

BRANCH: ETC (4TH SEM)