```
classdef LLOPRFPM < FFTIFO

    methods
        function obj = LLOPRFPM( varargin )
            obj@FFTIFO( varargin{:} );
        end

        function defineIFO( obj, varargin )

            rmloss = 1;
            ITMloss = 1;
            angleScale = 1;
            armScale = 1;
            armLoss = 1;
            RMSize = -8;
            TMMap = 1;
            BSNN = 2;
            invFocalY = 0;
            withArm = 1;

% Interacive parameter setting
%           eval( UtilTK.setVarsInputs('set params: ', 'angleScale', 'armScale',↙
'RMSize', 'TMMap', 'BSNN', 'withArm' ) );
            eval( UtilTK.setVars( who, true, varargin{:} ) );

            invFocalY = 1/(-82.00e3) * invFocalY;

            if TMMap ~= 0
                armLoss = 1;
                armLossPPM = 0;
            else
                armLoss = 0;
                armLossPPM = 35e-6;
            end

            dPRC12 = 16.6107; dPRC23 = 16.1647; dPRC3B = 19.5384;
            dCP = 0.10;  dBSCPX = 4.8046;  dBSCPY = 4.8497; dCPITM = 0.005;
            thetaPR2 = 0.79/180*pi*angleScale;  thetaPR3 = 0.615/180*pi*angleScale;
            thetaSR2 = 0.87/180*pi;  thetaSR3 = 0.785/180*pi;

            Atm = 0.326 * armScale;
            Apr3 = 0.262 * armScale;
            Abs = 0.3671 * armScale;
            dBS = 0.06;

            % add optics
            obj.addLaser( 'src' );
            obj.addMirror( 'PRM', 'invRoC', -1/11.009, 'Thick', 0.0749, ...
                'T', 3.1e-2, 'Lhr', (15+0.5)*1e-6*rmloss, 'Lar', (11.45+26)*1e-6*rmloss,↙
'Aperture', RMSize);
            obj.addMirror( 'PRM2', 'invRoC', -1/4.545, ...
                'T', 243e-6, 'Lhr', 8.6e-6*rmloss, 'inc', thetaPR2, 'Aperture', RMSize );
            obj.addMirror( 'PRM3', 'invRoC', 1/36.027, 'Aperture', Apr3, ...
                'T', 5.3e-6, 'Lhr', 17e-6*rmloss, 'incAngle', thetaPR3 );

            obj.addMirror( 'BS', 'Aperture', Abs, 'Thickness', dBS, ...
                'T', 0.5, 'Lhr', 8.6e-6, 'Lar', 31.7e-6, 'incAngle', pi/4 );
            obj.addMirror( 'ITMX', 'invRoC', 1/1934, 'Aperture', Atm, 'Thick', 0.2 + dCP,↙
...
                'T', 1.48e-2, 'Lhr', 10.4e-6*armLoss+armLossPPM/2, 'Lar', (164+42)*1e-↙
6*ITMloss, 'removeTilt', true );
            obj.addMirror( 'ITMY', 'invRoC', 1/1934, 'Aperture', Atm, 'Thick', 0.2 + dCP,↙
...
                'T', 1.48e-2, 'Lhr', 14.3e-6*armLoss+armLossPPM/2, 'Lar', (250+80)*1e-↙
6*ITMloss, 'removeTilt', true, ...
                'invFocal', invFocalY);

            obj.addDetector( 'dark' );

            % build BS maps
```

```
            switch BSNN
                case 2
                    obj.setHRfiles( 'BS', './Data/BS02_S1_CIT.dat', '', 0.16, 0, 1 );
                    obj.setARfiles( 'BS', './Data/BS02_R2Fig.dat', '', 0.16, 0, -1 );
                    obj.setTRfiles( 'BS', './Data/BS02_SPTWE.dat', '', 0.16, 0, -1 );
                case 5  % for L1
                    obj.setHRfiles( 'BS', './Data/BS05_S1_CIT.dat', '', 0.16, 0, 1 );
                    obj.setARfiles( 'BS', './Data/xBS_05_R2_Figure.dat', '', 0.16, 0, -1↙
);
                    obj.setTRfiles( 'BS', './Data/xBS_05_SPTWE_Figure.dat', '', 0.16, 0,↙
-1 );
                case 6  % for H1
                    obj.setHRfiles( 'BS', './Data/BS06_S1_CIT.dat', '', 0.16, 0, 1 );
                    obj.setARfiles( 'BS', './Data/xBS_06_R2_Figure.dat', '', 0.16, 0, -1↙
);
                    obj.setTRfiles( 'BS', './Data/xBS_06_SPTWE_Figure.dat', '', 0.16, 0,↙
-1 );
                case 100
                    obj.setRPYfiles( 'BS', './Data/BS02_RPY.dat', '', 0.22, 0 );
                    obj.setRSXfiles( 'BS', './Data/BS02_RSX.dat', '', 0.22, 0 );
                    obj.setTPXfiles( 'BS', './Data/BS02_TPX.dat', 'map(:,1:end-5)=map(:,↙
6:end);func=map;', 0.22, 0 );
                    obj.setTSYfiles( 'BS', './Data/BS02_TSY.dat', 'map(:,1:end-5)=map(:,↙
6:end);func=map;', 0.22, 0 );
                case 101
                    global BSmodel;
                    BSmodel = mphload('BS.mph');
                    obj.setRPYfiles( 'BS', '', '-sqrt(2)*calcBSatZ( ''w'', x(1,end:-1:1),↙
y(:,1), 0, 0 );', 0.22, 0 );
                    obj.setRSXfiles( 'BS', '', '-(calcBST( x(1,:), y(:,1), 0.06, 10, 1 )↙
+calcBST( x(1,:), y(:,1), 0.06, 10, -1 ));', 0.22, 0 );
                    obj.setTPXfiles( 'BS', '', '-calcBST( x(1,:), y(:,1), 0.06, 10, 1↙
);', 0.22, 0 );
                    obj.setTSYfiles( 'BS', '', '-calcBST( x(1,end:-1:1), y(:,1), 0.06,↙
10, -1 );', 0.22, 0 );
            end

            % implement BS baffle
            if armScale == 1
                obj.setModMap( 'BS', @obj.BSbaffle );
            end

            if withArm
                 % ETMX
                 obj.addMirror( 'ETMX', 'invRoC', 1/2245, 'Aperture', Atm, 'Thick', 0.2,↙
...
                     'T', 3.7e-6, 'Lhr', armLoss*10.9e-6+armLossPPM/2, 'resolution', 2e-3↙
);
                 % ETMY
                 obj.addMirror( 'ETMY', 'invRoC', 1/2245, 'Aperture', Atm, 'Thick', 0.2,↙
...
                     'T', 3.6e-6, 'Lhr', armLoss*9.3e-6+armLossPPM/2, 'resolution', 2e-3↙
);
            end

%% test mass maps
           if isempty( TMMap ) || TMMap ~= 0
               % ITMX
               obj.setHRfiles( 'ITMX', ...
                   './Data/ITM04_CITZygo.dat', ...     % phase map data file
                   'map + (x.^2+y.^2)/2*invRoC', ...   % formula to modify the map :↙
change the RoC for simple TCS
                   0.16, 1937.9, 0, ...                % BS RoC, 1937.9m, was measured in↙
the central 16cm
                   'invRoC', 0);                       % default value of the parameter↙
in the formula
               obj.setTRfiles( 'ITMX', './Data/itm04_sptwe.dat', '', 0.16, 302.41e3);
               obj.setTRfiles( 'ITMX', './Data/CP06_sptwe.dat', '', 0.16, 1e10);

               % ITMY
               obj.setHRfiles( 'ITMY', ...
```

```
                    './Data/ITM08_CITZygo.dat', ...     % phase map data file
                    '', ...
                    0.16, 1940.7, 0 )                % ITMY RoC 1940.7m, was measured in↙
the central 16cm
                obj.setTRfiles( 'ITMY', './Data/itm08_sptwe.dat', ...
                    'map - (x.^2+y.^2)/2*invFnum', 0.16, -82.00e3, 0, ... % CP thermal↙
lens
                    'invFnum', 0 );
                obj.setTRfiles( 'ITMY', './Data/CP08_sptwe.dat', '', 0.16, 1e10);

                if withArm
                     obj.setHRfiles( 'ETMX', './Data/ETM07_CIT.dat', 'map', 0.16, 2239.7);
                     obj.setHRfiles( 'ETMY', './Data/ETM09_CIT.dat', 'map', 0.16, 2242.4);
                end
            end

%% propagators
            % arms
            if withArm
                 obj.addProp( '', 'ITMX-HR', 'ETMX-HR', 3994.5);
                 obj.addProp( '', 'ITMY-HR', 'ETMY-HR', 3994.5);
            end

            obj.addProp( '', 'src', 'PRM-AR' );
            obj.addProp( '', 'PRM-HR', 'PRM2-HRx', 16.6107 );
            obj.addProp( '', 'PRM2-HRy', 'PRM3-HRx', 16.1647 );
            obj.addProp( '', 'PRM3-HRy', 'BS-HRx', 19.5384 );

            obj.addProp( '', 'BS-ARx', 'ITMX-AR', dBSCPX + dCPITM);
            obj.addProp( '', 'BS-HRy', 'ITMY-AR', dBSCPY + dCPITM);

            % SRC props : uncomment this to add SRC
%             obj.props.addProp( '', 'SRM',  1, 'SRM2', 1, 15.7566 );
%             obj.props.addProp( '', 'SRM2',  2, 'SRM3', 1, 15.4435 );
%             obj.props.addProp( '', 'SRM3',  2, 'BS', 4, 19.368 );
%             obj.props.addProp( '', 'SRM',  2, 'dark', 1 );
% make sure the following one to be commented out when SRC is added
            obj.props.addProp( '', 'BS-ARy', 'dark');
        end

        function setupBaseMode( obj, varargin )

            useArmMode = 1;
            eval( UtilTK.setVars( who, true, varargin{:} ) );

            if useArmMode == 0 || obj.optics.findNamedOptic('ETMX') <= 0
                cav = obj.cavities.closedCavs{1};
                ind = find( cav(2:end) == cav(1), 1 );
                cav = cav(1:ind+1);
                [qInv, rootID] = obj.cavities.findCavityMode( cav );
%               rootID = 1;
%               qInv = -0.132522 - i*0.0332622;
            else

                % adjust the PRM2-PRM3 length so that the PRC mode matches with the arm↙
mode
                ITMYHRoutID = obj.optics.findPortID( 'ITMY-HR-o' );
                ITMYHRinID = obj.optics.findPortID( 'ITMY-HR-i' );
                ITMYARoutID = obj.optics.findPortID( 'ITMY-AR-o' );
                ETMYHRoutID = obj.optics.findPortID( 'ETMY-HR-o' );
                BSHRxoutID = obj.optics.findPortID( 'BS-HRx-o' );
                PRM3xoutID = obj.optics.findPortID( 'PRM3-HRx-o' );
                PRM2xoutID = obj.optics.findPortID( 'PRM2-HRx-o'  );
                BSARyoutID = obj.optics.findPortID( 'BS-ARy-o' );
    %            SRM3xoutID = obj.optics.findPortID( 'SRM3', 1 );
    %            SRM2xoutID = obj.optics.findPortID( 'SRM2', 1 );

                % resonating mode in the arm.  rootArmY is ITMYHRoutID
                [qInv, rootID] = obj.cavities.findCavityMode( [ITMYHRoutID, ETMYHRoutID]↙
);
                sprintf('Mode used to define the modes of the entire IFO : portID = %d,↙
```

```
RoC = %g',...
                    rootID, 1/real(qInv) )

                % find the q value going out of the y arm
                % reverse the q value going into the ITMY HR, then let it goes through↙
ITMY toward AR side
                qInvArmYAR = MMTK.ABCD_qinv( obj.optics.findABCD( ITMYHRinID, ITMYARoutID↙
), MMTK.reverseDir( qInv ) );

                % find the adjustment of the PRM3-PRM2 length
                LadjP = obj.cavities.adjustModeMatch( qInvArmYAR, PRM3xoutID, 0, ...
                    [ITMYARoutID, BSHRxoutID, PRM3xoutID, PRM2xoutID ], true );

                % find the adjustment of the SRM3-SRM2 length
     %           LadjS = obj.cavities.adjustModeMatch( qInvArmYAR, SRM3xoutID, 0, ...
     %               [ITMYARoutID, BSARyoutID, SRM3xoutID, SRM2xoutID ], true );

    %            sprintf('PRM3-PRM2 is adjusted by %g and SRM3-SRM2 by %g', LadjP, LadjS↙
)
                sprintf('PRM3-PRM2 is adjusted by %g', LadjP )
            end

            obj.cavities.fillQvals( rootID, qInv );
        end
      end

    methods( Static )
        % implement a BS baffle
        function [reflOut, transOut] = BSbaffle( reflIn, transIn, x, y )

            baffleMask = x.^2/0.105^2+y.^2/0.13^2 < 1;

            reflOut{1} = baffleMask .* reflIn{1};
            reflOut{2} = reflIn{2} .* fliplr(baffleMask);
            reflOut{3} = reflIn{3} .* fliplr(baffleMask);
            reflOut{4} = baffleMask .* reflIn{4};
            transOut{1} = baffleMask .* transIn{1};
            transOut{2} = transIn{2} .* fliplr(baffleMask);
            transOut{3} = transIn{3} .* fliplr(baffleMask);
            transOut{4} = baffleMask .* transIn{4};
        end
    end
end
```