

Аннотация

Данная работа посвящена определению параметров СИС-смесителя на промежуточной частоте, а именно определению уровня отражения в зависимости от напряжения смещения на смесителе.

Предлагается экспериментальная методика, которая с помощью векторного анализатора цепей позволяет найти коэффициент отражения от смесителя, когда тот находится в "рабочем" состоянии. Используется особый метод калибровки измерений, где сам СИС-переход используется как калибратор, что позволяет учесть в расчетах его собственную емкость и индуктивность. Также предлагается проводить дополнительную калибровку по пику поглощения СИС-смесителя, так можно оценить точность расчетов, увидеть вклад комплексной части импеданса в общей коэффициент отражения и определить импеданс подводящей к СИС-переходу линии.

В результатах получено, что уровень отражений достаточно высок и варьируется от -4 до -20 дБ в зависимости от напряжения смещения смесителя. В "рабочем" состоянии отражение составляет около -5 дБ, что вынуждает использовать специальные вентили, для улучшения согласования в цепи.

Теоретические расчеты согласуются с экспериментальными данными.

Содержание

1 Обозначения и сокращения	3
2 Введение	4
3 Обзор литературы	5
3.1 СИС-смеситель	5
3.2 Гетеродинный прием	7
3.3 Измерение отражений от СИС-смесителя	9
4 Расчет и измерение отражения	11
4.1 Схема эксперимента	11
4.2 Калибровка цепи	14
4.3 Теоретический расчет отражения	17
4.4 Результаты измерений	20
4.4.1 Частотная характеристика	20
4.4.2 Зависимость от напряжения смещения	22
4.4.3 Пик поглощения	23
5 Заключение	25
6 Приложение	28

1 Обозначения и сокращения

В настоящей ВКР применяются следующие сокращения и обозначения:

- МФТИ — Московский физико-технический институт
- ФЭФМ — Физтех-школа электроники, фотоники и молекулярной физики
- ИРЭ — Институт радиотехники и электроники им. В.А. Котельникова РАН
- Nb — ниобий
- NbTiN — нитрид ниобий-титана
- СИС, SIS — туннельная структура сверхпроводник-изолятор-сверхпроводник (средоточенный)
- РДП, FFO — Распределенный джозефсоновский переход (англ. flux-flow oscillator)
- ВАЦ, VNA — векторный анализатор цепей (англ. Vector Network Analyser)
- LO — опорный генератор (англ. Local Oscillator)
- LNA — криогенный малошумящий усилитель (англ. Low Noise Amplifier)
- ПЧ, IF — промежуточная частота (англ. Intermediate Frequency)
- ВАХ — Вольт-Амперная характеристика

2 Введение

Радиоастрономия и физика атмосферы являются главными драйверами развития миллиметровых и суб-миллиметровых приемников. Данный диапазон черезвычайно информативен, однако, наименее изучен, и представляет на данный момент огромный интерес для космологии и астрофизики. В терагерцовом диапазоне отслеживают состав и стадии формирования звезд, протопланетарных систем, наблюдают детали реликтового излучения, а также изучают состав межзвездного вещества. Многие молекулы, в том числе и H_2O имеют вращательные и колебательные спектры поглощения в субмиллиметровом диапазоне.

Приемники на основе смесителей на тунNELьном переходе сверхпроводник – изолятор – сверхпроводник (СИС) [1] имеют рекордные шумовые характеристики в диапазоне частот 100-1200 ГГц, близкие к квантовому пределу [ссылки](#).

Среди СИС-приемников для наземных телескопов наибольшее распространение получили смесители с разделением боковых полос, которые имеют в составе два одиночных СИС-смесителя [2] [3]. Это позволяет уменьшить вклад атмосферных шумов в два раза.

Для эффективной работы приемника, а именно для достижения предельной чувствительности и для высокого качества разделения боковых полос, принципиально важно иметь СИС-смесители с низким уровнем отражения. Низкий уровень отражений нужен как по входу смесителя, т.е. на частоте принимаемого сигнала, на высокой частоте (ВЧ) [4] [5], так и по выходу, т.е. на промежуточной частоте (ПЧ, IF). В тракте ПЧ такая необходимость обусловлена возникновением стоячих волн между самим СИС-смесителем и криогенным малошумящим усилителем, так как последний имеет достаточно высокий уровень отражения до ~ -6 дБ. Стоячие волны увеличивают шумовую температуру и ухудшают качество разделения полос. В случае же матричных приемников важна компактность и нет возможности использовать вентили для устранения стоячих волн, поэтому минимизация отражений по ПЧ важна и для этого технического направления.

Данная работа посвящена определению уровня отражений от СИС-смесителя в рабочем режиме с целью минимизации этого уровня в будущем. Для решения поставленной задачи проведен теоретический расчет уровня отражения от СИС-смесителя по выходному тракту ПЧ, а также собрана экспериментальная схема, и проведено непосредственное измерение отраженного сигнала.

Полученные результаты могут быть применимы для лучшей характеризации СИС-смесителей и оптимизации их работы.

3 Обзор литературы

3.1 СИС-смеситель

СИС-смеситель представляет собой непосредственно сам туннельный СИС-переход с выходом на применую antennную, криогенный малошумящий усилитель (LNA) и опорный генератор гетеродина (LO) (рис. 1)

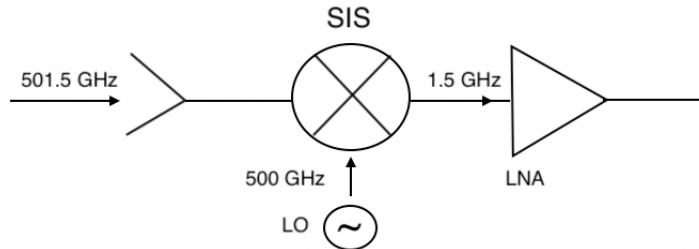


Рис. 1: Схема СИС-смесителя

На рис. 2 представлена интегральная согласующая структура, которая располагается полностью на одном чипе, включая СИС-переход, трансформаторы импеданса и опорный генератор на основе РДП.

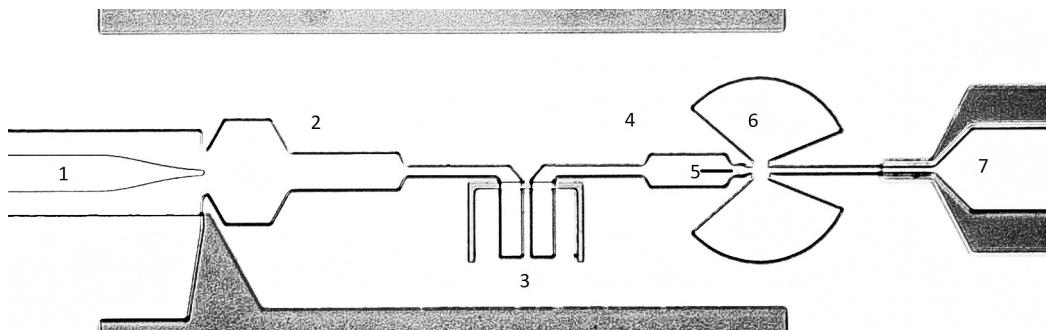


Рис. 2: Фото интегральной согласующей структуры. 1 — генератор на РДП, 2 — трёхступенчатый трансформатор импеданса, 3 — элемент разрыва по низким частотам, 4 — двухступенчатый трансформатор импеданса, 5 — СИС-переход, 6 — радиальный замыкатель, 7 — выходная копланарная линия.

На рис. 3 представлена конструкция «боевого» СИС-смесителя. Для данной конструкции внешний сигнал опорного генератора подводится через рупор.

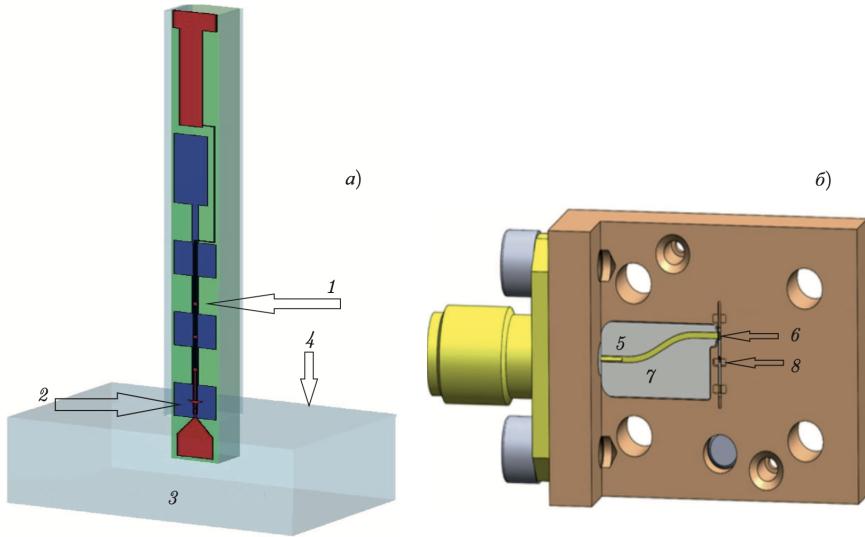


Рис. 3: [12] Конструкция смесительного элемента и центральной части блока смесителя:
Панель а : трёх- мерная модель дизайна смесительного элемента с размерами 3 250 × 150 × 125 мкм, размещённого в волноводе, 1 — фильтр высоких частот (ВЧ), 2 — СИС-переход, 3 — вход волновода, показано также расположение замыкающей стенки 4. Панель б : трёхмерная модель части смесительного блока с платой согласования по промежуточной частоте 5. В блоке установлен смесительный элемент 6, за которым в волноводе находится замыкающая плоскость; 7 — линия 50 Ом, 8 — сечение волновода, также показана область подключения волновода.

Сам туннельный СИС-переход является «сэндвич» структурой сверхпроводник-изолятор-сверхпроводник (рис. 4). При приложении отрицательного напряжения на левый берег, энергия электронов увеличивается и будет наблюдаться туннельный ток (синяя автономная ВАХ рис. 5). Если при этом подать сигнал опорного генератора (LO), то энергия фотона $\hbar\omega$ позволит электронам туннелировать компенсируя недостаток энергии и образуются так называемые ступени $\hbar\omega/e$ (красная "рабочая" ВАХ рис. 5)

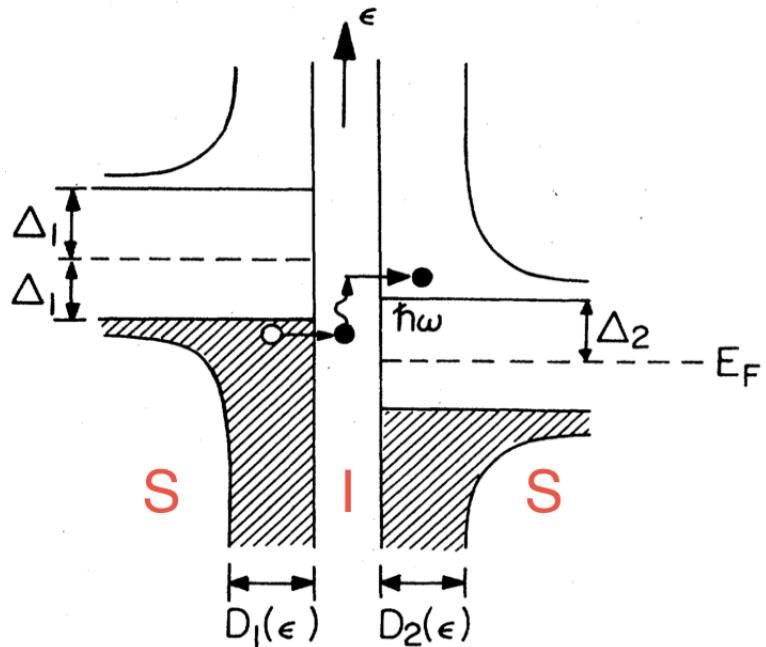


Рис. 4: Энергетическая диаграмма СИС-перехода [1]

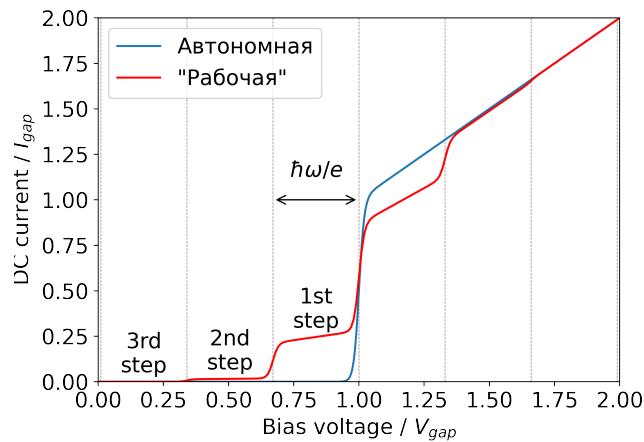


Рис. 5: Теоретическая ВАХ СИС-перехода; Автономная (синяя кривая) и "Рабочая" (красная кривая) при поданном напряжении опорного генератора LO. Расчет выполнен с помощью python-библиотеки QMix

[9].

3.2 Гетеродинный прием

Принцип гетеродинирования [8] заключается в складывании мощного сигнала опорного генератора и слабого детектируемого сигнала в смесителе с нелинейной ВАХ. Ввиду нелинейности возникают суммы и разности соответствующих частот:

$$|n \cdot f_S - m \cdot f_{LO}| \quad (1)$$

где f_S — частота слабого детектируемого сигнала, f_{LO} — частота мощного сигнала опорного генератора.

Для примера, возьмем квадратичную ВАХ: $I(V) \sim V^2$. Сигналы принимаемый и опорного генератора соответственно:

$$V_S = V_S \sin(2\pi f_S t); \quad V_{LO} = V_{LO} \sin(2\pi f_{LO} t) \quad (2)$$

Тогда ток смесителя при приложенных сигналах и напряжении смещения V_0 :

$$\begin{aligned} I &\sim (V_0 + V_{LO} \sin(2\pi f_{LO} t) + V_S \sin(2\pi f_S t))^2 \\ &\sim V_0^2 + 2V_0 V_S \sin(2\pi f_S t) + 2V_0 V_{LO} \sin(2\pi f_{LO} t) \\ &+ \frac{1}{2} V_S^2 \{1 - \cos(2\pi(2f_S)t)\} + \frac{1}{2} V_{LO}^2 \{1 - \cos[2\pi(2f_{LO})t]\} \\ &+ V_S V_{LO} \{\cos(2\pi(f_S - f_{LO})t) - \cos(2\pi(f_S + f_{LO})t)\} \end{aligned} \quad (3)$$

Итак, мы можем выделить компоненту сигнала $V_S V_{LO} \cos(2\pi(f_S - f_{LO})t)$. Эта разностная частота называется промежуточной частотой (ПЧ) f_0 :

$$f_S = f_{LO} \pm f_0 \quad (4)$$

Таким образом гетеродин позволяет "сбросить" частоту принимаемого сигнала f_S до промежуточной частоты f_0 , сохраняя пропорции величины сигнала. На рис. 6 проиллюстрирован принцип "сброса" частоты гетеродином.

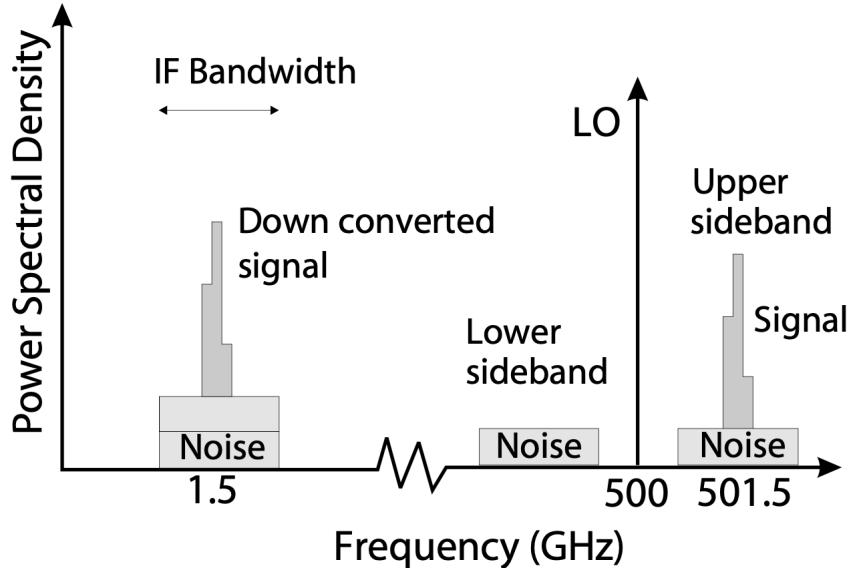


Рис. 6: Иллюстрация принципа гетеродинирования [8]. Детектируемый сигнал частоты $f_s = 501.5 \text{ GHz}$, сигнал LO $f_{LO} = 500 \text{ GHz}$, сигнал ПЧ $f_0 = 1.5 \text{ GHz}$.

3.3 Измерение отражений от СИС-смесителя

На данный момент достаточно много научных статей, посвященных изучению криогенных приемников на основе туннельного перехода СИС и их различных характеристик[ссылки]. Однако, вопрос непосредственного измерения уровня отражения от СИС-смесителя на промежуточной частоте затрагивает лишь несколько работ:

- В работе Serres et al. [7], посвященной определению выходного импеданса СИС-смесителя по промежуточной частоте, приведено сравнение теоретических расчетов и экспериментальных результатов. Цель данной статьи достаточно близка к нашей, поскольку определение импеданса необходимо для получения уровня отражения. Предложена модифицированная однопортовая экспериментальная схема с использованием циркулятора (рис. 7).

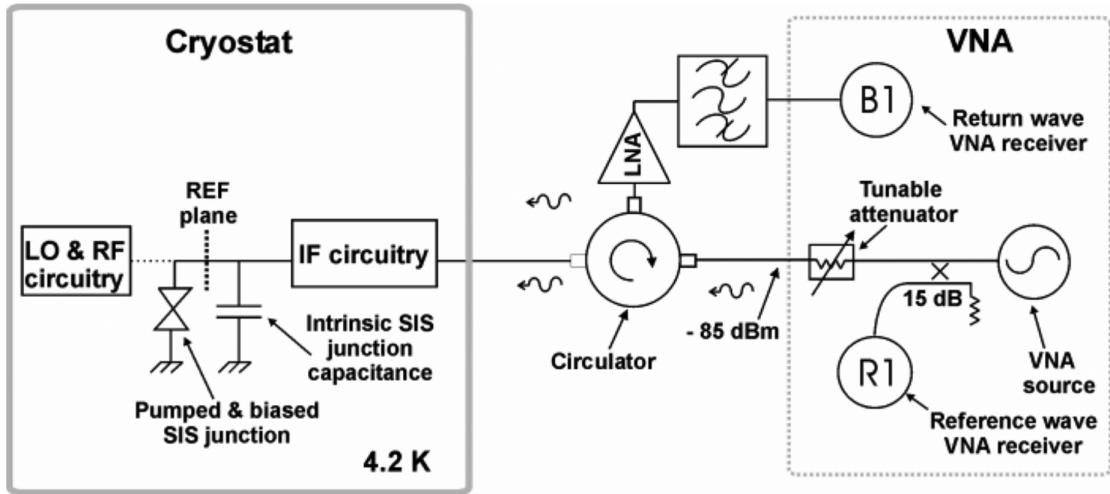


Рис. 7: Система измерения импеданса ПЧ СИС-смесителя с модифицированной конфигурацией однопортового ВАЦ. Криостат и ВАЦ показаны соответственно слева и справа. Посередине показаны направления падающей и обратной волн от ВАЦ, а также соединения циркулятора с криостатом и ВАЦ. Криостат содержит СИС-смеситель с его схемой ПЧ, а также внешние коаксиальные кабели инжектора Bias-Tee и ПЧ, которые соединяют выход ПЧ смесителя с пластиной криостата 300 К. [7]

Особенностью калибровки является то, что сам СИС-смеситель используется как калибратор, двигаясь по его вольт-амперной характеристике (ВАХ) можно получить три калибровочных стандарта (SOL).

Причем, такая калибровка позволяет учесть собственную емкость и индуктивность перехода. Численный расчет импеданса смесителя в данной работе опирается на теорию Tucker et al. [1].

Особенности: все в холоде, размеры маленькие

- В кандидатской диссертации (PhD thesis) А.М. Барышева (Barichev A.M) [8] приводится пример распространенной схемы для измерения отражений от СИС-смесителя по тракту ПЧ (рис. 8), на который мы будем ориентироваться при составлении схемы нашего эксперимента.

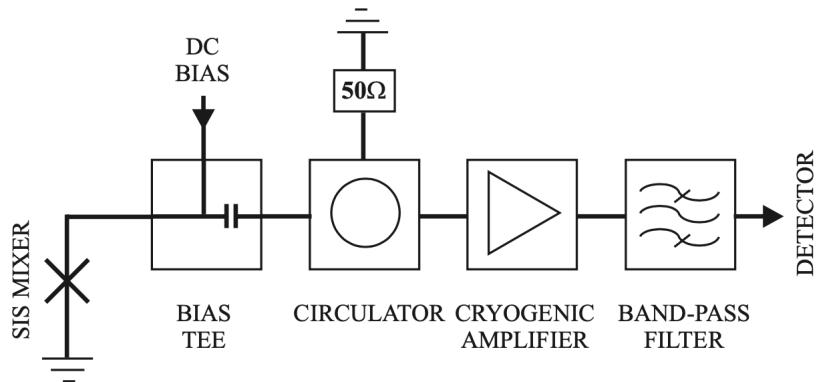


Рис. 8: Типичная схема соединения СИС-смесителя по выходу ПЧ [8]

4 Расчет и измерение отражения

4.1 Схема эксперимента

Основная задача данной работы заключается в измерении уровня отражения от СИС-смесителя по выходному каналу ПЧ, когда сам смеситель находится в рабочем состоянии, а именно на него подается напряжение смещения и приложен сигнал высокочастотного опорного генератора.

Измерение отражений от СИС-смесителя имеет ряд принципиальных особенностей:

- СИС-смеситель насыщается сигналом малой мощности, что вызывает искажение ВАХ и искажает режим его работы, поэтому уровень тестового сигнала VNA, приходящего на СИС по тракту ПЧ, не должна превышать -50дБм.
- отраженный от СИС-смесителя сигнал очень мал и должен быть усилен с помощью МШУ перед подачей на ВАЦ
- СИС-смеситель расположен в криостате при температуре 4К, а ВАЦ при комнатной температуре

Измерения проведены в диапазоне 4–8 ГГц; этот диапазон определен полосой используемого криогенного ПЧ усилителя. Схема эксперимента представлена на Рис.9; СИС-смеситель помещен в криостат замкнутого цикла при температуре около 4 К. Высокочастотный опорный генератор (LO) интегрирован с СИС-смесителем на одном чипе и представляет собой распределенный джозефсоновский переход с вязким течением магнитных вихрей (РДП, FFO). Векторный анализатор цепей (ВАЦ, VNA),

размещенный вне криостата, генерирует тестовый сигнал диапазона 4–8 ГГц на порте (P1, P2), который, проходя через аттенюатор -10 дБ, поступает в направленный ответвитель (Directional coupler), который направляет его на СИС-смеситель с коэффициентом связи около -10 дБ. Далее сигнал проходит специальный инжектор (Bias Tee), позволяющий беспрепятственно проходить ПЧ сигналу, задавая при этом напряжение на СИС-смесителе по постоянному току через большую индуктивность. Отразившись от СИС-смесителя, основная часть сигнала проходит напрямую через направленный ответвитель и поступает на вход криогенного малошумящего усилителя (LNA), который усиливает этот сигнал и направляет его на приемный порт (P2, P2) ВАЦ. Фактически, измеряемым параметром является отношение сигналов ВАЦ на портах P1 и P2, а точнее его спектр.

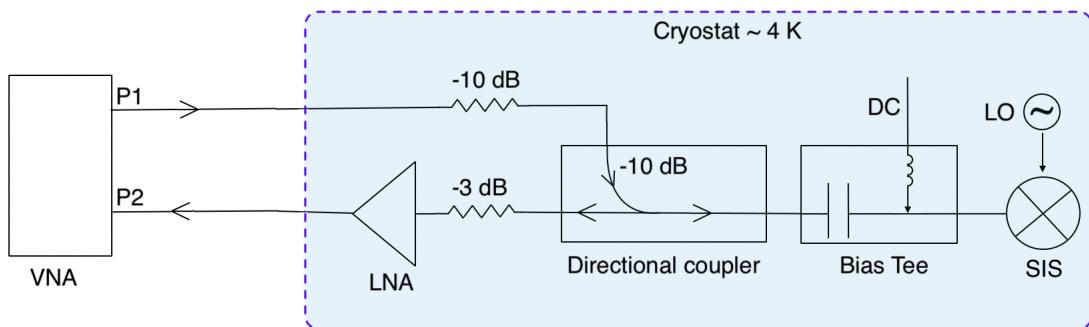


Рис. 9: Схема эксперимента по измерению отражения от СИС-смесителя по выходу ПЧ

Ниже представлены фото собранных измерительных установок (рис. 10 и 11 соответственно) с использованием интегральной согласующей структуры (схема на рис. 2) и волноводного СИС-смесителя диапазона 211-275 ГГц (схема на рис. 3).

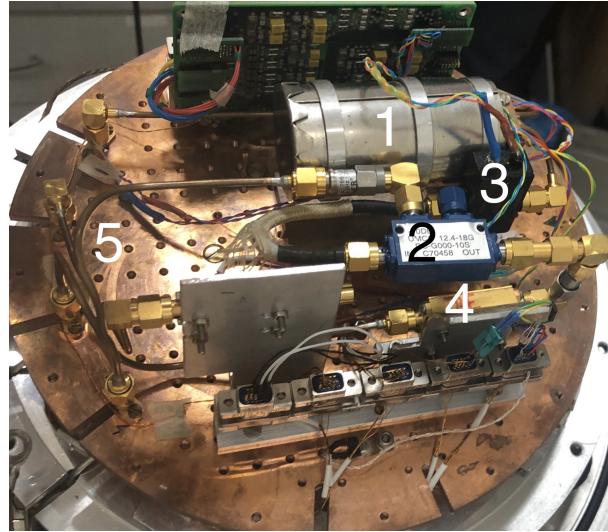


Рис. 10: Фото установки внутри криостата при измерении отражения от интегрального СИС-смесителя. 1 — Металлический экран (внутри чип с интегральной согласующей структурой), 2 — Направленный ответвитель, 3 — Инжектор, 4 — криогенный усилитель, 5 — плита криостата $\sim 4\text{ K}$.

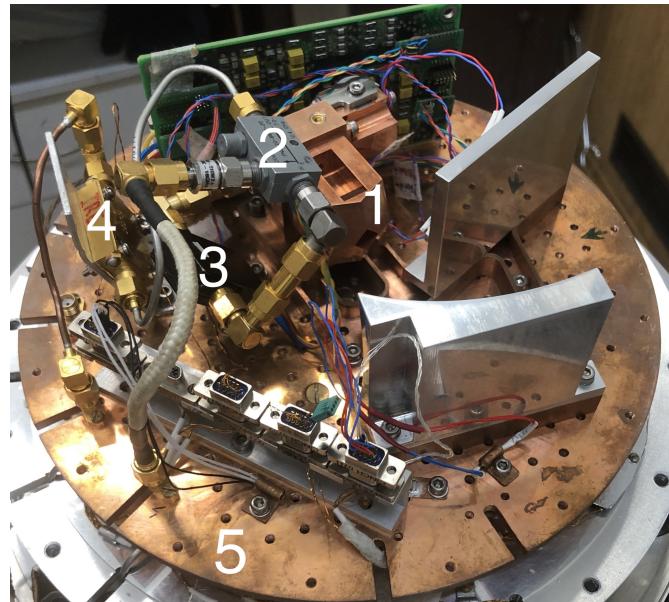


Рис. 11: Фото установки внутри криостата при измерении отражения от «боевого» волноводного СИС-смесителя диапазона 211-275 ГГц. 1 — Рупор СИС-приемника, 2 — Направленный ответвитель, 3 — Инжектор, 4 — криогенный усилитель, 5 — плита криостата $\sim 4\text{ K}$.

4.2 Калибровка цепи

Обязательным этапом для измерений отражения является калибровка цепи. Мы используем калибровку, эквивалентную стандартной однопортовой калибровке (Рис. 12), в основе которой лежит определение 3х параметров цепи: D - прямые утечки в цепи, R - внутренние отражения, M - рассогласование.

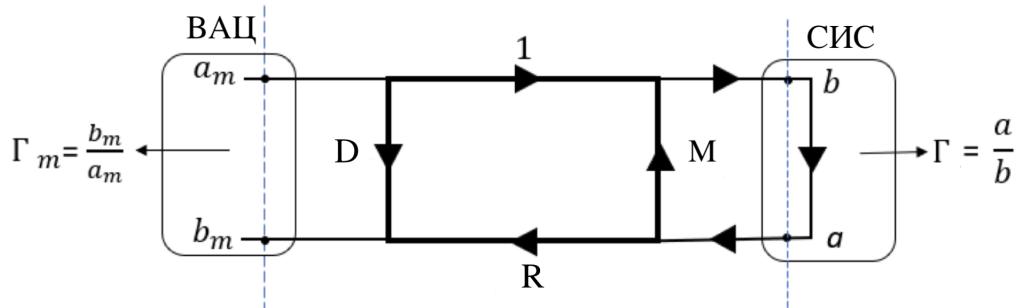


Рис. 12: Схема однопортовой калибровки.

Путем несложных преобразований (полные выкладки можно найти в [11]), используя последовательное (рис. 13) и параллельное (рис. 14) правила расчета коэффициента пропускания элементов цепи можно явно выразить фактический коэффициент отражения Γ через измеряемую величину Γ_m и 3 калибровочных параметра D, R, M:

$$\Gamma = \frac{\Gamma_m - D}{R + M(\Gamma_m - D)} \quad (5)$$

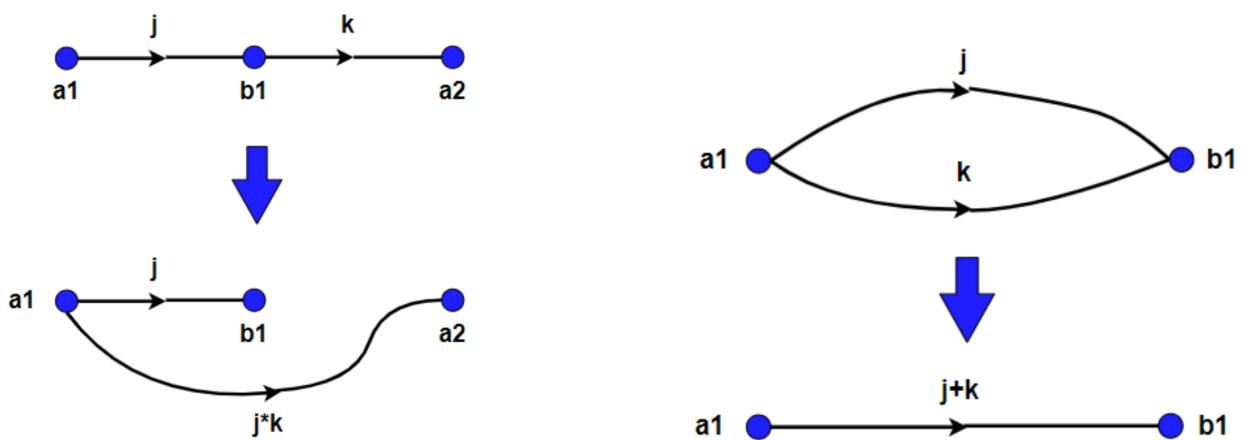


Рис. 13: Последовательное правило
[11]

Рис. 14: Параллельное правило [11]

Коэффициенты D, R, M могут быть определены по трем калибровочным измерениям,

путем составления и решения системы (6) из трех уравнений (5), где коэффициент отражения вычисляется теоретически по формуле (10), а величина непосредственно измеряется векторным анализатором цепей.

$$\begin{cases} \frac{\Gamma_{m_{open}} - D}{R + M(\Gamma_{m_{open}} - D)} = \Gamma_{open} & \text{Открытая цепь} \\ \frac{\Gamma_{m_{short}} - D}{R + M(\Gamma_{m_{short}} - D)} = \Gamma_{short} & \text{Короткое замыкание} \\ \frac{\Gamma_{m_{load}} - D}{R + M(\Gamma_{m_{load}} - D)} = \Gamma_{load} & \text{Нагруженная линия} \end{cases} \quad (6)$$

Представим систему (6) в матричном виде [11]:

$$\begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix} = (C^\dagger \cdot C)^{-1} \cdot C^\dagger \cdot V \quad (7)$$

где

$$C = \begin{bmatrix} \Gamma_{load} & 1 & \Gamma_{load}\Gamma_{m_{load}} \\ \Gamma_{open} & 1 & \Gamma_{open}\Gamma_{m_{open}} \\ \Gamma_{short} & 1 & \Gamma_{short}\Gamma_{m_{short}} \end{bmatrix} \quad \text{и} \quad V = \begin{bmatrix} \Gamma_{load} \\ \Gamma_{open} \\ \Gamma_{short} \end{bmatrix} \quad (8)$$

Откуда явно выражим неизвестные калибровочные параметры:

$$\begin{cases} D = E_2 \\ M = E_3 \\ R = E_1 + E_2 \cdot E_3 \end{cases} \quad (9)$$

В калибровочных измерениях при 4 К сам СИС-переход используется как калибратор [7]. Это позволяет избежать погрешностей, возникающих при калибровке при комнатной температуре и связанных с изменениями электрической длины и импеданса элементов цепи при охлаждении. СИС-смеситель находится в автономном состоянии, т.е. без приложения внешнего сигнала.

На Рис. 15 проиллюстрировано, какие напряжения смещения используются для калибровки: 1) при напряжении смещения в 2 мВ дифференциальное сопротивление становится порядка 1000 Ом, что близко к ситуации «открытой цепи», т.к. импеданс подводящей линии близок к величине Ом; 2) при напряжении смещения 3.8 мВ, т.е. посередине туннельного скачка тока, его дифференциальное сопротивление составляет около 3 Ом, что приближенно соответствует калибровке «короткое замыкание»; 3) при напряжении смещения 5–7 мВ дифференциальное сопротивление становится Ом, что близко к ситуации «нагруженной линии».

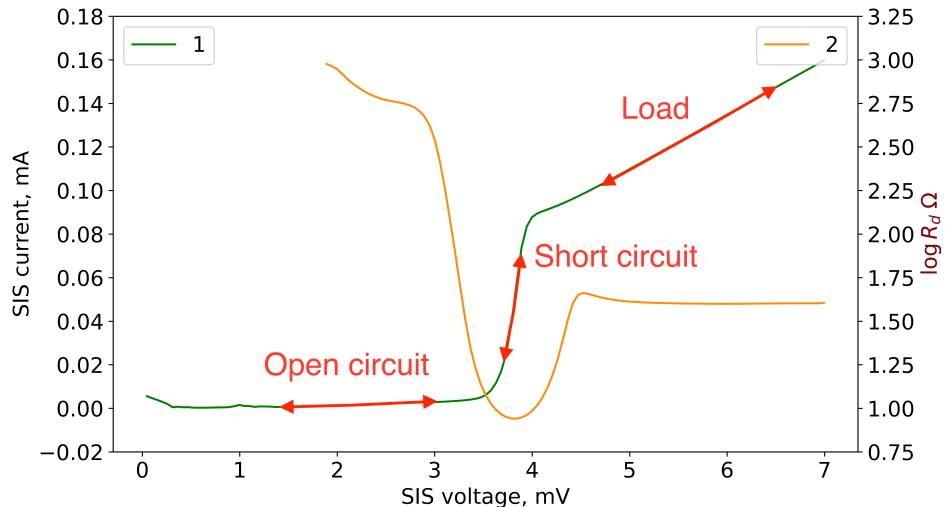


Рис. 15: Автономная Вольт-амперная характеристика СИС-смесителя №1 зеленая линия; соответствующее дифференциальное сопротивление показано оранжевой линией №2.

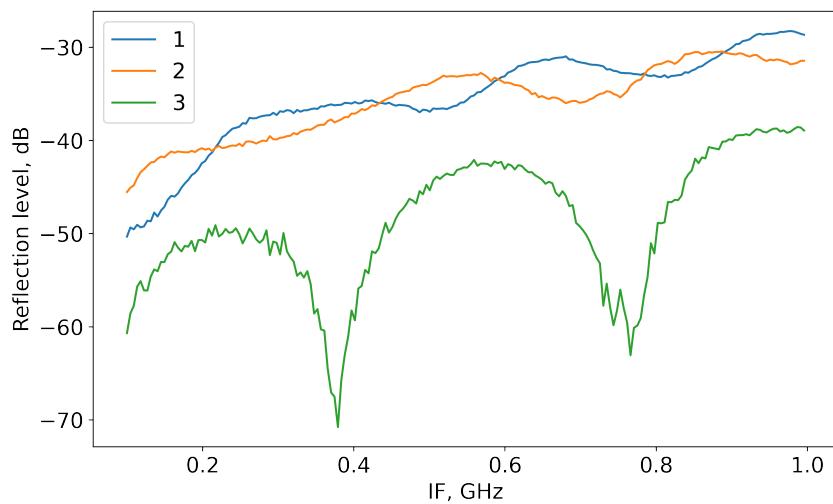


Рис. 16: Пример калибровочных измерений: №1 — "открытая цепь"; №2 — "короткое замыкание"; №3 — "нагруженная линия"

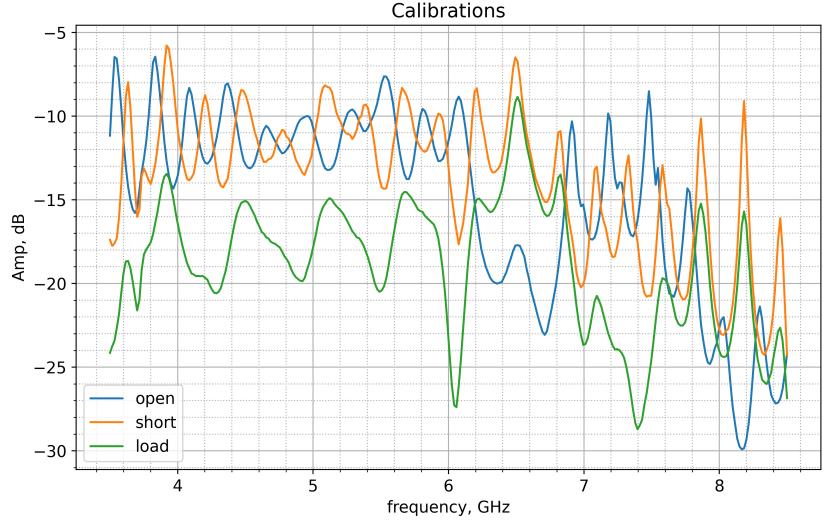


Рис. 17: Пример калибровочных измерений для волноводного СИС-смесителя диапазона 211-275 ГГц: open — "открытая цепь"; short — "короткое замыкание"; load — "нагруженная линия"

Обратим внимание, что кривые №1,2 и open, short на рис.16 и 17 соответственно находятся в противофазе, это объясняется тем, что при отражении электро-магнитной волны от среды с очень маленьким сопротивлением (короткое замыкание) происходит скачок фазы на π , в отличии от случая отражения от очень большого сопротивления (открытая цепь).

Этих трех калибровок достаточно, чтобы составить 3 уравнения (6) используя (5) и тем самым определить коэффициенты D, R и M. Это позволяет учесть все отражения и утечки в цепи и корректно измерить отражение от СИС-смесителя в рабочем режиме. Стоит отметить, что в приведенной калибровке внутренняя емкость СИС-смесителя выступает как часть внешней цепи и ее влияние также нивелируется калибровкой.

4.3 Теоретический расчет отражения

Величина отражений от СИС-смесителя по тракту ПЧ, характеризуемая параметром $S11_{IF}$ [6], может быть определена, если известен выходной импеданс ПЧ СИС-смесителя Z_{IF} и импеданс подводящей линии тракта ПЧ Z_L

$$S11_{IF} = \frac{Z_{IF} - Z_L}{Z_{IF} + Z_L} \quad (10)$$

Для расчета мы используем 3x частотное приближение к теории квантового смешения Tucker [1]. Рассмотрены сигналы на частотах $f_m = m \cdot f_{LO} + f_0$, где $m = 0, \pm 1$; f_{\pm}

- верхняя и нижняя полосы; f_{LO} - частота опорного генератора; f_0 - промежуточная частота. Полагаем, что более высокие гармоники шунтируются емкостью СИС-смесителя, составляющей в нашем случае порядка 100 фФ. Принимая во внимание взаимодействие портов смесителя, компоненты напряжений и токов слабых сигналов линейно связаны матрицей проводимости $i_m = \sum_{m'} Y'_{mm'} v_{m'}$:

$$Y'_{mm'}(V_{gap}, R_N, f_{LO}, \alpha, V_0) = \begin{bmatrix} Y_{11} + Y_S & Y_{10} & Y_{1-1} \\ Y_{01} & Y_{00} + Y_L & Y_{0-1} \\ Y_{-11} & Y_{-10} & Y_{-1-1} + Y_I \end{bmatrix} \quad (11)$$

где V_{gap} - напряжение смещения туннельного скачка тока, R_N - нормальное дифференциальное сопротивление ВАХ, $\alpha = eV_{LO}/hf_{LO}$ - безразмерный параметр накачки, пропорциональный напряжению опорного генератора V_{LO} , V_0 - напряжение смещения СИС-смесителя.

У каждого порта есть своя канальная проводимость: $Y_1 = Y_S$ - нагрузка порта с принимаемым сигналом, $Y_0 = Y_L$ - нагрузка порта ПЧ, $Y_{-1} = Y_I$ - нагрузка порта нижней полосы.

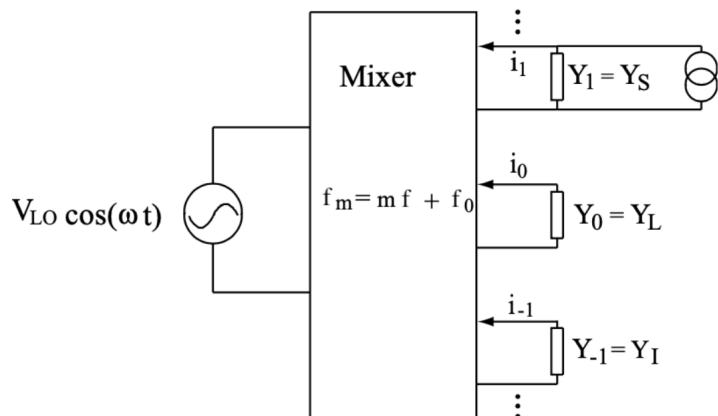


Рис. 18: Эквивалентная схема гетеродинного смесителя, с приложенным сигналом опорного генератора частоты, принимающим сигналом частоты и сигналом ПЧ

Для идеального двухполосного смесителя импедансы верхней и нижней полос равны $Y_1 = Y_{-1}$. В нашем случае $Y_{\pm 1}$ - результат преобразования проводимости опорного генератора несколькими микрополосковыми линиями.

Воспользовавшись результатом [1], получим $Y_{mm'} = G_{mm'} + iB_{mm'}$, где

$$G_{mm'} = \frac{e}{2\hbar\omega_{m'}} \cdot \sum_{n,n'=-\infty}^{\infty} J_n(\alpha) J_{n'}(\alpha) \delta_{m-m', n'-n} \\ \{ [I_{dc}(V_0 + n\hbar\omega/e + \hbar\omega_{m'}/e) - I_{dc}(V_0 + n\hbar\omega/e)] + \\ [I_{dc}(V_0 + n\hbar\omega/e) - I_{dc}(V_0 + n\hbar\omega/e - \hbar\omega_{m'}/e)] \} \quad (12)$$

$$B_{mm'} = \frac{e}{2\hbar\omega_{m'}} \cdot \sum_{n,n'=-\infty}^{\infty} J_n(\alpha) J_{n'}(\alpha) \delta_{m-m', n'-n} \\ \{ [I_{kk}(V_0 + n\hbar\omega/e + \hbar\omega_{m'}/e) - I_{kk}(V_0 + n\hbar\omega/e)] - \\ [I_{kk}(V_0 + n\hbar\omega/e) - I_{kk}(V_0 + n\hbar\omega/e - \hbar\omega_{m'}/e)] \} \quad (13)$$

Где $I_{dc}(V_0)$ - зависимость туннельного тока СИС-смесителя от его напряжения; $I_{kk}(V_0)$ - зависимость соотношения Крамерса-Кронига тока I_{dc} от напряжения СИС-смесителя; $J_n(\alpha)$ - функция Бесселя порядка от параметра накачки α . Тогда, в нашем случае импеданс по ПЧ:

$$Z_{IF} = \|Y_{mm'} + Y_m \delta_{mm'}\|_{00}^{-1} \quad (14)$$

На рис.19 продемонстрирована измеренная ВАХ СИС-перехода, которая является зависимостью $I_{dc}(V_0)$ и определяет I_{kk} . Автономная кривая показана зеленой линией, при этом, оранжевой кривой показана ВАХ для случая, когда приложен внешний сигнал генератора частотой 600 ГГц.

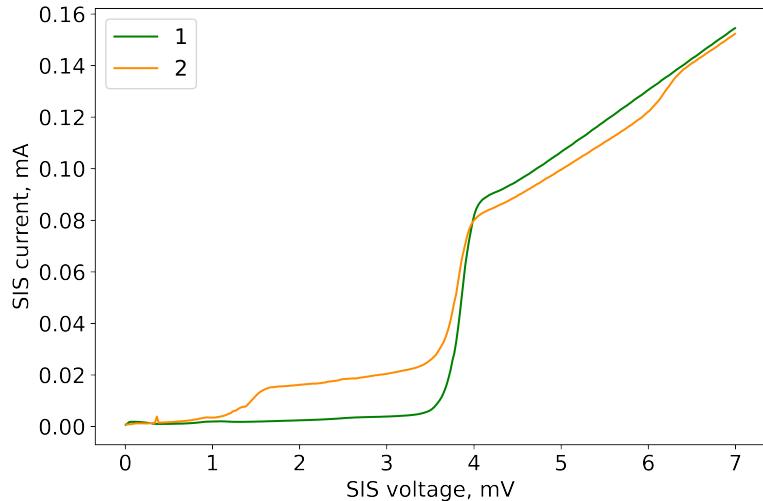


Рис. 19: Экспериментальная ВАХ автономная (№1 зеленая кривая) и с приложенным сигналом опорного генератора (№2 оранжевая кривая).

На рис. 20 приведен результат теоретического расчета импеданса с использованием ВАХ (рис.19) в диапазоне ПЧ 0.1-12 ГГц при различных напряжениях на СИС-переходе: 2.8, 3, 3.2 мВ, причем импеданс подводящей линии полагается ~ 50 Ом. Можно заключить, что действительная часть изменяется незначительно с промежуточной частотой, в отличие от мнимой части. Также видно, что импеданс сильно зависит от напряжения на СИС-переходе.

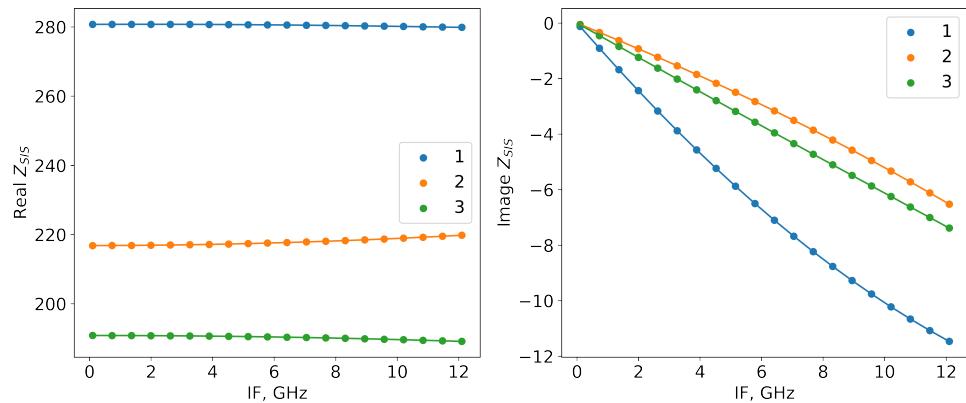


Рис. 20: Действительная (слева) и мнимая (справа) части расчетного импеданса СИС-смесителя от ПЧ при разных напряжениях на СИС-смесителе; №1 – 2.8 мВ, №2 – 3 мВ, №3 – 3.2 мВ.

4.4 Результаты измерений

4.4.1 Частотная характеристика

ВАХ интегрального СИС-смесителя, при приложении сигнала опорного генератора частотой ГГц в «рабочем» режиме, показана на рис.19 оранжевой кривой. Диапазон «рабочего» напряжения смещения лежит приблизительно в интервале 1.6–3.4 мВ. Эта область напряжений соответствует так называемой квазичастичной ступени, вызванной приложением к СИС-переходу сигнала гетеродина. На рис.21 приведены экспериментальные (непрерывные кривые) и теоретические (пунктирные кривые) частотные зависимости уровня отражения при различных напряжениях на СИС-смесителе. Из приведенных результатов видно, что в диапазоне 4–8 ГГц уровень отражения почти не меняется с частотой, что хорошо согласуется и с теоретическими предсказаниями.

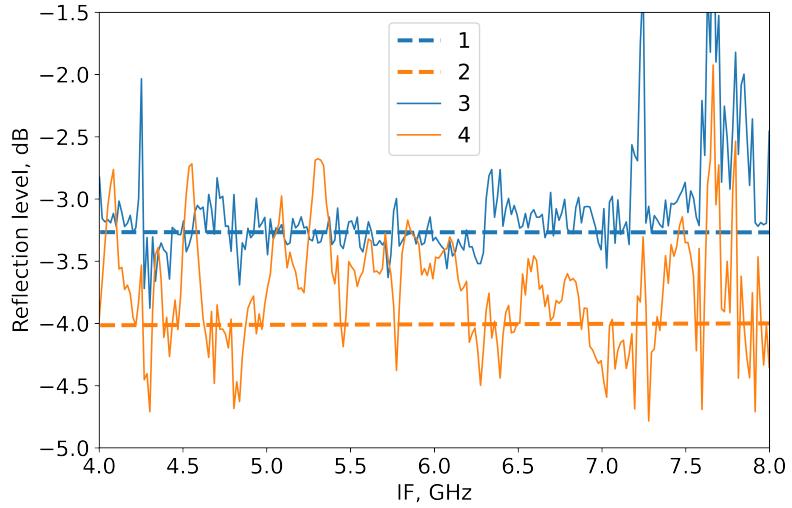


Рис. 21: Рассчитанный теоретически (№ 1,2 пунктирные линии) и определенный экспериментально (№ 3,4 сплошные кривые) уровень отражения от СИС-перехода в «рабочем режиме»; Напряжения СИС-смесителя № 1,3 – 2 мВ, № 2,4 – 2.6 мВ

Аналогично приведен пример измеренного отражения по тракту ПЧ от волноводного СИС-смесителя, при приложенном сигнале опорного генератора 230 ГГц на рис. 22. Можно заключить, что уровень отражения не имеет явной зависимости от ПЧ, однако в данном случае результаты искажены некорректной работой криогенного усилителя.

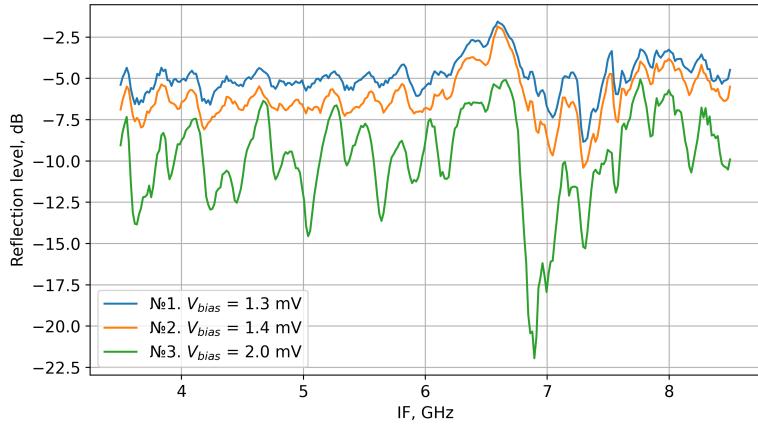


Рис. 22: Определенный экспериментально уровень отражения от волноводного СИС-смесителя в «рабочем режиме»; Напряжения СИС-смесителя № 1 – 1.3 мВ, № 2 – 1.4 мВ, № 3 – 2 мВ.

4.4.2 Зависимость от напряжения смещения

Изменение уровня отражений при варьировании напряжения смещения СИС-смесителя продемонстрировано на Рис.23. Здесь приведены результаты измерений и расчета отражения для середины диапазона ПЧ, а именно при частоте 6 ГГц. Синие точки - экспериментальные данные, красная сплошная кривая – теоретический расчет. Уровень отражения в среднем составляет около 4 дБ. При напряжении около 2 мВ имеет место искажение ВАХ вызванное краем джозефсоновской ступени, которая проявляется ввиду наличия неподавленного критического тока перехода. Это искажение вызывает аномалию в уровне отражения.

Также представлен результат измерения уровня отражения от «боевого» волноводного СИС-смесителя диапазона 211-275 ГГц в зависимости от напряжения смещения (рис.24)

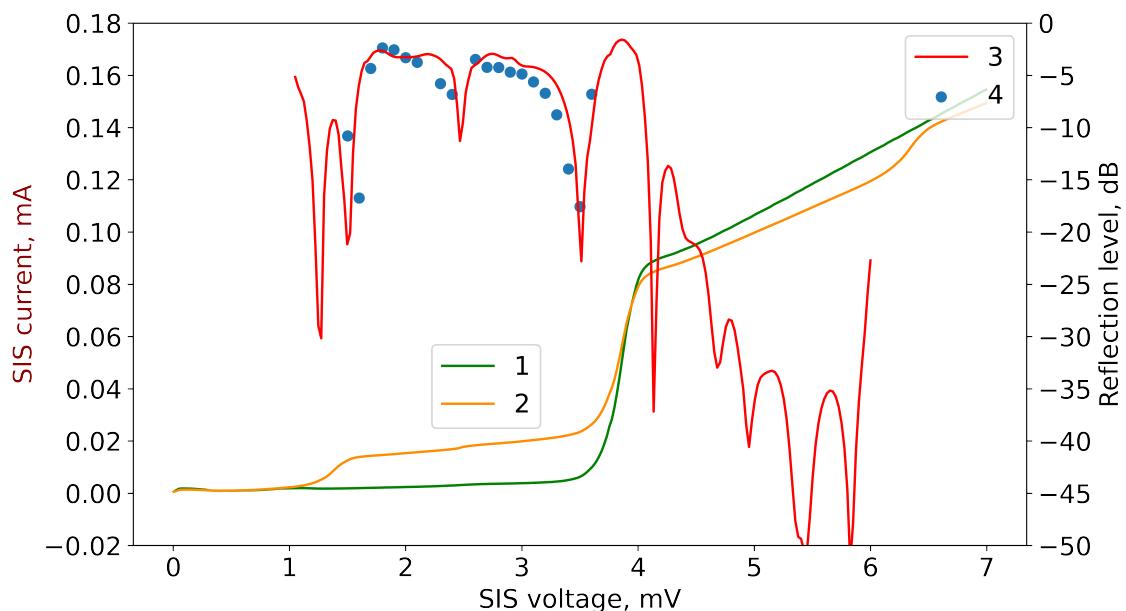


Рис. 23: ВАХ СИС-смесителя: автономная (№ 1 зеленая кривая), нагруженная сигналом опорного генератора (№ 2 оранжевая кривая). Теоретический расчет отражения (№ 3 красная кривая). Результаты измерений отражения (№ 4 синие точки). ПЧ равна 6 ГГц.

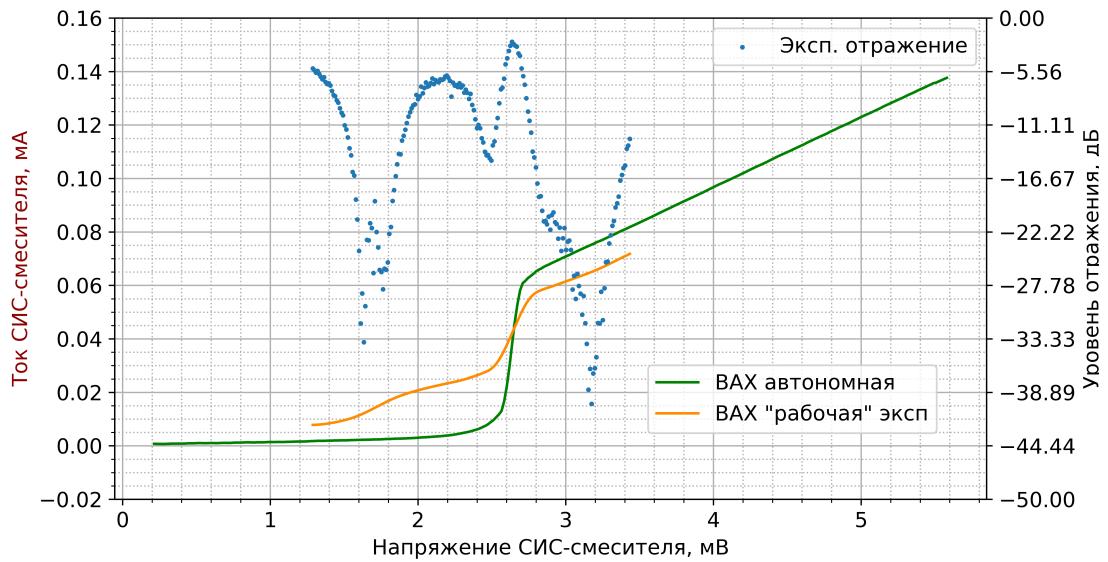


Рис. 24: ВАХ «боевого» СИС-смесителя: автономная (зеленая кривая), нагруженная сигналом опорного генератора (оранжевая кривая). Результаты измерений отражения (синие точки). ПЧ равна 6 ГГц.

4.4.3 Пик поглощения

Можно заметить, что при напряжении 3.5 мВ уровень отражения значительно снижается (детально на Рис.25). Этот пик поглощения можно объяснить тем, что импеданс СИС-смесителя становится почти равным импедансу подводящей линии ПЧ, формула (10). Точнее, в этой точке дифференциальное сопротивление СИС-смесителя становится равным действительной компоненте импеданса подводящей линии. Вычислив импеданс СИС-смесителя по формуле (10) мы сможем экспериментально определить импеданс подводящей линии . В данном случае, мы наблюдаем, что с высокой степенью точности составляет Ом. На Рис.25 приведено сравнение экспериментального уровня отражения и теоретического. Важно отметить, что уровень отражения в минимуме определяется модулем разности мнимых компонент импедансов СИС-смесителя и подводящей линии. Отличие глубины пика поглощения в измерении и в расчете позволяет проверить достоверность расчета, а также оценить величину комплексной части импеданса подводящей линии и за счет проведения измерений при различных частотах ПЧ. В нашем случае можно заключить, что мнимая часть импеданса подводящей линии не превышает 2 Ом. Таким образом, предложен и апробирован способ нахождения импеданса подводящей линии, используя особенности ВАХ СИС-смесителя. В целом, можно заключить, что уровень отражения достаточно высокий и составляет в среднем около -4.5dB в «рабочем»

диапазоне, что вынуждает нас использовать специальные вентили в канале ПЧ в СИС приемниках для минимизации стоячих волн в тракте ПЧ.

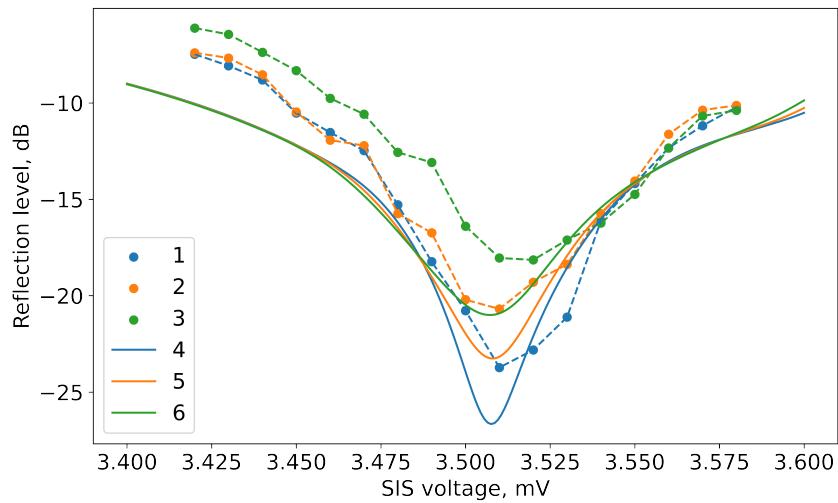


Рис. 25: Уровень отражения в «яме». Точки (№ 1,2,3) - экспериментальный уровень отражения для ПЧ 4,6,8 ГГц соответственно; Сплошные кривые (№ 4,5,6) - теоретический расчет уровня отражения для ПЧ 4,6,8 ГГц соответственно.

5 Заключение

Представлен метод экспериментального и теоретического определения ПЧ параметров СИС-перехода. Это позволяет исследовать зависимость уровня отражений от СИС-смесителя по ПЧ выходу от напряжения смещения и от мощности опорного сигнала. Определение параметров самого СИС-перехода совмещенное с моделированием элементов ПЧ канала позволит в будущем рассчитать с высокой точностью ПЧ характеристики самого смесителя, а также всего приёмника сконструированного на его основе. Дополнительная калибровка по пику поглощения при варьировании напряжения смещения позволяет повысить точность измерений, которые хорошо совпадают с теоретическим расчетом. Также представленный код позволяет вычислять импеданс (уровень отражения) по ПЧ СИС-смесителя с учетом импеданса внешней цепи, проводить измерение и калибровку в автоматизированном режиме.

В дальнейшем планируется исследование ПЧ параметров на частотах 4-12 ГГц и выше, а также при варьировании мощности опорного генератора в широком диапазоне.

Список литературы

- [1] Tucker J.R., Feldman M.J. Quantum detection at millimeter wavelengths. *Rev. Mod. Phys.* **57**, 4, 1055—1113, DOI:10.1103/RevModPhys.57.1055 (1985)
- [2] Belitsky, V., Bylund, M., Desmaris, V., Ermakov, A., Ferm, S.E., Fredrixon, M., Krause, S., Lapkin, I., Meledin, D., Pavolotsky, A. and Rashid, H., ALMA Band 5 receiver cartridge—Design, performance, and commissioning. *Astronomy & Astrophysics*, 611, p.A98 (2018)
- [3] Chenu, J.Y., Navarrini, A., Bortolotti, Y., Butin, G., Fontana, A.L., Mahieu, S., Maier, D., Mattiocco, F., Serres, P., Berton, M. and Garnier, O., The front-end of the NOEMA interferometer. *IEEE Transactions on Terahertz Science and Technology*, 6(2), 223 (2016)
- [4] Hesper, R., Khudchenko, A., Baryshev, A.M., Barkhof, J. and Mena, F.P., A high-performance 650-GHz sideband-separating mixer—design and results. *IEEE Transactions on Terahertz Science and Technology*, 7(6), 686 (2017)
- [5] Khudchenko, A., Hesper, R., Barkhof, J., Mena, F.P. and Baryshev, A.M., September. Comprehensive Description of Sideband Ratio of 2SB SIS Receiver. In 2019 44th International Conference on Infrared, Millimeter, and Terahertz Waves (IRMMW-THz) (pp. 1-2). IEEE. (2019)
- [6] Kooi J.W., Advanced Receivers for Submillimeter and Far Infrared Astronomy. Print Partners Ipskamps B.V., Enschede, The Netherlands, ISBN 978-90-367-3653-4 (2008)
- [7] Serres P. et al., The IF Output Impedance of SIS Mixers. *IEEE Transactions on Terahertz Science and Technology*, 5, 1, 27 (2015)
- [8] Barichev A.M., Superconductor-Insulator-Superconductor THz Mixer Integrated with a Superconducting Flux-Flow Oscillator. PhD thesis, Delft University of Technology, ISBN 90-9019220-4 (2005)
- [9] Garrett J.D., A 230 GHz Focal Plane Array Using a Wide IF Bandwidth SIS Receiver. PhD thesis, University of Oxford, (2018)
- [10] Shen T.M., Conversion gain in millimeter wave quasi-particle heterodyne mixers. *IEEE J. Quantum Electronics*, 17, 7, 1151 (1981)

- [11] Walker B., One-Port VNA Calibration: A Look Under the Hood. Copper Mountain Technologies. ResearchGate (2020)
- [12] Rudakov K., Dmitriev P., Baryshev A., Khudchenko A., Hesper R., Koshelets V. Low-Noise Sis Receivers for New Radio-Astronomy Projects. Radiophysics and Quantum Electronics. **62**. DOI: 10.1007/s11141-020-10001-7. (2020)

6 Приложение

Весь код доступен на github.com/yarvod/vna_cals

1. Код расчета импеданса СИС-смесителя; Mixer.py

```
from collections import defaultdict

import numpy as np
from matplotlib import pyplot as plt
import pandas as pd
import csv

from mpmath import besselj
from qmix.respfn import RespFnFromIVData
from qmix.mathfn.misc import slope
from scipy.constants import e, hbar
from scipy.optimize import curve_fit

# from .logger import *
# import logging
# logger = logging.getLogger(__name__)

def reim(df, num):
    data_z = np.zeros(shape=(num), dtype=np.complex128)
    data_z.real = np.array(df['re'], dtype=np.float64)
    data_z.imag = np.array(df['im'], dtype=np.float64)
    return data_z

def to_db(vec):
    return 20 * np.log10(np.abs(vec))

def moving_average(a, n=3):
    """
    :param list a: vector to average
```

```

:param int n: points window
:return: averaged vector
"""

a = np.array(a)
ret = np.cumsum(a)
ret[n:] = ret[n:] - ret[:-n]
return ret[n - 1:] / n

def power_aver(vec, n):
    return 10*np.log10(moving_average((np.abs(vec))**2, n))

def calc_offset(V, I):
    I = np.array(list(I))
    V = np.array(list(V))

    slp = slope(V, I)

    ind_v_pos = np.where(V>0.002)
    ind_v_neg = np.where(V<-0.002)

    ind_max_right = np.where(slp==np.max(slp[ind_v_pos]))
    ind_max_left = np.where(slp==np.max(slp[ind_v_neg]))

    aver = (V[ind_max_right] - V[ind_max_left]) / 2

    offset_v = aver - V[ind_max_right] # for addition

    V = V + offset_v
    v_nearest2zero = min(abs(V))
    ind_0 = np.where((V==v_nearest2zero) | (V== -v_nearest2zero))
    offset_i = - I[ind_0] # for addition

    return offset_v, offset_i

```

```

class Measure:

    def __init__(self, meas, cals, cal_impedance, point_num, freq_list,
                 rho=50):
        self.meas = meas
        self.cals = cals
        self.open_z = np.vectorize(cal_impedance.get('open')) or None
        self.short_z = np.vectorize(cal_impedance.get('short')) or None
        self.load_z = np.vectorize(cal_impedance.get('load')) or None
        self.rho = rho

        self.freq_list = freq_list / 1e9

        self.point_num = point_num

    def _get_z(self):
        self.cal_load_z = self.cals.iloc[:, 2]
        self.cal_open_z = self.cals.iloc[:, 0]
        self.cal_short_z = self.cals.iloc[:, 1]

        self.point_z = np.array(self.meas, dtype=np.complex128)

    @staticmethod
    def _E_matrix(C, V):
        C_H = np.matrix.getH(C)
        inv_CC_H = np.linalg.inv(np.dot(C_H, C))
        result = np.dot(np.dot(inv_CC_H, C_H), V.T)
        return result

    @staticmethod
    def _Error_Coeffs(vector, cals):
        cals['D'].append(vector[1])
        cals['S'].append(vector[2])
        cals['R'].append(vector[0] + vector[1] * vector[2])

```

```

@staticmethod

def _Gamma(att, cal_frame):
    gamma = (att - cal_frame['D']) / (cal_frame['R'] + cal_frame['S'] *
                                         att - cal_frame['S'] * cal_frame['D'])
    return gamma

def _gamma_cal(self, Z_n):
    return (Z_n - self.rho) / (Z_n + self.rho)

def calibrate(self):

    self._get_z()

    G_a_1 = self._gamma_cal(self.load_z(self.freq_list * 1e9)) # Actual
    match load
    G_a_2 = self._gamma_cal(self.open_z(self.freq_list * 1e9)) # Actual
    open
    G_a_3 = self._gamma_cal(self.short_z(self.freq_list * 1e9)) # Actual
    short

    cals = {'D': [], 'S': [], 'R': []}
    for i in range(self.point_num):
        G_m_1 = self.cal_load_z[i] # Measured match load
        G_m_2 = self.cal_open_z[i] # Measured open
        G_m_3 = self.cal_short_z[i] # Measured short

    C = np.array(
        [[G_a_1[i], 1, G_a_1[i] * G_m_1], [G_a_2[i], 1, G_a_2[i] *
                                                G_m_2], [G_a_3[i], 1, G_a_3[i] * G_m_3]])
    V = np.array([G_m_1, G_m_2, G_m_3])

    self._Error_Coeffs(self._E_matrix(C, V), cals)

    cals_frame = pd.DataFrame(data=cals, index=self.freq_list.round(4))
    self.point_calibrated = self._Gamma(self.point_z, cals_frame)

```

```

def plot(self, pic_name='SIS_IF_Ref', plot_phase=None, title='SIS IF
Reflection', pic_path='', save=False,
        start=None, stop=None):

    if plot_phase:
        plt.figure(figsize=(19, 6))
        plt.suptitle(title)

        plt.subplot(121)
        plt.plot(self.freq_list[start:stop], 20 *
                 np.log10(np.abs(self.point_calibrated))[start:stop])
        plt.xlabel('frequency, GHz')
        plt.ylabel('Amp, dB')
        plt.minorticks_on()
        plt.grid(which='minor', linestyle=':')
        plt.grid()

        plt.subplot(122)
        plt.plot(self.freq_list[start:stop],
                 np.angle(self.point_calibrated)[start:stop])
        plt.xlabel('frequency, GHz')
        plt.ylabel(r'phase, $rad$')
        plt.minorticks_on()
        plt.grid(which='minor', linestyle=':')
        plt.grid()

    if not plot_phase:
        plt.figure(figsize=(10, 6))
        plt.title(title)

        plt.plot(self.freq_list[start:stop], 20 *
                 np.log10(np.abs(self.point_calibrated))[start:stop])
        plt.xlabel('frequency, GHz')
        plt.ylabel('Amp, dB')
        plt.minorticks_on()
        plt.grid(which='minor', linestyle=':')

```

```

        plt.grid()

    if save:
        plt.savefig(pic_path + pic_name + '.pdf', dpi=400)
        plt.show()

def plot_cals(self, pic_name='Cals', title='Calibrations', pic_path='', save=False, plot_measured=False, plot_calibrated=False):

    plt.figure(figsize=(10, 6))
    plt.title(title)

    lgnd = ['open', 'short', 'load']
    self._get_z()
    plt.plot(self.freq_list, 20 * np.log10(np.abs(self.cal_open_z)))
    plt.plot(self.freq_list, 20 * np.log10(np.abs(self.cal_short_z)))
    plt.plot(self.freq_list, 20 * np.log10(np.abs(self.cal_load_z)))

    if plot_calibrated:
        plt.plot(self.freq_list, 20 *
                 np.log10(np.abs(self.point_calibrated)))
        lgnd.append('point_calibrated')
    if plot_measured:
        plt.plot(self.freq_list, 20 * np.log10(np.abs(self.point_z)))
        lgnd.append('point_measured')

    plt.legend(lgnd)

    plt.xlabel('frequency, GHz')
    plt.ylabel('Amp, dB')
    plt.minorticks_on()
    plt.grid(which='minor', linestyle=':')
    plt.grid()

    if save:

```

```

        plt.savefig(pic_path + pic_name + '.png', dpi=400)
        plt.show()

class Mixer:

    def __init__(self, meas_table, cal_table, V_bias, point_num, LO_rate,
                 offset=(0,0), rho=50):
        self.meas_table = meas_table
        self.cal_table = cal_table
        self.offset = offset

        self.IV_curve = dict(
            (
                float(s.split(';')[0]) + self.offset[0],
                float(s.split(';')[1]) + self.offset[1]
            )
            for s in self.cal_table.columns[:-1]
        )

        self.LO_rate = LO_rate

        self.I = np.array(list(self.IV_curve.values()))
        self.V = np.array(list(self.IV_curve.keys()))

        self.V_bias = V_bias
        self.rho = rho

        self.resp = RespFnFromIVData(self.Vn, self.In)
        self.point_num = point_num

        self.freq_list = np.array(self.meas_table.pop('freq'),
                                  dtype=np.float64)

        self._measures = defaultdict(Measure)
        self.cal_impedance = None

```

```

def calibrate(self):
    for key in self._measures.keys():
        self._measures[key].calibrate()

@property
def measures(self):
    return self._measures

def set_measures(self, recalculate=False):
    if recalculate or not self.cal_impedance:
        self.set_cal_impedance()
    for vi, meas in self.meas_table.items():
        key = f"{float(vi.split(';')[0]) + self.offset[0]};{float(vi.split(';')[1]) + self.offset[1]}"
        self._measures[key] = Measure(meas, self.cals, self.cal_impedance,
                                      point_num=self.point_num, rho=self.rho,
                                      freq_list=self.freq_list)

def remove_measures(self):
    self._measures = defaultdict(Measure)

@property
def cals(self):
    open = self.cal_table.filter(like=f"{self.V_bias['open']}").iloc[:,0]
    short = self.cal_table.filter(like=f"{self.V_bias['short']}").iloc[:,0]
    load = self.cal_table.filter(like=f"{self.V_bias['load']}").iloc[:,0]
    return pd.DataFrame((open, short, load), dtype=complex).T

@property
def Vn(self):
    return self.V / self.Vgap

@property
def In(self):
    return self.I / self.Igap

```

```

@property
def Igap(self):
    cond = slope(self.V, self.I)
    ind = np.where(cond == np.max(cond))
    return self.I[ind]

@property
def Vgap(self):
    cond = slope(self.V, self.I)
    ind = np.where(cond == np.max(cond))

    return self.V[ind]

@staticmethod
def kron(a, b):
    if a == b:
        return 1
    else:
        return 0

def set_cal_impedance(self):
    res = {
        'open': {'re': [], 'im': []},
        'short': {'re': [], 'im': []},
        'load': {'re': [], 'im': []}
    }
    par = {
        'open': {'opt_re': [], 'cov_re': [], 'opt_im': [], 'cov_im': []},
        'short': {'opt_re': [], 'cov_re': [], 'opt_im': [], 'cov_im': []},
        'load': {'opt_re': [], 'cov_re': [], 'opt_im': [], 'cov_im': []}
    }
    nu0_range = self.freq_list[::15]
    f_re = lambda x, a1, a2, a3, a4: a1 * x ** 3 + a2 * x ** 2 + a3 * x +
        a4
    f_im = lambda x, a1, a2, a3, a4: a1 * x ** 3 + a2 * x ** 2 + a3 * x +

```

a4

```
for nu0 in nu0_range:
    z_open = self.Z(nu=self.L0_rate, nu0=nu0, V0=self.V_bias['open'],
                    al=0.01)
    z_short = self.Z(nu=self.L0_rate, nu0=nu0,
                      V0=self.V_bias['short'], al=0.01)
    z_load = self.Z(nu=self.L0_rate, nu0=nu0, V0=self.V_bias['load'],
                     al=0.01)

    res['open']['re'].append(z_open.real)
    res['open']['im'].append(z_open.imag)
    res['short']['re'].append(z_short.real)
    res['short']['im'].append(z_short.imag)
    res['load']['re'].append(z_load.real)
    res['load']['im'].append(z_load.imag)

for key in par.keys():
    par[key]['opt_re'], par[key]['cov_re'] = curve_fit(f_re,
                                                          nu0_range, res[key]['re'])
    par[key]['opt_im'], par[key]['cov_im'] = curve_fit(f_im,
                                                          nu0_range, res[key]['im'])

imp = {
    'open': lambda x: f_re(x, *par['open']['opt_re']) + f_im(x,
                                                               *par['open']['opt_im']) * 1j,
    'short': lambda x: f_re(x, *par['short']['opt_re']) + f_im(x,
                                                               *par['short']['opt_im']) * 1j,
    'load': lambda x: f_re(x, *par['load']['opt_re']) + f_im(x,
                                                               *par['load']['opt_im']) * 1j
}
self.cal_impedance = imp

def _G(self, nu, nu0, V0, al, lim=10, mrange=[0]):
    """
    :param float nu: FFO rate
```

```

:param float nu0: IF rate
"""

om = nu * np.pi * 2
om0 = nu0 * 2 * np.pi
omm = lambda m: m * om + om0

g = np.zeros((len(mrange), len(mrange)))
d = max(mrange)

for m in mrange:
    for m1 in mrange:
        for n in np.arange(-lim, lim + 1, 1):
            for n1 in np.arange(-lim, lim + 1, 1):
                g[m + d][m1 + d] += float(
                    besselj(n, al) * besselj(n1, al) * self.kron(m - m1,
                    n1 - n) *
                    ((self.resp.idc((V0 + n1 * hbar * om / e + hbar *
                    omm(m1) / e) / self.Vgap) - self.resp.idc(
                    (V0 + n1 * hbar * om / e) / self.Vgap)) +
                    (self.resp.idc((V0 + n * hbar * om / e) /
                    self.Vgap) - self.resp.idc(
                    (V0 + n * hbar * om / e - hbar * omm(m1) / e) /
                    self.Vgap))) * self.Igap
                )
    g[m + d][m1 + d] *= e / (2 * hbar * om0)
return g

def _B(self, nu, nu0, V0, al, lim=10, mrange=[0]):
"""

:param float nu: FFO rate
:param float nu0: IF rate
"""

om = nu * np.pi * 2
om0 = nu0 * 2 * np.pi
omm = lambda m: m * om + om0

```

```

b = np.zeros((len(mrange), len(mrange)))

d = max(mrange)

for m in mrange:

    for m1 in mrange:

        i = 0

        for n in np.arange(-lim, lim + 1, 1):

            for n1 in np.arange(-lim, lim + 1, 1):

                b[m + d][m1 + d] += float(
                    besselj(n, al) * besselj(n1, al) * self.kron(m - m1,
                    n1 - n) * \
                    ((self.resp.idc((V0 + n1 * hbar * om / e + hbar *
                    omm(m1) / e) / self.Vgap) - self.resp.idc(
                    (V0 + n1 * hbar * om / e) / self.Vgap)) - \
                    (self.resp.idc((V0 + n * hbar * om / e) /
                    self.Vgap) - self.resp.idc(
                    (V0 + n * hbar * om / e - hbar * omm(m1) / e) /
                    self.Vgap))) * self.Igap
                )

                b[m + d][m1 + d] *= e / (2 * hbar * om0)

return b


def Z(self, nu, nu0, V0, al, lim=10, mrange=[0]):

    """
    :param float nu: FFO rate
    :param float nu0: IF rate
    :param float V0: V bias
    :param float al: pumping level
    """

    g = np.array(self._G(nu, nu0, V0, al, lim,
                         mrange)[max(mrange)][max(mrange)])
    b = np.array(self._B(nu, nu0, V0, al, lim,
                         mrange)[max(mrange)][max(mrange)])
    y = g + b * 1j

return 1 / y

```

```

def Ip(self, V0, al=0.2, nu=600 * 10 ** 9, lim=10):
    """
    :param float V0: SIS Bias
    :param float al: Pumping level
    :param float om: FFO rate
    :param int lim: summ limit
    """
    res = 0
    om = 2 * np.pi * nu
    for n in np.arange(-lim, lim + 1, 1):
        res += besselj(n, al) ** 2 * self.resp.idc((V0 + n * hbar * om /
            e) / self.Vgap) * self.Igap
    return res

def plot_Ip(self, al, nu, V=None):
    """
    :param float al: pumping parameter
    :param float nu: Heterodyne rate
    :param float V: bias voltages range
    """
    if not V:
        V = self.V

    plt.figure(figsize=(10, 6))
    plt.plot(self.V * 1000, self.I * 1000, label='autonomus')
    plt.plot(self.V * 1000, self.Ip(V, al=al, nu=nu) * 1000,
             label='pumped')
    plt.ylabel('I, mA')
    plt.xlabel('V, mV')
    plt.legend()
    plt.grid()
    plt.show()

def mixing(meas_table, cal_table, V_bias, LO_rate, offset=(0,0),

```

```

    point_num=300, rho=50):
    mixer = Mixer(
        meas_table=meas_table,
        cal_table=cal_table,
        V_bias=V_bias,
        point_num=point_num,
        offset=offset,
        LO_rate=LO_rate,
        rho=rho
    )
    mixer.set_measures()
    mixer.calibrate()
    return mixer

```

2. Код взаимодействия с блоком смещения по напряжению; scpi.py

```

import socket
from collections import defaultdict
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from vna import get_data
import time
import csv
import logger as mylogger
import logging

from config import BLOCK_IP

logger = logging.getLogger(__name__)

ST_OK = 'OK'

class Block:

    HOST = BLOCK_IP

```

```

PORT = 9876

IV = defaultdict(list)

@classmethod
def manipulate(cls, cmd: str) -> str:
    if type(cmd) != bytes:
        cmd = bytes(cmd, 'utf-8')
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.connect((cls.HOST, cls.PORT))
        s.sendall(cmd)
        data = s.recv(1024)

    return data.decode().rstrip()

def get_current(self):
    return self.manipulate('BIAS:DEV2:CURR?')

def get_voltage(self):
    return self.manipulate('BIAS:DEV2:VOLT?')

def set_voltage(self, volt: float):
    self.manipulate(f'BIAS:DEV2:VOLT {volt}')

def measure_IV(self, v_from: float, v_to: float, points: int):
    iv = defaultdict(list)
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        s.connect((self.HOST, self.PORT))

        s.sendall(bytes(f'BIAS:DEV2:VOLT?', 'utf-8'))
        init_v = float(s.recv(1024).decode().rstrip())
        for v in np.linspace(v_from, v_to, points):

            s.sendall(bytes(f'BIAS:DEV2:VOLT {v}', 'utf-8'))
            if v == v_from or (2.1e-3 < v < 2.5e-3) or (-2.5e-3 < v <
-2.1e-3):

```

```

        time.sleep(0.2)

        status = s.recv(1024).decode().rstrip()

        i = 0

        while 1:

            try:

                time.sleep(0.1)

                s.sendall(b'BIAS:DEV2:CURR?')

                i = s.recv(1024).decode().rstrip()

                i = float(i)

                break

            except ValueError:

                logger.error(f'Error with v = {v}; i = {i}')

                continue

        iv['I'].append(float(i))

        iv['V'].append(v)

        logger.info(f"volt {v}; curr {float(i)}")

        s.sendall(bytes(f'BIAS:DEV2:VOLT {init_v}', 'utf-8'))

    return iv


def write_IV_csv(self, path, iv):

    with open(f'{path}', 'w') as f:

        writer = csv.writer(f)

        writer.writerow(['I', 'V'])

        for i, v in zip(iv['I'], iv['V']):

            writer.writerow([i, v])


def write_refl_csv(self, path, refl):

    df = pd.DataFrame(refl)

    df.to_csv(path, index=False)


def calc_offset(self):

    iv1 = self.measure_IV(v_from=0, v_to=7e-3, points=300)

    iv2 = self.measure_IV(v_from=7e-3, v_to=0, points=300)


def measure_reflection(self, v_from: float, v_to: float, v_points: int,
                      f_from: float, f_to: float, f_points: int, s_par,

```

```

        exp_path, avg: int):

refl = defaultdict(list)

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((self.HOST, self.PORT))

    s.sendall(bytes(f'BIAS:DEV2:VOLT?', 'utf-8'))
    init_v = float(s.recv(1024).decode().rstrip())
    for v in np.linspace(v_from, v_to, v_points):
        s.sendall(bytes(f'BIAS:DEV2:VOLT {v}', 'utf-8'))
        if v == v_from or (2.1e-3 < v < 2.5e-3) or (-2.5e-3 < v <
-2.1e-3):
            time.sleep(0.2)
        status = s.recv(1024).decode().rstrip()
        i = 0
        while 1:
            try:
                time.sleep(0.2)
                s.sendall(b'BIAS:DEV2:CURR?')
                i = s.recv(1024).decode().rstrip()
                i = float(i)
                break
            except ValueError:
                logger.error(f'Error with v = {v}; i = {i}')
                continue

res = get_data(
    param=s_par,
    plot=False,
    plot_phase=False,
    freq_start=f_from, freq_stop=f_to,
    exp_path=exp_path,
    freq_num=f_points or 201,
    avg=int(avg)
)
logger.info(f"volt {v}; curr {float(i)}")
refl[f"{v};{i}"] = res['trace']

```

```

    s.sendall(bytes(f'BIAS:DEV2:VOLT {init_v}', 'utf-8'))


    refl['freq'] = res['freq']

    return refl


def plot_iv(self, iv):
    plt.figure(figsize=(10, 6))
    plt.scatter(np.array(iv['V'])*1e3, np.array(iv['I'])*1e3, s=2)
    plt.xlabel('SIS Voltage, mV')
    plt.ylabel('SIS Current, mA')
    plt.minorticks_on()
    plt.grid(which='minor', linestyle=':')
    plt.grid()
    plt.show()

```

3. Код взаимодействия с VNA; vna.py

```

import matplotlib.pyplot as plt
import numpy as np
import csv, os
import qcodes.instrument_drivers.rohde_schwarz.ZNB as ZNB
import qcodes

from config import VNA_IP


def get_data(param, exp_path, freq_start=3.5e9, freq_stop=8.5e9,
            freq_num=201, vna_power=-30, avg=None, aver=False, num=None,
            mov_aver=False, span=None, plot=False, plot_phase=False,
            save=True):
    title = 'IF Reflection'
    plot_phase = plot_phase
    exp_path = exp_path
    IP = VNA_IP
    ZNB.ZNB.close_all()
    vna = ZNB.ZNB('VNA', f"TCPIPO:::{IP}::INSTR", init_s_params=False)

```

```

station = qcodes.Station(vna)

vna.add_channel(param)

vna.cont_meas_on()
vna.display_single_window()
vna.rf_on()

freq_start = freq_start
freq_stop = freq_stop
freq_num = freq_num
freq = np.linspace(freq_start, freq_stop, freq_num)

vna.channels.format('Complex')
vna.channels.start(freq_start)
vna.channels.stop(freq_stop)
vna.channels.npts(freq_num)
vna.channels.power(vna_power)

if avg:
    vna.channels.avg(avg)
vna.channels.autoscale()

if aver:
    trace = av(num, vna)
    db = 20 * np.log10(np.abs(trace))

if mov_aver:
    trace = vna.channels.trace.get()[0]
    db = 20 * np.log10(np.abs(mov_av(list(trace), span)))
    freq = freq[span-1:]

else:
    trace = vna.channels.trace.get()[0]
    db = 20 * np.log10(np.abs(trace))

if plot and plot_phase:
    plt.figure(figsize=(18, 6))
    plt.suptitle(title)

```

```

plt.subplot(121)
plt.plot(freq, db)
plt.xlabel('frequency, GHz')
plt.ylabel('Amp, dB')
plt.minorticks_on()
plt.grid(which='minor', linestyle=':')
plt.grid()

plt.subplot(122)
plt.plot(freq, np.angle(trace))
plt.xlabel('frequency, GHz')
plt.ylabel(r'phase, $rad$')
plt.minorticks_on()
plt.grid(which='minor', linestyle=':')
plt.grid()
plt.show()

elif plot:
    plt.figure(figsize=(10, 6))
    plt.plot(freq, db)
    plt.xlabel('frequency, GHz')
    plt.ylabel('Amp, dB')
    plt.minorticks_on()
    plt.grid(which='minor', linestyle=':')
    plt.grid()
    plt.show()

if not os.path.exists(f'{exp_path}/current'):
    os.mkdir(f'{exp_path}/current')

if save:
    with open(f'{exp_path}/current/data.csv', 'w') as f:
        writer = csv.writer(f)
        writer.writerow(['freq-Hz', 're', 'im'])
        for i in range(len(freq)):

```

```
writer.writerow([freq[i], trace[i].real, trace[i].imag])

return {'freq': freq, 'trace': trace}

def save_data(pic_path, exp_path):
    if os.path.exists(f'{exp_path}/current/data.csv') and pic_path:
        os.replace(f'{exp_path}/current/data.csv', f"{pic_path}")

def save_calibrated_data(data_path, exp_path):
    if os.path.exists(f'{exp_path}/current/calibrated_data.csv') and
       data_path:
        os.replace(f'{exp_path}/current/calibrated_data.csv', f"{data_path}")
```